A First Look at the Impact of Distillation Hyper-Parameters in Federated Knowledge Distillation

Norah Alballa
KAUST

Marco Canini
KAUST

Abstract

Knowledge distillation has been known as a useful way for model compression. It has been recently adopted in the distributed training domain, such as federated learning, as a way to transfer knowledge between already pre-trained models. Knowledge distillation in distributed settings promises advantages, including significantly reducing the communication overhead and allowing heterogeneous model architectures. However, distillation is still not well studied and understood in such settings, which hinders the possible gains. We bridge this gap by performing an experimental analysis of the distillation process in the distributed training setting, mainly with non-IID data. We highlight some elements that require special considerations when transferring knowledge between already pre-trained models: the transfer set, the temperature, the weight, and the positioning. Appropriately tuning these hyper-parameters can remarkably boost learning outcomes. In our experiments, around two-thirds of the participants require settings other than commonly used default settings in literature, and appropriate tuning can reach more than five times improvement on average.

CCS Concepts: • **Computing methodologies** → **Machine learning**; *Distributed algorithms.*

Keywords: Knowledge Distillation, Joint Distillation, Decentralized Learning

ACM Reference Format:
Norah Alballa and Marco Canini. 2023. A First Look at the Impact of Distillation Hyper-Parameters in Federated Knowledge Distillation. In *3rd Workshop on Machine Learning and Systems (EuroMLSys '23), May 8, 2023, Rome, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3578356.3592590

EuroMLSys '23, May 8, 2023, Rome, Italy

ACM ISBN 979-8-4007-0084-2/23/05…$15.00
https://doi.org/10.1145/3578356.3592590

1 Introduction

Knowledge Distillation (KD) is a technique that was initially proposed for model compression, where a larger teacher model is used to train a more compact student model. A smaller student model learns a high-fidelity representation of a larger teacher model through the teacher's output (soft targets) [4, 10, 23]. The teacher is usually pre-trained while the student model learns by mimicking the teacher's soft targets on the same training set or a separate transfer set.

Besides model compression settings that employ offline distillation (i.e., from a trained teacher to an untrained student), knowledge distillation has been recently employed to transfer knowledge between already pre-trained models in distributed training settings [5, 11–13, 16, 17, 22]. These models might be trained using different data samples, might have diverse performances resulting from heterogeneity, including statistical heterogeneity (i.e., non-IID data distributions), where participants typically have different distributions and quantities of local data, and system heterogeneity, where participants might have different amounts of bandwidth and computational power. Learning across these systems and unbalanced datasets under those constraints can be challenging. We call knowledge distillation between already pre-trained models "*joint distillation.*"

Joint distillation can be useful for distributed training and federated learning (FL) and can deliver great advantages, including (1) significant communication reduction [12, 16, 21, 22], and (2) model architecture flexibility [5, 11, 13, 17], as participants can pick the model architecture that suits their capabilities. The communication reduction acquired from knowledge distillation can be obtained either by communicating the pre-trained models only once instead of communicating the models updates at every round [9, 16], or by communicating the model outputs only, which have negligible size compared to the model updates. This was found to reduce the communication overheads by up to 99% while achieving similar model performance relative to a FL benchmark, even with non-IID data distributions [12]. However, joint distillation is still not well studied and understood in the literature, hindering the possible gains.

We focus on the problem of joint distillation, where pre-trained models learn directly from each other to produce models that combine their knowledge. In other words, we assume a peer-to-peer system of collaborating participants. We

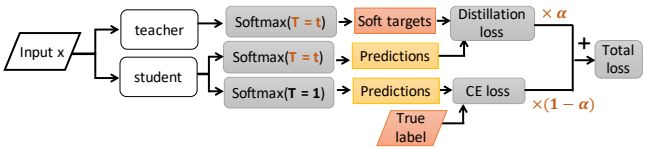

Figure 1. Knowledge distillation pipeline. The hyper-parameters of interest in this work are the temperature T, the weight α, the transfer set (from which input x comes), and the position (teacher vs. student).

highlight some critical factors that require careful consideration in joint distillation (i.e., the transfer set, the weight, the temperature, and the positioning) compared with the other distillation approaches (cf. Figure 1). Our analysis indicates that appropriately tuning some hyper-parameters in joint distillation can substantially improve knowledge transfer outcomes.

We make the following contributions:

- We are the first to analyze and study the impact of distillation hyper-parameters when transferring knowledge between already pre-trained models, i.e., joint distillation.
- We identify and highlight different factors (the transfer set, the temperature, the weight, and the positioning) that require special consideration in joint distillation, unlike offline and online distillation.
- We empirically demonstrate that appropriately tuning those hyper-parameters can significantly improve performance and the amount of knowledge transfer, compared to commonly used default settings.

We hope that our study can shed some light on the future research of distributed training and federated learning with non-IID data, allowing to exploit the great communication reduction and model architecture flexibility offered by knowledge distillation. Indeed, hyper-parameter tuning is a recurring, hard problem in machine learning applications; our work contributes a quantification of the extent to which a range of hyper-parameters affect performance in the joint distillation setting.

2 Background

Preliminaries. Knowledge distillation was first proposed as a way to compress a large model, the teacher, into a smaller model, the student, without a significant drop in performance [1, 4, 10, 23]. The idea is that the student can learn faster and more efficiently by mimicking the output of the teacher model (soft probabilities or soft targets) than just from the class label. Soft targets are the class probabilities p_i derived from applying the softmax operation to the logits z_i as follows:

$$p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)} \quad (1)$$

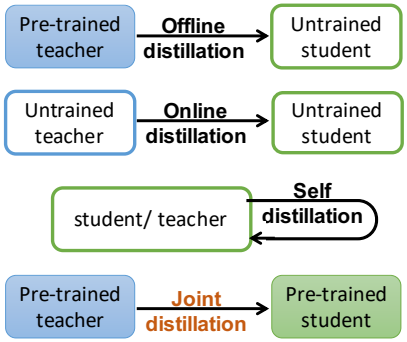

Figure 2. Distillation approaches.

where T is the "temperature" parameter that is usually set to 1 in the regular softmax, whereas in the distillation setting, it is used to control the softness of the probability distribution over classes.

There are several variants of distillation, using different types of loss functions and different options for what dataset is used to distill the knowledge to the student model (called transfer set or proxy dataset). For example, the transfer set could be a large public unlabeled dataset, a public labeled dataset, or even the teacher's original training data. If the transfer set is labeled, the student can be trained using a linear combination of two loss functions:

$$\mathcal{L} = (1-\alpha)\ \mathcal{L}_{CE}(p^S, y) + \alpha\ \mathcal{L}_{KL}(p^S, p^T) \quad (2)$$

where α and $(1-\alpha)$ are the weights assigned to the Kullback–Leibler (KL) divergence (i.e., distillation loss) and the cross entropy (CE) loss, respectively, y is the true label hot encoded, p^S and p^T are the class probabilities (soft targets) predicted by the student model and the teacher model, respectively. Figure 1 illustrates the knowledge distillation pipeline.

Distillation approaches. The distillation literature has investigated three learning approaches of knowledge distillation: offline distillation, online distillation, and self-distillation. Figure 2 summarizes the distillation approaches. We provide a brief description below; for details, we refer the reader to a recent survey [8].

The **offline distillation** is a traditional way commonly used for model compression. A pre-trained teacher is used to train an untrained student using a loss term that encourages the student's predictions to match the predictions of the teacher model [10]. Most previous knowledge distillation works use the offline method [8]. In the **online distillation**, both the teacher and the student are untrained, and they learn collaboratively and teach each other throughout the training process [1, 27]. In **self-distillation**, the same networks are used for the teacher and the student models, and the knowledge is transferred in the same model, from the deeper layers to the shallow layers [26].

Participant #	Local model accuracy
0	18.93%
1	10.00%
2	74.54%
3	19.21%
4	53.79%
5	42.68%
6	18.79%
7	55.90%
8	25.13%
9	69.12%

Table 1. Per-participant accuracy of the pre-trained model. For reference, a model trained centrally over the entire dataset achieves 82.68% accuracy.

However, there is another promising approach that is not yet well studied and investigated in literature; when both the teacher and the student are already pre-trained on disjoint subsets of the data, and the goal is to directly transfer the knowledge from one to another and combine the knowledge previously learned by both models. We call this approach "**joint distillation**."

Distributed distillation. Joint distillation can deliver prominent advantages, particularly in distributed training and FL settings.

Participants can do the full local training, then transfer knowledge to each other. This can enormously reduce the communication overhead, rather than sending model updates at every round. Moreover, unlike model merging [20] that requires models to share a common architecture and initialization, distillation allows model heterogeneity, and models are no longer restricted to having the same architecture. This is especially important for cross-device FL settings where devices have various computing and storage capabilities.

Knowledge distillation has recently grabbed increasing attention in the distributed training and federated learning community, and some works employed distillation to transfer knowledge to pre-trained models (e.g., [5, 11, 12, 16, 17, 22]). However, transferring knowledge between already pre-trained models (i.e., joint distillation) is not well studied and understood in the literature. Our study found that some critical elements (i.e., the transfer set, the weight, the temperature, and the position) should be carefully tuned in joint distillation compared to the other distillation approaches, and that appropriately tuning these elements can remarkably affect the learning outcomes.

3 Methodology

Task, datasets and models. For this preliminary study, we focus on a standard computer vision task: image classification. As a dataset, we use CIFAR10 [14], which is commonly used in the FL literature (e.g., [2, 3, 7, 15, 18, 19]).

Name	Symbol	Values
Temperature	T	[0.1, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 6, 7]
Weight	α	[0.1, 0.25, 0.5, 0.75, 0.9]
Transfer set	S	[Student, Public, Public + Student model]

Table 2. Space of hyper-parameters.

We assume a distributed cross-device training scenario with 10 participants.

The CIFAR-10 dataset consists of 32x32 color images of 10 classes. There are 50K training images and 10K test images. We use 10% of the training set as a validation set. The same testing set is used to measure the accuracy of the participants' models after distillation.

We split the training dataset into partitions so that the data for each participant is non-IID. To split the training data, we assign a random number of classes to every participant, then assign random samples from each class to the participants that have the class. Different participants may have different portion sizes of the classes.

To exclude the effects of different model architecture, we use ResNet-18 for each participant. We plan to revisit this choice in follow-up work.

For each participant, we pre-train the model over the local dataset for at most 100 epochs. We employ early stopping after the validation performance plateaus for 10 epochs. We use standard settings for hyper-parameters: an Adam optimizer is used with a learning rate of $1e{-}3$, weight decay of $1e{-}4$, and a batch size of 32. These hyper-parameters are kept constant over all tasks. Table 1 lists the baseline accuracy of each participant on the testing dataset.

KD setup. The space of experiments for KD is listed in Table 2. The transfer set may be the student dataset or a public unlabelled dataset. In the latter case, we also explore a setting in which we add the student's pre-trained model as a teacher for itself.

A KD experiment runs a KD configuration. By configuration, we mean the 5-tuple given by teacher participant, student participant, and a valuation of the three hyper-parameters drawn from the space defined above. Since we exclude self-transfer, there are 90 possible teacher-student pairs and 3,060 KD configurations.

We consider the configurations with $T = 1, \alpha = 0.5$ as baselines since this choice of parameters is the most common one in the literature [5–7, 13, 16, 21, 22]. By default, we use the student's training set as transfer set. To characterize the effects of each hyper-parameter, we vary the hyper-parameter of interest and set the others to their baseline values. We also explore optimizing T and α sequentially (in both orders), to gauge the importance of joint optimization.

For simplicity, and in line with our assumption of a peer-to-peer collaborating system, we consider the setting with a single student and a single teacher. We leave to future work

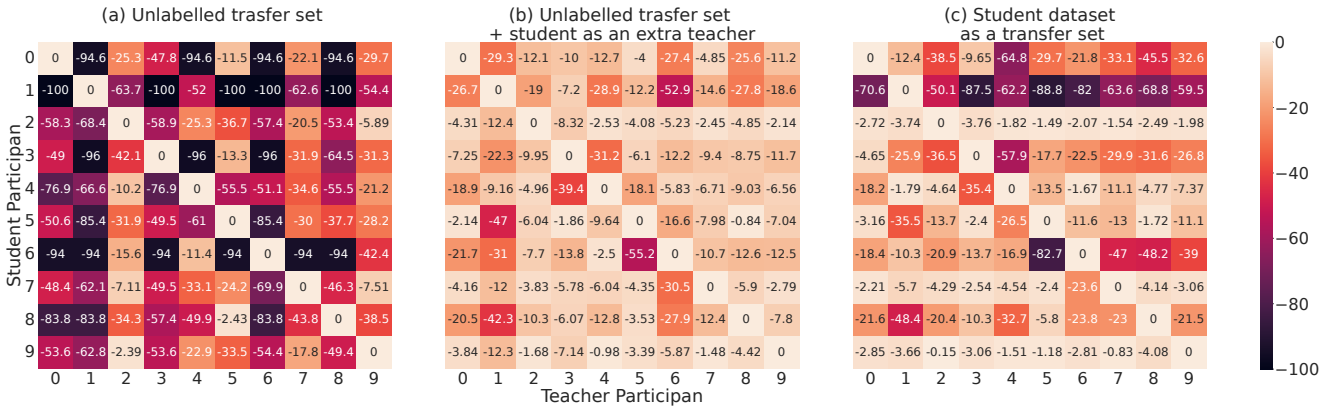

Figure 3. Adding the student as an additional teacher for itself when using an unlabelled transfer set, or using the student dataset as a transfer set, can remarkably mitigate the student's forgetting.

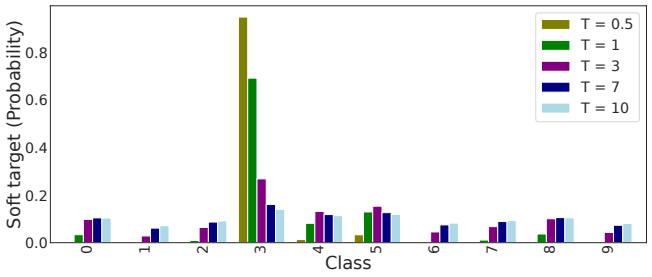

Figure 4. Example soft targets distribution with different T. The input image is from the CIFAR10 training set, and the ground truth label is 3 (cats).

studying the case of joint distillation with multiple teachers and a single student.

Metrics. The main metric of interest is the *accuracy gain* in percentage points after distillation. That is, we consider the accuracy at the participant holding the role of student and we subtract the accuracy of that participant's pre-trained model. Note that this difference could be negative in case KD isn't beneficial for a particular pair of participants. Another metric of interest is the *forgetting* which we measure by considering the negative values resulting from the difference between the student models' per-class accuracy after and before distillation.

4 Joint Distillation

We highlight the effect of some important factors that can remarkably affect the performance and should be tuned carefully in joint distillation, compared to other distillation approaches.

Transfer Set. There are several variants of distillation, using different options for what dataset is used to distill the knowledge to the student model (called transfer set or proxy dataset). One common approach is to use a large unlabelled dataset, as it is far easier to collect than a labeled dataset [3, 5–7, 12, 18, 21, 22]. However, we find that this approach alone is sub-optimal for joint distillation. Besides the significant performance degradation that can happen with the increase in the distribution variation between the public dataset and participants' datasets that are usually non-IID [25], our experiments reveal that using such unlabelled transfer set can cause the student to significantly or completely forget all the knowledge it has previously learned, specifically when the student and the teacher have different class distributions.

To mitigate the forgetting problem, we consider two approaches:

1. Add a copy of the student model as an additional teacher to itself when using an unlabelled transfer set.
2. Use the student dataset as a transfer set with the addition of the cross-entropy loss (CE) between the student's predictions and the ground truth label.

We observe that these two approaches can reduce forgetting and help the student remember the knowledge it was previously trained for, yet learn new knowledge from the teacher (even with the second option when the student dataset doesn't include any samples of the teacher's classes).

Figure 3 shows a comparison of the three approaches in terms of the student's percentage points forgetting. Each sub-figure involves knowledge transfers of 90 pairs of participants. The unlabelled transfer set used for this experiment contains 10 percent of each of the classes. Although the size of the unlabelled transfer set is more than most of the participants' datasets in the setting and despite the lack of distribution variation, the forgetting is significant when using the unlabelled public dataset alone (cf. Figure 3 (a)).

It is worth mentioning that forgetting is sometimes inevitable, especially when students and teachers have different class distributions. Thus, learning from participants who were pre-trained with some similarities in their objectives

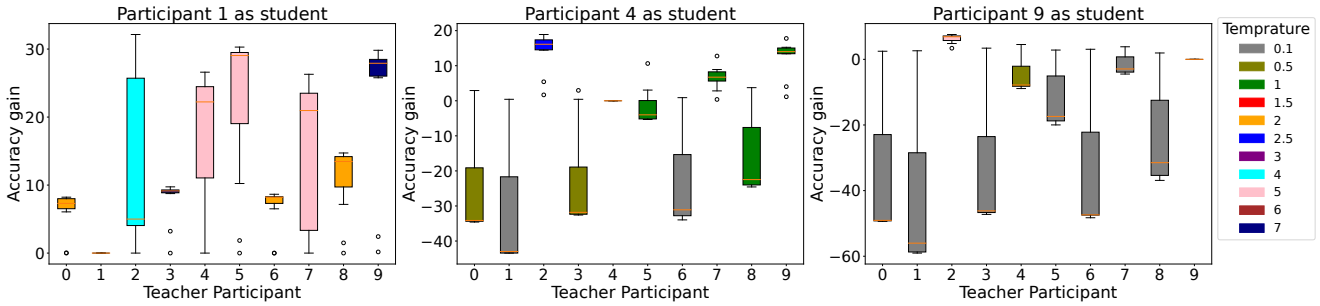

Figure 5. No single choice of temperature T is always optimal. The boxplot shows the after-distillation accuracy gain for three participants learning from different teachers for all the choices of T. The boxes are colored according to the T that achieves the maximum gain.

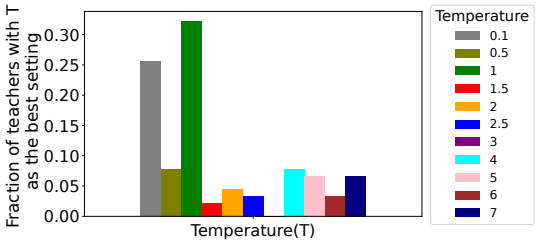

Figure 6. Occurrence of best choice of T for all possible combinations of participant pairs.

yields better learning and less forgetting outcomes, which is related to teacher selection, and we leave for future work. However, for a given pair of participants, adding a copy of the student model as an additional teacher for itself when using an unlabelled transfer set, Figure 3 (b), or using the student data as a transfer set with the CE loss in addition to the distillation loss, Figure 3 (c), can remarkably mitigate the possible forgetting and improve overall accuracy. Also note that the second approach, Figure 3 (c), is sometimes worse than the first approach, Figure 3 (b), (e.g, student participant 1 and in some cases student participants 3 and 6), which usually happens with weak student participants that have a poor pre-trained model accuracy and limited data. However, the right positioning of the models (i.e., which model is the student and which model is the teacher) does mitigate this issue and improve the overall performance, according to our experimental analysis. For example, changing the position of participant 1 from a student to a teacher when paired with participant 5 reduced the forgetting (of participant 1 classes) from 88% to only 16%. Similarly, changing the position of participant 1 from a student to a teacher when paired with participant 3 reduced the forgetting (of participant 1 classes) from 87% to only 13%.

Note that both proposed approaches are applicable for privacy-sensitive settings such as FL, as the participants never share their private datasets. For the rest of the paper, we continue our experiments using the student dataset as a transfer set, because suitable public datasets might not always be available in real scenarios [1] and require prior knowledge of participants' private data and careful consideration (i.e., to avoid the significant performance degradation caused by distribution variation between the public dataset participants' datasets) [18, 25].

Temperature Effect. Recall that in knowledge distillation, the student learns from the soft-target outputs of the teacher model (Eq. 1). Temperature T is usually set to 1. A higher value of T produces a softer probability distribution, while a lower temperature creates a sharper probability distribution over classes [10]. The effect of the different temperature choices is illustrated in Figure 4.

The scores in the logits can be thought of as an inherent similarity between the corresponding label and the input samples [10]. Thus, it is common to use temperatures higher than one in offline distillation literature [10], while in distributed distillation literature, it is common to set the temperature to 1 [5, 11, 12, 16, 17, 22].

However, note that in the distributed distillation setting with non-IID data, we might want to learn all or only part of what the teacher knows. We observe that temperature has an effect on controlling this. For example, in the case of a teacher model trained with data that involves most or all of the classes, the student would want to learn about what it knows about the other classes by setting a higher temperature. Instead, in the case of a teacher model with limited knowledge, the student would not want to learn misleading information, so a lower temperature should be chosen. Thus, unlike prior work that uses a static temperature, we show that the choice of the temperature in distributed distillation should be adapted to the context of the participants in the knowledge distillation process (in a nutshell, depending how much they already know).

We now illustrate that setting the right temperature can boost learning outcomes. Figure 5 shows the accuracy gain (after KD) of three different participants when learning different teachers (covering all possible teachers). The three

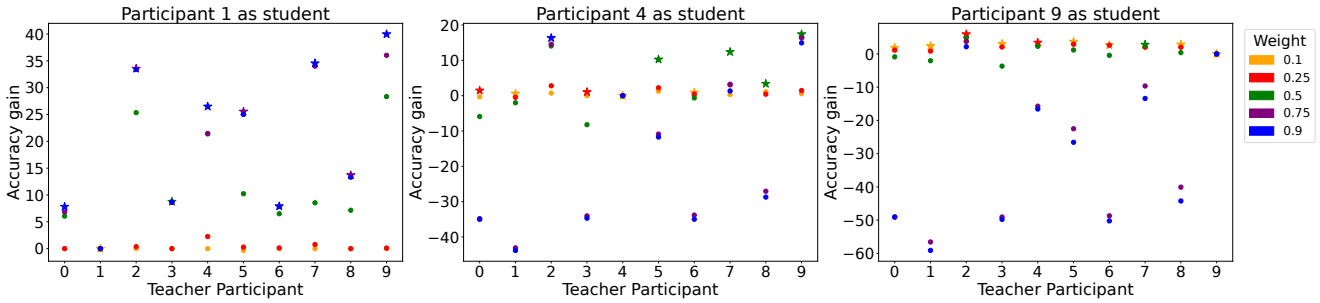

Figure 7. Setting an appropriate weight is important. The after-distillation accuracy gain for three participants learning from different teachers with all the weight settings.

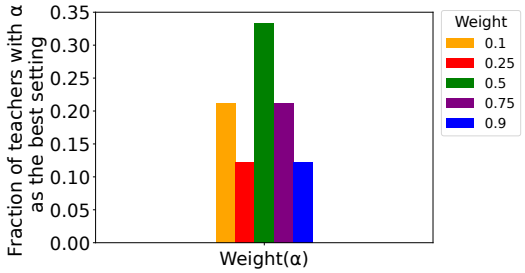

Figure 8. Occurrence of best choice of α for all possible combinations of participant pairs.

participants are chosen based on the accuracy of the pre-trained models: one with low accuracy, one with medium and one with high accuracy. The figure shows a boxplot depicting the distribution of accuracy gain while experimenting with all possible choices of T. The color of each boxplot bar illustrates the best choice of T (see the legend for the color scheme). The results clearly demonstrate that the choice of T affects the performance of the student model after distillation in all cases. As expected, there is more room for accuracy gain with a lower initial accuracy. We note that the accuracy gain is overall sensitive to T: a good choice can bring substantial gains, whereas a poor choice makes KD undesirable. Notably, this observation holds for all the three types of student participants (having low, medium or high initial pre-trained accuracy). In a few cases, the sensitivity is only modest. This happens when the models have close data distributions.

We also analyze whether a particular choice of T is overall good enough in most cases. Figure 6 quantifies the proportion of cases in which each particular value of T is the best one across all the possible combinations of participant pairs. The results indicate that about 2/3 of the pairs require T other than 1, which comes with significant improvement in accuracy gain (up to around 32 percentage points). We conclude that the choice of T should not be static in distributed joint distillation settings.

Our results further corroborate some previous findings from the offline distillation literature. Besides the effect of the participants' data distributions, prior work found that different datasets [10] and different model architectures [24] do affect the temperature choices. Thus, different temperatures might be optimal for different settings, and we need to find the best temperature for a given setting.

Finally, we posit that grid searching or manually tuning this hyper-parameter is impractical, particularly in the distributed training setting, where knowledge transfers may frequently happen with a wide range of participants. Hence, automating the process is necessary. Devising automatic tuning techniques is part of our ongoing work.

Weight Effect. In offline distillation, the student is usually trained using a linear combination of two losses (i.e., CE loss and KL diversions loss), as in Eq. 2, where the weight α is usually assigned statically. However, in joint distillation, where models are already pre-trained, different weights might work better for different pairs. Our analysis shows that the weight given to the teacher model (versus the student model when using the transfer set option 1 or versus the CE loss when using the transfer set option 2) can significantly impact the learning outcome.

Similar to the previous experiment with T, we analyze the accuracy gain for the same three different participants while varying α. Figure 7 shows the accuracy gain (after KD) across all possible teachers. Points marked as $\star$ indicate the best accuracy gains and are shown color coded with the value of α (see the legend for the color scheme). The results clearly indicate that the choice of α affects the performance of the student model after distillation.

Figure 8 shows the proportion of cases in which each particular value of α is the best one across all the possible combinations of participant pairs. Similar to the temperature analysis, the results indicate that 2/3 of the pairs require weight other than the default 0.5, in order to obtain a significant improvement. We also conclude that the choice of α should not be static in distributed joint distillation.

Dual Tuning (T and α). We now analyze the effect of tuning the temperature and the weight, one after another,

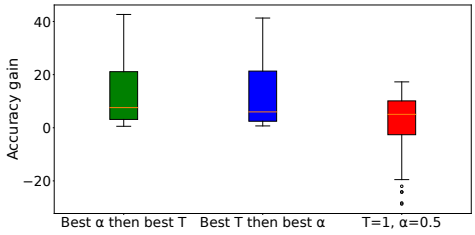

Figure 9. Accuracy gain with the two different approaches of dual tuning compared with the baseline (red bar).

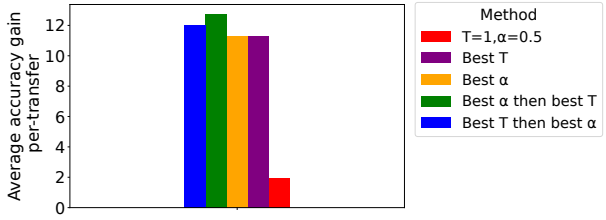

Figure 10. Average accuracy gain (per-transfer) with different tuning approaches.

to see how they together affect the learning process and to determine if the order matters and which order may achieve more gain.

Due to the large number of all possible KD configurations, we consider two orders for sequential tuning: (1) we first pick the best α for each pair, and then search for the best T; (2) we first pick the best T for each pair and then search for the best α.

Figure 9 presents the results. Each bar in the boxplot represents the accuracy gain from KD across all the participant pairs when tuning the α first, then the T (green bar), and the other way around (blue bar), compared with the baseline (red). The two experiments achieve similar results, with tuning the weight first being slightly better.

Figure 10 shows the average per-transfer accuracy gain with different tuning approaches. The results demonstrate a significant improvement (more than 5 times better) compared to the baseline (T=1 and α=0.5). Besides, it indicates that the sequential dual tuning outperforms the single parameter tuning by around 2 percentage points per transfer, on average. Considering the whole search space of the two-parameter combinations instead of the sequential tuning could possibly provide further improvement, which merits further investigation.

Position Effect. Another factor that can deliver further improvement is to appropriately set the position of participants (i.e., to be a student or teacher). Unlike the other distillation approaches where the student position is obvious (i.e., the untrained model in the offline distillation), in joint distillation models are already pre-trained, and we need to set the

With participant	Participant 1		Participant 4		Participant 9	
	As student	As teacher	As student	As teacher	As student	As teacher
0	8.22	11.70	3.31	-9.74	3.85	-13.47
1	-	-	1.26	-11.52	3.59	-17.87
2	42.65	67.16	19.86	24.37	7.09	8.92
3	9.59	10.51	2.79	-13.39	4.00	-11.27
4	32.27	45.05	-	-	4.46	0.30
5	28.61	33.97	8.91	0.07	4.21	-3.19
6	8.33	9.88	1.07	-12.57	3.86	-14.58
7	37. 10	48.02	10.57	10.50	3.30	-0.05
8	14.06	16.09	0.77	-4.92	3.91	-10.64
9	41.25	62.71	15.63	19.80	-	-

Figure 11. Setting the right position is important. The accuracy gain (ppt) of the participants at the top when being students versus when being teachers. The green-shaded cells represent the best positions.

right position for each model. Our analysis finds that the position can notably affect the learning process. If participant k wants to learn from participant j, it is not always optimal to set it as a student; it might be better as a teacher: participant k teaches participant j and adopts the resulting model.

Figure 11 shows examples of three participants learning from or teaching some other participants. The tables compare the accuracy gain in percentage points after tuning T and α when the participant listed at the top is a student versus when it is a teacher, paired with each of the rest of the participants. Clearly, the choice of position should depend on the characteristics of the pair participating in KD. The results also indicate that setting the right position remarkably affects the learning outcomes (up to around 25 percentage points).

5 Conclusion

Unlike offline and online distillation, employing knowledge distillation to transfer knowledge between already pre-trained models (i.e., joint distillation) requires careful consideration. Joint distillation can offer great communication reduction and model architecture flexibility in distributed training and FL settings, and we found that appropriately tuning some hyper-parameters can significantly improve learning outcomes and maximize the accuracy gain.

Acknowledgments

This publication is based upon work supported by King Abdullah University of Science and Technology (KAUST) under Award No. ORA-CRG10-2021-4699.

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. 2018. Large scale distributed neural network training through online distillation. arXiv:1804.03235 [cs.LG]

[2] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane.

2022. Flower: A Friendly Federated Learning Research Framework. arXiv:2007.14390 [cs.LG]
[3] Ilai Bistritz, Ariana Mann, and Nicholas Bambos. 2020. Distributed distillation for on-device learning. In *NeurIPS*.
[4] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*.
[5] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2019. Cronus: Robust and Heterogeneous Collaborative Learning with Black-Box Knowledge Transfer. arXiv:1912.11279 [stat.ML]
[6] Hong-You Chen and Wei-Lun Chao. 2021. FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning. arXiv:2009.01974 [cs.LG]
[7] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. 2021. Ensemble attention distillation for privacy-preserving federated learning. In *ICCV*.
[8] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021).
[9] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-Shot Federated Learning. arXiv:1902.11175 [cs.LG]
[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML]
[11] Li Hu, Hongyang Yan, Lang Li, Zijie Pan, Xiaozhang Liu, and Zulong Zhang. 2021. MHAT: An efficient model-heterogenous aggregation training scheme for federated learning. *Information Sciences* 560 (2021).
[12] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. 2021. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing* 22, 1 (2021).
[13] Shivam Kalra, Junfeng Wen, Jesse C. Cresswell, Maksims Volkovs, and Hamid R. Tizhoosh. 2021. ProxyFL: Decentralized Federated Learning through Proxy Model Sharing. arXiv:2111.11343 [cs.LG]
[14] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. University of Toronto.
[15] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. 2022. Fedscale: Benchmarking model and system performance of federated learning at scale. In *ICML*.
[16] Chengxi Li, Gang Li, and Pramod K Varshney. 2021. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal* 9, 2 (2021).
[17] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous Federated Learning via Model Distillation. arXiv:1910.03581 [cs.LG]
[18] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*.
[19] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data. In *NeurIPS*.
[20] Michael Matena and Colin Raffel. 2022. Merging Models with Fisher-Weighted Averaging. arXiv:2111.09832 [cs.LG]
[21] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. 2021. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems* (2021).
[22] Felix Sattler, Arturo Marban, Roman Rischke, and Wojciech Samek. 2021. Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Transactions on Network Science and Engineering* 9, 4 (2021).
[23] Jürgen Schmidhuber. 1991. *Neural sequence chunkers*. Inst. für Informatik.
[24] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. 2021. Does knowledge distillation really work?. In *NeurIPS*.
[25] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*.
[26] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. 2022. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 8 (2022).
[27] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *CVPR*.