Cen-FedRAL: Bridging Centralized Reinforcement Learning Training and Federated Active Learning for Dynamic Batch Acquisition on Resource-Constrained Devices

Ioannis Lazaridis[1], Athanasios Psaltis[1], Anastasios Dimou[1], and Petros Daras[1]

Visual Computing Lab, Information Technologies Institute,
Centre for Research & Technology Hellas (CERTH), Thessaloniki, Greece
{lazaioan, at.psaltis, dimou, daras}@iti.gr

Abstract. This paper addresses the critical challenges of data sparsity, label scarcity, and resource constraints in distributed vision environments, which directly impact the effectiveness and annotation efficiency of federated systems. We introduce Cen-FedRAL, a novel architecture optimized for cost-sensitive edge deployment. Adopting a Federated Reinforced Active Learning approach, we propose a centralized training strategy for the reinforcement learning agent, enhanced by Prioritized Experience Replay. By aggregating experience buffers across all participating clients, the framework enables a global agent to synthesize multi-client insights into a unified, dynamic batch acquisition policy. This collaborative approach enables a unified and adaptive sampling policy across the federated network. We benchmark Cen-FedRAL across three vision domains: Image Classification (CIFAR-100 with ResNet-18), and Object Detection and Semantic Segmentation (PASCAL VOC with Faster R-CNN and U-Net). The robustness is demonstrated through zero-shot experiments, exhibiting strong policy generalization and effective cross-dataset transfer without extra training. Experimental results show that Cen-FedRAL offers a versatile, cost-aware, and robust framework for real-world distributed systems. Compared to decentralized FedRAL, Cen-FedRAL improves CIFAR-100 ACC by 7.1 percentage points, PASCAL VOC segmentation mIoU by 10.47 points, and object detection mAP by 3.5 points, under identical annotation budgets.

Keywords: Active learning · Reinforcement Learning · Federated Learning · Annotation Budget · Dynamic Batch Acquisition

1 Introduction

The expansion of edge and Internet of Things (IoT) architectures has generated unprecedented volumes of decentralized data. However, their effective use for Deep Learning (DL) is constrained by high annotation costs and the need to keep raw data local. Federated Learning (FL) enables collaborative model optimization without exchanging raw training data, but does not address the labeling bottleneck.

Federated Active Learning (FAL) combines Active Learning (AL) with federated optimization to prioritize informative samples. Frameworks such as FedRAL [29] employ Reinforcement Learning (RL) for adaptive batch selection, but train agents from sparse local trajectories. This experience isolation can yield unstable policies, slow convergence, and locally biased acquisition decisions.

Problem Statement: In distributed AL systems, local RL agents frequently lack the breadth of experience necessary to navigate the complex trade-off between model performance gain and annotation cost. When agents learn in isolation, they are susceptible to local biases and noise, resulting in suboptimal batch selection policies that either prematurely exhaust the budget or fail to reach target accuracy (ACC) levels. In practice, this leads to shallow learning curves where local agents overfit their own client distribution. The fundamental challenge lies in distilling collective decision-making intelligence from across the network into a unified policy while preserving the decentralized training setting.

We introduce Cen-FedRAL, a framework designed to bridge the gap between local experience and global policy optimization by aggregating RL experience centrally to learn a single acquisition policy that is cost-efficient while keeping raw training data decentralized. Our main contributions are summarized as follows:

- **Centralized Training (CT) for Federated Reinforcement Learning (FRL):** We introduce a novel architecture that aggregates RL trajectories at a central server to train a global Deep Q-Network (DQN) [57] agent. We show that centralizing the DQN eliminates experience isolation without sharing raw images or labels, as only transition tuples $(state, action, reward)$—rather than raw client data—are shared to create a diversified, global pool of experience.
- **Prioritized Global Experience Replay (PGER):** By adapting Prioritized Experience Replay (PER) [46] to a centralized-federated setting, Cen-FedRAL prioritizes transitions with large TD-error across all participating clients. We demonstrate that this mechanism effectively accelerates convergence and improves sample efficiency as empirically validated on CIFAR-100 [28] and PASCAL VOC 2012 [14].
- **Cost-Aware Multi-Task Framework:** We provide a unified, cost-aware framework validated across Image Classification, Object Detection, and Semantic Segmentation. By employing a single RL configuration for all tasks, we demonstrate that Cen-FedRAL is robust to task transitions and achieves strong zero-shot generalization across unseen datasets without further training, suggesting its scalability for real-world distributed systems.
- **Centralized RL Training Strategy:** Clients independently optimize their local DQNs to generate experience trajectories, while a separate centralized DQN is trained exclusively on the server using the aggregated replay buffers. At the beginning of each federated round, the centralized DQN is redis-

tributed to all clients, replacing the local DQN parameters and providing a common policy initialization.

Cen-FedRAL overcomes experience isolation by centralizing RL training and distilling global insights through PGER. This creates a unified, hardware-efficient policy that adaptively optimizes the performance-cost trade-off. By targeting dense prediction tasks such as semantic segmentation on PASCAL VOC 2012 with realistic client budgets, Cen-FedRAL directly addresses the practical constraints of deploying AL in federated vision systems—a key concern for real-world edge applications.

2 Related Work

2.1 Active Learning and Resource Constraints

AL prioritizes high-utility samples to mitigate prohibitive labeling costs [65]. While traditional uncertainty metrics suffice for classification [56], complex tasks like detection and segmentation require sophisticated strategies for localization [18] and pixel-level complexity [22]. However, scaling these heuristics often incurs high computational overhead [65], hindering resource-constrained edge devices [36]. Despite advances in annotator expertise [24] and screening [52], traditional methods often misrepresent model uncertainty [49], necessitating adaptive, learned sampling policies to optimize the performance-cost trade-off [36].

2.2 Reinforced Active Learning

RAL treats sample selection as a sequential decision-making process [16], employing adaptive policies across NLP, meta-learning, and vision [16]. These frameworks dynamically adjust to remaining capacity [20], adaptive batching [17], and stream-based recognition [11]. In complex vision tasks, RAL mitigates metric misalignment; MGRAL [32] maximizes mean Average Precision (mAP) for object detection via efficient look-up tables [32], while semantic segmentation policies prioritize informative sub-regions to reduce labeling costs and class imbalance [6]. This paradigm extends to multi-agent RL for medical imaging [2], with learned stopping criteria identifying points of diminishing returns [25].

2.3 Federated Learning

FL enables decentralized training while keeping raw data local via Federated Averaging (FedAvg) [37]. Recent shifts toward representation learning utilize contrastive objectives [9, 67] and distillation [21] to mitigate non-IID data [41]. In segmentation, unsupervised grouping [7] and adaptive aggregation via fuzzy logic [68] or uncertainty maps [19] have been explored. Similarly, federated detection targets edge efficiency through layer-freezing [3], blockchain-secured asynchronous architectures [55], and domain-specific YOLO [43] initialization [1]. Despite these advances, optimizing acquisition under resource constraints remains an open challenge, often necessitating deep RL-based dynamic assessment [8].

2.4 Federated Reinforcement Learning

FRL synergizes FL and RL to address privacy, scalability, and efficiency [40, 42, 51]. By localizing data, FRL optimizes network performance and security [34, 62, 66] via algorithms such as FedAvg [37], PPO [47], and SARSA [53] in heterogeneous environments [58, 64]. Recent work targets personalization [61], resource-aware offloading [33,70], client diversity [15,60], and multi-agent scaling across robotics and UAVs [31,35,38]. To our knowledge, Cen-FedRAL is the first framework to centralize the RL-based acquisition policy in FAL while keeping task model updates decentralized.

2.5 Federated Active Learning

FAL mitigates annotation bottlenecks in decentralized systems [37], with frameworks like FEDALV [5] and LoGo [27] addressing data imbalances via domain generalization. Beyond classification and medical tasks [13], FAL reduces costs in spatially-aware tasks such as object detection [4] and 3D segmentation [54]. FedRAL [29] integrates RL for autonomous batch selection in resource-limited networks and is the closest work to ours. However, while FedRAL trains RL agents independently on each client, Cen-FedRAL centralizes training and introduces PGER, directly addressing experience isolation.

3 Proposed Method: Cen-FedRAL

Cen-FedRAL extends the FedRAL [29] framework by addressing experience isolation through CT with PER. We assume that: (i) clients can share RL trajectories (states, actions, rewards) but not raw images, without transmitting raw images or labels; (ii) communication between clients and the server is synchronous at the round level; and (iii) client datasets are non-overlapping and potentially non-IID, reflecting realistic federated environments. These mechanisms are implemented through three core components:

- **Global Experience Prioritization:** Cen-FedRAL introduces centralized training by offloading local trajectories to a central server. By employing PER, the system identifies and prioritizes critical transitions from the entire network. This approach enhances RL stability and ensures the sampling policy is refined using the most informative trajectories across all clients, overcoming the limitations of isolated local learning.
- **Collaborative Experience Sharing:** Clients contribute decision-making metadata—states, actions, and rewards—to a shared global buffer. These states are derived exclusively from model outputs (e.g., margins, entropies, confidences), without transmitting raw images or labels. This allows the central DQN to distill collective knowledge into a unified batch selection policy. Such collaboration overcomes local data sparsity, enabling clients to leverage diverse trajectories from across the network regardless of individual data distributions.

– **Federated Model Aggregation:** While the RL agent optimizes the selection strategy centrally, the underlying task-specific models (e.g., U-Net [45] for segmentation or Faster R-CNN [44] for detection) are synchronized via FL. This ensures that the predictive power of the global model is distributed back to the clients, providing more accurate uncertainty estimates that guide the next round of active sampling.

As illustrated in Fig. 1, the Cen-FedRAL process forms a closed loop: each client utilizes its local task-specific model and the shared global DQN policy to select informative data batches. These local experiences are then aggregated on the central server to update the global policy via PER, while the model weights are simultaneously updated via FedAvg. This dual-stream aggregation ensures that both the task model and the decision-making agent benefit from the collective intelligence of the federated network.

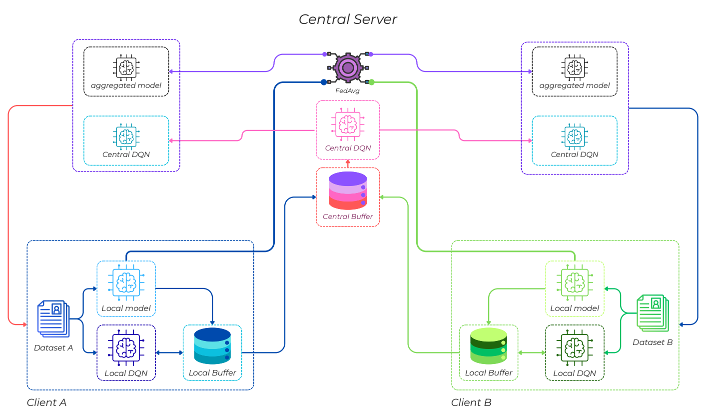

Fig. 1: Overview of the Cen-FedRAL Framework. The figure illustrates the dual-stream update: model aggregation via FedAvg [37] (purple arrows) and centralized DQN [57] update from aggregated replay buffers (pink arrows). In each round, clients utilize their local models—ResNet-18 [23], Faster R-CNN [44], or U-Net [45] depending on the specific vision task—to generate RL trajectories, which are stored in local replay buffers. These buffers are aggregated into a global priority buffer at the central server to train a central DQN via PER [46]. Simultaneously, local model parameters are synchronized through FedAvg. The refined global policy and aggregated models are redistributed to clients to initialize the subsequent federated round.

This approach is particularly tailored for high-complexity vision tasks, including Image Classification, Object Detection, and Semantic Segmentation, where annotation costs and computational resources must be meticulously managed. By intelligently distributing the workload and prioritizing global experiences, Cen-FedRAL makes high-performance distributed learning feasible under strict budget constraints.

3.1 Adaptive Batch Selection via Federated Reinforcement Learning

The Cen-FedRAL annotation strategy is modeled as a Markov Decision Process (MDP), employing an RL agent as a high-level controller. Moving beyond fixed heuristics, the agent learns an adaptive batch selection policy that optimizes the trade-off between precision gain and labeling cost. At each iteration t, the agent analyzes the local model's uncertainty through the state representation s_t and executes an action a_t, which defines the size and composition of the next batch to be annotated. This sequential process is refined via a global-to-local loop: edge-side actions inform a centrally optimized policy leveraging collective network experiences, ensuring task-awareness and robustness against client-side data distribution shifts.

State Space Representation The state s_t at each iteration represents the current knowledge and uncertainty of the local model over a fixed reference subset of the data, denoted as $\mathcal{D}_{state}$. By using a fixed subset, the RL agent can consistently monitor the model's evolution. The state vector is constructed based on task-specific uncertainty metrics:

- **Image Classification:** For classification, the state is composed of the margin scores of the samples in $\mathcal{D}_{state}$. For a sample x_i, the margin score [48] $m(x_i)$ is defined as:

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

  where $P(y^*|x_i)$ and $P(y^{**}|x_i)$ are the highest and second-highest predicted class probabilities, respectively. The final state vector is obtained by sorting these scores to ensure permutation invariance.
- **Semantic Segmentation:** In the segmentation task, the state captures pixel-wise uncertainty via Shannon Entropy [50]. For each image in $\mathcal{D}_{state}$, the entropy map is calculated, and the average entropy per image is used as a feature:

$$E(x_i) = -\frac{1}{H \times W} \sum_{h,w} \sum_{c} P(y^c_{h,w}|x_i) \log(P(y^c_{h,w}|x_i) + \epsilon) \tag{2}$$

  where H, W are the image dimensions, c is the class index, and ϵ is a small constant for numerical stability. The state vector is the sorted list of mean entropies across $\mathcal{D}_{state}$.
- **Object Detection:** For object detection (using Faster R-CNN), the state incorporates multi-dimensional features from predicted bounding boxes. For each image, we extract a triplet of features $\phi(x_i) = [\mu_{conf}, max_{conf}, H_{conf}]$, where μ_{conf} is the mean confidence score, max_{conf} is the maximum confidence, and H_{conf} is the entropy of the confidence distribution:

$$H_{conf}(x_i) = -\sum_{j} \bar{p}_j \log(\bar{p}_j + \epsilon), \quad \bar{p}_j = \frac{conf_j}{\sum conf} \tag{3}$$

The features are flattened into a single vector and sorted based on the mean confidence to maintain consistency.

Action Space At each iteration t, the DQN agent executes an action $a_t \in A$, which defines the number of samples k to be selected from the unlabeled pool $\mathcal{U}_t$. More specifically, we define a discrete action set $A = \{k_1, \ldots, k_L\}$, where each k_i corresponds to a fraction of the current unlabeled pool (e.g., $1\%, 2\%, \ldots, 10\%$). In all our experiments, the maximum fraction is set to 10%. This adaptive approach allows the agent to request larger batches when the model is highly uncertain and smaller, more surgical updates as the model converges. Once the batch k is decided, the top-k samples are selected based on the specific uncertainty criteria (e.g., Margin, Entropy, or Confidence) defined for each task.

Cost-Aware Reward The objective of the RL agent is to maximize the efficiency of the annotation process relative to the consumed budget. Unlike standard RAL rewards that only consider absolute performance gains, we define a cost-aware reward signal r_t that normalizes the performance metric $\mathcal{M}$ by the cumulative batch size B_t used up to iteration t:

$$r_t = \frac{\mathcal{M}(B_{t+1})}{B_{t+1}} - \frac{\mathcal{M}(B_t)}{B_t} \tag{4}$$

where $\mathcal{M}(B_t)$ represents the model's performance after annotating B_t samples. This differential efficiency reward forces the agent to favor actions that yield the steepest learning curve per annotated sample, effectively penalizing redundant labeling. The metric $\mathcal{M}$ is dynamically assigned per task: **Image Classification** uses Weighted Precision Score, **Segmentation** employs mean Intersection over Union (mIoU) for pixel-level overlap, and **Object Detection** utilizes mAP for box localization and classification.

Warm-Start and Calibration To provide the DQN with a baseline policy, we initiate the process with *Warm-Start Episodes*. During these episodes, the agent explores various batches to observe their impact on the model's performance metrics. This phase serves a dual purpose: (1) it populates the local Replay Buffer $\mathcal{B}_i$ with initial transitions, and (2) it establishes the Target Performance ($\mathcal{M}_{target}$) and Target Budget ($\mathcal{B}_{target}$). The process terminates once the agent reaches the target precision or exhausts the budget, ensuring that subsequent training episodes are focused on optimizing these established thresholds.

Local Agent Training & Experience Generation Following the warm-start phase, the agent enters the main AL loop. At each iteration, the local DQN observes state s_t and selects an adaptive batch a_t. Upon receiving the cost-aware reward r_t (Eq. 4), the transition (s_t, a_t, r_t, s_{t+1}) is stored in the local replay buffer. While clients perform periodic local Q-network optimization [57] to adapt to local distributions, these transitions are offloaded to the central node at each round's end. This aggregation enables the global agent to distill a unified policy that transcends individual data silos, facilitating the global optimization.

Convergence and Termination Criteria The AL trajectory is governed by a dual-constraint policy balancing model maturity and label economy. The loop terminates upon satisfying either: (i) **Performance Saturation**, where the sampling concludes once the task-specific metric $\mathcal{M}$ reaches the threshold $\mathcal{M}_{target}$ established during warm-start calibration to avoid marginal gains; or (ii) **Budget Exhaustion**, where the process is capped by a maximum allowance $\mathcal{B}_{target}$, forcing the agent to prioritize informative samples within strict resource limits.

3.2 Centralized Server Operations: A Dual-Stream Strategy

In our experimental setup, the global dataset is partitioned among N decentralized clients with unique, non-overlapping subsets. As formalized in Algorithm 1, each federated round follows a synchronized sequence: local model optimization and experience generation at the client level, followed by dual-stream aggregation and policy refinement at the central server. This structured workflow maintains data privacy while progressively optimizing both the vision task and the sampling policy.

The central server coordinates the collaborative process by performing two distinct updates per round: the aggregation of task-specific model parameters and the centralized training of the RL agent using offloaded trajectories.

Stream I: Federated Model Aggregation To maintain a robust global predictive model (e.g., ResNet-18, Faster R-CNN, or U-Net), the server aggregates the weights from all N clients. This is achieved using the FedAvg algorithm. The global parameters θ_{global} are computed as the weighted average of the local parameters θ_n based on the number of samples $|\mathcal{D}_n|$ contributed by each client:

$$\theta_{global} = \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \theta_n \tag{5}$$

where $|\mathcal{D}|$ represents the total number of samples across all clients. This unified model is then redistributed to provide each client with an improved baseline for the next round.

Stream II: Centralized Policy Training with PER Simultaneously, the server enhances the batch selection policy by training a central DQN. Although classifier models are aggregated to ensure predictive consistency, the true novelty of Cen-FedRAL lies in the centralized training of the RL agent. In traditional FedRAL, agents learn in isolation, leading to suboptimal and inconsistent sampling policies. We propose a centralized training scheme in which the central server acts as a global experience aggregator, distilling knowledge from the diverse trajectories of all participating clients.

Algorithm 1 Cen-FedRAL: Federated Reinforced Active Learning with Centralized Training

Require: Global model θ_0, Central DQN $\mathcal{Q}_0$, Number of clients K, Rounds T

Initialize global parameters $\theta_{glob} \leftarrow \theta_0$ and policy $\mathcal{Q}_{glob} \leftarrow \mathcal{Q}_0$
for each federated round $t = 1, \ldots, T$ **do**
  Server Broadcast:
  Send θ_{glob} and $\mathcal{Q}_{glob}$ to all clients $k \in \{1, \ldots, K\}$
  for each client $k = 1, \ldots, K$ **in parallel do**
    Synchronize local model: $\theta_k \leftarrow \theta_{glob}$
    Initialize local replay buffer $\mathcal{B}_k$
    while budget not exhausted **and** target performance not met **do**
      Observe state s from fixed reference subset $\mathcal{D}_{state}$
      Select batch a using $\mathcal{Q}_{glob}$ (ϵ-greedy)
      Annotate samples and update local model θ_k
      Compute cost-aware reward r based on task metric $\mathcal{M}$
      Store transition (s, a, r, s') in local buffer $\mathcal{B}_k$
    end while
    Upload local model weights θ_k and buffer $\mathcal{B}_k$ to Server
  end for
  Centralized Server Operations (Dual-Stream):
  {*Stream I: Model Aggregation*}
  The central server aggregates the updated model weights from all clients using FedAvg:

$$\theta_{global} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \theta_k$$

  {*Stream II: Policy Optimization*}
  Aggregate all $\mathcal{B}_k$ into a Global Priority Pool
  Train $\mathcal{Q}_{glob}$ using PER based on TD-errors
  Global Update:
  Update $\mathcal{Q}_{glob}$ and θ_{glob} for the next round
end for

Global Experience Aggregation At the end of each federated round, instead of only sharing model weights, each client $n \in \{1, \ldots, N\}$ transmits its local replay buffer $\mathcal{B}_n$ to the central server. The server initializes a global experience pool by collecting these distributed transitions. This aggregation allows the central agent to observe a significantly broader state-space than any individual client, effectively overcoming local data sparsity and heterogeneous data distributions.

Prioritized Experience Replay To optimize the central DQN policy, we employ PER, ensuring that the agent focuses on the most "instructive" transitions from across the network. Unlike uniform sampling, we assign a priority p_i to each transition i based on its magnitude of the temporal difference (TD) error δ_i:

$$p_i = |\delta_i| + \epsilon \tag{6}$$

where ϵ is a small positive constant to ensure all transitions have a non-zero probability of being sampled. The probability of the sampling transition i is then defined as:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{7}$$

where α determines the strength of prioritization. In our implementation, we utilize this distribution to sample mini-batches that are then used to train the central DQN estimator.

Centralized DQN Update The central server executes the training step for the global DQN by minimizing the Mean Squared Error (MSE) of the TD-error across the aggregated mini-batches. The target Q-value Y_t is computed as:

$$Y_t = r_t + \gamma \max_{a'} \hat{Q}(s_{t+1}, a'; \theta_{target}) \tag{8}$$

where $\hat{Q}$ represents the target network, which is slowly updated via a soft-copy factor τ (Target Copy Factor), where $\hat{Q}$ denotes the target network, which is updated using a soft-update coefficient τ, where $\tau \in (0, 1)$ controls the interpolation between the local and target networks:

$$\theta_{target} \leftarrow \tau\theta_{local} + (1 - \tau)\theta_{target} \tag{9}$$

In our implementation, $\tau = 0.01$.

Federated Model and Policy Synchronization Post-centralized training, the server performs a dual synchronization: (i) **Model Aggregation**, where task-specific parameters (ResNet-18, Faster R-CNN, or U-Net) are aggregated via FedAvg, weighted by local sample counts; and (ii) **Policy Redistribution**, where the parameters of the centralized DQN are broadcast to all clients and replace the local DQN parameters, providing a common initialization for the next federated round. This ensures every client begins the next federated round with a state-of-the-art global model and a refined, collective batch selection policy.

4 Experiments

In this section, we evaluate the performance of Cen-FedRAL across three computer vision tasks: Image Classification, Object Detection, and Semantic Segmentation. We benchmark our framework using the CIFAR-100 dataset with a ResNet-18 backbone for image classification, and the PASCAL VOC dataset utilizing Faster R-CNN and U-Net architectures for object detection and semantic segmentation, respectively. All reported results represent the mean performance across 10 independent experimental runs.

4.1 Experimental Setup

To ensure a fair comparison across different domains, we maintain a unified RL configuration and data partitioning scheme for all experiments.

Data Partitioning: Each client's local dataset is partitioned into four distinct subsets: (i) State data (10%) used to represent the environment for the DQN agent; (ii) Warm-start data (10%) for initial classifier training; (iii) DQN training data (60%) for global policy optimization; and (iv) Evaluation data (20%) for benchmarking performance. To evaluate the framework's versatility, we test under different data distributions: our Image Classification experiments follow an IID setup, whereas Object Detection and Semantic Segmentation tasks are conducted under Non-IID conditions to simulate the data heterogeneity typical of real-world edge deployments.

Implementation Details: We use a centralized Double DQN with a PER buffer (5×10^4, exponent 3.0), learning rate 10^{-3}, and $\tau = 0.01$. The system comprises 5 clients over 10 rounds. Each round includes 10 DQN epochs (5 episodes per epoch, and 50 network updates per episode), initiated by 10 warm-start episodes for stability. We compare Cen-FedRAL against: (i) the decentralized FedRAL baseline [29], where DQN agents are trained locally on each client without sharing experience; and (ii) a non-RL Federated Active Learning baseline, which utilizes fixed batch sizes and standard uncertainty sampling. This multi-level comparison allows us to quantify the gains achieved through global trajectory aggregation and prioritized global experience replay, while also justifying the use of reinforcement learning over heuristic-based sampling strategies.

4.2 Image Classification

Task Configuration: For the image classification experiments, we utilize the CIFAR-100 dataset, which consists of 60,000 32×32 color images across 100 classes. We employ a ResNet-18 backbone pre-trained on ImageNet [12] as the classifier. Each client trains with a local batch size of 32 for 5 epochs per federated round.

Results: Table 1 shows Cen-FedRAL outperforms FedRAL by +7.14% ACC with similar budget (<1% difference). Crucially, it surpasses Non RL-FAL (+1.51% ACC) while using an 83.8% smaller budget, proving centralized RL efficiency.

4.3 Object Detection

Task Configuration: For the object detection task, we evaluate our framework on the PASCAL VOC 2012 dataset, which encompasses 20 object categories (21 including background) across 11,530 images. We employ a Faster R-CNN architecture with a ResNet-based backbone as the detection model. This task is significantly more complex than classification, as the acquisition policy must take into account both accurate location of the bounding box and category identification. Each client performs local training with a batch size of 32 for 5 epochs per federated round.

Table 1: Performance Comparison on CIFAR-100 [28] for Image Classification with ResNet-18 [23]. We compare Cen-FedRAL against the decentralized FedRAL [29] and a Non-RL Federated Active Learning (FAL) baseline using fixed-size uncertainty sampling. The table reports mean values for ACC, weighted Average Precision (AP), and annotation Budget (as a percentage of the total available unlabeled data).

Method	**ACC (%)**	**AP (%)**	**Budget (%)**
Non RL - FAL	34.90	35.75	64
FedRAL (Decentralized) [29]	29.27	30.48	**10.16**
Cen-FedRAL (Ours)	**36.41**	**37.18**	10.36

Results: On PASCAL VOC 2012 (Table 2), Cen-FedRAL achieves +3.50% mAP over FedRAL while reducing the budget by 22.4%. Compared to Non RL-FAL, it maintains performance (+0.07% mAP) with 44.3% less labeling effort, showing superior localization sample selection.

Table 2: Performance Comparison on PASCAL VOC 2012 [14] for Object Detection with Faster R-CNN [44]. We compare Cen-FedRAL against the decentralized FedRAL [29] and a Non-RL Federated Active Learning (FAL) baseline using fixed-size uncertainty sampling. The table reports mean values for mAP and annotation Budget (as a percentage of the total available unlabeled data).

Method	**mAP (%)**	**Budget (%)**
Non RL - FAL	50.96	12.9
FedRAL (Decentralized) [29]	47.53	9.27
Cen-FedRAL (Ours)	**51.03**	**7.19**

4.4 Semantic Segmentation

Task Configuration: We evaluate pixel-level classification on PASCAL VOC 2012 using a U-Net with ResNet-18 encoder. Performance is measured via mean Intersection over Union (mIoU).

Results: Table 3 shows Cen-FedRAL achieves a +10.47% mIoU leap over decentralized FedRAL with a near-identical budget (11%). Compared to Non RL-FAL, it provides a +18.56% absolute mIoU improvement despite an 86.4% smaller budget. This shows that for high-dimensional state spaces like segmentation, centralized experience sharing is vital for developing efficient labeling policies.

Table 3: Performance Comparison on PASCAL VOC 2012 [14] for Semantic Segmentation with U-Net [45]. We compare Cen-FedRAL against the decentralized FedRAL [29] and a Non-RL Federated Active Learning (FAL) baseline using fixed-size uncertainty sampling. The table reports mean values for IoU (mIoU) and annotation Budget (as a percentage of the total available unlabeled data).

Method	mIoU (%)	Budget (%)
Non RL - FAL	47.38	81.2
FedRAL (Decentralized) [29]	55.47	11.09
Cen-FedRAL (Ours)	**65.94**	**11.03**

4.5 Zero-Shot Generalization and Policy Universality

To validate that the learned acquisition policy transcends specific tasks and domains, we evaluated the central DQN agent in a zero-shot setting. Crucially, the policy was used directly without any retraining or fine-tuning on the target datasets, isolating the agent's ability to identify universal indicators of sample utility.

Image Classification: The CIFAR-trained policy was applied to a SimpleCNN across three benchmarks (Table 4). It achieved high ACC on MNIST [30] (90.18%), Fashion-MNIST [59] (88.11%), and SVHN [39] (80.59%) using minimal budgets (1.07%–13.86%), proving robustness across diverse visual distributions.

Object Detection: The VOC-trained policy was transferred to a YOLOv8 [26] detector on VisDrone [69]. As shown in Table 5, it recovered over 75% of peak performance (28.0% mAP) using only 8.8% of labels, generalizing across complex architectures and viewpoints.

Semantic Segmentation: Evaluated on Cityscapes [10] using a UNet-ResNet18, the VOC-trained policy reached 41.62% mIoU with a restricted 10.5% budget (Table 6). This demonstrates the agent's capacity to navigate high-dimensional state spaces in disparate urban environments.

Table 4: Zero-Shot Transferability Results for Image Classification. The DQN [57] policy, pre-trained on CIFAR-100 [28], was applied to a SimpleCNN architecture across different datasets without retraining. The table reports values for ACC, and Budget (as a percentage of the total available data).

Dataset	ACC (%)	Budget (%)
MNIST [30]	90.18	1.07
Fashion-MNIST [59]	88.11	5.13
SVHN [39]	80.59	13.86

Table 5: Zero-Shot Transferability Results on VisDrone [69] for Object Detection. The DQN [57] agent was trained on PASCAL VOC 2012 [14] and tested on VisDrone using YOLOv8 [26], without retraining the RL policy. The table reports mean values for Average Precision (mAP) and Budget (as a percentage of the total available data).

Method	mAP (%)	Budget (%)
Full Supervision [63]	**37.00**	100.0
with Cen-FedRAL (Ours)	28.00	**8.80**

Table 6: Zero-Shot Transferability Results on Cityscapes [10] for Semantic Segmentation. The DQN [57] agent was trained on PASCAL VOC 2012 [14] and tested on Cityscapes using a UNet [45] model, without retraining the RL policy. The table reports values for IoU (mIoU) and Budget (as a percentage of the total available data).

Dataset	mIoU (%)	Budget (%)
Cityscapes [10]	41.62	10.50

5 Conclusions

We introduced Cen-FedRAL, a framework addressing experience isolation in FAL by centralizing RL training and employing PGER. Our evaluation across classification, detection, and segmentation demonstrates that aggregating fragmented trajectories into a cohesive policy significantly enhances sampling efficiency, and boosts performance. Notably, the learned policy exhibits robust zero-shot generalization, effectively transferring to unseen datasets without retraining. Despite these gains, Cen-FedRAL assumes trajectory sharing does not violate privacy, which may not hold in all sensitive domains. Additionally, the centralized buffer introduces a potential bottleneck in massive-scale deployments. Future work will investigate the integration of Differential Privacy mechanisms. Finally, we plan to extend Cen-FedRAL to multi-modal tasks and architecture-agnostic settings, enabling the fusion of heterogeneous uncertainty signals to optimize acquisition across complex, cross-domain environments.

Acknowledgements

This work is part of the TESTUDO project, which has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101121258. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Akuma, C.F., Tighe, B., Iwendi, C., et al.: Initialisation strategies and convergence behaviour in federated yolov8-based object detection for intelligent transportation systems. In: International Conference on Connected Intelligence for Industrial Applications (CI2A) (2026)
2. Allioui, H., et al.: A multi-agent deep reinforcement learning approach for enhancement of covid-19 ct image segmentation. Journal of Personalized Medicine (2022)
3. Amato, F., Della Bruna, C., Savoia, M., Prezioso, E., Piccialli, F.: Fedcom: Federated communication-efficient object detection model. IEEE Wireless Communications (2026)
4. van Bommel, J.: Active learning during federated learning for object detection. In: University of Twente, Bachelor Thesis (2024)
5. Caramalau, R., Bhattarai, B., Stoyanov, D.: Federated active learning for target domain generalisation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
6. Casanova, A., Pinheiro, P.O., Rostamzadeh, N., Pal, C.J.: Reinforced active learning for image segmentation. In: ICLR (2020)
7. Charalampakis, E., Mygdalis, V., Pitas, I.: Federated unsupervised semantic segmentation. arXiv preprint arXiv:2505.23292 (2026)
8. Charalampakis, E., Mygdalis, V., Pitas, I.: A trust-energy-aware scheduling framework for latency-constrained federated edge computing. Neurocomputing **674** (2026)
9. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: ICML. pp. 2089–2099 (2021)
10. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
11. Cui, Y., Hiremath, S.K., Ploetz, T.: Reinforcement learning based online active learning for human activity recognition. In: Proceedings of the 2022 ACM ISWC (2022)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
13. Deng, Z., Yang, Y., Suzuki, K.: Federated active learning framework for efficient annotation strategy in skin-lesion classification. Journal of Investigative Dermatology **145**(2), 303–311 (2025)
14. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
15. Fan, F.X., Ma, Y., et al.: Federated heterogeneous q-learning. In: AAMAS (2023)
16. Fang, M., Li, Y., Cohn, T.: Learning how to active learn: A deep reinforcement learning approach. In: EMNLP (2017)
17. Faulds, A.: Adaptive batch size selection in active learning for regression. Mathematics in Computer Science (MCS) (2022)
18. Garcia, D., Carias, J., Adão, T., Jesus, R., Cunha, A., Magalhães, L.G.: Ten years of active learning techniques and object detection: A systematic review. Applied Sciences **13**(19) (2023). https://doi.org/10.3390/app131910667
19. Godavarthi, D., Jose, D., Mohanty, S.N., et al.: Federated learning-based semantic segmentation framework for sustainable development. Egyptian Informatics Journal **30** (2025)

20. Hacohen, G., Dekel, A., Weinshall, D.: Active learning on a budget: Opposite strategies suit high and low budgets. arXiv preprint (2022)
21. Han, S., Park, S., et al.: Fedx: Unsupervised federated learning with cross knowledge distillation. In: ECCV (2022)
22. Han, Y.R., Park, D., Han, Y.S., Jung, J.Y.: Cost-efficient active transfer learning framework for object detection from engineering documents. Processes (2025)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
24. Henkel, J., Hoxha, G., Sumbul, G., Möllenbrok, L., Demir, B.: Annotation cost efficient active learning for content based image retrieval. In: IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium. pp. 4994–4997. IEEE (2023)
25. Ishibashi, H., Hino, H.: Stopping criterion for active learning based on error stability. arXiv preprint (2017)
26. Jocher, G., Chaurasia, A., Jing, Q.: Ultralytics YOLOv8. `https://github.com/ultralytics/ultralytics` (2023), version 8.0.0
27. Kim, S., Bae, S., Song, H., Yun, S.Y.: Logo: Local-global composition for federated active learning. arXiv preprint (2023)
28. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report (2009)
29. Lazaridis, Y., Kastellos, A., Psaltis, A., Daras, P.: Fedral: Cost-effective distributed annotation via federated reinforced active learning. In: 2024 IEEE International Conference on Big Data (BigData) (2024). `https://doi.org/10.1109/BigData62323.2024.10930759`
30. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998)
31. Lee, W.: Federated reinforcement learning-based uav swarm system for aerial remote sensing. Wireless Communications and Mobile Computing (2022)
32. Liang, Z., Zeng, X., Zhao, R., Luo, P.: Performance-guided reinforced active learning for object detection. In: ICASSP (2026)
33. Liu, S., Yang, S., Zhang, H., Wu, W.: A federated learning and deep reinforcement learning-based method for computation offload. Sensors (2023)
34. Liu, W., Xu, X., Wu, J., Jian, J.: Federated meta reinforcement learning for personalized tasks. Tsinghua Science and Technology (2024)
35. Liu, Z., Garg, N., Ratnarajah, T.: Multi-agent federated reinforcement learning strategy for mobile virtual reality delivery networks. IEEE Transactions on Network Science and Engineering (2023)
36. Mahmood, J.R., Lucas, J., Alvarez, J.M., Fidler, S., Law, M.T.: Optimising data collection for machine learning. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS) (2022)
37. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics (PMLR) (2017)
38. Na, S., Rouček, T., et al.: Federated reinforcement learning for collective navigation of robotic swarms. IEEE Transactions on Cognitive and Developmental Systems (2023)
39. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning (2011)

40. Pinto Neto, E.C., Sadeghi, S., Zhang, X., Dadkhah, S.: Federated reinforcement learning in IoT: Applications, opportunities and open challenges. Applied Sciences **13**(11), 6497 (2023). https://doi.org/10.3390/app13116497
41. Psaltis, A., Kastellos, A., et al.: Fedlid: Self-supervised federated learning for leveraging limited image data. In: ICCV (2023)
42. Qi, J., Zhou, Q., Lei, L., Zheng, K.: Federated reinforcement learning: Techniques, applications, and open challenges. Intelligence & Robotics **1**(1), 18–57 (2021). https://doi.org/10.20517/ir.2021.02
43. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
45. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
46. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. In: International Conference on Learning Representations (ICLR) (2016)
47. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017), https://arxiv.org/abs/1707.06347
48. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
49. Shankaranarayana, S.M.: In model uncertainty based active learning on tabular data using boosted trees. In: The Third International Conference on Artificial Intelligence and Machine Learning Systems (ACM). pp. 1–9 (2023)
50. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948)
51. Sheng, X., Gao, Z., Cui, X., Yu, C.: Federated reinforcement learning technology and application in edge intelligence scene. In: EIDWT (2023)
52. Song, D., Joseph, V.R.: Efficient active learning strategies for computer experiments. Technometrics (2025). https://doi.org/10.1080/00401706.2025.2546366
53. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge, Massachusetts, second edn. (2018)
54. Tenescu, A., Avramescu, C., Bercean, B., Marcu, M.: Lung nodule segmentation using federated active learning. In: Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA). pp. 17–21 (2023)
55. Tian, Y., Su, Y., Wang, Y., et al.: Privacy-preserving personnel detection in substations via federated learning with dynamic noise adaptation. Computers, Materials & Continua **86**(3) (2026)
56. Vo, T.N.: Active learning for multi-class image classification. arXiv preprint arXiv:2505.06825 (2025)
57. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**(3), 279–292 (1992)
58. Wong, Y.J., Tham, M.L., Kwan, B.H., Owada, Y.: Fedddrl: Federated double deep reinforcement learning for heterogeneous iot. Sensors (2023)
59. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

60. Xie, Z., Song, S.: Fedkl: Tackling data heterogeneity in federated reinforcement learning by penalizing kl divergence. IEEE Journal on Selected Areas in Communications (2023)
61. Xiong, W., Liu, Q., Li, F., Wang, B.: Personalized federated reinforcement learning: Balancing personalization and experience sharing. Expert Systems with Applications (2024)
62. Xu, M., Peng, J., Gupta, B.B., Kang, J., Xiong, Z., Li, Z.: Multiagent federated reinforcement learning for secure incentive mechanism in intelligent cyber-physical systems. IEEE Internet of Things Journal (2022)
63. Yan, H., Li, J., Shen, Q.: Air-yolo: A new small object detection model of uav. In: 2025 International Conference on Algorithm, Artificial Intelligence and Computer Vision (AAICV) (2025). https://doi.org/10.1109/AAICV66571.2025.00008
64. Yu, S., Chen, X., Zhou, Z., Gong, X., Wu, D.: When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management. In: IEEE (2021)
65. Zhan, X., Wang, Q., Huang, K.h., Xiong, H., Dou, D., Chan, A.B.: A comparative survey of deep active learning. arXiv preprint arXiv:2203.13450 (2022), https://arxiv.org/abs/2203.13450
66. Zhang, C., Wang, H., Mitra, A., Anderson, J.: Finite-time analysis of on-policy heterogeneous federated reinforcement learning. In: International Conference on Learning Representations (ICLR) (2024)
67. Zhang, F., Kuang, K., et al.: Federated unsupervised representation learning. Frontiers of Information Technology and Electronic Engineering (2023)
68. Zhang, M., Huang, T., Chen, W., Sun, X., Xi, X.: Feddw: an adaptive weight aggregation federated learning framework for multi-institutional collaborative polyp segmentation. Engineering Research Express **8**(1) (2026)
69. Zhu, P., Wen, L., Du, D., et al.: Visdrone-det2021: The vision meets drone object detection challenge results. ICCV Workshops (2021)
70. Zhu, T., Zhou, W., Ye, D., Cheng, Z., Li, J.: Resource allocation in iot edge computing via concurrent federated reinforcement learning. IEEE Internet of Things Journal (2021)