FedRAL: Cost-Effective Distributed Annotation via Federated Reinforced Active Learning

Yannis Lazaridis, Anestis Kastellos, Athanasios Psaltis, and Petros Daras

Centre for Research and Technology Hellas, Thessaloniki, Greece

{lazaioan, kastellosa, at.psaltis, daras}@iti.gr

***Abstract*—This paper addresses the challenge of reducing annotation costs in distributed learning environments, particularly in systems with limited data and computational resources, such as those found in edge devices. We propose Federated Reinforced Active Learning, a framework that integrates Federated Learning with Reinforced Active Learning to optimize data labeling under strict cost constraints. The method is designed for small-scale networks where data is sparse, and minimal training epochs are available. By utilizing reinforcement learning within active learning, the system selects the most informative data samples, allowing for efficient training while significantly reducing the need for extensive annotations. This approach is particularly suited for environments where minimizing both annotation and computational costs is critical, such as in applications where cost efficiency and resource limitations are top priorities. The proposed method is evaluated on the CIFAR-10 and CIFAR-100 datasets using ResNet18, across 5 and 10 clients. Results demonstrate that the method significantly reduces annotation costs and improves learning outcomes, making it an ideal solution for cost-sensitive distributed systems.**

***Index Terms*—Active learning, Reinforcement Learning, Federated Learning, Distributed Learning, Annotation Budget.**

I. Introduction

In recent years, artificial intelligence (AI) has become predominant in many fields. Algorithms powered by deep neural networks have achieved remarkable success across numerous domains, and the demand for more complex frameworks capable of solving and generalizing over a wide range of cases continues to grow. This need for creating powerful AI systems is directly correlated with the requirement for more annotated data, especially for deep networks that must learn complex relationships. However, it has been observed that human annotators are unable to meet the data demands of these types of systems. Consequently, Active Learning (AL) has become central to machine learning (ML), and researchers are continually exploring new methods to minimize the labor involved in data annotation. At the same time, the growing number of edge devices, such as smartphones and IoT sensors, has created a need for methods that can not only process but also annotate data at the edge, where it is generated. This is crucial for managing the large amounts of data these systems produce, without depending on centralized infrastructure, which may not work well in environments with limited resources.

AL addresses the challenge of labeling large datasets by selecting the most informative samples for annotation, thus reducing the total labeling effort. While AL effectively reduces annotation costs, the need for further optimization becomes apparent in resource-constrained environments, where reducing both data annotations and computational effort is critical.

Building on AL, Reinforced Active Learning (RAL) incorporates Reinforcement Learning (RL) to optimize the selection of informative samples dynamically. RAL adapts the annotation process by using RL to identify which data points are most uncertain, ensuring that only the most critical samples are labeled. This leads to further reductions in annotation costs by preventing redundant labeling and focusing on the most impactful data points.

On the other hand, Federated Learning (FL) offers a distributed approach to model training, where multiple clients collaborate to build a global model without sharing their raw data, thus preserving privacy. FL has emerged as a promising framework for distributed learning in privacy-sensitive environments, such as edge devices or IoT networks. Integrating RAL into an FL framework introduces several unique challenges. First, determining the optimal batch for labeling on each client, especially when data distributions and model uncertainty vary across clients, remains an open problem. Second, efficiently aggregating models trained with varying batches in a federated environment is complex, as the local adaptations might lead to inconsistencies in the global model. Third, communication efficiency and privacy must be balanced, ensuring that frequent model updates do not overwhelm the network or expose sensitive information.

Problem Statement: To address the limitations of both AL and FL, we introduce Federated Reinforced Active Learning (FedRAL), a method specifically designed for distributed annotation. In FedRAL, each client runs a RAL algorithm independently, using RL to select the most valuable batches for annotation based on model uncertainty. This allows each client to focus on the most informative data, reducing unnecessary annotations. Specifically, the proposed method aims to:

- **Minimize Annotation Costs:** In systems with tight cost constraints, it is crucial to reduce the number of labeled samples required for model training without sacrificing performance.
- **Preserve Privacy:** FL allows clients to collaboratively train models while keeping their data local, ensuring that sensitive information is not shared.

- **Adapt to Resource-Constrained Environments:** Many applications, such as those involving edge devices, have limited processing power, memory, and bandwidth. Training models in these environments requires a strategy that reduces both the number of annotations and the number of training epochs.

FedRAL integrates these elements to create an efficient system for FL in distributed environments, where cost, privacy, and resource constraints are top priorities. The effectiveness of this approach is demonstrated through experiments on image classification tasks, showing that it achieves remarkable performance with minimal annotations and training epochs, making it particularly suitable for systems with stringent resource and cost limitations.

II. Related Work

A. Active Learning

AL is an ML technique that reduces labeling costs by selecting only the most informative samples for annotation. This approach is particularly beneficial when labeled data is limited or expensive to obtain. In uncertainty-based AL [19]–[21], the model queries uncertain samples for labeling, allowing it to focus on data points that are likely to improve its performance.

Hybrid methods [22], [29], [66] combine various strategies to achieve improved results by utilizing the strengths of different approaches. Interpolation-based AL selects samples that are located near labeled ones, enhancing the model's ability to generalize from its existing labeled dataset. Furthermore, batch methods [23] concentrate on efficiently labeling groups of data, which can significantly reduce the total annotation cost.

Despite the involvement of human annotators, optimization techniques [24] aim to lower costs while maintaining high performance, demonstrating the potential of AL to make the labeling process more efficient and cost-effective.

B. Reinforced Active Learning

RL enhances AL by selecting the most informative samples for labeling, making it a powerful tool in various applications. This technique finds utility in areas such as natural language processing, image classification, and meta-learning [1], [5].

Deep Reinforcement Learning (DRL) plays a significant role in image selection processes, stream-based AL, and refining models specifically for tasks like person re-identification [6], [8], [9], [11]. Recent advances in this field have expanded its applications to critical areas such as medical image analysis and multi-agent systems, demonstrating its versatility and effectiveness [12], [14], [16].

Moreover, RL-based AL is particularly beneficial in addressing challenges such as cost reduction and class imbalance, which are common in real-world datasets. This approach is being actively explored in domains like molecular design and efficient classification, highlighting its potential for innovation and improvement in various scientific and industrial fields [17], [18].

C. Federated Learning

FL is an emerging paradigm for decentralized training that enables the creation of a unified model generalized across numerous clients, each containing heterogeneous types of data. By allowing individual clients—often operating on different devices—to train a shared model locally on datasets exclusive to them, FL addresses the limitations associated with centralized data collection. A key advantage of FL is its ability to maintain data privacy, as raw data remains on the clients and only model updates, such as weights or gradients, are communicated to a central server. Aggregating these updates using algorithms like Federated Averaging (FedAVG) [25] results in a global model that encapsulates the collective knowledge derived from all participating clients without compromising individual data privacy. Furthermore, integrating various neural network architectures into FL frameworks has enhanced the potential of federated training, enabling more complex and effective models to be trained in a decentralized manner. In particular, the application of representation learning techniques within FL systems, especially when combined with contrastive learning, has demonstrated promising results by utilizing the strengths of both methodologies to improve model performance, especially in scenarios where clients possess significantly different data distributions. A framework in [48] achieves this by creating unified representations while keeping local models. The work of [49] shows that contrastive learning boosts local client performance, even with imbalanced and scattered data. FedX [50] uses contrastive learning to handle diverse data without sharing features, improving performance even with limited data. FedCA [51] aligns client representations with a public model to ensure consistency.

D. Federated Reinforcement Learning

Federated Reinforcement Learning (FRL) continues to be a promising approach, integrating the strengths of FL and RL to address key challenges such as privacy, scalability, and resource efficiency [62]–[65]. FRL enables collaborative learning across distributed devices while keeping data localized, thus preserving user privacy and improving efficiency across applications like network optimization, edge computing, and security [20], [52], [70]. Methods such as weighted averaging [25] and advanced algorithms like PPO [26] and SARSA [71] have been successfully applied to heterogeneous and dynamic environments [22], [24].

Recent works have explored FRL across various domains. Personalization [52], [53], resource efficiency [54], [55], and heterogeneity [56], [57] have been key focuses, with solutions optimizing tasks such as offloading and handling diverse client data. Multi-agent systems [58], robotics [59], [61], and UAVs [60] have also benefited from FRL approaches.

E. Federated Active Learning

Federated Active Learning (FAL) has emerged as an important approach to address the challenges of data annotation in FL systems [76], [77], [80]. [78] propose FEDALV, which combines AL with federated domain generalization, enabling image classification on unseen target domains with minimal data annotation from clients. [81] introduce LoGo, a sampling strategy that integrates global and local models to handle class diversity and data imbalance across clients, outperforming existing AL strategies. [79] apply FAL to medical image analysis, reducing annotation needs while maintaining high performance in skin-lesion classification. Another method by [75] integrates AL into FL to improve annotation efficiency, demonstrating superior performance compared to random sampling in image classification.

Building upon these foundational efforts, FedRAL further improves FAL by integrating RL. This innovation optimizes the learning process, enhancing privacy and data efficiency, and enabling FAL to be more robust in dynamic and heterogeneous environments.

III. Proposed Method

In distributed environments, where data is spread across various clients, the proposed approach provides an efficient method for distributed annotations by employing two key mechanisms:

- **AL via RAL**: RAL dynamically selects the most informative and uncertain data samples for annotation, allowing each client to focus on labeling only the most valuable data. This significantly reduces the number of samples that need to be annotated, thereby minimizing the overall annotation cost.
- **FL**: FedRAL uses FL to aggregate locally trained models into a global model, ensuring that the benefits of annotated data from one client are shared across all clients. This prevents redundant annotation, as other clients can utilize the insights from the global model, further reducing the annotation efforts.

Fig. 1 provides an overview of the FL process combined with RAL for batch selection across three example clients. Each client operates independently on its local dataset, utilizing its own classifier and the DQN agent to actively select informative batches of data for training.

FedRAL delivers a solution designed for systems where annotation costs are a priority. By using RAL to intelligently select batches of data and utilizing FL to distribute the annotation workload across clients, our method optimizes budget selection in real-world applications. This approach is particularly useful in environments where each client operates under strict budget constraints, such as in medical diagnostics or industrial inspections, where annotation costs and computational resources are closely managed.

A. Batch Selection with Reinforcement Learning

At the client side, we employ a RAL method to enhance the annotation process by concentrating on the most informative batches. The framework modifies the batch itself by trying different batches based on model uncertainty and performance, ensuring efficient resource use while maintaining accuracy. An RL agent drives this approach, treating the annotation task as a sequential decision-making problem. At each step, the agent observes the current state, and decides on the optimal batch.

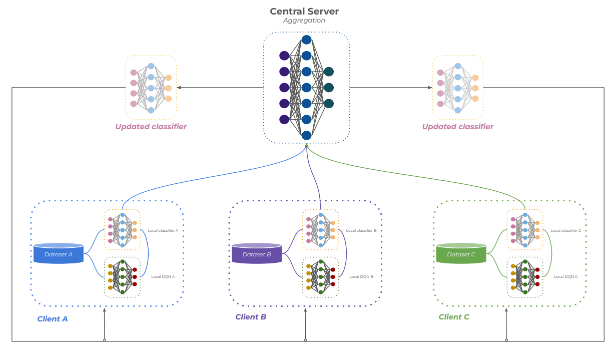

Fig. 1. Overview of the FedRAL process. Three clients each possess individual datasets, classifiers, and DQN agents. The DQN agent determines the optimal batch samples to annotate in each AL run. After each training round, the classifier weights, reflecting the local learning, are sent to the central server for aggregation. The aggregated classifier is then redistributed to the clients to guide the next FL as well as RAL round.

1) State Representation: The state at each iteration is represented by the margin scores of the current unlabeled samples. The margin score $m(x_i)$ for a sample x_i is defined as the difference between the highest and the second-highest predicted class probabilities:

$$m(x_i) = P(y^*|x_i) - P(y^{**}|x_i) \tag{1}$$

where $P(y^*|x_i)$ is the probability of the most likely class and $P(y^{**}|x_i)$ is the probability of the second most likely class. The RL agent uses these margin scores to assess the uncertainty of the samples.

2) Action and Reward: At each iteration t, the agent selects an action a_t, which corresponds to a batch b_t. The agent aims to select batches that maximize the improvement in model performance, while minimizing the annotation costs. The reward function r_t is defined as:

$$r_t = P_{t+1} - P_t \tag{2}$$

where P_t is the precision achieved at iteration t.

3) Warm-Start Episodes: To initialize the RL agent, warm-start episodes are conducted in which the agent tests different batches of samples, both smaller and larger, to observe their impact on model performance. These episodes assist the agent in identifying an appropriate starting point for batch selection. A warm-start episode concludes when the agent experiences two consecutive declines in rewards, ensuring that it does not waste annotation resources.

4) Post-Warm-Start / Batch Selection: After the warm-start episodes, the agent moves to the main AL process, where it selects batches based on its past experiences. In each annotation round, the agent assesses the current state (margin scores of unlabeled data) and chooses a batch b_t expected to improve precision with minimal cost.

5) End of Learning: Termination Criteria: The AL process continues until a predefined convergence criterion is met. The stopping conditions include:

- **Target Precision:** The agent continues selecting batches and annotating samples until the model reaches a target precision level, established during the warm-start episodes.
- **Annotation Budget:** The process ends when the agent has exhausted the predefined annotation budget, ensuring that the AL process remains within resource constraints.

B. Federated Learning Process with Active Learning

The FedRAL framework uses FL to allow several clients (such as mobile devices or decentralized organizations) to collaboratively train an ML model without needing to share their private, local datasets. Each client applies the RAL strategy to choose which data points should be labeled. At each FL round, clients independently train their own local models using their data. The model being aggregated across the clients is the classification model (referred to as the global classifier), which is responsible for making predictions, generating performance metrics like precision scores, and determining the current state for the RL agent. This state information is crucial for guiding the agent's decision-making process, allowing it to optimize batch selection effectively based on model performance.

1) Detailed Federated Learning Steps for Each Client: The proposed FL architecture consists of a central server and multiple clients, each holding local data. The server distributes the global model to the clients, aggregates their local model updates, and redistributes the updated global model for continued training.

Once all clients complete their local training, the server aggregates their model updates using the FedAvg algorithm. For each layer of the model, the parameters are averaged across clients, weighted by the number of samples at each client, as follows:

$$\theta_{\text{global}} = \frac{1}{N} \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \theta_n \tag{3}$$

where θ_n represents the parameters of client n, and $|\mathcal{D}_n|$ is the size of the dataset held by client n.

In the global setup, the dataset is distributed among N clients, and each client receives a unique portion of the dataset. The FL process unfolds in iterative rounds, as shown in Algorithm 1. Each round involves a series of key steps, both on the server and the clients' side.

This iterative process continues for several rounds until the global model converges. The architecture ensures privacy preservation by keeping the raw data decentralized, with only model updates being shared between the clients and the central server.

2) Federated Learning with Reinforced Active Learning Integration: Each client autonomously applies RAL to intelligently select and label the most informative data points. The critical feature of this approach is that each client employs

Algorithm 1 Federated Learning Process

Initialize global classification model c_0 and set number of clients K
for each federated round $t = 1, 2, \ldots, T$ **do**
  Model Distribution: The central server distributes the global classifier c_{t-1} to all clients.
  for each client $k = 1, 2, \ldots, K$ **in parallel do**
    Load local dataset D_k for client k.
    Initialize local model $c_k^t = c_{t-1}$.
    Perform AL to select batch b_k^t using RL.
    Train local model c_k^t on selected batch of data from D_k.
    Send updated model weights c_k^t to the central server.
  end for
  Model Aggregation: The central server aggregates the updated model weights from all clients using FedAvg:

$$c_{\text{global}} = \frac{1}{K} \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} c_k^t$$

  Global Update: The server redistributes the aggregated global model c_{global} back to the clients for the next round of training.
end for

its own Deep Q-Network (DQN) [38] agent for AL, meaning that each client develops a unique RL policy for data selection, while only the classifier model weights are shared and aggregated by the central server. The DQN weights, which guide the AL decisions, remain local and independent across clients. However, at each federated learning round, the global aggregated model influences the policy update by redefining the reinforcement learning state and its associated metric.

In Algorithm 2, the entire process is broken down into several key components and phases, all of which work together to ensure efficient training and data selection across the distributed network of clients.

IV. Experiments

In this section, the performance of the proposed FedRAL framework is evaluated on two commonly used benchmarks: CIFAR-10 [72] and CIFAR-100 [72]. The results are presented in terms of the annotation budget, accuracy, and precision. Both mean and maximum values are reported for each experiment, with configurations of 5 and 10 clients.

A. Experimental Setup

Datasets: We use the CIFAR-10 and CIFAR-100 datasets. Each dataset consists of 60,000 32x32 RGB images, with 50,000 used for training and 10,000 for testing. CIFAR-10 has 10 classes, while CIFAR-100 has 100 classes. For each dataset, the evaluation data comprise 20% of the total training data.

Data Partitioning: The experiments are conducted using the following partitioning scheme:

Algorithm 2 Federated Learning with Reinforced Active Learning

Initialize global classifier model c_0 at the central server.
Set number of clients K and distribute unique data partitions $D_1, D_2, \ldots, D_K$ to each client.
for each federated round $t = 1, 2, \ldots, T$ **do**
  Model Distribution: The central server distributes the global classifier model c_{t-1} to all clients.
  for each client $k = 1, 2, \ldots, K$ **in parallel do**
    Dataset Partitioning: Client k loads its local dataset D_k.
    Warm-Start Phase: Each client begins by training the initial classifier model on a small, labeled subset of its local data. This phase ensures that the model has a foundation to make informed predictions during the subsequent AL phases.
    Agent Phase with DQN: After the warm-start phase, each client uses its own DQN agent to guide the batch selection process. The DQN agent interacts with a simulated environment $\mathcal{E}$, which reflects the current state of the classifier model and data distribution. The state provided to the DQN reflects the model uncertainty.
    Batch Selection: In each round, the DQN agent selects a batch of unlabeled data from the local dataset for annotation. The agent's goal is to maximize the classifier's performance (measured by precision). The DQN receives rewards based on how much precision is improved relative to the selected batch, optimizing the agent's decision-making over time.
    Local Training: Using the selected and labeled batch, client k trains its local classifier model c_k^t. The classifier is updated based on the new annotations, improving the model's predictions on the unlabeled data.
    Weight Update: After training, the updated classifier model weights c_k^t are sent back to the central server. It is important to highlight that only the classifier model weights are communicated—the DQN weights remain local and are never shared or aggregated. This ensures that each client maintains its own policy for data selection.
  end for
  Model Aggregation: The central server aggregates the classifier weights from all clients using the FedAvg algorithm, which combines the local updates into a global classifier model:

$$c_t = \frac{1}{K} \sum_{k=1}^{K} c_k^t$$

  Global Model Update: The aggregated global classifier model c_t is redistributed to all clients, serving as the starting point for the next round of local training and RAL.
end for

- 10% of the training data are used as state data for the DQN agent.
- 10% of the training data are used as warm-start data.
- 60% of the training data are used for DQN training.
- 20% of the training data are used as evaluation data.

Classifier: We use a ResNet-18 [74] model pre-trained on ImageNet [73] as the classifier for both CIFAR-10 and CIFAR-100 datasets.

DQN: We implement RL using a non-linear Q-function approximation based on DQN [38]. Key techniques include a target network with a slow update rate of 0.01 [38], a replay buffer of size 50,000 [38], and Double DQN to reduce overestimation bias [38]. Prioritized Experience Replay [38] is used, with a prioritization exponent of 3 to balance exploration and prioritization.

BatchAgent Parameters: The BatchAgent, which is used for the RAL method in each client, is trained for 25 epochs, with 5 episodes per epoch and 100 updates per episode.

B. Results on CIFAR-10

Table I presents the performance of the proposed FedRAL framework on the CIFAR-10 dataset with configurations of 5 and 10 clients. The results highlight the efficiency of the method in balancing annotation budget, accuracy, and precision across multiple clients in an FL environment.

TABLE I
RESULTS FOR CIFAR-10 WITH 5 AND 10 CLIENTS USING THE FEDRAL FRAMEWORK. THE TABLE REPORTS THE MEAN AND MAXIMUM VALUES FOR THE ANNOTATION BUDGET (AS A PERCENTAGE OF THE TOTAL EVALUATION DATA), THE ACCURACY, AND THE WEIGHTED AVERAGE PRECISION.

	5 clients		**10 clients**	
	mean	***max***	***mean***	***max***
Budget	3.45%	6.39%	2.65%	3.99%
Accuracy	49.10%	50.07%	55.98%	56.96%
Precision	50.57%	51.69%	56.82%	57.80%

The FedRAL framework effectively enhances annotation budget efficiency, accuracy, and precision. Notably, by allowing each client to select informative batches through the DQN agent, the model attains improved performance with significantly fewer labeled samples, i.e. reaching approximately 50% accuracy with just 3.5% of the available training data.

Accuracy serves as a crucial metric for evaluating the global model's performance. An increase in the number of clients contributes to better accuracy, primarily due to the diverse data contributions from multiple clients. This diversity helps the aggregated model generalize more effectively, capturing variations within the dataset and resulting in a more robust overall performance.

Similarly, precision, which measures the correctness of positive predictions, also improves as the DQN agent's batch selection strategy prioritizes high-quality samples. This capability enhances the model's ability to make accurate predictions during training.

As the client count rises, we see consistent gains in both accuracy and precision, alongside a more efficient use of the annotation budget. It is evident that by utilizing intelligent batch selection, FedRAL enhances model performance while reducing the labeling burden on each client.

C. Results on CIFAR-100

Table II presents the performance of the proposed FedRAL framework on the CIFAR-100 dataset. Due to the increased complexity of CIFAR-100 (100 classes instead of 10), we observe lower accuracy and precision compared to CIFAR-10. However, the FedRAL framework continues to show significant performance improvements with additional clients, demonstrating its effectiveness even in challenging data environments.

TABLE II
RESULTS FOR CIFAR-100 WITH 5 AND 10 CLIENTS USING THE FEDRAL FRAMEWORK. THE TABLE REPORTS THE MEAN AND MAXIMUM VALUES FOR THE ANNOTATION BUDGET (AS A PERCENTAGE OF THE TOTAL EVALUATION DATA), THE ACCURACY, AND THE WEIGHTED AVERAGE PRECISION.

	5 clients		**10 clients**	
	mean	***max***	***mean***	***max***
Budget	10.16%	13.75%	13.33%	18.40%
Accuracy	29.27%	30.00%	34.41%	35.44%
Precision	30.48%	30.66%	35.95%	36.74%

The CIFAR-100 dataset, which contains 100 classes, presents a more challenging environment for model training compared to CIFAR-10. The FedRAL framework demonstrates its effectiveness in this context through enhanced annotation budget efficiency, accuracy, and precision.

The batch selection process of the FedRAL framework ensures that even with a higher annotation budget, the batches chosen are informative, leading to significant improvements in model performance. As the number of clients increases, the data diversity enhances, allowing the global model to generalize better across a more complex dataset.

Precision, reflecting the accuracy of positive predictions, also shows notable improvements (30% for only 10% of the available data). The DQN agent's AL strategy aids in selecting batches that refine the classifier's decision boundaries, enabling the model to maximize the effectiveness of each training round. The ability to prioritize high-quality samples directly contributes to better overall performance.

While the results on CIFAR-100 are generally lower than those on CIFAR-10 due to the dataset's complexity, FedRAL still achieves meaningful performance enhancements. The framework effectively manages datasets with numerous classes, demonstrating high precision and accuracy even with increased annotation budgets. FedRAL achieves high accuracy and precision while keeping the annotation budget low. Although CIFAR-100 requires a higher budget due to its complexity, the DQN-based batch selection strategy ensures that only the most informative samples are chosen for training. In addition, by increasing the number of clients significantly enhances overall model performance. By spreading data and learning across more clients, FedRAL captures a broader range of features, which is crucial in FL settings where data may be non-iid.

D. Comparison with Federated Active Learning (F-AL)

The proposed FedRAL framework is compared with F-AL [75] to evaluate its effectiveness in minimizing annotation costs and improving performance, especially in resource-constrained settings where efficiency is crucial.

1) Key Differences Between FedRAL and F-AL: The following points highlight the key differences and unique contributions of FedRAL compared to F-AL:

- **Efficiency in Training Epochs:** By focusing on the most informative batches through RAL, FedRAL cuts down the number of AL runs needed.
- **Prioritization of Cost Constraints:** FedRAL is tailored for environments where annotation costs are critical, optimizing data selection to keep resource expenditures below set thresholds.

2) Comparison of Annotation Budgets for CIFAR-10 and CIFAR-100: We set a target accuracy that reflects a real-world scenario of a device with limited computational resources. We then evaluate the performance of FedRAL and F-AL in terms of the total annotation budget required to achieve this accuracy on the CIFAR-10 and CIFAR-100 datasets.

Table III shows that FedRAL requires an annotation budget of only 3.45% to achieve the same accuracy (∼50%) as F-AL, which needs approximately 5-6% of the data to be labeled for the CIFAR-10 dataset.

TABLE III
APPROXIMATE ANNOTATION BUDGETS LABELED FOR FEDRAL AND F-AL TO ACHIEVE THE TARGET ACCURACY (50%)

Method	**Annotation Budget**
FedRAL (5 clients)	3.45%
F-AL (5 clients)	∼5-6%

Similarly, in Table IV, FedRAL requires an annotation budget of 10.16% to reach an accuracy of around 30% for the CIFAR-100 dataset. In comparison, F-AL needs a much larger annotation budget of ∼30% to achieve a similar accuracy level.

TABLE IV
APPROXIMATE ANNOTATION BUDGETS LABELED FOR FEDRAL AND F-AL TO ACHIEVE THE TARGET ACCURACY (30%)

Method	**Annotation Budget**
FedRAL (5 clients)	10.16%
F-AL (5 clients)	∼30%

These results illustrate the clear advantage of FedRAL in terms of annotation efficiency. By using RL to intelligently select the most informative batches, FedRAL reduces the number of labeled instances required to achieve high accuracy. In contrast, F-AL, while effective, requires significantly higher annotation budgets to reach the same level of performance. This makes FedRAL a more suitable option for environments where minimizing annotation costs is crucial, such as resource-constrained systems or scenarios with limited labeling capacity.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduce FedRAL, a framework that integrates FL with RAL to optimize batch selection in distributed machine learning. By utilizing RL, FedRAL identifies the most informative data batches based on model uncertainty, significantly reducing labeling costs while maintaining privacy. The results on CIFAR-10 and CIFAR-100 demonstrate that

FedRAL improves both accuracy and precision, all while minimizing the need for labeled data. This highlights FedRAL's efficiency, making it an ideal solution for systems with strict resource and cost constraints.

For future work, we plan to enhance FedRAL's flexibility by enabling each client to adopt completely different policies, regarding custom rewards and termination conditions tailored to their specific data. We will also extend FedRAL to manage more complex, non-iid data distributions. Additionally, we aim to improve scalability by investigating more efficient RL algorithms and reducing communication overhead. These advancements will make FedRAL more robust for diverse real-world applications.

Acknowledgements. This research was supported by the European Commission under contract No. 101121329 ARIEN.

References

[1] Usman Ahmad Usmani, Junzo Watada, Jafreezal Jaafar, Izzatdin Abdul Aziz, and Arunava Roy, "A reinforced active learning algorithm for semantic segmentation in complex imaging," in IEEE, 2021.

[2] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," in Proceedings of the 8th International Conference on Learning Representations (ICLR), 2019.

[3] Wenbin Cai, Muhan Zhang, and Ya Zhang, "Batch mode active learning for regression with expected model change," in IEEE Transactions on Neural Networks and Learning Systems, 2017.

[4] Xiaofeng Cao, and Ivor W. Tsang, "Bayesian active learning by disagreements: A geometric perspective," in CoRR, 2021.

[5] Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal, "Reinforced active learning for image segmentation," in International Conference on Learning Representations (ICLR), 2020.

[6] Shayok Chakraborty, Vineeth Balasubramanian, and Sethuraman Panchanathan, "Adaptive batch mode active learning," in IEEE Transactions on Neural Networks and Learning Systems, 2014.

[7] Xaolin Chun, "Active learning with reinforcement learning for data-efficient classification," 2023.

[8] Yulai Cui, Shruthi Kashinath Hiremath, and Thomas Ploetz, "Reinforcement learning based online active learning for human activity recognition," in Proceedings of the 2022 ACM International Symposium on Wearable Computers, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[10] Michael Dodds, Jeff Guo, Thomas Löhr, Alessandro Tibo, Ola Engkvist, and Jon Paul Janet, "Sample efficient reinforcement learning with active learning for molecular design," in Chemical Science, 2023.

[11] Meng Fang, Yuan Li, and Trevor Cohn, "Learning how to active learn: A deep reinforcement learning approach," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017.

[12] Anthony Faulds, "Adaptive batch size selection in active learning for regression," in Mathematics in Computer Science (MCS), 2022.

[13] Nikos Fazakis, Vasileios G. Kanas, Christos K. Aridas, Stamatis Karlos, and Sotiris Kotsiantis, "Combination of active learning and semi-supervised learning under a self-training scheme," in Entropy, 2019.

[14] Guy Hacohen, Avihu Dekel, and Daphna Weinshall, "Active learning on a budget: Opposite strategies suit high and low budgets," 2022.

[15] Mojtaba Haghighatlari, Gaurav Vishwakarma, Doaa Altarawy, Ramachandran Subramanian, Bhargava U. Kota, Aditya Sonpal, Srirangaraj Setlur, and Johannes Hachmann: Chemml, "A machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data," in ChemRxiv, 2020.

[16] Hanane Allioui, Mazin Abed Mohammed, Narjes Benameur, Belal Al-Khateeb, Karrar Hameed Abdulkareem, Begonya Garcia-Zapirain, Robertas Damaševičius, and Rytis Maskeliūnas, "A multi-agent deep reinforcement learning approach for enhancement of covid-19 ct image segmentation," in Journal of Personalized Medicine, 2022.

[17] Hideaki Ishibashi, and Hideitsu Hino, "Stopping criterion for active learning based on error stability," 2017.

[18] Michael Katz, and Eli Kravchik, "Reinforced meta active learning," 2022.

[19] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun, "Task-aware variational adversarial active learning," 2020.

[20] Minrui Xu, Jialiang Peng, B. B. Gupta, Jiawen Kang, Zehui Xiong, Zhenni Li, "Multiagent Federated Reinforcement Learning for Secure Incentive Mechanism in Intelligent Cyber–Physical Systems," in IEEE Internet of Things Journal, 2022.

[21] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua, "Discovering general-purpose active learning strategies," 2018.

[22] Shuai Yu, Xu Chen, Zhi Zhou, Xiaowen Gong, and Di Wu, "When deep reinforcement learning meets federated learning: Intelligent multi-timescale resource management for multiaccess edge computing in 5G ultradense network," in IEEE, 2021.

[23] David D. Lewis, and William A. Gale, "A sequential algorithm for training text classifiers," in ACM SIGIR proceedings on Research and Development in Information Retrieval, 1994.

[24] Yi Jie Wong, Mau-Luen Tham, Ban-Hoe Kwan, and Yasunori Owada: FedDdrl, "Federated double deep reinforcement learning for heterogeneous IoT with adaptive early client termination and local epoch adjustment," in Sensors, 2023.

[25] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," 2017.

[27] David Nilsson, Aleksis Pirinen, Erik Gärtner, and Cristian Sminchisescu, "Embodied learning for lifelong visual perception," 2021.

[28] Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales, "Meta-learning transferable active learning policies by deep reinforcement learning," in AutoML Workshop at ICML, 2018.

[29] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi, "Active learning by feature mixing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[30] Rinyoichi Takezoe, Xu Liu, Shunan Mao, Marco Tianyu Chen, Zhanpeng Feng, Shiliang Zhang, and Xiaoyu Wang, "Deep active learning for computer vision: Past and future," in APSIPA Transactions on Signal and Information Processing, 2023.

[31] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang, "A survey of deep active learning," in ACM Computing Surveys (CSUR), 2020.

[32] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell, "Variational adversarial active learning," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.

[33] Ozan Sener, and Silvio Savarese, "Active learning for convolutional neural networks: A core-set approach," 2017.

[34] Emma Slade, and Kim M. Branson, "Deep reinforced active learning for multi-class image classification," 2022.

[35] Le Sun, and Yihong Gong: Active learning for image classification, "A deep reinforcement learning approach," in 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), IEEE, 2019.

[36] Jingwen Wang, Yuguang Yan, Yubing Zhang, Guiping Cao, Ming Yang, and Michael K. Ng, "Deep reinforcement active learning for medical image classification," in Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, 2020.

[37] Sarah Wassermann, Thibaut Cuvelier, and Pedro Casas, "Ral – improving stream-based active learning by reinforcement learning," in European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Database, Workshop on Iterative Adaptive Learning, 2019.

[38] Christopher J. C. H. Watkins, and Peter Dayan, "Q-learning," in Machine learning, 1992.

[39] Mark Woodward, and Chelsea Finn, "Active one-shot learning," in NIPS Deep RL Workshop, 2017.

[40] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan, "A comparative survey of deep active learning," 2022.

[41] Deepraj Chowdhury, Soham Banerjee, Madhushree Sannigrahi, Arka Chakraborty, Anik Das, Ajoy Dey, and Ashutosh Dhar Dwivedi, "Federated learning based covid-19 detection," 2023.

[42] Ishan Rajendrakumar Dave, Chen Chen, and Mubarak Shah: Spact, "Self-supervised privacy preservation for action recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[43] Keval Doshi, and Yasin Yilmaz, "Federated learning-based driver activity recognition for edge devices," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[44] Qiang Meng, Feng Zhou, Hainan Ren, Tianshu Feng, Guochao Liu, and Yuanqing Lin, "Improving federated learning face recognition via privacy-agnostic clusters," 2022.

[45] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi, "Federated unlearning via class-discriminative pruning," in Proceedings of the ACM Web Conference 2022, 2022.

[46] Chun-Han Yao, Boqing Gong, Yin Cui, Hang Qi, Yukun Zhu, and Ming-Hsuan Yang, "Federated multi-target domain adaptation," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022.

[47] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov, "See through gradients: Image batch recovery via gradinversion," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[48] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai, "Exploiting shared representations for personalized federated learning," in International conference on machine learning. pp. 2089–2099. PMLR, 2021.

[49] Athanasios Psaltis, Anestis Kastellos, Charalampos Z. Patrikakis, and Petros Daras, "Fedlid: Self-supervised federated learning for leveraging limited image data," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.

[50] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha, "Fedx: Unsupervised federated learning with cross knowledge distillation," in European Conference on Computer Vision. Springer, 2022.

[51] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li, "Federated unsupervised representation learning," in Frontiers of Information Technology and Electronic Engineering, 2023.

[52] Wentao Liu, Xiaolong Xu, Jintao Wu, and Jielin Jian, "Federated meta reinforcement learning for personalized tasks," in Tsinghua Science and Technology, 2024.

[53] Weicheng Xiong, Quan Liu, Fanzhang Li, and Bangjun Wang, "Personalized federated reinforcement learning: Balancing personalization and experience sharing via distance constraint," in Expert Systems with Applications, 2024.

[54] Song Liu, Shiyuan Yang, Hanze Zhang and Weiguo Wu, "A federated learning and deep reinforcement learning-based method with two types of agents for computation offload," in Sensors, 2023.

[55] Zhu Tianqing, Wei Zhou, Dayong Ye, Zishuo Cheng, and Jin Li, "Resource allocation in iot edge computing via concurrent federated reinforcement learning," in IEEE Internet of Things Journal, 2021.

[56] Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Cheston Tan, Bryan Kian Hsiang Low, and Roger Wattenhofer, "Federated heterogeneous q-learning," in Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 2023.

[57] Zhijie Xie, and Shenghui Song, "FedKL: Tackling data heterogeneity in federated reinforcement learning by penalizing KL divergence," in IEEE Journal on Selected Areas in Communications, 2023.

[58] Zhikai Liu, Navneet Garg, and Tharmalingam Ratnarajah. "Multi-agent federated reinforcement learning strategy for mobile virtual reality delivery networks," in IEEE Transactions on Network Science and Engineering, 2023.

[59] Seongin Na, Tomáš Rouček, Jiří Ulrich, Jan Pikman, Tom s Krajník, Barry Lennox, and Farshad Arvin, "Federated reinforcement learning for collective navigation of robotic swarms," in IEEE Transactions on Cognitive and Developmental Systems, 2023.

[60] Woonghee Lee, "Federated reinforcement learning-based UAV swarm system for aerial remote sensing,"in Wireless Communications and Mobile Computing, 2022.

[61] Boyi Liu, Lujia Wang, and Ming Liu, "Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems," in IEEE Robotics and Automation Letters, 2019.

[62] Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng, "Federated reinforcement learning: Techniques, applications, and open challenges," 2021.

[63] Xuanzhu Sheng, Zhiqiang Gao, Xiaolong Cui, and Chao Yu, "Federated Reinforcement Learning Technology and Application in Edge Intelligence Scene," in Advances in Internet, Data & Web Technologies, Proceedings of the 11th International Conference on Emerging Internet, Data & Web Technologies (EIDWT-2023), 2023.

[64] Euclides Carlos Pinto Neto, Somayeh Sadeghi, Xichen Zhang and Sajjad Dadkhah, "Federated reinforcement learning in IoT: Applications, opportunities and open challenges," in Applied Sciences, 2023.

[65] Euclides Carlos, Pinto Neto, Somayeh Sadeghi, Xichen Zhang and Sajjad Dadkhah, "Federated Reinforcement Learning in IoT: Applications, Opportunities and Open Challenges," in Research on Security and Privacy in IoT and Big Data, 2023.

[66] Santiago A. Moreno-Acevedo, Juan Camilo Vasquez-Correa, Juan M. Martín-Doñas, and Aitor Álvarez, "Stream-based Active Learning for Speech Emotion Recognition via Hybrid Data Selection and Continuous Learning," in International Conference on Text, Speech, and Dialogue, 2024.

[67] 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, 2016.

[68] Krizhevsky Alex, and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[69] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition,", in Proceedings of the IEEE, 1998.

[70] Chenyu Zhang, Han Wang, Aritra Mitra, and James Anderson, "Finite-time analysis of on-policy heterogeneous federated reinforcement learning," 2024.

[71] Richard S. Sutton, and Andrew G. Barto, "Reinforcement learning: An introduction," 2018.

[72] Alex Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[73] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in IEEE, 2009.

[74] 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, Las Vegas, 2016.

[75] Jin-Hyun Ahn, Kyungsang Kim, Jeongwan Koh, and Quanzheng Li, "Federated active learning (FAL): An efficient annotation strategy for federated learning," in IEEE Access, 2024.

[76] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

[77] SangMook Kim, Sangmin Bae, Hwanjun Song, Se-Young Yun, "Re-thinking Federated Active Learning based on Inter-class Diversity," 2023.

[78] Razvan Caramalau, Binod Bhattarai, Danail Stoyanov, "Federated Active Learning for Target Domain Generalisation," 2023.

[79] Zhipeng Deng, Yuqiao Yang, and Kenji Suzuki, "Federated Active Learning Framework for Efficient Annotation Strategy in Skin-Lesion Classification," 2024.

[80] Usman Ahmed, Jerry Chun-Wei Lin and Philippe Fournier-Viger, "Federated deep active learning for attention-based transaction classification," 2023.

[81] SangMook Kim, Sangmin Bae, Hwanjun Song, and Se-Young Yun, "Re-thinking Federated Active Learning based on Inter-class Diversity," 2023.