Knowledge-Aware Federated Active Learning with Non-IID Data

Yu-Tong Cao[1], Ye Shi[2], Baosheng Yu[1], Jingya Wang[2], Dacheng Tao[1]
[1] Sydney AI Centre, School of Computer Science, The University of Sydney
[2] ShanghaiTech University
ycao5602@uni.sydney.edu.au, shiye@shanghaitech.edu.cn, baosheng.yu@sydney.edu.au,
wangjingya@shanghaitech.edu.cn, dacheng.tao@gmail.com

In this supplementary, we provide additional experimental details and visualizations that provide further insights into the main content presented in the paper. Furthermore, we discuss the limitations of our work and highlight possible future directions for improvement.

A. Additional Experimental Details and Visualizations

In this section, we provide additional details of our experiments in the paper to further support our main content. We include visualizations of data distributions, hardware information, and detailed result numbers to help readers better understand our experimental setup. Furthermore, we present results from extra ablation studies and on additional datasets to demonstrate the effectiveness and robustness of our proposed KAFAL approach. Moreover, we introduce a visualization that illustrates the mismatch of active sampling goals between the global model and the client models in federated active learning, highlighting the importance of knowledge-specialized sampling for better performance. We also provide a detailed demonstration of the Knowledge-Specialized KL-Divergence using a toy example, to help readers better understand this key component of our approach.

A.1. Experiment Details

The experiments are conducted with one NVIDIA GeForce GTX 1080 Ti GPU. Each client is trained for 40 epochs locally. The batch size is 128 and the learning rate $\eta = 0.1$. We run $T = 50$ communication rounds before active sampling and evaluation. $\boldsymbol{\beta}$ is sampled from $\text{Beta}(2, 2)$ and $\nu = 0.5$. We present the exact result numbers of our main results on CIFAR10 (Tab. 1) and CIFAR100 (Tab. 2). To present a complete picture of our KAFAL performance, we evaluate each client using the test set and show the results in Fig. 2. The non-IID data distributions used in our main results on CIFAR10 are show in Fig. 1(a).

Table 1. Detailed results on CIFAR10.

Method	10%	15%	20%	25%	30%	35%
Random	50.60	54.29	59.76	62.85	65.16	66.52
Core-Set		58.98	67.48	68.85	69.04	71.05
Entropy		58.45	65.76	68.61	69.59	71.68
Margin		58.19	63.50	66.75	68.66	71.13
LL4AL		57.48	60.87	63.79	65.31	66.94
QBC		58.45	62.10	65.81	66.49	68.22
BADGE		57.46	63.57	67.39	70.42	71.67
Alfa-Mix		56.75	61.90	64.98	66.57	67.81
KAFAL (ours)		60.88	67.47	70.82	72.94	74.60

Table 2. Detailed results on CIFAR100.

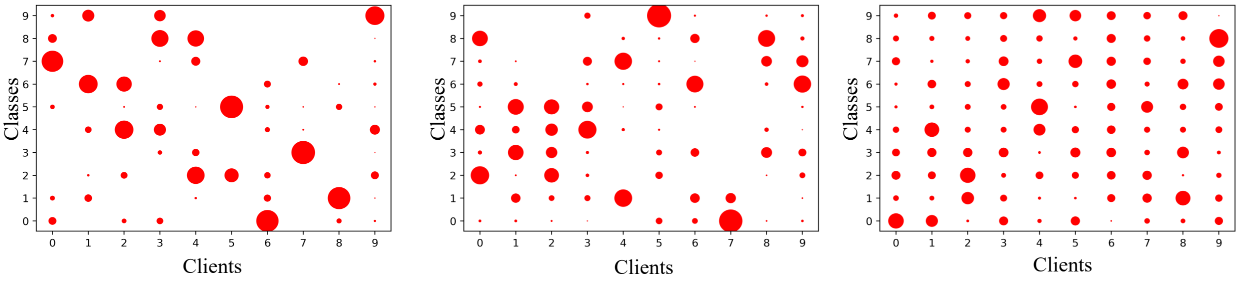
Method	10%	15%	20%	25%	30%	35%
Random	17.67	20.10	24.28	27.85	29.21	30.41
Core-Set		22.78	26.10	28.49	30.11	30.86
Entropy		20.79	23.48	26.41	28.07	30.01
Margin		22.65	25.50	28.56	29.77	30.88
LL4AL		22.18	24.05	27.14	27.99	28.41
QBC		22.41	24.86	27.15	29.95	30.39
BADGE		22.93	26.19	28.61	29.72	31.26
Alfa-Mix		21.14	24.54	27.79	29.15	30.56
KAFAL (ours)		23.63	26.13	28.89	30.79	32.04

A.2. Knowledge Specialization Alternatives

Given that knowledge specialization of KL-Divergence is achieved via score-level reweighting (as detailed in Eq. (1)-(3) of the paper) in our KAFAL, an interesting question arises: Can other reweighting techniques also enable knowledge specialization in federated active learning? To answer this question, we compare our method with two knowledge specialization alternatives, namely probability-level specialization and KL-Divergence-level specialization.

To conduct probability-level specialization, we can rewrite Eq. (1) as follows:

$$P^i_y(\boldsymbol{x}) = \frac{\exp\Big(\nu^\lambda_{i,y} \cdot g_y(\boldsymbol{x}; \boldsymbol{\omega}_i)\Big)}{\sum_{c \in \mathbb{C}} \exp\Big(\nu^\lambda_{i,c} \cdot g_c(\boldsymbol{x}; \boldsymbol{\omega}_i)\Big)}$$



(a) $\alpha = 0.1$ (b) $\alpha = 0.3$ (c) $\alpha = 1$

Figure 1. Illustration of CIFAR10 non-IID data distributions over clients with $\alpha = 0.1$, $\alpha = 0.3$, and $\alpha = 1$. The x-axes represents the client names. The y-axes represents the class labels. The dot sizes represent the number of data.

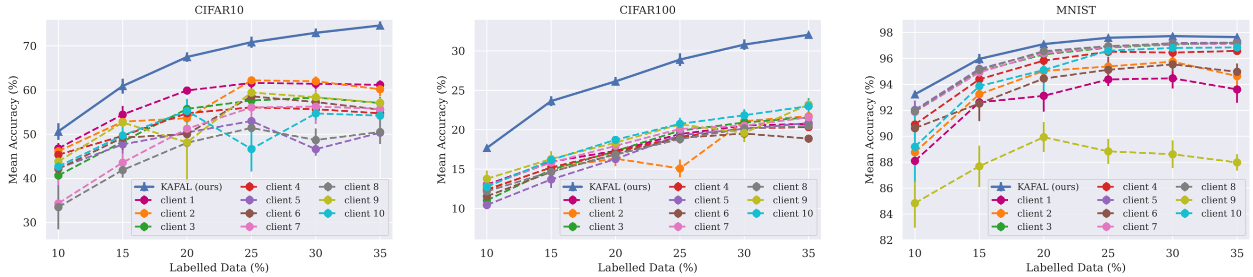


Figure 2. Evaluation on each client on CIFAR 10/100 and MNIST using KAFAL.

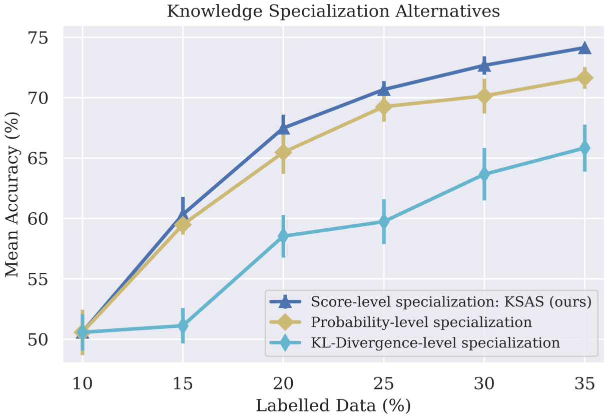


Figure 3. Results on CIFAR using different knowledge specialization techniques.

where $\nu_{i,y} = \frac{n_{i,y}}{\sum_{c \in \mathbb{C}} n_{i,c}}$ is the normalized knowledge weight. Note that we did not normalize the knowledge weight in our score-level knowledge specialization (KAFAL) because it can be easily proved that the results are equivalent with or without normalization. And similarly, Eq. (2) is replaced with:

$$Q_y^i(\boldsymbol{x}) = \frac{\exp\Big(\nu_{i,y}^{\lambda} \cdot g_y(\boldsymbol{x}; \boldsymbol{\Omega})\Big)}{\sum_{c \in \mathbb{C}} \exp\Big(\nu_{i,c}^{\lambda} \cdot g_c(\boldsymbol{x}; \boldsymbol{\Omega})\Big)}.$$

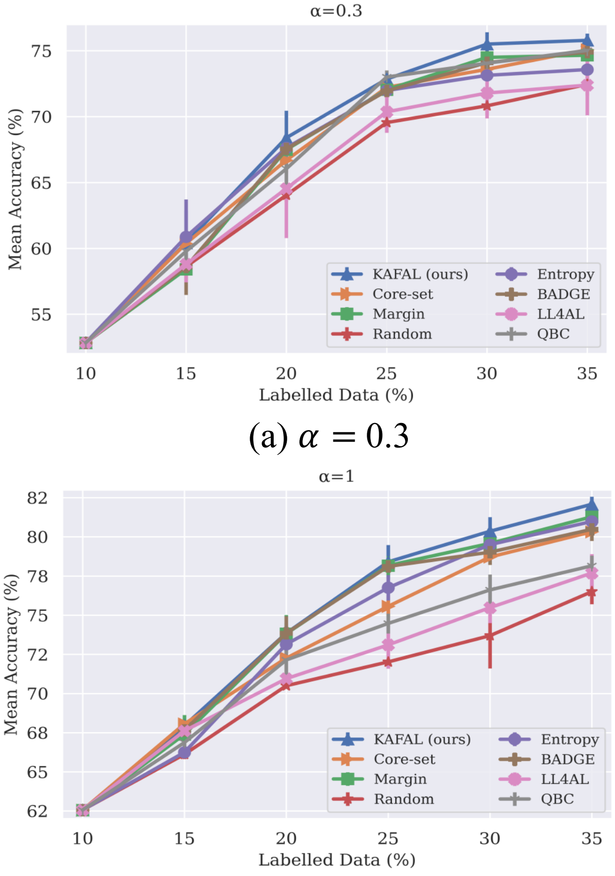
This knowledge specialization alternative still involves the computation of the KL-Divergence as described in Eq. (3). This knowledge specialization alternative reweights the logits during the calculation of the predicted probability, hence the name.

To conduct KL-Divergence-level specialization, we replace Eq. (3) with:

$$D^i(\boldsymbol{x}) = \sum_{y \in \mathbb{C}} \left[\nu_{i,y}^{\lambda} \cdot \left(P_y^i(\boldsymbol{x}) \ln \frac{P_y^i(\boldsymbol{x})}{Q_y^i(\boldsymbol{x})} + Q_y^i(\boldsymbol{x}) \ln \frac{Q_y^i(\boldsymbol{x})}{P_y^i(\boldsymbol{x})} \right) \right].$$

This knowledge specialization alternative reweights the sum while calculating KL-Divergence.

In Fig. 3, we present the results of the two alternatives as well as our KAFAL. The experimental results show that KAFAL outperforms both of the alternative methods. While probability-level specialization yields an acceptable outcome, KL-Divergence-level specialization fails to produce a reasonable result. One possible reason for this difference is that the probability-level specialization method, like our KAFAL, uses a moderate level of reweighting to adjust the results. In contrast, the KL-Divergence-level specialization method directly reweights the summation in the KL-Divergence calculation, potentially resulting in a stronger level of reweighting. Our score-level specialization approach may outperform probability-level specialization because reweighting the raw logits may not have a natural interpretation, whereas reweighting normalized results as in



(a) $\alpha = 0.3$

(b) α =1

Figure 4. Results from using $\alpha = 0.3$ and $\alpha = 1$ for the non-IID coefficient on CIFAR10.

our KAFAL can be interpreted as adjusting the likelihood of the results.

A.3. Different Non-IID Levels

We further explore federated active learning with the non-IID coefficient $\alpha = 0.3$ and $\alpha = 1$ on CIFAR10. The data distributions are shown in Fig. 1(b) and (c) respectively. We show the experiment results in Fig. 4. A larger α value provides less non-IID distributions for clients, i.e., the distributions across different clients are more similar. Unsurprisingly, compared to our CIFAR10 with $\alpha = 0.1$ results, the results are overall better for $\alpha = 0.3$ and $\alpha = 1$. Our KAFAL is still state-of-the-art, but the margins between the results of KAFAL and the rest methods are relatively smaller. This experiment demonstrates that our KAFAL is more competitive with higher levels of non-IID. It validates that intensifying knowledge-specialized data in KAFAL can handle the non-IID distributed data in federated active learning. The margins between Random and other methods become larger with larger α values, possibly because the mismatch problem in federated active learning becomes less significant with a lower level of non-IID in data. And the rest methods can benefit from the actively sampled data.

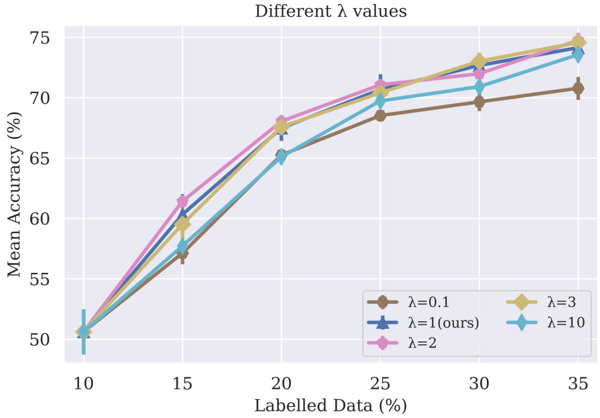


Figure 5. Results on CIFAR10 from using five different values of λ for the intensification of specialized knowledge in federated active learning with non-IID data.

A.4. Different Values of λ For Knowledge-Specialized Intensification

The coefficient λ in eq. (1)(2) controls the knowledge-specialized level in KSAS. With larger values of λ, the clients intensify more on their specialized knowledge in active sampling. As we stated in the paper, we simply use $\lambda = 1$ in our main experiments. Here we explore more values of λ on CIFAR10 and show the results in Fig. 5. For λ of values 1, 2, and 3, the difference is not significant. However, for more extreme λ values 0.1 and 10, the results are clearly poorer. Specifically, $\lambda = 0.1$ produces the worst results of the five. When the λ value approaches zero, the active sampling purely depends on the disagreement between the clients and the global model. The results gradually approach the results from using vanilla KL-Divergence in Subsec. 4.3.1 in the paper. When the λ value goes to infinity, the active sampling process almost ignores the less frequent classes and tries to compute the disagreement solely based on the most common class (or classes) of each client. Therefore, when applying KAFAL, the λ value should be neither too small nor too large.

A.5. Learning With More Decentralized Clients

In the paper, we explored federated active learning with $N = 10$ clients. To better analyze the problem, we run experiments on CIFAR10 with $N = 20$ and $N = 100$ while keeping the rest setup the same. The labelled data amount still starts with 10% of each local training set, meaning that with $N = 20$ the data available for each client is half of that

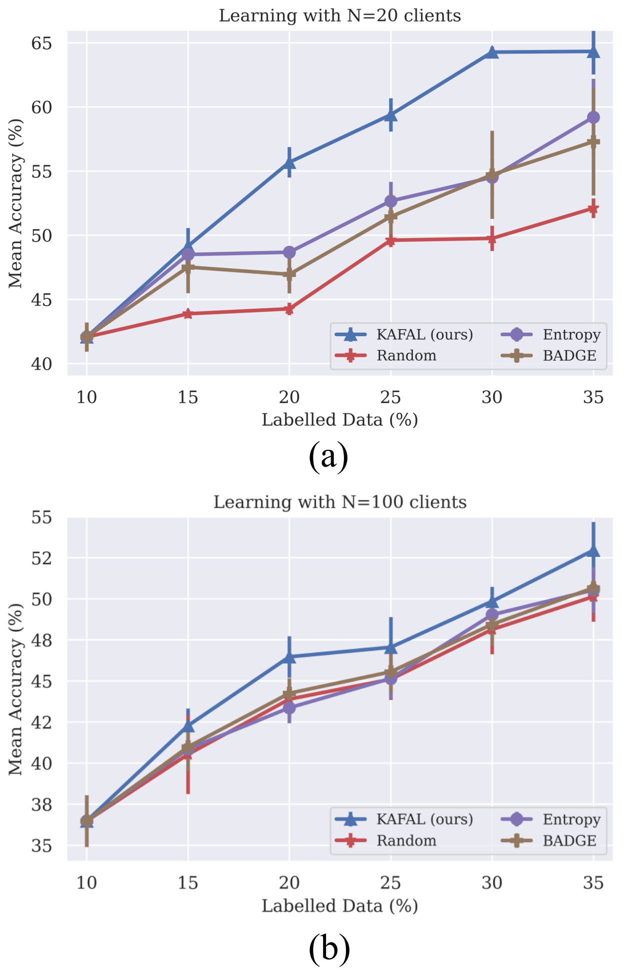


Figure 6. Results on CIFAR10 from using (a) $N = 20$ and (b) $N = 100$ clients in federated active learning with non-IID data.

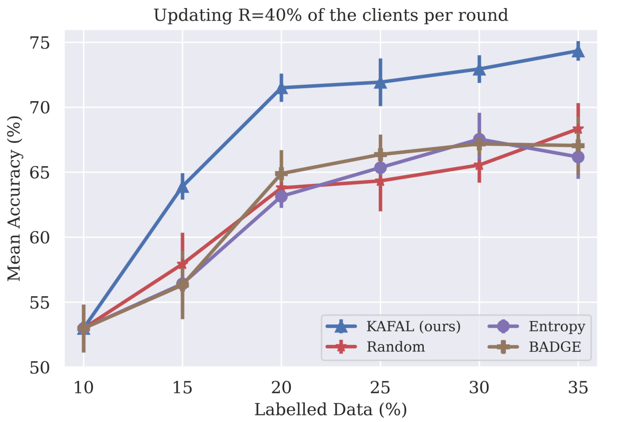


Figure 7. Results on CIFAR10 from updating 40% of clients per communication round in federated active learning with non-IID data.

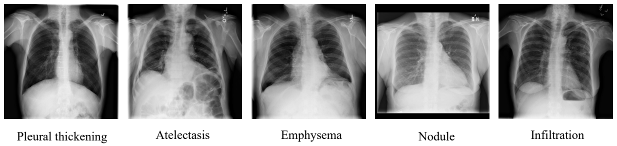


Figure 8. Selected images in NIH Chest X-Ray dataset.

in the previous experiments, and with $N = 100$ the data available for each client is only $\frac{1}{10}$ of that in the previous experiments. The results are shown in Fig. 6. Compared with the previous results from using $N = 10$ clients, results for all methods reduce due to the smaller local datasets for both $N = 20$ and $N = 100$. Our KAFAL still outperforms the rest methods by a clear margin. This shows the superiority of our method when more decentralized clients are involved in federated active learning. The result lines are more jiggly compared with previous results, the possible reason is that the fewer labelled data and the $T = 50$ communication rounds may not be enough for the convergence to be achieved. With $N = 100$ clients, the margin is less significant compared to using $N = 20$, this is possibly due to the extremely small local dataset size. Each local dataset starts with on average 500 images and adds about 250 images at each active round for $N = 100$. This also explains why Entropy and BADGE generate similar results compared with Random. The limited training data lead to poor classification ability and deteriorates the credibility of the model statistics.

A.6. A Smaller Ratio of Clients to Update per Round

We used $R = 80\%$ in previous experiments. To test how our KAFAL performs with a smaller ratio of clients updated in each communication round, we use $R = 40\%$ instead and present the results on CIFAR10 in Fig. 7. All the rest setup is kept the same. $R = 40\%$ means that only 40% of the clients are updated in each communication round. Surprisingly, our KAFAL performs even better using $R = 40\%$ compared with using $R = 80\%$, while results from the rest methods all drop. This is possible because our KAFAL compensates for the knowledge of clients with the global model using KCFU along with actively sampling data by intensifying specialized knowledge using KSAS. The two together enable a faster convergence in global aggregations. Using $R = 40\%$ means each client is trained less compared to using $R = 80\%$ when the communication rounds T is fixed. The rest methods which still actively sample harder data that are likely from less frequent classes cannot utilize these data in training with the smaller R value. Although KCFU is also used for other methods for a fair comparison, it cannot be fully utilized without the knowledge-specialized intensification of KSAS.

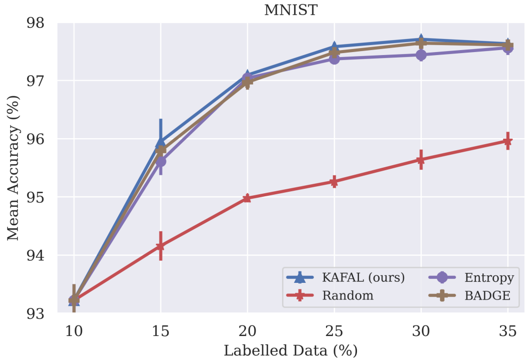


Figure 9. Results on MNIST in federated active learning with non-IID data.

A.7. Medical Image Classification

We further conduct experiments in a more realistic scenario of X-ray image classification using NIH Chest X-Ray dataset [4]. Some examples are shown in Fig. 8. The task is to categorize thorax diseases using chest X-ray images. The dataset consists of more than 112k images of size 1024 × 1024. We follow the official training and testing splits. And we exclude images tagged with 'no findings'. The rest data have 14 for different thorax diseases as labels. The training split includes 36024 images and the testing split includes 15735 images. We use ResNet-50 [1] as the backbone of the clients and the global model. We still use $\alpha = 0.1$ as the non-IID coefficient to distribute the client data. 5 clients are used, and 80% are selected for the update at each communication round. We start with 10% labels and use 5% of the whole dataset as the budget. We train for 2 epochs in each communication round with learning rate $\eta = 0.0005$ and run 5 communication rounds before sampling. The mean AUC score is used to evaluate each method's performance. The results are presented in Tab. 3. We compare with four baseline methods (Random, Core-Set, Entropy, and Margin) that the dataset can easily fit in considering the image size and model size. Our KAFAL still achieves state-of-the-art results on this dataset.

Table 3. mAUC scores on NIH Chest X-Ray dataset.

Method	10%	15%	20%
Random	56.12	60.62	62.77
Core-Set		62.55	63.24
Entropy		63.13	63.80
Margin		60.19	62.81
KAFAL (ours)		63.61	64.48

A.8. Results on MNIST

We also run experiments on MNIST [2]. MNIST is a 10-class image dataset that contains handwritten images of 10 digits. We use the MNIST 2NN proposed by McMahan et al. [3] as the clients' and the global model's architecture. We train for 10 epochs each communication round and we repeat for 10 communication rounds. We split the MNIST dataset with $\alpha = 1$. The results are shown in Fig. 9. This is a fairly simple dataset, so all the results are quite high. But random is still far behind compared to the other methods. On this dataset, our KAFAL still outperforms the other methods, but with a quite small margin.

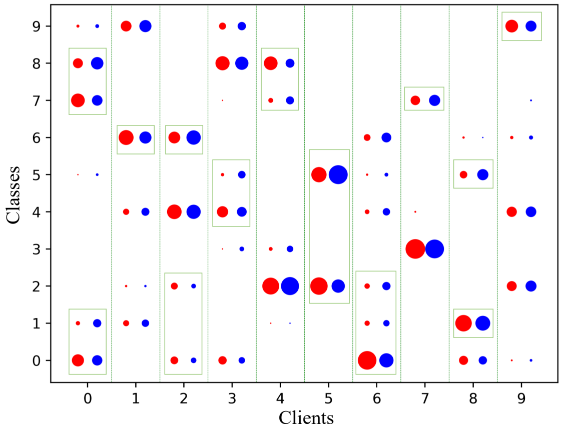


Figure 10. An example of the sampling goal mismatch between the global model and the clients. The green bounding boxes highlight class distributions that are clearly different for active sampling results using the global model (red) and active sampling results using the client model (blue).

A.9. Visualizing the Mismatch Problem

In the paper, we mentioned that the main challenge of federated active learning is the mismatch between the active sampling goal of the global model on the server and that of the asynchronous local clients. To demonstrate this problem with an experiment, we actively sample with the global model and the clients respectively and show the class distributions of the sampled data in Fig. 10. We use the same sampling method Core-set for both the clients and the global model for a fair comparison. With the bounding boxes, we show the differences between the sampling results. The original data distributions on clients with $\alpha = 0.1$ are shown in Fig. 1(a). Also, note that this figure only shows the class distributions. If we further consider specific data points within each class, the difference in sampled results will be more significant.

A.10. Demonstration of Knowledge-Specialized KL-Divergence in a Toy Example with Details

To better visualize how Knowledge-Specialized KL-Divergence intensifies specialized knowledge compared to KL-Divergence, we use continuous distributions to simulate model predictions and compute the divergences (Fig. 11).

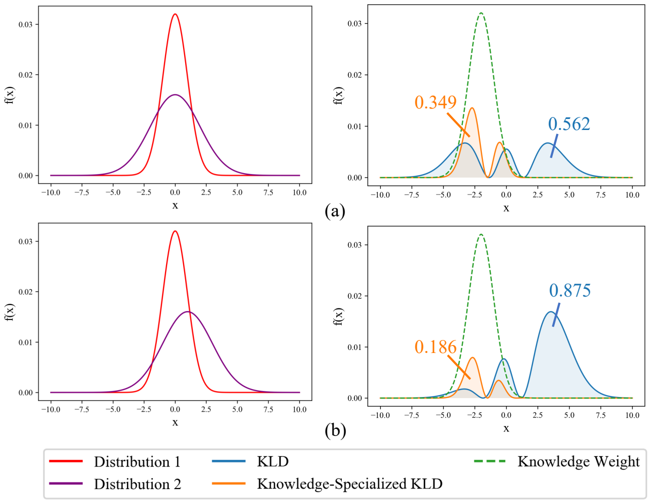


Figure 11. Illustration of how Knowledge-Specialized KL-Divergence intensifies specialized knowledge compared to standard KL-Divergence. On the left, we show two distribution curves. On the right, the blue and orange lines integrate to be KL-Divergence and the knowledge-specialized KL-Divergence computed from the left distributions. The blue and orange numbers show the integrated areas of the blue and orange curves in each image, respectively.

Note that the knowledge weight curves serve as a continuous version of our knowledge weights. In the figure, we present the distribution curves on the left and the corresponding KL-Divergence and Knowledge-Specialized KL-Divergence curves on the right, which have been calculated accordingly. The KL-Divergence curve is formulated as:

$$p(x)\ln\frac{p(x)}{q(x)}+q(x)\ln\frac{q(x)}{p(x)},$$

where $p(x)$ and $q(x)$ are the two distribution functions (presented on the left of Fig. 11). The KL-Divergence value is obtained by integrating this function with respect to x. The Knowledge-Specialized KL-Divergence curve is formulated as:

$$p_w(x)\ln\frac{p_w(x)}{q_w(x)}+q_w(x)\ln\frac{q_w(x)}{p_w(x)},$$

where $p_w(x)=\frac{w(x)\cdot p(x)}{Z_p}$ and $q_w(x)=\frac{w(x)\cdot q(x)}{Z_q}$. The normalization constants $Z_p=\int p(x)\cdot w(x)\mathrm{d}x$ and $Z_q=\int q(x)\cdot w(x)\mathrm{d}x$. The weight curve $w(x)$ is shown with green dashed lines in the figure. The Knowledge-Specialized KL-Divergence value is obtained by integrating this function with respect to x. The right-hand side of Fig. 11 can be viewed as global-local discrepancies from two different inputs on the same client model since the KL-Divergence values are different and the knowledge weights are the same. On the left-hand side, distributions 1 and 2 simulate the outputs of the client model and the global model. Notably, while (a) has a smaller KL-Divergence, its Knowledge-Specialized KL-Divergence is larger, suggesting it is less likely to be sampled than (b) if KL-Divergence is the sampling criterion. However, using our proposed Knowledge-Specialized KL-Divergence, (a) is more likely to be sampled than (b). This difference in sampling results is due to the knowledge weight, which intensifies the client's specialized knowledge while dampening the contribution of unfamiliar knowledge. Importantly, in (a), more of the model difference arises from specialized knowledge (as indicated by the peak area of the knowledge weight) compared to (b).

B. Limitations and Future Work

Our federated active learning paradigm KAFAL includes KSAS, a novel active sampling method to sample informative data using intensified discrepancies between the server and clients based on the specialized knowledge of each client, and KCFU, a federated update method to deal with data heterogeneity by compensating weak classes with the help from the global model. Although the experimental results demonstrate that KAFAL can perform well on the federated active learning task, we also want to highlight the potential drawbacks of this method. In KSAS, the specialized knowledge is extracted based on the class distributions of labelled local data. We may explore other ways to find a more comprehensive solution to represent the specialized knowledge, either, possibly not only considering the class distributions but also taking the training dynamics into account. In KCFU, the compensation is achieved through sampling the unlabelled data and then weighting them using the class distributions. Unfortunately, the data from weak classes may not be enough even though we include the unlabelled data. We may utilize the data generation techniques to generate more weak-class data for better knowledge compensation in the future. In addition to the potential drawbacks mentioned, another area for future work is to extend KAFAL to handle the case of long-tailed distribution in the federated active learning setting. In a long-tailed scenario, the local data can distribute globally long-tailed with some classes being rare for all clients. To consider active learning in such a scenario, additional resampling techniques and an improved version of knowledge-specialized KL-Divergence that takes the long-tailed distribution into account need to be included.

References

[1] 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*, pages 770–778, 2016. 5

[2] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 5

[3] 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*, pages 1273–1282. PMLR, 2017. 5

[4] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, M Bagheri, and R Summers. Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE CVPR*, volume 7, 2017. 5