Adaptive Compression Becomes Unreliable: Fixed, Rule-Based, and Tabular-Q-Learning Policies for Federated ECG Learning

Abdul Haseeb Shaikh[1], Iram Mahar[1], Haleema Mahar[1], Sadaf Bibi[1*], and Fahmeeda Sobdar Mahar[2]

[1]Faculty of Emerging Sciences and Technology, Department of Artificial Intelligence,Aror University of Art, Architecture, Design and Heritage, Sukkur, Sindh, Pakistan

[2]Faculty of Emerging Sciences and Technology, Department of Multimedia and Gaming,Aror University of Art, Architecture, Design and Heritage, Sukkur, Sindh, Pakistan

*__Correspondence:__f24ari184@aror.edu.pk,sadaf.bibi.dev@gmail.com,ahaseeb.faculty@aror.edu.pk, f24ari34@aror.edu.pk, f24ari33@aror.edu.pk, & f24mmg43@aror.edu.pk

Citation | Mahar. I , Mahar. H , Bibi. S , Shaikh. A. H., Mahar. F. S,"Adaptive Compression Becomes Unreliable: Fixed, Rule-Based, and Tabular-Q-Learning Policies for Federated ECG Learning",IJIST, Vol. Vol. 8 Issue.9 pp 3595-3618, October 2026

DOI:

Received| Oct 03, 2026 **Revised|** Oct 16, 2026 **Accepted|** Oct 22, 2026 **Published|** Oct 24, 2026.

Wearable ECG monitoring needs federated learning for privacy and compression for bandwidth, and adaptive compression policies are often assumed to beat fixed ones. We tested that assumption on the MIT-BIH Arrhythmia Database with record-level, non-IID partitioning (Dirichlet alpha = 0.5, four eligible clients, 30 rounds, five paired seeds). Seven configurations were compared: FP32 FedAvg (E1), fixed 4-bit update quantization (E2), tabular Q-learning policies that adapt quantization (E3), sparsification (E4) or both (E5, E5s), and a hand-written rule (B3). Fixed 4-bit quantization cut total communication by 43.7% (9.067 to 5.107 MB). Its macro-F1 was lower than E1's in all five seeds (mean difference 0.033, 95% t-interval -0.012 to 0.078, exact Wilcoxon p = 0.0625), a consistent direction that does not establish a difference at this sample size. B3 had the highest mean composite minority F1 (0.2790) and the lowest communication (4.670 MB); its macro-F1 was close to E2's (0.4936 against 0.4958). No comparison against E5 was significant (all exact Wilcoxon p >= 0.31), and the two nominal paired-t results involving composite minority F1 did not survive correction for 18 exploratory tests (smallest adjusted p = 0.28). Two audits change how the results should be read. The uncompressed FP32 downlink carried 50% of E1's traffic and 85% to 97% of every compressed configuration's, which limits any uplink-only saving to about 11% relative to E2 in this setup. And the composite minority F1 is driven by ventricular beats: F1 for supraventricular and fusion beats stayed below 0.013 in all 35 runs. For practice, fixed 4-bit quantization is the safer default here, learned sparsification control needs ablation before use, and downlink compression and better supraventricular and fusion learning come first.

Keywords: Federated Learning; ECG Classification; Model Compression; Reinforcement Learning; Non-IID Data; Wearables; Communication Efficiency.

Introduction:

Arrhythmias are often intermittent, so one clinic recording can miss them, and a wearable that records ECG continuously can catch what a clinic visit does not. Training the models that read those recordings raises two problems at once. Raw ECG is sensitive medical data, so pooling it on a server is often not an option. A wearable also has little battery and bandwidth, so the training protocol has to be cheap to run and cheap to communicate. Federated Learning (FL) addresses the first problem: clients train locally and send only model updates, never raw data[1]. Healthcare reviews describe it as a practical route to learning across institutions [2],[3]. Federated ECG studies cover arrhythmia classification from 12-lead recordings [4], federated learning combined with differential privacy on multi-hospital ECG data [5]and asynchronous training for arrhythmia detection [6]. A scoping review of federated learning for cardiovascular disease detection collects the promises and open challenges of this setting[7].

Compression addresses the second problem. Quantization, sparsification with error feedback, and combinations of the two reduce the bytes a client sends in each round [8],[9],[10][11]. More recent work adapts the compression level during training: the bit-width can change across rounds and clients[12], [13], the server can supply side information that sets the compression rate [14], and weights can be clustered before transmission[15] Learning-based controllers have mostly been applied to who participates rather than to how much each participant compresses: multi-agent Reinforcement Learning (RL) has been used for client selection [16][17]. In federated ECG, communication has been attacked with knowledge distillation [18] and with hierarchical aggregation for very small models[19].

These studies leave a practical question open for wearable ECG. When a per-client controller chooses both the bit-width and the keep-ratio of every round, does it do better than a fixed setting or a short hand-written rule? Answering that needs one protocol in which only the policy changes. It also needs two safeguards that headline numbers can hide. A split made beat by beat can put one patient's beats on both sides of the train/test boundary. And a composite minority F1 can look reasonable while two of its three classes are never detected.

We built that protocol on MIT-BIH [20] All seven configurations share one 9,445-parameter convolutional model, one record-level, Dirichlet-partitioned non-IID split, the same five seeds, and the same held-out test records. They are an uncompressed FP32 baseline (E1), a fixed 4-bit update-quantization baseline (E2), four tabular-RL policies that adapt quantization, sparsification, or both for each client and round (E3 to E5s), and a hand-written rule that sees the same state as the RL agents (B3). Each configuration runs for 30 rounds. We report macro-F1, a composite minority F1 (the mean of the S, V and F class scores), per-class F1, and communication split into uplink and downlink. Figure 1 shows the pipeline; each stage lists its input, processing, and output.

Research Objectives:

The study has four objectives. Each has a stated measure and is tied to the experiments and result sections that address it.

To quantify what fixed 4-bit update quantization (E2) costs and saves relative to FP32 FedAvg (E1), measured as total communication in MB, final macro-F1, and final composite minority F1 over five paired seeds (In Section Results, Overview, Fixed 4-bit quantization against FP32, and Statistical Comparisons).

To test whether adaptive tabular-RL control (E3 to E5s) or the rule-based policy (B3) improves on E2 in macro-F1, composite minority F1, or total communication, using paired per-seed differences with intervals instead of comparing means alone (In Results Section, adaptive quantization against fixed 4-bit, and Statistical Comparisons)

To split total communication into uplink and downlink for every configuration and report the share of each, so that the ceiling for uplink-only savings is stated instead of assumed (In Results Section, Uplink and Downlink).

To audit what the composite minority F1 and the learned policies actually reflect, using per-class F1 for N, S, V, F and Q and per-agent Q-table coverage (In Results Section, per class results and Discussion Section, why the rule based policy is competitive?).

We call a difference statistically supported only if it survives the multiplicity correction described in section Statistical Analysis. All other differences are reported as descriptive, including differences that are consistent across all five seeds.



Figure 1. Experimental pipeline. Arrows show the order in which data and models pass between stages. Stage 1 supplies 109,464 annotated beats from 48 MIT-BIH records. Stage 2 converts them into normalized, class-labeled beat windows. Stage 3 separates 34 training records from 14 test records so that no record appears on both sides. Stage 4 turns the training records into five Dirichlet-skewed client datasets and drops the one client that fails the eligibility rule, leaving four. Stage 5 runs 30 federated rounds for each of the seven policies on those four clients, and stage 6 scores every round on the same 14 test records over five paired seeds. Stage 7 analyzes the saved per-seed results and logs. Stages 5 and 6 serve objectives O1 and O2, and stage 7 serves O3 and O4 in Research Objectives.

Novelty Statement:

This paper does not propose a new compression algorithm. Its contribution is a controlled comparison. In the studies reviewed in Related Work, we did not find one that evaluates a per-client learned controller for both quantization and sparsification against a fixed baseline and a hand-written rule on record-level non-IID ECG, with the same model, partition, seeds and test records for every policy. The comparison differs from the federated ECG studies in section Federated ECG under communication constraints, which concentrate on distillation, hierarchical aggregation, asynchrony, or privacy, and from the adaptive-compression studies in section communication-efficient federated learning and adaptive compression, which, as far as their abstracts indicate, are not evaluated on clinical ECG. Two measurements accompany the comparison and are not routinely reported together: uplink and downlink traffic counted jointly, and per-class F1 reported beside the composite minority score.

Contributions:

A matched comparison of seven fixed, learned and rule-based compression policies on record-level, non-IID MIT-BIH data, with five paired seeds, a common model, and common test records in Sections Methodology and Results.

Evidence that fixed 4-bit update quantization saves 43.7% of total communication and that FP32 FedAvg had the higher macro-F1 in all five seeds (mean gap 0.033). The direction is consistent, but the gap is not statistically established in sections Fixed 4-bit quantiation against FP32 and Statistical Comparisons.

A descriptive result that the rule-based policy had the highest mean composite minority F1 and the lowest total communication of the seven configurations, together with the finding that no comparison against E5 is significant and the composite differences involving B3 are nominal only.

A bidirectional communication account showing that the uncompressed downlink carries 50% to 97% of total traffic, which limits uplink-only savings to about 11% relative to E2 in this four-client, 9,445-parameter setup.

A per-class audit showing that F1 for supraventricular and fusion beats stays below 0.013 in all 35 runs, so the composite minority F1 is driven by ventricular beats.

Related Work:

ECG Classification and Class Imbalance:

The AAMI EC57 standard groups beats into five classes (N, S, V, F, Q) and is the usual label scheme for MIT-BIH work [21],[20]. Deep networks trained on much larger recordings have reached cardiologist-level rhythm classification [22] and have been applied to automatic diagnosis from 12-lead recordings [23], which suggests that the difficulty reported here may owe something to the size and label balance of MIT-BIH. In MIT-BIH the N class is 82.8% of beats, so overall accuracy mostly reflects N and per-class scores are more informative. Remedies for class imbalance include reweighted or focal losses [24] and resampling[25]. We use class-weighted cross-entropy only and test none of the other remedies.

Federated learning in Healthcare and ECG:

FedAvg [1] is the basic protocol: clients train locally and the server averages their models by sample count. Surveys place federated learning among the main routes to privacy-preserving medical AI and catalog its open problems [2], [3],[7] Federated ECG studies include arrhythmia classification from 12-lead recordings [4], asynchronous updates [6], and federated learning combined with differential privacy across hospitals [5]. Their emphasis is on accuracy, staleness, or privacy, not on a per-round compression policy. Secure aggregation, which hides individual updates from the server [26], is a separate layer that we do not implement.

Non-IID Data and Normalization:

FedAvg degrades when client data are non-IID[27] . Label skew is commonly simulated with a Dirichlet distribution over class proportions[28], which we follow. Proposed remedies include a proximal term in the local objective[29], control variates [30], dynamic regularization [31], and adaptive server optimizers [32]. FedBN argues for keeping batch-normalization statistics local when client features differ [33], and Group Normalization avoids batch statistics altogether [34], which is why ECGNet uses it. We use plain FedAvg without drift correction, so our results describe compression policies on a standard baseline and may change under a drift-correcting optimizer.

Communication-Efficient Federated learning and Adaptive Compression:

Early work reduced upload size with structured and sketched updates [8]. QSGD[9] and FedPAQ[10] analyze quantized training, and sparsification with a memory of the dropped coordinates, known as error feedback, keeps top-k selection from discarding information permanently [11].

Sparse ternary compression has been studied under non-IID clients[35]. Adaptive schemes change the compression with time or client. DAdaQuant adapts the quantization level over rounds and across clients [12], and Mao et al. adjust the level during training [13]. Isik et al. use information the server already holds to set the compression [14], and Tsouvalas et al. cluster weights and add server-side distillation [15]. Amiri et al. compress the global model that is broadcast as well as the uploads [36], which is the direction our accounting shows matters most in a small federation. As far as their abstracts indicate, none of these studies is evaluated on clinical ECG, and none compares a learned per-client controller with a hand-written rule under one protocol. Their datasets, partitions and metrics also differ, so their reported numbers cannot be compared directly; this is why we run every policy under one protocol instead of quoting published results.

Reinforcement Learning for Federated Control:

FedMarl uses multi-agent RL to choose which clients take part in a round, balancing accuracy, latency and communication [16], and FedQMIX uses multi-agent RL to select nodes [17]. Heuristic client selection under resource limits is older [37]. These controllers decide who participates, whereas the controller studied here decides how much a participating client compresses. Tabular Q-learning [38] is small enough to run on a wearable, but the number of observations it needs grows with the number of state-action cells, and that becomes the central issue.

Federated ECG Under Communication Constraints:

BiFedKD replaces weight exchange with bidirectional knowledge distillation for non-IID, long-tailed ECG monitoring [18], and Family-FL groups clients hierarchically for models under 5 KB on constrained wearables [19]; both are 2026 preprints. Both attack communication cost through the training protocol, not through a per-round compression policy. Reporting the full energy and compute cost of a method, beyond its accuracy, is a broader call in efficient machine learning [39]; our energy term is a proxy for that reason.

Comparison and Research Gap:

Table 1 places the closest studies next to this one. The limitations that matter for our question are the same across rows: the adaptive-compression studies use general benchmarks and no learned per-client controller, the RL studies control participation and not compression, and the ECG studies change the protocol instead of the per-round compression policy. In the studies reviewed here we did not find one that (i) compares fixed, learned and rule-based compression under one protocol, (ii) uses record-level non-IID ECG with a class-imbalanced test set, and (iii) reports uplink and downlink together with per-class F1. This does not show that no such study exists, only that we did not find one.

Table 1. Closest prior studies compared with this study. Entries summarize each study's stated contribution as given in its abstract and may omit details. RL means a reinforcement-learning controller.

Study	Domain	What is adapted or compressed	RL	Relation to this study
[10]	General FL	Quantized uploads with periodic averaging	No	Fixed-quantization counterpart of E2
[11]	Distributed SGD	Top-k sparsification with error feedback	No	Same error-feedback mechanism as E4 to B3
[12]	General FL	Quantization level over rounds and clients	No	Adaptive quantization without learned control
[13]	General FL	Quantization level during training	No	Adaptive quantization without learned control
[14]	General FL	Compression guided by server-side information	No	Adaptive compression with a different signal
[36]	General FL	Quantized global-model broadcast	No	Addresses the downlink, which dominates here
[16]	General FL	Client selection	Yes (multi-agent)	Learned control of participation, not compression
[17]	General FL	Node selection	Yes (multi-agent)	Learned control of participation, not compression
[18]	ECG	Bidirectional knowledge distillation	No	ECG; communication cut by protocol redesign
[19]	ECG	Hierarchical aggregation for sub-5 KB models	No	ECG; communication cut by topology
This study	**ECG**	**Per-client quantization and sparsification**	**Yes (tabular Q), plus a rule and fixed baselines**	**Matched comparison with bidirectional accounting and per-class audit**

Problem Formulation:

Let C be the set of eligible clients, n_i the number of local training beats of client i, and w the shared model. Each round, client i trains locally and returns an update, and the server aggregates by sample count:

$$w_{t+1} = \sum_{i \in C} \frac{n_i}{\sum_{j \in C} n_j} \cdot w_{t+1}^{i} \quad (1)$$

Compression changes what is transmitted for each client but not these weights. Beats carry one of five AAMI classes, Y = {N, S, V, F, Q}. Macro-F1 is the unweighted mean of the five per-class F1 scores, and the composite minority F1 is the mean of the S, V and F scores. Section per-class results shows that the composite is dominated by V.

Each client observes a state $s_t = (b_t, n_t, u_t)$ made of battery level, network quality and prediction uncertainty, discretized into 5, 3 and 5 bins (75 states). The action $a_t = (q_t, r_t)$ chooses a bit-width q_t in {2, 4, 6, 8} and a keep-ratio r_t in {1%, 5%, 20%}, which gives 12 joint actions. Section Agent design, simulated gives the reward and update equations. The rule-based baseline maps the same state to an action through a fixed decision list instead of a learned table.

Methodology:

Figure 2 shows the components. Each eligible client trains ECGNet on its private beats, reads its context (battery, network, prediction uncertainty), lets a policy choose a quantization level and keep-ratio, compresses the update with error feedback, and uploads it. The server reconstructs the updates, averages them by sample count, and broadcasts the updated model as dense FP32. Raw ECG never leaves a client.

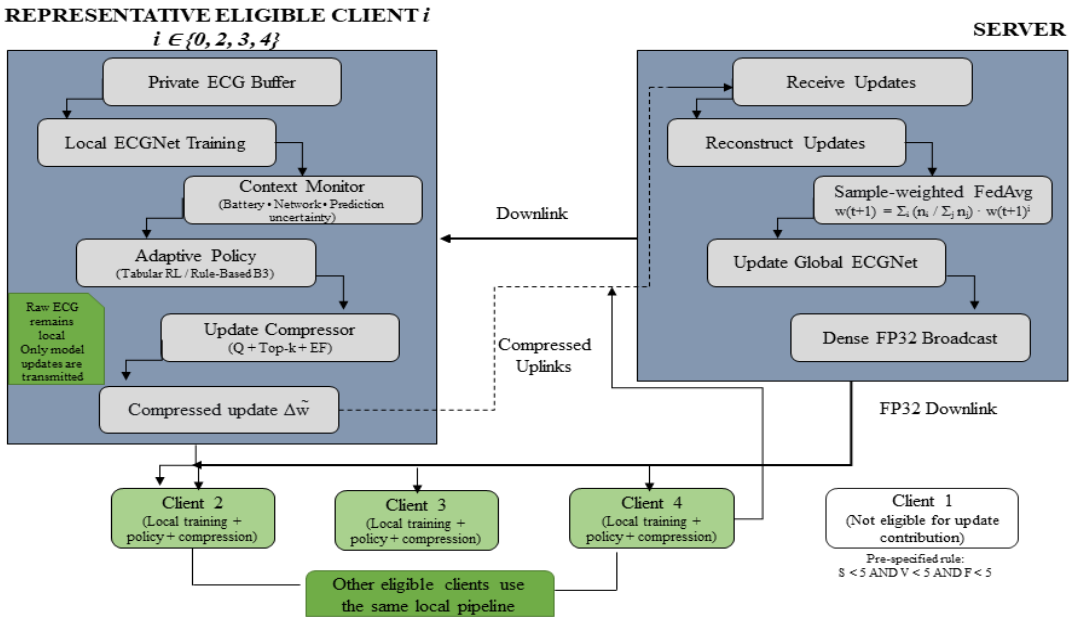

Figure 2. System architecture. A representative eligible client (clients 0, 2, 3 and 4 follow the same pipeline; clients 2 to 4 are drawn) trains ECGNet on its private ECG buffer, monitors battery, network and prediction uncertainty, lets the adaptive policy (tabular Q-learning or the rule-based B3) choose the quantization level and keep-ratio, compresses the update with error feedback, and sends it to the server. The server reconstructs the updates, applies sample-weighted FedAvg, and broadcasts the updated global model as dense FP32. Client 1 fails the eligibility rule in Client Eligibility and takes no part.

Dataset:

We use the MIT-BIH Arrhythmia Database [20]through a public Kaggle mirror that provides raw per-record signal and annotation files; this is what makes record-level splitting

possible. The database has 48 half-hour records from 47 subjects, sampled at 360 Hz. After mapping beat annotations to the five AAMI EC57 classes[21] and discarding non-beat symbols, we extract 109,464 beats: N 90,606 (82.77%), S 2,779 (2.54%), V 7,235 (6.61%), F 802 (0.73%), and Q 8,042 (7.35%). Each beat is a 200-sample window, 90 samples before and 110 after the annotated R peak, normalized by its own mean and standard deviation so that no statistic computed on test data enters training. Only the first signal column of each record is used.

Record-level Split and Client Partition:

The unit of separation is the record, not the beat. Fourteen records (100, 101, 102, 108, 113, 114, 115, 121, 208, 214, 217, 219, 232 and 233) form the held-out test set. They were chosen so that the test set holds at least 100 beats of each of S, V and F. The other 34 records form the training pool of 79,460 beats; the test set has 30,004 beats (N 21,894, S 1,457, V 2,370, F 391, Q 3,892). MIT-BIH has 48 records from 47 subjects because records 201 and 202 are documented as coming from the same person. Both are in the training pool, so no subject appears on both sides of the split; we still describe the split as record-level because we did not check subject identity beyond that known pair. The training records are assigned to five clients with a Dirichlet distribution over class proportions (alpha = 0.5) [28], which makes the class mix differ strongly between clients (Figure 3).

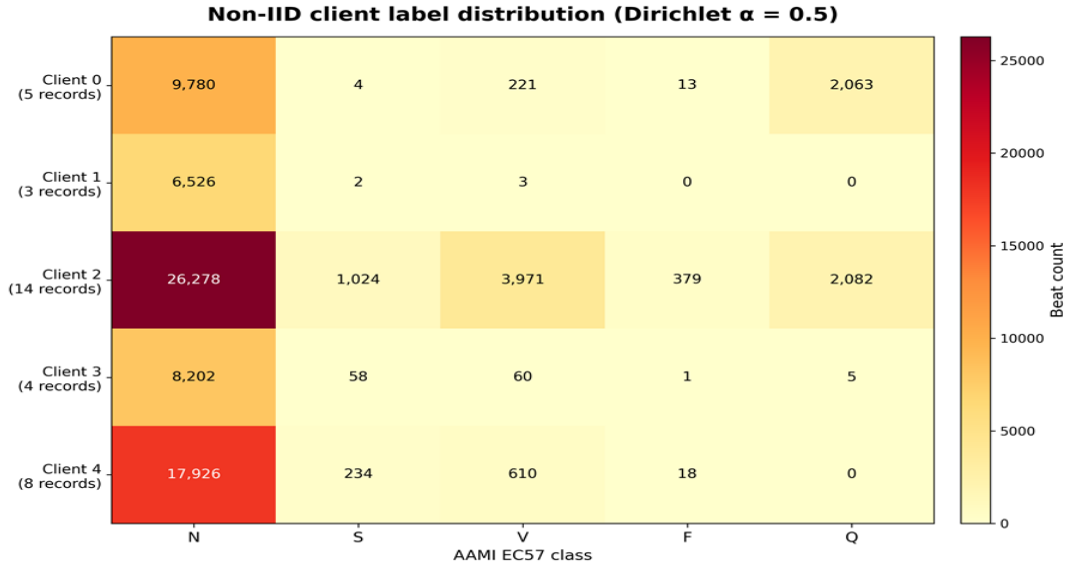

Figure 3. Class composition of the five client datasets after the Dirichlet partition (alpha = 0.5). Client 1 holds 2 S, 3 V and 0 F beats and is excluded by the eligibility rule in Section 4.4; the other four clients take part in training.

Model and local Training:

ECGNet is a three-block one-dimensional CNN followed by a linear classifier; Table 2 lists the layers. Each block applies a convolution, Group Normalization, ReLU, and pooling. Batch normalization keeps running statistics that depend on each client's label mix and do not average meaningfully across clients [33], whereas Group Normalization computes statistics within each example and has no such state [34]. Each client trains for two local epochs per round with Adam (learning rate 5e-4, batch size 64) on a cross-entropy loss weighted by the inverse square root of the client's class frequencies, normalized and clipped to the range 0.3 to 3.

Client Eligibility:

A client is excluded when it holds fewer than 5 beats of every one of the S, V and F classes; otherwise it is eligible. The threshold was fixed before the final run and applied identically to every seed and configuration. Table 3 shows the effect on this partition. Client 1

has 3 records and 6,531 beats, which is 8.2% of the 79,460 training beats, but only 2 S, 3 V and 0 F beats. That is 5 of the 6,598 training S, V and F beats (0.08%), so its exclusion removes almost no minority-class signal and the per-class findings in per-class results cannot plausibly be an artifact of it. The exclusion still has three consequences. The federation has four clients, not five, and both uplink and downlink totals would be roughly one quarter larger with a fifth participant, since each term scales with the number of participants. The rule is permissive: client 3 stays in with 58 S and 60 V beats but a single F beat, so eligibility does not guarantee usable data for every minority class. And a deployment would need its own rule, because the threshold of 5 is a simulation convenience that was not tuned or validated. We did not rerun the study with client 1 included or with other thresholds.

Table 2. ECGNet architecture. The input is one 200-sample beat. Parameter counts include convolution or linear weights and biases and the Group Normalization scale and shift.

Block	Layers	Parameters
1	Conv1d (1 to 16 channels, kernel 5), GroupNorm (4 groups), ReLU, MaxPool (2)	128
2	Conv1d (16 to 32 channels, kernel 5), GroupNorm (8 groups), ReLU, MaxPool (2)	2,656
3	Conv1d (32 to 64 channels, kernel 3), GroupNorm (8 groups), ReLU, adaptive average pooling	6,336
Classifier	Linear (64 to 5 classes)	325
Total	**36.9 KB at FP32**	**9,445**

Table 3. Client partition and eligibility (training beats per class). A client is excluded when it has fewer than 5 beats of each of S, V and F.

Client	Records	N	S	V	F	Q	Total	Eligible
0	5	9,780	4	221	13	2,063	12,081	Yes
1	3	6,526	2	3	0	0	6,531	No
2	14	26,278	1,024	3,971	379	2,082	33,734	Yes
3	4	8,202	58	60	1	5	8,326	Yes
4	8	17,926	234	610	18	0	18,788	Yes
All	**34**	**68,712**	**1,322**	**4,865**	**411**	**4,150**	**79,460**	**4 of 5**

Compression, Action Space and Warm-up:

Quantization is symmetric min-max at 2, 4, 6 or 8 bits, applied to the model update (the difference between the local and global weights). Sparsification keeps the top-k coordinates of that update by magnitude at keep-ratios of 1%, 5% or 20%. A persistent per-client error-feedback residual carries dropped and rounded-off components into the next round [11]. E1 sends the update dense at 32 bits and E2 sends it dense at 4 bits. E3 chooses only the bit-width and always sends dense updates (keep-ratio 1.0). E4 chooses only the keep-ratio and always sends 32-bit values. E5 and E5s choose both. E5s also ranks coordinates by an importance-weighted score built from class-wise gradients on the calibration data; this is a heuristic proxy for class-aware sparsification, not a per-class quota. B3 selects both from a fixed decision list. Table 4 lists the configurations as they were run.

The learned configurations use a warm-up of ten rounds. According to the saved action logs, E3, E5 and E5s are restricted to 6 or 8 bits during rounds 1 to 10, and E4, which has no bit-width choice, is restricted to keep-ratios of 5% or 20%. E5 and E5s do not restrict the keep-ratio during warm-up: a 1% keep-ratio was selected at least once in those rounds. B3 follows its rule from round 1 with no warm-up restriction. Comparisons between B3 and the RL configurations should be read with that difference in mind.

Table 4. The seven configurations as run, checked against the saved action logs. E2 is per-round model-update quantization, not post-training quantization of weights. All configurations except E1 use error feedback.

ID	Configuration	Bit-width	Keep-ratio	Control	Warm-up restriction (rounds 1 to 10)
E1	FP32 FedAvg	32	1.0 (dense)	None	None
E2	Fixed 4-bit update quantization	4	1.0 (dense)	None	None
E3	Adaptive quantization only	2, 4, 6, 8	1.0 (dense)	Tabular Q	Bits 6 or 8
E4	Adaptive sparsification only	32	1%, 5%, 20%	Tabular Q	Keep-ratio 5% or 20%
E5	Joint adaptive control	2, 4, 6, 8	1%, 5%, 20%	Tabular Q	Bits 6 or 8
E5s	Joint, importance-weighted sparsification proxy	2, 4, 6, 8	1%, 5%, 20%	Tabular Q	Bits 6 or 8
B3	Rule-based decision list	2, 4, 6, 8	1%, 5%, 20%	Fixed rule	None

Table 5. Main results at the final round, mean ± SD and median over five seeds. Communication is cumulative over 30 rounds with four eligible clients (MB = 10^6 bytes); the per-round column divides by 30. Composite minority F1 is the mean of the S, V and F class F1.

Config	Macro-F1 (mean ± SD)	Median	Composite minority F1 (mean ± SD)	Median	Total (MB)	Per round (MB)
E1	0.5285 ± 0.0201	0.5335	0.2630 ± 0.0305	0.2744	9.067	0.3022
E2	0.4958 ± 0.0358	0.4937	0.2540 ± 0.0258	0.2649	5.107	0.1702
E3	0.5007 ± 0.0358	0.4873	0.2550 ± 0.0335	0.2673	5.354	0.1785
E4	0.4333 ± 0.1515	0.5359	0.2412 ± 0.0581	0.2664	5.173	0.1724
E5	0.4909 ± 0.0989	0.5535	0.2488 ± 0.0683	0.2831	4.759	0.1586
E5s	0.4653 ± 0.0494	0.4429	0.2776 ± 0.0340	0.2903	4.767	0.1589
B3	0.4936 ± 0.0674	0.4842	0.2790 ± 0.0273	0.2822	4.670	0.1557

Table 6. Paired comparisons of E5 with each other configuration (E5 minus X), five seeds. Intervals are 95% t-intervals for the mean paired difference; p is the exact two-sided Wilcoxon signed-rank p-value. No p-value is below 0.05 (the smallest attainable with five pairs is 0.0625).

X	Δ macro-F1	95% interval	p	Δ composite F1	95% interval	p
E1	-0.0376	[-0.1677, +0.0925]	0.8125	-0.0142	[-0.1215, +0.0931]	1.0000
E2	-0.0048	[-0.0976, +0.0880]	1.0000	-0.0052	[-0.1087, +0.0982]	1.0000
E3	-0.0097	[-0.1431, +0.1236]	0.8125	-0.0062	[-0.0959, +0.0836]	1.0000
E4	+0.0576	[-0.2276, +0.3429]	0.6250	+0.0076	[-0.1391, +0.1543]	1.0000
E5s	+0.0256	[-0.0879, +0.1392]	0.6250	-0.0288	[-0.1448, +0.0872]	0.6250
B3	-0.0027	[-0.0797, +0.0743]	1.0000	-0.0302	[-0.1208, +0.0603]	0.3125

Agent Design, Simulated State and Reward:

E3 to E5s use independent tabular Q-learning[38]. Each eligible client keeps its own table with 75 states and 4 (E3), 3 (E4) or 12 (E5, E5s) actions, initialized to 1 plus Gaussian noise (SD 0.01). Tables are not shared across clients or seeds. After each round the client updates the entry for its previous state s and action a using the reward R it received and the new state s':

$$Q(s,a) \leftarrow Q(s,a) + \eta\,[\,R + \gamma\, max_{a'}\, Q(s',a') - Q(s,a)\,] \quad (2)$$

with learning rate $\eta = 0.1$ and discount $\gamma = 0.95$; the last round is treated as terminal. Actions are chosen epsilon-greedily with $\varepsilon_t = \max(0.05, 0.35 \times 0.99^t)$ at round t. B3 reads the same state and applies a decision list: if battery < 0.2, use 4-bit at 1%; otherwise, if network < 0.33, use 2-bit at 1%; otherwise, if uncertainty > 0.7, use 8-bit at 20%; otherwise use 6-bit at 5%. Battery, network and uncertainty are produced for each client and round as follows. Battery level is

$$b_t = clip(\,1 - t/T - (\,0.005\, q_{t-1}/8 + 0.002\, r_{t-1}/0.2\,) + \xi_b, 0.05, 1\,) \quad (3)$$

where T = 30, q_{t-1} and r_{t-1} are the previous round's bit-width and keep-ratio, and ξ_b is Gaussian noise with SD 0.03. The battery drains steadily and drains slightly faster when the client sent more bits. Network quality is

$$n_t = clip(\,0.5 + 0.35\, sin(t/5) + \xi_n, 0, 1\,) \quad (4)$$

with Gaussian noise ξ_n of SD 0.1, a slow oscillation that stands in for changing link conditions. Prediction uncertainty is not simulated: it is the normalized Shannon entropy of the softmax output (divided by log 5) of the current global model, averaged over the client's S, V and F calibration beats (over all its calibration beats if it has fewer than five). Each client draws its own noise sequence from a seeded generator.

The bins are five equal-width bands for battery, three for network (below 1/3, 1/3 to 2/3, above 2/3), and five equal-width bands for uncertainty. We chose 5, 3 and 5 before the final run to keep the state space at 75 states, small enough that the 30 observations each client produces can visit a meaningful share of it, and because the bands resemble the coarse indicators that devices commonly expose, such as battery in steps of about 20% and link quality as poor, fair or good. These are design choices, not values fitted to device traces. We did not test other bin edges, and the simulated battery and network processes are not calibrated against real wearable telemetry.

Each client sets aside a class-stratified calibration set before training: up to 30 beats per class, capped at one fifth of the client's beats of that class. These beats are excluded from the client's gradient updates, so they are disjoint at the beat level, but they come from the same training records and therefore share patients with the client's training data. They are not a cross-patient measure. The union over the four eligible clients is the pooled calibration set of 391 beats (N 120, S 72, V 102, F 36, Q 61). The calibration split is fixed across seeds and configurations. The reward is

$$R_{i,t} = (1-\alpha)\, Q_{i,t}^{loc} + \alpha\, Q_t^{glob} - \lambda_c \frac{B_{i,t}}{B_{ref}} - \lambda_e \frac{n_i}{n_{ref}} \quad (5)$$

where Q = 0.5 × macro-F1 + 0.5 × composite minority F1. The local term Q^{loc} is computed on client i's calibration beats with the model reconstructed from its compressed update, so it reflects compression damage. The global term Q^{glob} is the same quantity for the aggregated global model on the pooled calibration set, evaluated after aggregation and shared by all clients in that round. $B_{i,t}$ is the number of bits client i transmitted (values, indices and scales), $B_{ref} = 5 \times 10^6$ bits, n_i is the client's number of local training beats, and $n_{ref} = 20{,}000$. The weights are $\alpha = 0.5$, $\lambda_c = 0.15$ and $\lambda_e = 0.05$. The energy term depends only on the amount of local data, not on the action, so it shifts all action values of a client equally and cannot change that client's preferred action; it is a constant offset, not an energy cost. The reward is a weighted scalarization, not a constrained formulation.

Communication Accounting:

Uplink cost of a dense update is the number of parameters (9,445) times the bits per value, plus a 32-bit scale for each of the 14 parameter tensors when the values are quantized; a 32-bit update has no scale term. A sparse update costs k × (bits per value + ⌈log2 9,445⌉) bits plus the scales, where k is the number of selected coordinates taken from the sparsifier. We do not recount k from nonzero values, because a selected coordinate can round to zero and still needs its index sent. Downlink is the dense FP32 model sent to every eligible client each round, 9,445 × 32 × 4 bits per round. We report megabytes (MB, 10^6 bytes, equal to bits divided by 8,000,000) summed over the 30 rounds ("cumulative") and divided by 30 ("per round"). Megabits are not used anywhere in this paper.

Evaluation Protocol:

All metrics are measured on the pooled 14 held-out records after the final (30th) round; we report that round, not the best round. Each configuration is run with five seeds (42, 123, 2024, 7 and 99). A seed fixes model initialization, local data order, exploration, and the simulated device noise; the train/test split, the client partition and the calibration split are fixed across seeds, so seed-to-seed variability does not include variability from the partition. The same seeds are used for every configuration, so comparisons are paired. The full suite ran on a single GPU in about 2.4 hours.

Statistical Analysis:

For each paired comparison we report the mean paired difference, its 95% t-interval, the exact two-sided Wilcoxon signed-rank p-value, the paired-t p-value, the standardized paired effect size d_z (mean difference divided by the SD of the differences), and the number of seeds in which the first configuration was higher. With five pairs the smallest attainable exact Wilcoxon p-value is 0.0625, so that test cannot reach 0.05 for any effect size. We therefore read it together with the interval and the effect size, and treat the paired-t p-value as a nominal check whose normality assumption cannot be verified with five pairs. Percentile bootstrap intervals are not used, because they are known to be too narrow at this sample size. All tests are exploratory. We apply a Holm correction across the 18 tests in Table 6 and Table 7. A difference is called statistically supported only if its corrected p-value is below 0.05; every other difference is descriptive.

Reproducibility:

Code and result files are available at the repository listed on the title page. The per-seed values behind every table are in Appendix Table A2. A first execution of the same code and seeds was retained for E4; its final macro-F1 differed from the canonical run by up to 0.078 for identical seeds (Appendix Table A3). We did not force deterministic GPU kernels and did not isolate the source of this variability, which we describe as implementation-level variability and not as a finding. All results in this paper come from the canonical run only.

Results:

Overview:

Table 5 reports the final-round macro-F1, the composite minority F1 and the total communication of all seven configurations, as mean ± SD with the median over five seeds. Figure 4 shows the round-by-round curves. E1 to E3 are close to their final level by about round 10, whereas E4, E5, E5s and B3 climb later and fluctuate more.

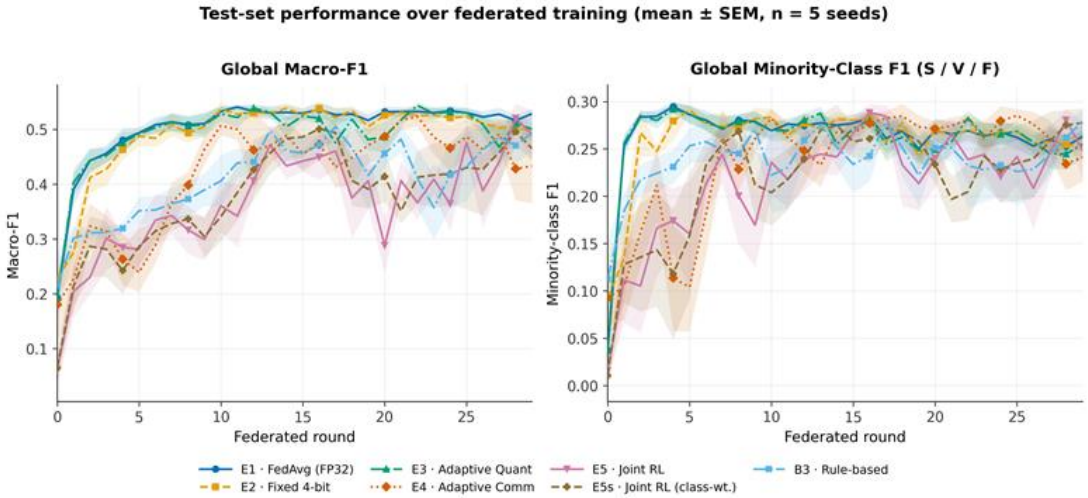

Figure 4. Test-set macro-F1 (left) and composite minority F1 (right) over 30 federated rounds, mean ± SEM over five seeds, for all seven configurations. Composite minority F1 is the mean of the S, V and F class F1.

Fixed 4-bit Quantization Against FP32:

E2 reduces total communication by 43.7% relative to E1 (5.107 against 9.067 MB). Its final macro-F1 is lower than E1's in all five seeds, by 0.033 on average (0.4958 against 0.5285). The gap is larger than E1's seed-to-seed SD (0.020) and about as large as E2's own (0.036). The 95% t-interval for the paired difference runs from -0.012 to 0.078, the exact Wilcoxon p-value is 0.0625 (the smallest value five pairs allow) and the paired-t p-value is 0.113. The data therefore show a consistent direction but do not establish a difference (Table 7). For composite minority F1 the gap is 0.009 (0.2630 against 0.2540), smaller than either configuration's SD; the paired-t p-value is 0.016 before correction and does not survive the Holm correction across 18 tests (smallest adjusted p = 0.28). E2's seed-to-seed SD (0.0358) is close to E3's and well below those of E4, E5, E5s and B3 (0.049 to 0.152); only E1's (0.0201) is lower. We read E2 as a low-variance efficiency baseline with a small macro-F1 cost that appears in every seed but is not statistically confirmed.

Adaptive Quantization Against fixed 4-Bit:

E3's mean macro-F1 (0.5007) is numerically above E2's (0.4958) and its mean composite minority F1 (0.2550) is above E2's (0.2540). E3 never sparsifies (keep-ratio 1.0 throughout; only the bit-width adapts), so its communication tracks the dense cost of E2 and comes in 4.8% higher (5.354 against 5.107 MB). Because E3 uses more communication than E2, it is not a Pareto improvement. The paired difference in macro-F1 (E2 minus E3) is -0.005 with a 95% t-interval from -0.058 to +0.048 and E2 is higher in 2 of 5 seeds, so adaptive quantization is competitive with a fixed 4-bit policy in this protocol but not shown to be better or worse.

Adaptive Sparsification:

E4 has the largest seed-to-seed spread of any configuration (macro-F1 SD 0.1515, 7.5 times E1's 0.0201). Two of its five seeds finish below 0.30 (seed 42 at 0.284 and seed 2024 at 0.251) after reaching 0.566 at round 22 and 0.564 at round 17, while the other three finish between 0.536 and 0.550 (Figure 5). In the retained first execution of the same code and seeds, the same two seeds again finished below 0.31 (0.278 and 0.306), although the other seeds differed by up to 0.078 (Appendix Table A3). We therefore regard the collapse of these two seeds as repeatable and the exact trajectory of the others as not repeatable.

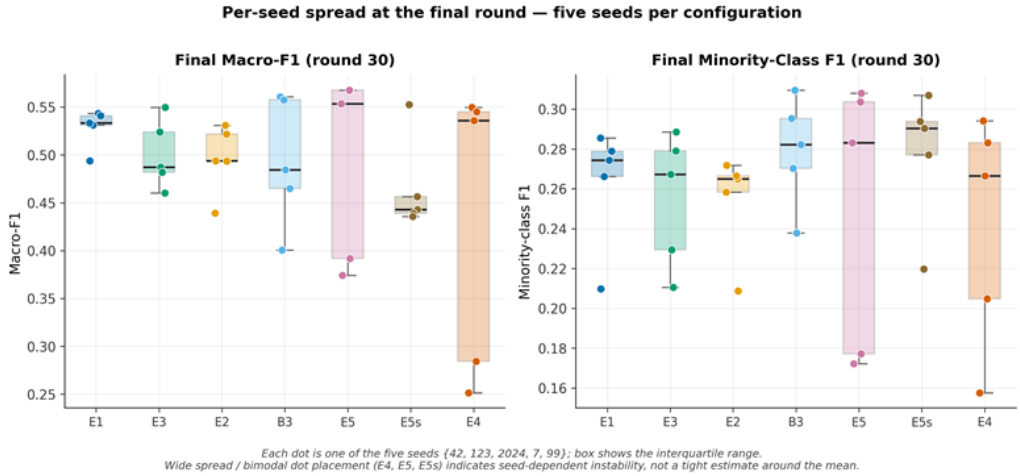

Figure 5. Final-round macro-F1 (left) and composite minority F1 (right) for each of the five seeds (points) and the interquartile range (boxes) for all seven configurations. Wide spread, as for E4 and E5, means that the mean is a poor summary of a single run.

Joint Adaptive Control:

E5 shows a separated high/low pattern across seeds: three seeds finish at 0.554 to 0.568 and two at 0.374 and 0.392 (Figure 5). E5s, which adds the importance-weighted sparsification proxy, has the highest mean composite minority F1 of the four RL configurations (0.2776) but a lower mean macro-F1 than E5 (0.4653 against 0.4909); one of its seeds finishes at 0.553 and the other four between 0.436 and 0.456. This is a trade-off between the two metrics, not a clear improvement.

Rule-Based Policy:

B3 has the highest mean composite minority F1 of the seven configurations (0.2790, against 0.2776 for E5s) and the lowest total communication (4.670 MB), while its mean macro-F1 (0.4936) is within 0.003 of E2's (0.4958). It learns nothing from experience. Its saved action logs show how it behaves: across 600 client-round decisions it chooses 2-bit 18.5% of the time, 4-bit 18.3%, 6-bit 56.7% and 8-bit 6.5% (mean 5.0 bits), with keep-ratios of 1% in 36.8% of decisions, 5% in 56.7% and 20% in 6.5% (mean 4.5%). The 6-bit, 5% branch of the rule dominates. Against E2, the paired difference in macro-F1 is -0.002 (95% t-interval -0.069 to +0.065), so the two are numerically close but not shown to be equivalent. The difference in composite minority F1 is +0.025 (interval +0.005 to +0.045, B3 higher in 4 of 5 seeds, paired-t $p = 0.027$), which does not survive correction. B3's macro-F1 SD (0.0674) is higher than that of E1, E2 or E3, and some of its seeds also swing widely (seed 42 falls to 0.232 at round 20 before recovering to 0.561). We describe B3 as the strongest descriptive low-communication operating point in this study, not as a statistically better policy. Figure 6 gives the per-class view behind the composite scores (analyzed in per-class results), and Figure 7 places each configuration on the communication and macro-F1 plane.

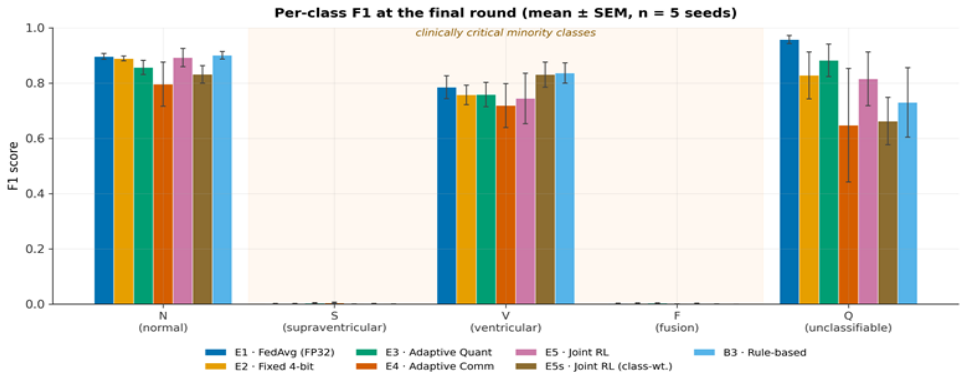

Figure 6. Final-round F1 for each AAMI class, mean ± SEM over five seeds, for the seven configurations. The S and F bars are close to zero for every configuration.

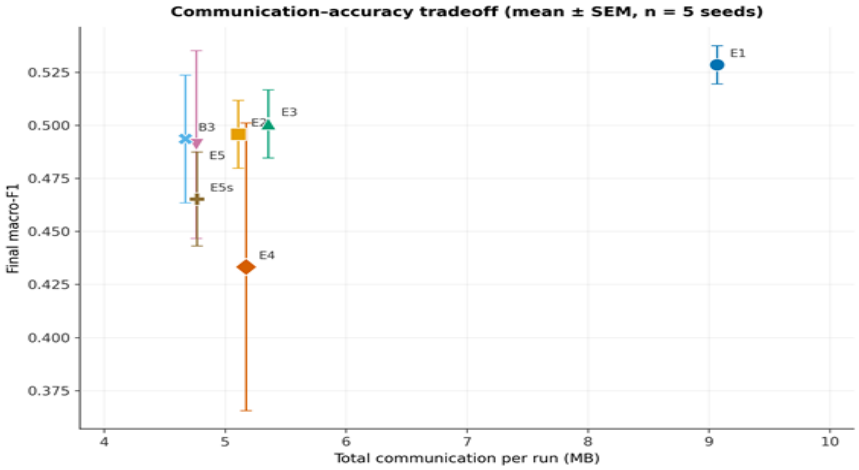

Figure 7. Total communication (cumulative MB) against final macro-F1, mean ± SEM over five seeds, for the seven configurations.

Statistical Comparisons:

Table 6 compares E5 with each other configuration across the same five seeds. No comparison against E5 reaches conventional significance: every exact Wilcoxon p-value is 0.31 or larger and every paired-t p-value is above 0.4. This limits what the study can say about E5 specifically; it does not show that the configurations perform alike. Table 7 adds the three direct comparisons readers are most likely to ask about, which do not involve E5. The two nominal paired-t results (E1 against E2 and B3 against E2, both on composite minority F1) do not survive Holm correction across the 18 tests (smallest adjusted p = 0.28). Because the exact Wilcoxon test cannot go below 0.0625 with five pairs, these p-values are better read together with the intervals than as pass or fail results.

A sensitivity calculation shows how little five seeds can resolve. For a paired t-test with 80% power at a two-sided 5% level, five pairs detect only a standardized effect d_z of about 1.7. With the observed SD of paired differences (0.036 for E1 against E2, 0.062 for E5 against B3, and 0.23 for E5 against E4), that corresponds to macro-F1 differences of about 0.06, 0.10 and 0.39. Ten seeds would lower these to about 0.04, 0.06 and 0.23. These figures assume normally distributed differences and use the observed variance, so they are rough guides and not a formal power analysis. Most differences among the configurations are smaller than these thresholds.

Table 7. Direct paired comparisons that do not involve E5 (first minus second), five seeds. Wins is the number of seeds in which the first configuration was higher; pW is the exact Wilcoxon p-value, pT the paired-t p-value, and d_z the standardized paired effect. No result is statistically supported after Holm correction across 18 tests.

Comparison	Metric	Mean Δ	95% interval	pW	pT	dz	Wins
E1 − E2	Macro-F1	+0.0327	[-0.0122, +0.0777]	0.0625	0.113	0.90	5 of 5
E1 − E2	Composite F1	+0.0089	[+0.0028, +0.0151]	0.0625	0.016	1.81	5 of 5
E2 − E3	Macro-F1	-0.0049	[-0.0582, +0.0483]	0.8125	0.811	-0.11	2 of 5
E2 − E3	Composite F1	-0.0009	[-0.0253, +0.0235]	1.0000	0.921	-0.05	2 of 5
B3 − E2	Macro-F1	-0.0022	[-0.0691, +0.0647]	0.8125	0.932	-0.04	3 of 5
B3 − E2	Composite F1	+0.0250	[+0.0047, +0.0454]	0.1250	0.027	1.53	4 of 5

Uplink and Downlink:

Table 8 splits communication into its two directions. Downlink is a dense FP32 broadcast in every configuration, so it is identical (4.534 MB cumulative) and only the uplink

varies, from 4.534 MB for E1 to 0.136 MB for B3. Downlink is 50.0% of E1's total traffic and 84.7% (E3) to 97.1% (B3) of every compressed configuration's (Figure 8). The uplink of E2 is 0.573 MB, which is 11.2% of E2's total of 5.107 MB. No uplink-only scheme could therefore reduce E2's total by more than about 11%, however well it adapts, because the downlink term does not change. This ceiling holds for four eligible clients, a 9,445-parameter model, a dense FP32 downlink sent to every eligible client each round, and 30 rounds. With more clients, a larger model or a compressed broadcast the balance would shift, and we do not claim the 11% figure beyond this setting.

Table 8. Communication decomposition, mean over five seeds (MB = 10^6 bytes). Cumulative values sum 30 rounds; per-round values divide by 30. Downlink share is downlink divided by total.

Config	Uplink (MB)	Downlink (MB)	Total (MB)	Total per round (MB)	Downlink share
E1	4.534	4.534	9.067	0.3022	50.0%
E2	0.573	4.534	5.107	0.1702	88.8%
E3	0.821	4.534	5.354	0.1785	84.7%
E4	0.639	4.534	5.173	0.1724	87.6%
E5	0.226	4.534	4.759	0.1586	95.3%
E5s	0.234	4.534	4.767	0.1589	95.1%
B3	0.136	4.534	4.670	0.1557	97.1%

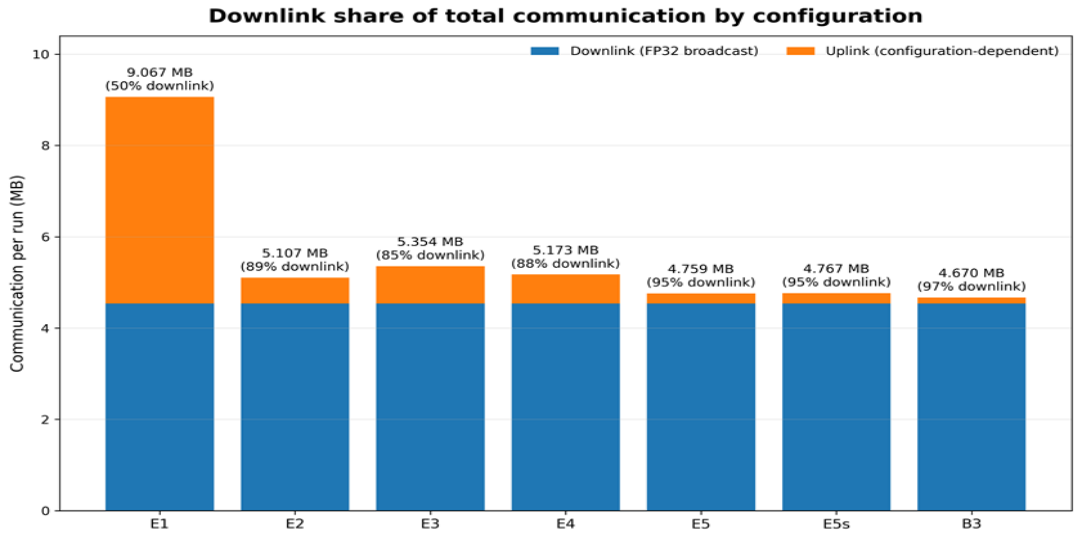

Figure 8. Total communication per run split into the fixed dense FP32 downlink and the configuration-dependent uplink, mean over five seeds. The downlink share ranges from 50% (E1) to 97% (B3).

Per-class Results:

Table 9 reports the final-round F1 of each class. Across all 35 seed-configuration runs, F1 for the S (supraventricular) class ranges from 0.000 to 0.0125 (mean 0.0021) and F1 for the F (fusion) class ranges from 0.000 to 0.0097 (mean 0.0015). Every run stays under 0.013 on both classes. We describe this as near zero, not as exactly zero and not as below chance, since we did not compute a chance baseline. F1 for V (ventricular) is far higher and varies more across configurations, from a mean of 0.719 (E4) to 0.836 (B3), with individual runs between about 0.47 and 0.93. Because the composite minority F1 is the mean of the S, V and F scores, it is approximately the V score divided by three: for E1, (0.0020 + 0.785 + 0.0023) / 3 = 0.263.

This is not a compression effect. It is present in E1, the uncompressed baseline that no adaptive or rule-based policy touches. Class-weighted cross-entropy with weights capped

at 3 is not enough to learn S and F under this imbalance (S is 2.54% of beats and F is 0.73%). Every statement in this paper about minority-class performance therefore concerns ventricular beats in practice, and the minority-class conclusions should be read with that in mind.

Table 9. Final-round per-class F1, mean over five seeds. The S and F columns are given to four decimals; the range over all 35 runs is in the text.

Config	N	S	V	F	Q
E1	0.896	0.0020	0.785	0.0023	0.958
E2	0.889	0.0021	0.757	0.0027	0.828
E3	0.857	0.0035	0.758	0.0030	0.882
E4	0.796	0.0045	0.719	0.0003	0.647
E5	0.893	0.0002	0.744	0.0017	0.816
E5s	0.831	0.0017	0.831	0.0003	0.662
B3	0.901	0.0008	0.836	0.0000	0.730

Discussion:

Why the Rule-Based Policy is Competitive:

The independent tables used by E3 to E5s are visited only 30 times each: one decision per client per round over 30 rounds. Table 10 and Figure 9 show the consequence. A single E5 or E5s agent visits a mean of 19.3 of its 900 state-action cells (2.1%, range 15 to 24), and E3 and E4, with 300 and 225 cells, visit 5.6% and 7.7%. Most visited cells are visited once. Pooling the four clients of seed 42 into one tally gives 68 of 900 cells (7.6%) for E5, but that figure overstates coverage because the four tables are independent: each agent learns from its own 30 decisions only. With at most 29 temporal-difference updates per agent, most state-action pairs an agent will consider have never been tried, so its behavior stays close to its initial table plus exploration. A rule that maps state to action needs no experience at all, which is a plausible reason why B3 competes with, and on some metrics matches, the learned policies. We did not test this explanation directly. A shared table across clients would multiply the observations per cell by up to four, and we did not run that variant.

Table 10. Q-table coverage per agent. Each of the 20 agents per configuration (4 eligible clients × 5 seeds) makes 30 decisions, one per round. Cells are the 75 states times the number of actions.

Config	Actions	Cells per table	Agents	Mean cells visited	Range	Share visited
E3	4	300	20	16.7	12 to 23	5.6%
E4	3	225	20	17.3	13 to 24	7.7%
E5	12	900	20	19.3	15 to 24	2.1%
E5s	12	900	20	19.4	14 to 23	2.2%

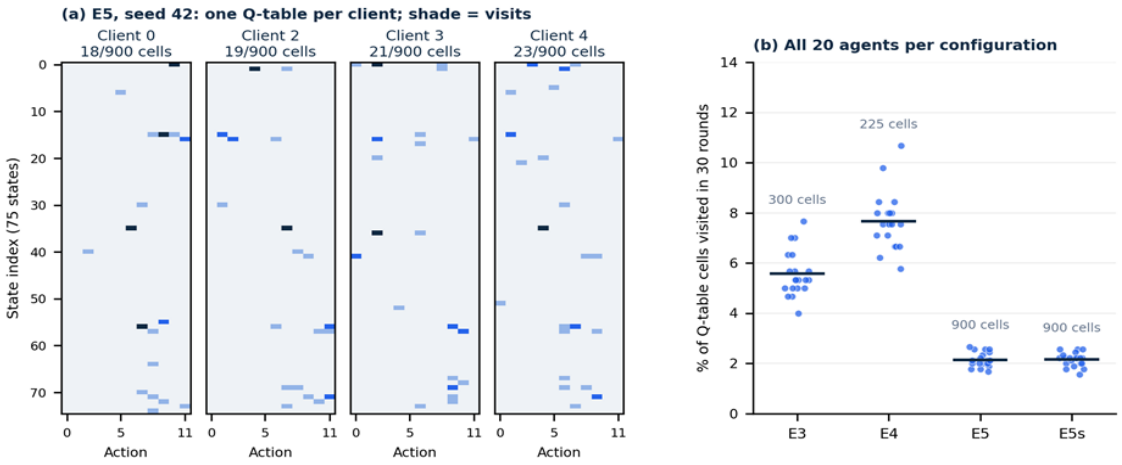

Figure 9. Q-table coverage. (a) Visited (state, action) cells of the four independent E5 Q-tables in seed 42, one panel per client; darker shade means more visits. (b) Share of cells visited by each of the 20 agents in E3, E4, E5 and E5s (points) with the mean (bar). Every agent makes 30 decisions, so coverage is capped at 30 cells and is far lower for the 900-cell tables.

Why Sparsification is Harder to Stabilize than Quantization:

E4 (sparsification only, 32-bit values) is much less stable than E3 (quantization only, dense). The two do not share an action set, since E3 chooses among four bit-widths and E4 among three keep-ratios, so the difference is not a matter of action-space size. Magnitude-based top-k favors coordinates with large gradients, which under heavy class imbalance tend to belong to the majority class, and an error-feedback residual lets dropped coordinates accumulate until they are sent together. A policy that changes the keep-ratio between rounds may make those releases irregular. We offer this as a hypothesis, not a finding.

The existing comparisons narrow the candidates without isolating them. E3 uses the same reward and the same kind of learner as E4 and is the stable adaptive policy (macro-F1 SD 0.036). B3 sparsifies at a mean keep-ratio of 4.5% without any learning and has an SD of 0.067. E4 and E5 combine learning with sparsification and have SDs of 0.152 and 0.099. E5s also combines them and has an SD of 0.049, although its lower mean macro-F1 (0.465) means it is not a clean counterexample either. The pattern is compatible with an interaction between learned control and sparsification, but it confounds the sparsification level, the action-space size and the learning signal. Separating them would need controlled runs (E4 with sparsification disabled, a shared Q-table, and a constant reward), which we did not perform.

Downlink Dominance:

The downlink result follows from the accounting and needs no statistical test: with four clients and a 9,445-parameter model, the dense broadcast is large compared with any plausible compressed upload. It is a property of this setup, not a general law. A federation with many more clients per round, a larger model, or a multicast broadcast would shift the balance toward the uplink. Compressing the broadcast global model has been studied [36]. Our results suggest that in a small wearable federation it is where most of the remaining budget lies, and that reports of compression gains should state which direction they cover.

Calibration Data and the Reward Signal:

The reward is computed from a pooled calibration set of 391 beats that comes from the training records. It holds only 36 F beats and 72 S beats, so per-class estimates are noisy, and because the beats share patients with the training data it can keep improving while cross-patient test performance does not. In the logs, the calibration quality of the global model tracks test macro-F1 only moderately: the median Spearman correlation over the 20 RL runs is 0.57 (range 0.12 to 0.84). In the E4 collapse of seed 42 the two moved in opposite directions. Calibration quality was 0.394 at round 20, when test macro-F1 stood at 0.560, and rose to 0.646 at round 30, when test macro-F1 had fallen to 0.284. In seed 2024, calibration quality was highest at round 20 (0.627), after test macro-F1 had already dropped from its peak of 0.564 at round 17 to 0.427. We cannot say whether this mismatch causes the collapses, only that the signal the agents were rewarded on was a weak proxy for test performance at those moments.

Practical Implications:

For a system that must be built now, fixed 4-bit update quantization is the most defensible choice in this study. It saved 43.7% of total communication, its seed-to-seed variability was low, and its macro-F1 was lower than FP32 FedAvg in every seed by a small margin that we could not confirm statistically. A hand-written rule such as B3 is a reasonable second option where some awareness of battery and network is wanted without the sample cost of a learned table. Its composite minority advantage is descriptive and driven by

ventricular beats, and its macro-F1 variability is higher, so it should be checked on the target data before use.

We would not deploy learned sparsification control in its present form. E4 collapsed in two of five seeds and E5 split into high and low seeds, and we could not isolate the cause. We also would not describe any of the seven configurations as protecting minority classes, because supraventricular and fusion beats are not detected by any of them. The engineering priorities that follow from the results are, in order: compressing the downlink, since it carries most of the traffic; fixing supraventricular and fusion learning before any claim of minority protection, for example with focal loss [24]or resampling [25], which we did not test; and measuring the on-device cost of the compression and control logic. None of this amounts to clinical readiness.

Limitations and Future Work:

Limitations:

Scope. The study uses one dataset, one 9,445-parameter model, one non-IID level (alpha = 0.5), one client configuration (five nominal clients, four eligible), one draw of the partition, and 30 rounds. The five seeds vary initialization, exploration and simulated device noise but not the partition, so seed-to-seed variability understates the variability that a different assignment of records to clients would add. We did not test other heterogeneity levels, other client counts, or longer horizons, and the findings may change under them. Excluding client 1 also reduces the federation.

Statistics. Five seeds are too few to separate differences of a few hundredths of macro-F1. All tests are exploratory, the exact Wilcoxon test cannot fall below 0.0625 with five pairs, and the normality assumption behind the t-intervals cannot be checked. Non-significance is not equivalence, so a comparison that is not significant here does not show that two policies perform alike. and are not calibrated against real wearable traces, and the energy term is a data-size proxy that does not depend on the action. We measured no energy, memory or latency on real hardware, including the cost of a Q-table lookup or a rule evaluation, so any statement about deployment cost is inferred from bit counts only.

Learning. The independent Q-tables see about 30 decisions each. The learning rate, discount, exploration schedule, bin edges and reward weights were fixed in advance and not tuned or varied, and E5s is a heuristic proxy for class-aware sparsification. No ablation was run to separate sparsification, limited experience and the reward function as causes of the instability in E4 and E5. Observed differences between two executions of the same code and seeds (Appendix Table A3) show implementation-level variability that we did not isolate. Minority classes and data. No configuration detects supraventricular or fusion beats, and we tested no alternative training method for them. The calibration beats share patients with the training data. The split is record-level, and we verified subject separation only for the known pair of records 201 and 202. We report no external validation, no clinical evaluation, and no privacy or security evaluation: secure aggregation [26]and differential privacy were not implemented. The downlink is an uncompressed FP32 broadcast in every configuration.

Future Work:

We see six experiments, listed in order of how much they would strengthen the present conclusions. First, a controlled ablation of E4 with sparsification disabled, with a shared Q-table, and with a constant reward would separate the candidate causes of instability. Second, ten or more seeds and several partition draws would narrow the intervals in Table 7 (sensitivity thresholds). Third, alpha values of 0.1 and 1.0, client counts of 3 and 10, and inclusion of client 1 would test whether the findings hold beyond the present configuration. Fourth, training methods aimed at supraventricular and fusion beats should be evaluated before compression is layered on top. Fifth, downlink compression such as quantized or delta broadcast should be

added to the policy space. Sixth, profiling on an embedded or wearable platform would replace the simulated battery, network and energy terms with measurements.

Conclusion:

Some findings need no statistical inference because they are measurements or arithmetic on this protocol. Fixed 4-bit update quantization reduced total communication by 43.7% relative to FP32 FedAvg. The uncompressed downlink carried 50% of E1's traffic and 85% to 97% of every compressed configuration's, so uplink-only savings are capped at about 11% relative to E2 for four clients and a 9,445-parameter model. F1 for supraventricular and fusion beats stayed below 0.013 in all 35 runs, so the composite minority F1 mainly measures ventricular beats.

Other findings are descriptive and not statistically established. FP32 FedAvg had a higher macro-F1 than fixed 4-bit quantization in all five seeds (mean gap 0.033), but the interval includes zero and the exact Wilcoxon p-value is 0.0625. The rule-based policy had the highest mean composite minority F1 and the lowest communication, and its macro-F1 was close to that of fixed 4-bit quantization; its composite advantage did not survive correction for multiple tests. No comparison against the joint RL policy E5 was significant. The collapse of two E4 seeds appeared in two executions of the same code, and we have not isolated its cause. We make no claim about other heterogeneity levels, other client counts, real hardware costs, or clinical use.

In practice, fixed 4-bit quantization is the safer default in this setting, a simple state-aware rule is a reasonable alternative, and learned sparsification control needs the ablations listed in Future Work before it is used. Compressing the downlink and learning supraventricular and fusion beats come before any claim that a compression policy protects minority classes.

References:

[1] B. A. y A. H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv:1602.05629*, vol. 54, 2017, doi: https://doi.org/10.48550/arXiv.1602.05629.

[2] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, "The future of digital health with federated learning," *npj Digit. Med.*, vol. 3, 2020, [Online]. Available: https://www.nature.com/articles/s41746-020-00323-1

[3] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021, doi: 10.1561/2200000083.

[4] D. M. Jimenez Gutierrez, H. M. Hassan, L. Landi, A. Vitaletti, and I. Chatzigiannakis, "Application of federated learning techniques for arrhythmia classification using 12-lead ECG signals," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 14053 LNCS, pp. 38–65, Aug. 2022, doi: 10.1007/978-3-031-49361-4_3.

[5] A. Vikhyat *et al.*, "Federated Learning and Differential Privacy Techniques on Multi-hospital Population-scale Electrocardiogram Data," *ACM Int. Conf. Proceeding Ser.*, pp. 143–152, Apr. 2024, doi: 10.1145/3673971.3673990.

[6] S. Sakib, M. M. Fouda, Z. Md Fadlullah, K. Abualsaud, E. Yaacoub, and M. Guizani, "Asynchronous Federated Learning-based ECG Analysis for Arrhythmia Detection," *2021 IEEE Int. Mediterr. Conf. Commun. Networking, MeditCom 2021*, pp. 277–282, 2021, doi: 10.1109/MEDITCOM49071.2021.9647636.

[7] Q. Z. S. Sricharan Donkada, Seyedamin Pouriyeh, Reza M. Parizi, Meng Han, Nasrin Dehbozorgi, Nazmus Sakib, "Uncovering Promises and Challenges of Federated Learning to Detect Cardiovascular Diseases: A Scoping Literature Review," *Comput. Sci. > Mach. Learn.*, doi: https://doi.org/10.48550/arXiv.2308.13714.

[8] J. Konečn, H. Brendan McMahan, F. X. Yu, A. Theertha Suresh, D. Bacon Google,

and P. Richtárik, "Federated Learning: Strategies for Improving Communication Efficiency," Oct. 2016, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1610.05492

[9] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 1710–1721, Oct. 2016, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1610.02132

[10] A. Reisizadeh, A. Jadbabaie, A. Mokhtari, H. Hassani, and R. Pedarsani, "FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization," *Proc. Mach. Learn. Res.*, vol. 108, pp. 2021–2031, Sep. 2019, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1909.13014

[11] S. U. Stich, J. B. Cordonnier, and M. Jaggi, "Sparsified SGD with Memory," *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, pp. 4447–4458, Sep. 2018, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1809.07599

[12] R. Hönig, Y. Zhao, and R. Mullins, "DAdaQuant: Doubly-adaptive quantization for communication-efficient Federated Learning," *Proc. Mach. Learn. Res.*, vol. 162, pp. 8852–8866, Oct. 2021, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2111.00465

[13] Y. Mao *et al.*, "Communication Efficient Federated Learning with Adaptive Quantization," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, Apr. 2021, doi: 10.1145/3510587.

[14] B. Isik, F. Pase, D. Gunduz, S. Koyejo, T. Weissman, and M. Zorzi, "Adaptive Compression in Federated Learning via Side Information," *Proc. Mach. Learn. Res.*, vol. 238, pp. 487–495, Jun. 2023, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2306.12625

[15] V. Tsouvalas, A. Saeed, T. Ozcelebi, and N. Meratnia, "Communication-Efficient Federated Learning through Adaptive Weight Clustering and Server-Side Distillation," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 5805–5809, Jan. 2024, doi: 10.1109/ICASSP48485.2024.10447174.

[16] S. Q. Zhang, J. Lin, and Q. Zhang, "A Multi-agent Reinforcement Learning Approach for Efficient Client Selection in Federated Learning," *Proc. 36th AAAI Conf. Artif. Intell. AAAI 2022*, vol. 36, pp. 9091–9099, Jan. 2022, doi: 10.1609/aaai.v36i8.20894.

[17] S. Cao *et al.*, "FedQMIX: Communication-efficient federated learning via multi-agent reinforcement learning," *High-Confidence Comput.*, vol. 4, no. 2, p. 100179, Jun. 2024, doi: 10.1016/J.HCC.2023.100179.

[18] Z. Shu, T. Cao, and H.-W. Huang, "BiFedKD: Bidirectional Federated Knowledge Distillation Framework for Non-IID and Long-Tailed ECG Monitoring," May 2026, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2605.14886

[19] H. Wu, "Towards Family-Grouped Hierarchical Federated Learning on Sub-5KB Models: A Feasibility Study of Privacy-Preserving ECG Monitoring for Ultra-Resource-Constrained Wearables," May 2026, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2605.18862

[20] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001, doi: 10.1109/51.932724.

[21] "ANSI/AAMI EC57:2012/(R)2020; Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," *ANSI/AAMI EC572012/(R)2020; Test. Report. Perform. results Card. Rhythm ST Segm. Meas. algorithms*, Jan. 2013, doi: 10.2345/9781570204784.

[22] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nat. Med. 2019 251*, vol.

25, no. 1, pp. 65–69, Jan. 2019, doi: 10.1038/s41591-018-0268-3.

[23] A. H. Ribeiro *et al.*, "Automatic diagnosis of the 12-lead ECG using a deep neural network," *Nat. Commun. 2020 111*, vol. 11, no. 1, pp. 1760-, Apr. 2020, doi: 10.1038/s41467-020-15432-4.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár., "Focal Loss for Dense Object Detection," *arXiv*, vol. 1, 2017, doi: https://doi.org/10.48550/arXiv.1708.02002 Focus to learn more.

[25] Justin M. Johnson & Taghi M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, 2019, [Online]. Available: https://link.springer.com/article/10.1186/s40537-019-0192-5

[26] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1175–1191, Oct. 2017, doi: 10.1145/3133956.3133982

[27] J. Zhang, C. Luo, M. Carpenter, and G. Min, "Federated Learning for Distributed IIoT Intrusion Detection Using Transfer Approaches," *IEEE Trans. Ind. Informatics*, vol. 19, no. 7, pp. 8159–8169, Jul. 2023, doi: 10.1109/TII.2022.3216575.

[28] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification," Sep. 2019, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1909.06335

[29] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," Dec. 2018, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1812.06127

[30] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF168147-7, pp. 5088–5099, Oct. 2019, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/1910.06378

[31] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated Learning Based on Dynamic Regularization," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, Nov. 2021, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2111.04263

[32] S. J. Reddi *et al.*, "Adaptive Federated Optimization," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, Feb. 2020, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2003.00295

[33] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated Learning on Non-IID Features via Local Batch Normalization," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, Feb. 2021, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2102.07623

[34] Y. Wu and K. He, "Group normalization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11217 LNCS, pp. 3–19, 2018, doi: 10.1007/978-3-030-01261-8_1/TABLES/8.

[35] F. Sattler, S. Wiedemann, K. R. Muller, and W. Samek, "Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020, doi: 10.1109/TNNLS.2019.2944481.

[36] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated Learning With Quantized Global Model Updates," Jun. 2020, Accessed: Oct. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2006.10672

[37] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," *IEEE Int. Conf. Commun.*, vol. 2019-May, May 2019, doi: 10.1109/ICC.2019.8761315.

[38] Sutton, R. S., & Barto, A. G, "Reinforcement learning: An introduction, 2nd ed.," *MIT Press*, 2018, [Online]. Available: https://psycnet.apa.org/record/2019-19679-000

[39] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Commun. ACM*, vol. 63, no. 12, pp. 54–63, Nov. 2020, doi: 10.1145/3381831.

Appendix A: Verification Checklist

Each item was checked against the saved result files of the canonical run (data provenance, configuration, history and action logs, validation report).

Table A1. Verification checklist for the canonical run.

Check	Status	Evidence
Dataset size	Verified	48 records, 109,464 beats, class counts as
Record-level split	Verified	14 test and 34 training records with no overlap; records 201 and 202 both in the training pool
Test minority coverage	Verified	S 1,457, V 2,370, F 391 test beats
Label mapping	Verified	15 beat symbols mapped to 5 AAMI classes; non-beat symbols excluded
Normalization	Verified	Per-beat z-score; Group Normalization in ECGNet
Eligibility rule	Verified	Fewer than 5 beats of each of S, V, F excludes a client; client 1 (S 2, V 3, F 0) excluded
E1 and E2	Verified	E1 dense 32-bit without error feedback; E2 dense 4-bit with error feedback
E3	Verified	Bit-width adapts; keep-ratio is 1.0 in all 600 logged decisions
E4	Verified	Bit-width is 32 in all 600 logged decisions; keep-ratio adapts
Warm-up	Verified	E3, E5, E5s: bits 6 or 8 in rounds 1 to 10; E4: keep-ratio 5% or 20%; E5, E5s select 1% during warm-up; B3 unrestricted
Calibration split	Verified	Beat-level disjoint from local training; same records; fixed seed offset per client; 391 pooled beats
Aggregation	Verified	Sample-weighted FedAvg without precision weighting
Q-learning	Verified	Independent per-client tables, 75 states, not shared across clients or seeds
Communication accounting	Verified	k from the sparsifier; downlink FP32 to 4 eligible clients each round
Seeds	Verified	42, 123, 2024, 7, 99 for every configuration; split and partition fixed across seeds
Run-to-run variability	Disclosed	A retained first execution of E4 differs from the canonical run by up to 0.078 (Table A3)

Appendix B: Per-Seed Results

Table A2. Final-round macro-F1 and composite minority F1 for every seed and configuration (canonical run).

Config	Metric	Seed 42	Seed 123	Seed 2024	Seed 7	Seed 99	Median
E1	Macro-F1	0.5435	0.5311	0.4937	0.5335	0.5408	0.5335
	Composite F1	0.2744	0.2662	0.2098	0.2855	0.2789	0.2744
E2	Macro-F1	0.5216	0.4392	0.4934	0.4937	0.5310	0.4937
	Composite F1	0.2649	0.2583	0.2087	0.2717	0.2665	0.2649
E3	Macro-F1	0.4603	0.4873	0.4820	0.5240	0.5498	0.4873
	Composite F1	0.2790	0.2294	0.2106	0.2673	0.2885	0.2673
E4	Macro-F1	0.2843	0.5497	0.2514	0.5451	0.5359	0.5359
	Composite F1	0.2048	0.2940	0.1576	0.2830	0.2664	0.2664
E5	Macro-F1	0.5535	0.3741	0.5677	0.3919	0.5676	0.5535
	Composite F1	0.2831	0.1722	0.3079	0.1772	0.3036	0.2831
E5s	Macro-F1	0.5525	0.4429	0.4563	0.4390	0.4357	0.4429
	Composite F1	0.2937	0.3070	0.2771	0.2903	0.2197	0.2903
B3	Macro-F1	0.5606	0.4648	0.4842	0.4006	0.5577	0.4842
	Composite F1	0.2954	0.2822	0.2379	0.2703	0.3095	0.2822

Appendix C: E4 First Execution Against the Canonical Run

Table A3. Final-round macro-F1 of E4 in a retained first execution of the same code and seeds and in the canonical run. First-execution values are transcribed from its console log.

Seed	First execution	Canonical run	Difference (canonical − first)
42	0.2779	0.2843	+0.0064
123	0.5435	0.5497	+0.0062
2024	0.3061	0.2514	-0.0547
7	0.5535	0.5451	-0.0084
99	0.4577	0.5359	+0.0782



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.