FedE2A: Large Language Model Agents for Energy-Aware Robust Federated Learning

Muhammad Usman
Dept. of Engineering
University of Sannio
Benevento, Italy
m.usman1@studenti.unisannio.it

Mario Luca Bernardi
Dept. of Engineering
University of Sannio
Benevento, Italy
bernardi@unisannio.it

Marta Cimitile
Dept. of Law and Digital Society
UnitelmaSapienza University
Rome, Italy
marta.cimitile@unitelmasapienza.it

Samiksha Bisht
Dept. of Engineering
University of Sannio
Benevento, Italy
s.bisht@studenti.unisannio.it

***Abstract*—Federated learning in resource-constrained environments poses a complex multi-objective optimization problem characterized by extreme data heterogeneity, stochastic client availability, and fluctuating energy constraints. Traditional client selection mechanisms, relying predominantly on static heuristics or single-objective optimization, lack the flexibility to navigate the dynamic Pareto frontier between data utility, system robustness, and energy efficiency. We propose FedE2A, a framework that reframes client selection as a context-aware decision process governed by a Large Language Model (LLM). In this architecture, clients are abstracted into structured Quality Profile Vectors (QPV) that encapsulate strictly local states—including epistemic uncertainty via Monte Carlo Dropout, system health, and historical reliability. The server-side LLM acts as a reasoning engine, interpreting these high-dimensional profiles to dynamically parameterize the selection policy based on the current training phase and convergence trajectory. Unlike rigid rule-based systems, this neuro-symbolic approach allows for semantic interpretation of system anomalies and adaptive balancing of competing objectives. Results achieved on CIFAR-10 under non-IID settings and label noise demonstrate that FedE2A yields improvements in convergence stability and energy efficiency, starting from a certain number of clients onwards, compared to state-of-the-art baselines.**

***Index Terms*—Federated Learning, Large Language Models, Energy Efficiency, Client Selection, Non-IID Data**

I. Introduction

Federated learning (FL) enables distributed model training across heterogeneous clients without centralizing raw data [2]. A central server orchestrates iterative rounds of local training and global aggregation, preserving data privacy while leveraging collective computational resources. This paradigm is particularly compelling for mobile and IoT deployments where data are naturally distributed and privacy regulations are stringent. However, the transition from controlled experiments to realistic deployments exposes fundamental limitations in how existing systems select participating clients.

The orchestration of these competing objectives presents a fundamental control problem. A client offering high information gain (e.g., data from the tail of a distribution) might simultaneously exhibit high latency or energy costs. In non-stationary environments, the optimal trade-off among these factors shifts as training progresses: high-variance, energy-intensive gradients may be desirable for exploration in early phases, whereas stability and efficiency become paramount near convergence.

Existing selection protocols typically reduce this dimensionality through simplification. Standard approaches such as FedAvg [14] employ uniform sampling, effectively treating all contributions as equiprobable in value. Variance-reduction methods like SCAFFOLD [16] or robustness-oriented schemes like FedProx [15] refine the aggregation step but generally inherit agnosticism toward client selection. Conversely, resource-aware strategies [9], [13] often optimize for system efficiency at the expense of data diversity. The core limitation of these methods lies in their reliance on fixed scoring functions or static heuristics, which lack the semantic capacity to reason about the evolving context of the federated system. They cannot distinguish, for instance, whether a drop in client stability represents a transient network anomaly or an emerging Byzantine behavior, nor can they autonomously re-weight objectives when the global model enters a new regime of the loss landscape.

To bridge this gap, we introduce FedE2A, a framework that integrates Large Language Models (LLM) into the control loop of federated systems. We postulate that the complexity of client selection is best managed not by rigid algebraic rules, but by a semantic reasoning engine capable of interpreting structured state representations. In our proposed architecture, each client computes a Quality Profile Vector (QPV) aggregating estimates of data uncertainty (via Monte Carlo Dropout), computational health, and historical alignment. The server-side LLM does not replace the deterministic mechanics of aggregation; rather, it parameterizes the selection policy. By ingesting the current training context—including round progress, accuracy trajectory, and resource budgets—

the LLM dynamically adjusts the constraints and weights of the selection function. This design preserves the deterministic nature of the underlying optimization while enabling a high-level policy that adapts to unseen conditions and generates interpretable rationales for its decisions.

Our contribution is threefold. First, we formalize the Quality Profile Vector as a unified state representation that makes client heterogeneity interpretable for automated reasoning. Second, we propose a neuro-symbolic governance protocol where an LLM modulates the hyperparameters of a robust selection algorithm, effectively mapping semantic system states to optimal control parameters. Third, we provide empirical evidence on the CIFAR-10 dataset demonstrating that this semantic orchestration achieves a superior Pareto trade-off between accuracy and energy consumption compared to static baselines, particularly under adversarial conditions characterized by non-IID data and label noise.

The remainder of this paper proceeds as follows. Section II reviews related work while Section III and Section IV present the FedE2A framework and the details of the LLM-governed selection. Finally Section V reports experiments to validate the approach. Conclusions and final remarks are reported in Section VI.

II. Related Work

The deployment of Federated Learning in realistic environments necessitates balancing the competing objectives of model convergence, system robustness, and energy efficiency. While recent literature has addressed these dimensions individually, the integration of multi-objective optimization with semantic reasoning has received limited attention, particularly when combined with LLM-based controllers.

Energy efficiency is paramount for edge devices constrained by battery life. Approaches such as dynamic resource-aware scheduling [9] typically employ reinforcement learning or convex optimization to manage the trade-off between energy consumption and training latency. Techniques including wireless power transfer optimization [8] and quantization-aware training [7] further reduce the per-round energy footprint. However, these methods largely treat energy minimization as an orthogonal problem to data quality, potentially prioritizing efficient but uninformative clients, thereby degrading the global model's convergence rate.

Conversely, heterogeneity-aware client selection focuses on mitigating the impact of non-IID data and system variance. Methods like TiFL [5] and Oort [4] stratify clients based on response latency or data utility, improving time-to-accuracy. Quality-driven frameworks [1] refine this by weighting contributions according to statistical properties of local datasets. While these strategies advance beyond uniform sampling, they rely on fixed heuristics or static scoring functions. Such rigid policies lack the adaptability to handle non-stationary environments where the relative importance of data diversity versus system stability shifts across training phases. Furthermore, robustness-oriented aggregation schemes [6], [10] generally operate reactively, filtering outliers after updates are received, rather than proactively managing client participation to prevent resource wastage.

Recent work has begun exploring LLMs for resource allocation and scheduling in distributed systems. LLMs have been applied to multi-agent coordination [18], efficient scheduling in manufacturing [19], and device selection in federated contexts [17]. Research on LLM interaction with structured tabular data [20], [21] demonstrates that text-based encodings preserving row-column structure enable effective reasoning over multi-dimensional records. Hybrid frameworks combining rule-based systems with LLM reasoning [23] show that language models can parameterize control policies while maintaining determinism.

FedE2A is inspired from these research threads to build an unified framework. Unlike prior work that addresses energy, quality, or robustness in isolation, FedE2A integrates all dimensions into a structured QPV representation and employs LLM-based reasoning to navigate their trade-offs. The LLM does not replace deterministic selection logic but governs it, adjusting weights and constraints based on contextual understanding while the server retains final control over client sets.

III. FedE2A Framework

This section presents the FedE2A architecture: problem formulation, Quality Profile Vector construction, energy modeling, and governance workflow.

We consider an FL system with N clients coordinated by a central server. Each client i owns a local dataset $\mathscr{D}_i$ drawn from distribution P_i. In non-IID settings, distributions exhibit severe heterogeneity through label skew (Dirichlet parameter $\alpha \ll 1$), quantity imbalance, and label noise with corruption rates up to 40%. Each client maintains time-varying state $s_i(t)$ capturing battery level, CPU load, memory pressure, and network quality. The server aims to learn a global model w minimizing the population loss

$$L(w) = \mathbb{E}_{(x,y)\sim P}[\ell(w;x,y)] \tag{1}$$

while managing cumulative energy expenditure across training rounds.

The foundational abstraction in FedE2A promotes FL clients from passive workers to autonomous agents. Each agent performs self-assessment and reports a Quality Profile Vector to the governance engine:

$$\text{QPV}_i = (QD_i, QS_i, QH_i, E_i), \tag{2}$$

where $QD_i \in [0,1]$ quantifies data quality, QS_i is a vector of system health metrics, QH_i encodes historical reliability, and $E_i \in [0,1]$ represents normalized energy cost. This multi-dimensional representation provides the LLM with structured input for contextual reasoning about client suitability.

The data quality component QD_i captures informational value through uncertainty estimation. Client i maintains model f_{θ_i} with dropout layers and performs $K = 10$ stochastic forward passes on local validation data. Let $\bar{p}(y) = K^{-1}\sum_k p_k(y)$

denote the mean predictive distribution. The data quality score is the normalized predictive entropy:

$$QD_i = \min\left(\frac{-\sum_y \bar{p}(y)\log\bar{p}(y)}{\log C}, 1\right), \tag{3}$$

where C is the number of classes. Higher values indicate data from regions where the model lacks confidence, suggesting informative contributions. The system health component:

$$QS_i = (\text{cpu}_i, \text{mem}_i, \text{bat}_i, \text{rtt}_i, \text{jitter}_i)$$

provides real-time operational assessment through lightweight probes measuring CPU utilization, memory pressure, battery level, network latency, and jitter. Each metric is normalized via min-max scaling and aggregated into a scalar:

$$\begin{aligned} g(QS_i) = w_{\text{cpu}}\text{cpu}_i + w_{\text{mem}}\text{mem}_i + w_{\text{bat}}\text{bat}_i \\ - w_{\text{rtt}}\text{rtt}_i - w_{\text{jit}}\text{jitter}_i, \end{aligned} \tag{4}$$

where weights encode the intuition that higher capacity and lower latency improve reliability. The historical reliability component $QH_i = (\text{stab}_i, \text{punct}_i, \text{count}_i)$ encodes long-term behavioral reputation through stability, punctuality, and participation count. Stability tracks alignment with a golden update Δ_{gold} computed from a small trusted holdout dataset. For each round, the server updates an exponential moving average of cosine similarity:

$$\text{stab}_i^{(t+1)} = \alpha\frac{\langle\Delta_i, \Delta_{\text{gold}}\rangle}{\|\Delta_i\|\|\Delta_{\text{gold}}\|} + (1-\alpha)\,\text{stab}_i^{(t)}, \tag{5}$$

with $\alpha = 0.3$. Punctuality adjusts based on response latency; the counter tracks completed rounds. These aggregate into:

$$\begin{aligned} h(QH_i) = u_{\text{stab}}\,\text{stab}_i + u_{\text{punct}}\,\text{punct}_i \\ + u_{\text{count}}\log(1+\text{count}_i). \end{aligned} \tag{6}$$

A precise estimation of energy consumption is paramount. Theoretically, the energy cost E_i for a training round is the sum of computation and communication components [24]:

$$E_i^{\text{theoretical}} = \underbrace{\kappa N_i C_{pp} f_i^2}_{\text{Computation}} + \underbrace{P_i^{tx}\frac{|\theta|}{R_i}}_{\text{Communication}} \tag{7}$$

where κ is the effective switched capacitance, N_i samples, C_{pp} cycles per sample, f_i frequency, P_i^{tx} transmission power, and R_i uplink rate. Here we adopt the energy cost component E_i providing a normalized proxy combining computation and communication. For each round, client i records training time T_i, average CPU utilization u_i, and bytes transmitted/received $(B_{\text{tx}}, B_{\text{rx}})$. A computation cost $\hat{E}_{\text{comp}} \propto T_i \cdot u_i$ and communication cost $\hat{E}_{\text{comm}} \propto (B_{\text{tx}} + B_{\text{rx}})$ are computed using nominal scaling constants, then min-max normalized and combined:

$$E_i = \alpha_E\, E_{\text{comp}} + (1-\alpha_E)\, E_{\text{comm}}. \tag{8}$$

interpreted as a dimensionless proxy.

Given QPVs, the LLM-governed engine computes client scores using weights $(\lambda_Q, \lambda_S, \lambda_H, \lambda_E)$ determined through contextual reasoning (detailed in Section IV):

$$\begin{aligned} \text{score}_i = \lambda_Q\, QD_i + \lambda_S\, g(QS_i) \\ + \lambda_H\, h(QH_i) - \lambda_E\, E_i. \end{aligned} \tag{9}$$

The governance workflow proceeds through five phases each round. First, available clients compute and transmit QPV components (QD_i, QS_i, E_i). Second, the engine reconstructs complete QPVs by retrieving stored QH_i vectors. Third, the LLM-governed module determines policy parameters and computes scores via Eq. (9). Fourth, selected clients execute local training in parallel, returning updates for weighted aggregation:

$$\theta^{(t+1)} = \theta^{(t)} + \sum_{i\in S_t}\frac{n_i}{\sum_{j\in S_t} n_j}\Delta_i^{(t)}, \tag{10}$$

where n_i is client i's sample count and S_t the selected set. Fifth, the engine updates QH_i based on contribution similarity to Δ_{gold} per Eq. (5). Figure 1 illustrates this architecture whereas Algorithm 1 provides pseudocode.

Algorithm 1: LLM-based clients selection

Input: Model $\theta^{(0)}$, clients $\{1,\ldots,N\}$, holdout $\mathscr{D}_{\text{hold}}$, rounds T
Output: Trained model $\theta^{(T)}$, profiles $\{QH_i\}$

Init: Train reference on $\mathscr{D}_{\text{hold}}$; compute Δ_{gold}; init QH_i;
for $t = 1,\ldots,T$ **do**
  Collect (QD_i, QS_i, E_i) from available clients;
  Reconstruct QPV_i for each client via Eq. (2);
  // LLM-Governed Selection
  Serialize QPVs and context into structured prompt;
  Query LLM for weights $(\lambda_Q, \lambda_S, \lambda_H, \lambda_E)$;
  if *response invalid* **then**
    Use default parameters;
  Compute score_i via Eq. (9); select top-k: S_t;
  // Training and Aggregation
  for $i \in S_t$ ***in parallel*** **do**
    $\Delta_i^{(t)} \leftarrow \text{LocalTrain}(\theta^{(t)}, \mathscr{D}_i)$;
  Aggregate via Eq. (10);
  // Profile Update
  for $i \in S_t$ **do**
    Update stab_i via Eq. (5); update punct_i;
return $\theta^{(T)}$, $\{QH_i\}$

IV. LLM-Governed Client Selection

The LLM governance module is the distinguishing component of FedE2A, transforming client selection from a fixed policy into an adaptive reasoning process. This section describes the module's design principles, interaction protocol, and integration with the deterministic governance workflow.

Traditional selection mechanisms operate through scoring functions with predetermined weights. While the formulation in Eq. (9) is mathematically tractable, using fixed weights $(\lambda_Q, \lambda_S, \lambda_H, \lambda_E)$ has fundamental limitations. The weights require manual tuning and remain static regardless of training dynamics. The relative importance of data quality versus energy efficiency shifts across training phases: early rounds benefit from diverse, informative clients even at higher energy cost, while later rounds should favor stable, efficient contributors. Fixed weights cannot capture these phase-dependent trade-offs. Similarly, appropriate responses to anomalous client behavior—declining stability scores, sudden energy spikes,

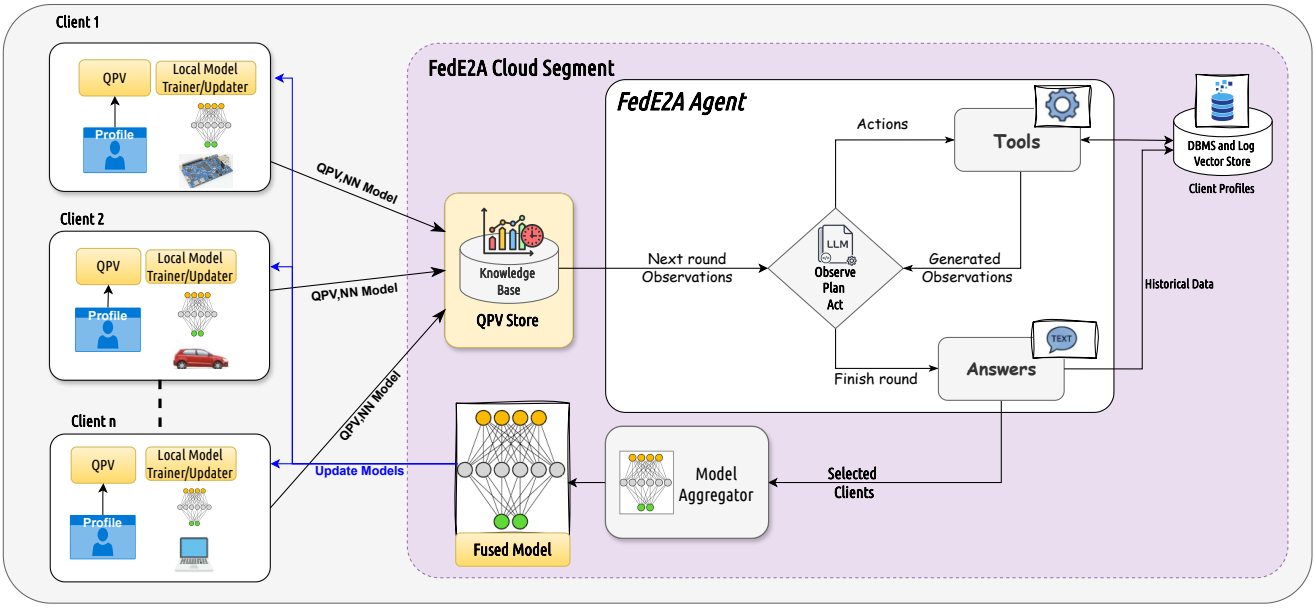


Fig. 1: An overview of the FedE2A architecture. Clients compute Quality Profile Vectors locally and report them to the server. The LLM-governed engine interprets QPVs in context, determines selection policy parameters, and produces client rankings. Selected clients perform local training and return updates for aggregation. Historical reliability profiles are updated based on contribution quality and punctuality.

network degradation—require contextual judgment that static rules cannot provide.

The LLM addresses these limitations by interpreting QPV data in light of training context and stated objectives. Rather than replacing the scoring function, the LLM governs its parameters: it suggests weights and constraints based on reasoning about the current system state. This design preserves the determinism and auditability of numeric selection while enabling adaptive policy generation.

The interaction protocol proceeds as follows. At each round, the governance engine constructs a structured representation of available clients by serializing their QPVs into a tabular format. Each client appears as a row with columns for QD_i, aggregated system health $g(QS_i)$ per Eq. (4), aggregated reliability $h(QH_i)$ per Eq. (6), energy cost E_i, and notable flags (low battery, declining stability, high latency). This tabular encoding follows best practices for LLM reasoning over structured data [20], [21], preserving row-column relationships that enable comparative analysis across clients.

A. LLM Implementation

We employ Gemma 3-4B [22], a compact open-source language decoder-only transformer model developed by Google DeepMind, as the governance engine for client selection. The selection of this particular variant stems from architectural considerations specific to federated learning server constraints. With 3.88 billion parameters (675M embedding, 3.21B non-embedding), this LLM provides a practical balance between contextual understanding and computational tractability for per-round inference tasks.

Several factors motivated this choice over alternatives. Large frontier LLMs such as GPT-4 or Claude, despite their sophisticated reasoning abilities, impose inference latencies exceeding several seconds per query—incompatible with the time budget of typical federated training rounds where aggregation itself completes within 25-40 seconds. Conversely, sub-2B parameter models exhibit unreliable structured output generation, frequently producing malformed structure or semantically inconsistent weight assignments during exploratory trials. Gemma 3-4B demonstrates 94% first-attempt valid JSON generation in controlled evaluations, substantially outperforming smaller alternatives while maintaining inference times under 500 milliseconds [22].

The LLM architectural characteristics prove well-suited to our governance task. Its 5:1 ratio of local-to-global attention layers reduces key-value cache memory consumption during inference—a design originally intended for long-context processing but beneficial here for minimizing server-side resource overhead. Supporting 128K token context windows ensures the model handles comprehensive QPV tables even in deployments with hundreds of participating clients, though typical federated scenarios rarely exceed 8K tokens in our prompts.

We deploy Gemma 3 4B locally on the federation server rather than via external API calls, preserving the privacy guarantees central to federated learning. The model operates in 16-bit floating-point precision on a single NVIDIA A100 GPU with 8GB allocated memory per inference session. Although quantized variants exist (INT4, switched-FP8) that reduce memory footprint to 2.6-4.4GB, preliminary experiments showed minimal latency benefits given our single-query-

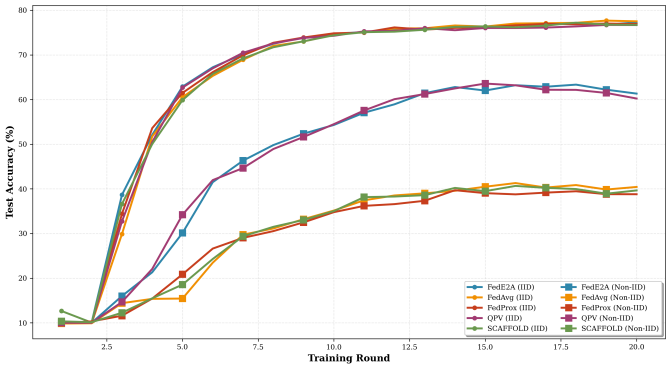


Fig. 2: IID vs non-IID performance across 20 rounds: Test Accuracy

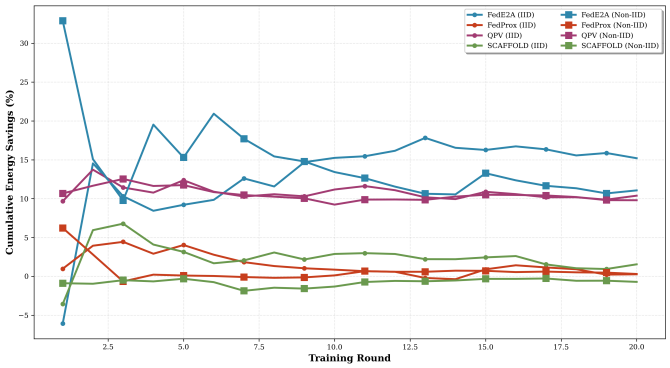


Fig. 3: IID vs non-IID performance: per-round energy

per-round pattern, and we prioritized numerical stability over memory compression.

The prompt comprises two parts. The system prompt establishes the LLM's role: "*You are a client selection advisor for federated learning. Your goal is to select clients that maximize model accuracy while respecting energy constraints and avoiding unreliable participants. Consider training phase, convergence state, and client quality profiles when determining selection priorities.*" The round-specific prompt contains the QPV table, current round number, total rounds, recent accuracy trajectory, cumulative energy expenditure, and any explicit constraints. The prompt concludes with a structured output request: "*Provide selection policy as JSON with fields: lambda_Q, lambda_S, lambda_H, lambda_E (weights in [0,1]), min_battery (threshold), min_stability (threshold), reasoning (brief explanation).*"

The LLM response is parsed and validated before use. Weights must be non-negative and are normalized to sum to one. Thresholds must fall within valid ranges. The reasoning field is logged for interpretability but does not affect selection. If parsing fails or constraints are violated, the system applies default parameters $(\lambda_Q, \lambda_S, \lambda_H, \lambda_E)$ with standard thresholds, ensuring robustness against LLM failures.

Once parameters are established, selection proceeds deterministically. Clients violating hard constraints (battery below threshold, stability below threshold) are excluded. Remaining clients are scored using Eq. (9) with LLM-suggested weights, and the top-k by score form the selection set S_t. The final selection is thus a deterministic function of QPVs and LLM-provided parameters, enabling reproducibility and debugging.

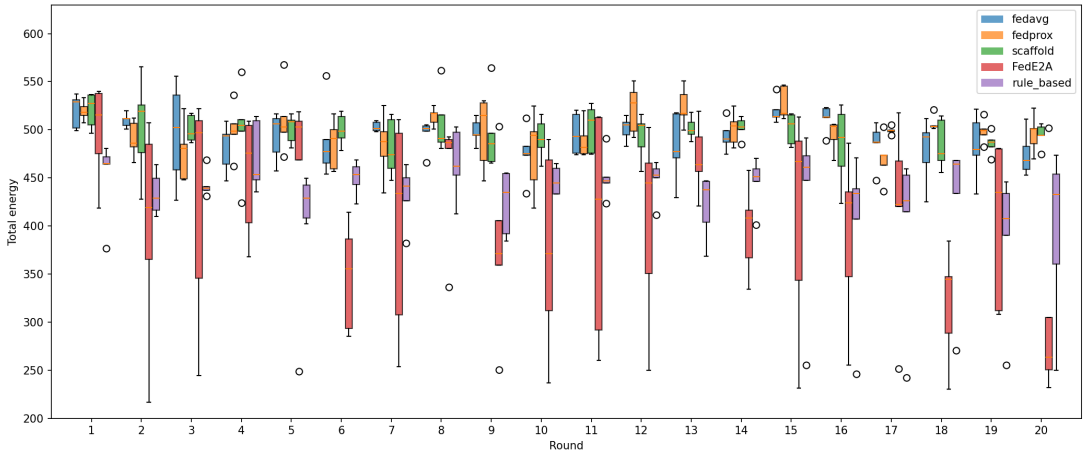
This architecture provides three key capabilities that fixed policies cannot match. First, the LLM adapts emphasis across training phases. In early rounds with low accuracy and high variance, it typically increases λ_Q to prioritize informative clients. As accuracy stabilizes, it shifts weight toward λ_H and λ_E to favor reliable, efficient contributors. This phase adaptation emerges from the LLM's interpretation of training context rather than explicit programming. Second, the LLM responds to anomalous patterns. When a client's stability score drops sharply, the LLM may raise the stability threshold or increase λ_H, effectively quarantining potentially compromised participants. When energy budgets tighten, it adjusts λ_E rather than arbitrarily excluding clients. Third, the LLM provides interpretable rationales. Each selection decision includes a reasoning field explaining the policy choice, enabling operators to understand system behavior and build trust in autonomous decisions. The LLM operates exclusively on aggregated, non-sensitive metadata: normalized quality scores, utilization percentages, and participation counts. No raw data, model parameters, or gradient information is exposed, preserving FL's privacy guarantees. For deployments requiring additional isolation, local LLM deployment eliminates external communication entirely.

V. Evaluation

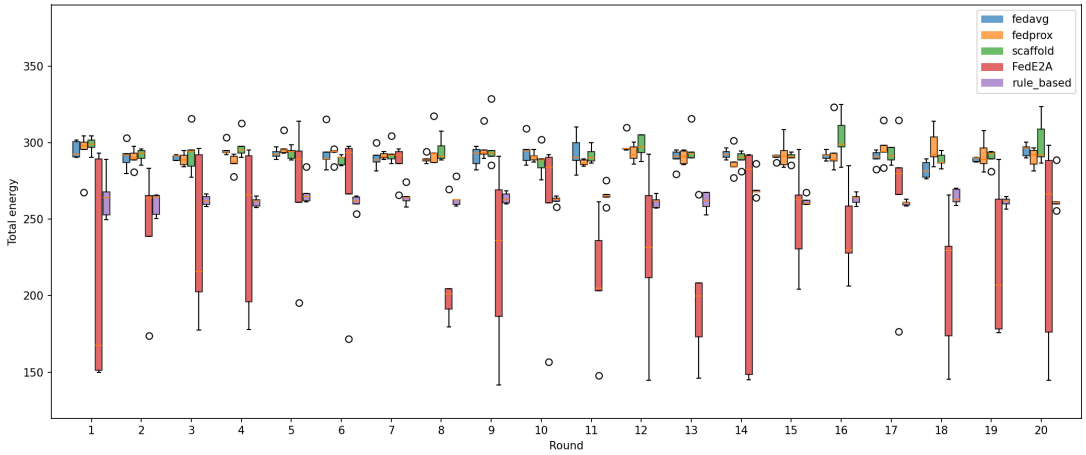
This section presents experimental results demonstrating that LLM-governed selection achieves superior accuracy-energy trade-offs compared to traditional FL algorithms and rule-based policies. All energy metrics use the normalized proxy defined in Eq. (8) rather than physical Joule measurements.

Experiments employ CIFAR-10 distributed across 10 clients under IID and extreme non-IID configurations. Non-IID scenarios use Dirichlet partitioning with $\alpha = 0.1$, producing severe label skew. Label noise is injected by flipping labels with 40% probability on selected clients. Each client's battery level is tracked and categorized as low (<30%), medium (30–60%), or high (>60%). The model is ResNet-18 adapted for 32×32 images, trained with Adam (learning rate 0.01) for 20 rounds. We compare FedE2A against FedAvg [14], FedProx [15], SCAFFOLD [16], and a rule-based variant using fixed thresholds without LLM governance. To ensure reproducibility, reported results use a deterministic FedE2A configuration with LLM-suggested parameters averaged across multiple query sessions.

Table I summarizes IID results. FedE2A achieves 77.24% accuracy with energy proxy 6517.8, yielding 15.2% savings relative to FedAvg while reducing accuracy by only 0.28 percentage points. All methods converge to similar accuracy between 76.71% and 77.52% by round 20, confirming that under IID conditions, aggregation dominates and selection primarily affects efficiency. FedE2A's advantage lies in adaptive



(a) IID setting



(b) Non-IID setting

Fig. 4: Total energy consumption over 10 runs: comparison between IID and Non-IID settings.

TABLE I: Performance on IID CIFAR-10

Method	Acc (%)	Energy	Savings (%)
FedE2A (Proposed)	77.24	6517.8	15.2
Rule-Based	76.68	6888.3	10.4
FedAvg	77.52	7686.3	—
FedProx	76.91	7666.1	0.3
SCAFFOLD	76.71	7567.5	1.5

TABLE II: Performance on Non-IID CIFAR-10 with 40% Label Noise

Method	Acc (%)	Energy	Savings (%)
FedE2A (Proposed)	61.33	7140.2	11.1
Rule-Based	60.23	7241.9	9.8
FedAvg	40.43	8028.3	—
FedProx	38.80	8004.0	0.3
SCAFFOLD	39.65	8085.8	−0.7

participation: it selects 5–10 clients per round, reducing to 5–7 as convergence stabilizes, while baselines select all 10 clients every round. The rule-based variant achieves 10.4% savings but lacks the phase adaptation that enables FedE2A's superior efficiency.

Table II presents the more challenging non-IID setting with 40% label noise. FedE2A achieves 61.33% accuracy—over 20 percentage points above FedAvg's 40.43%. Traditional methods collapse to near-random performance because they aggregate noisy gradients without quality filtering. FedE2A's LLM-governed selection implicitly identifies and excludes unreliable clients: the *QH* stability component flags clients whose updates diverge from Δ_{gold}, while the *QD* component distinguishes informative uncertainty from noise-induced entropy. The LLM interprets these signals in context, raising stability thresholds when reliability variance increases.

Despite the hostile environment, FedE2A maintains 11.1% energy savings. Absolute consumption increases across all methods relative to IID conditions, reflecting the harder optimization landscape, but quality-aware selection prevents the wasted computation that occurs when noisy clients participate. The rule-based variant achieves similar accuracy (60.23%) but with lower savings (9.8%), lacking the adaptive threshold adjustment that LLM governance provides.

Figures 2 and 3 visualize training dynamics. Under IID

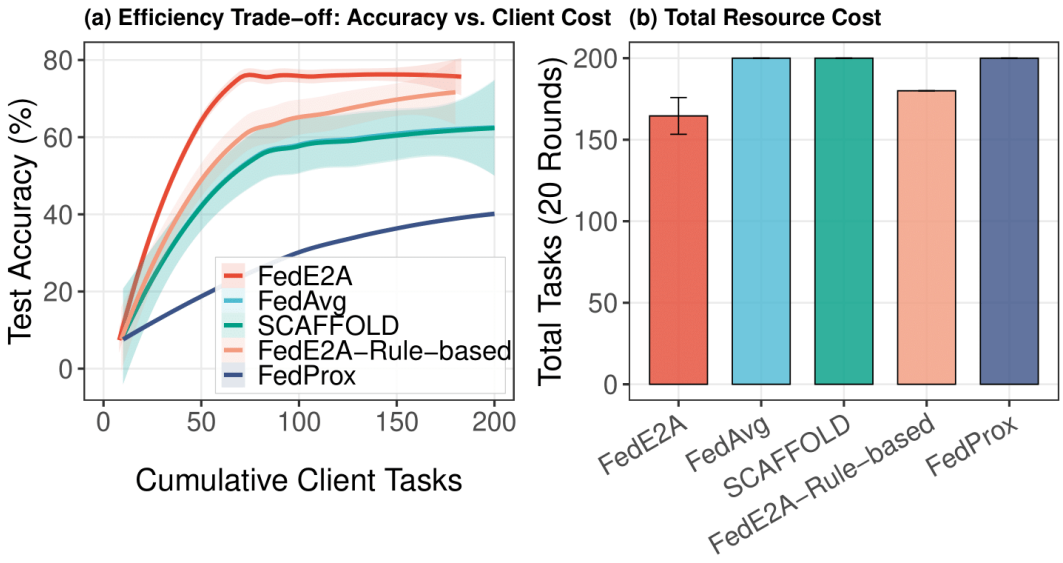


Fig. 5: Selected clients in IID and Non-IID settings over N_R=10 runs.

and non-IID conditions, all methods converge smoothly, but FedE2A's per-round energy drops as the LLM reduces participation near convergence. Under non-IID noise, traditional methods plateau at round 10 while FedE2A continues improving, demonstrating the sustained benefit of quality-aware selection. Figures 4a and 4b show the per-round total energy consumption with each boxplot representing the distribution over 10 runs in IID and non-IID settings, respectively. The number of clients selected by round, reported in Fgure 5, reveals FedE2A's adaptivity: it uses more clients in early noisy rounds for robustness, then reduces participation as reliable contributors are identified. Figure 5-(a) illustrates FedE2A's better Pareto efficiency, where a steeper learning trajectory indicates maximized information gain per resource unit compared to FedAvg and SCAFFOLD. Similarly, Figure 5-(b) confirms this aggregate saving: unlike static baselines that exhaust the maximum task budget (200), FedE2A's dynamic pruning of low-utility clients significantly reduces the total system load without compromising convergence.

These results demonstrate that LLM-governed selection achieves Pareto-superior outcomes in both settings. Under IID conditions, it trades negligible accuracy for substantial efficiency. Under adversarial noise, it improves both accuracy and efficiency simultaneously, surpassing the traditional trade-off between quality and cost. The LLM's contextual reasoning enables adaptive policies that fixed rules cannot replicate.

A natural concern regarding the proposed approach is whether the energy cost of LLM inference can be justified by improved client selection. To address this question, we conduct a scalability analysis examining how the trade-off between inference overhead and selection efficiency evolves as federation size increases.

We experimented federations ranging from $N = 1$ to $N = 1000$ clients under the same non-IID conditions and 40% label

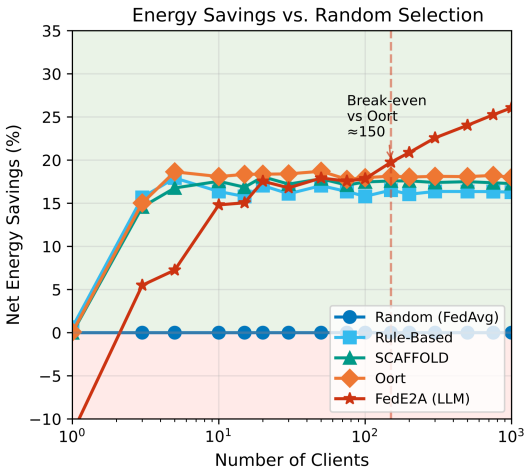


Fig. 6: Scalability Analysis: LLM Overhead

noise used in our main experiments. For client selection, we deploy Gemma 3n on a server-side GPU, while federated clients perform on-device training using NVIDIA Jetson Orin AGX 64GB edge devices operating at 60W in MAXN mode. We compare FedE2A against four baselines: random selection (FedAvg) [14], rule-based filtering with static thresholds, SCAFFOLD [16], and Oort [4].

Figure 6 presents the net energy savings relative to random selection as a function of client population. At small federation sizes ($N < 50$), FedE2A achieves comparable but not superior performance to Oort, as the fixed cost of LLM inference offsets the marginal improvements in selection quality. The break-even point against Oort occurs at approximately $N = 150$ clients, beyond which FedE2A consistently outperforms all baselines.

The divergence between methods becomes pronounced at

TABLE III: Energy consumption and model accuracy across federation sizes (20 rounds, non-IID data distribution, 40% label noise).

Method	$N = 50$			$N = 200$			$N = 1000$		
	E_{proxy}	Acc.	Sav.	E_{proxy}	Acc.	Sav.	E_{proxy}	Acc.	Sav.
Random	566	50.8	—	2253	50.8	—	11237	50.9	—
Rule-Based	470	55.5	17.0%	1891	55.2	16.0%	9407	55.4	16.3%
SCAFFOLD	465	56.8	17.8%	1857	56.6	17.6%	9296	56.6	17.3%
Oort	460	57.3	18.7%	1847	57.4	18.0%	9215	57.5	18.0%
FedE2A	**465**	**57.9**	17.9%	**1783**	**57.8**	**20.9%**	**8308**	**58.2**	**26.1%**

scale. At $N = 200$, FedE2A achieves 20.9% energy reduction compared to 18.0% for Oort. This gap widens further as federation size grows: at $N = 1000$, FedE2A reduces energy consumption by 26.1% while Oort achieves 18.0%, representing an 8.1 percentage point improvement. The scaling behavior arises from FedE2A's adaptive selectivity, which leverages larger client pools to identify higher-quality subsets while the relative cost of LLM inference diminishes from 3.8% of total energy at $N = 10$ to 0.3% at $N = 1000$.

These results suggest that LLM-governed client selection is most suitable for federations exceeding approximately 150 clients, where the inference overhead is amortized across sufficient training energy savings. For smaller deployments, conventional methods such as Oort provide comparable efficiency without requiring LLM infrastructure.

VI. Conclusion

This work introduced FedE2A, a federated learning framework where an LLM governs client selection by reasoning over structured Quality Profile Vectors. Unlike traditional approaches that rely on fixed policies or single-objective optimization, FedE2A leverages language model reasoning to navigate complex, dynamic trade-offs between data quality, system reliability, and energy efficiency. The QPV representation provides a principled interface for multi-dimensional client assessment, while the LLM governance module adapts selection policies based on training phase, convergence state, and anomaly detection.

Experiments demonstrate that LLM-governed selection achieves 15.2% energy savings under IID conditions with negligible accuracy impact, and over 20 percentage points accuracy improvement under 40% label noise compared to traditional baselines. These results establish that the contextual reasoning capabilities of LLMs provide meaningful advantages for multi-objective optimization in heterogeneous federated systems.

Several directions merit future investigation. Scaling to larger client populations will test whether LLM reasoning remains effective when QPV tables grow. Evaluation under explicit Byzantine threat models will clarify whether stability-based filtering provides robust defense or requires augmentation. Comparison across LLM architectures and sizes will illuminate the trade-offs between reasoning quality and inference cost. Finally, deployment in real-world edge environments will validate the energy proxy against physical measurements and expose practical challenges in LLM integration.

VII. Acknowledgments

The authors acknowledge the use of Gemini to refine the technical phrasing and improve the grammatical structure of this manuscript; however, the authors maintained full control over the conceptual development, data analysis, and final interpretative conclusions, and remain fully responsible for the content and accuracy of the work.

References

[1] M. L. Bernardi, M. Cimitile, and M. Usman, "DQFed: A federated learning strategy for non-IID data based on a quality-driven perspective," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 2024.
[2] F. Amato *et al.*, "Towards one-shot federated learning: Advances, challenges, and future directions," *Neurocomputing*, vol. 664, 2026.
[3] D. Song *et al.*, "Fast heterogeneous federated learning with hybrid client selection," in *Proc. UAI*, 2023.
[4] F. Lai *et al.*, "Oort: Efficient federated learning via guided participant selection," in *Proc. USENIX OSDI*, 2021.
[5] Z. Chai *et al.*, "TiFL: A tier-based federated learning system," arXiv:2001.09249, 2020.
[6] J. Yi *et al.*, "Robust quantity-aware aggregation for federated learning," arXiv:2205.10848, 2023.
[7] M. Rahmati, "Federated learning-based resource allocation for 6G NTN networks," *Satellite Communications*, vol. 2, 2025.
[8] Y. Zhang *et al.*, "A secure aggregation approach for wireless federated learning," *EURASIP J. Wireless Commun. Netw.*, 2025.
[9] J. Xia *et al.*, "Towards energy-aware federated learning via MARL," in *Proc. IEEE/ACM ICCAD*, 2025.
[10] A. Karakulev *et al.*, "Bayesian robust aggregation for federated learning," arXiv:2505.02490, 2025.
[11] S. Q. Zhang *et al.*, "A multi-agent reinforcement learning approach for efficient client selection," arXiv:2201.02932, 2022.
[12] M. L. Bernardi *et al.*, "A resource-aware adaptation approach for heterogeneous federated learning," in *Proc. IJCNN*, 2025.
[13] F. Maciel *et al.*, "Federated learning energy saving through client selection," *Pervasive Mobile Comput.*, vol. 103, 2024.
[14] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.
[15] T. Li *et al.*, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, 2020.
[16] S. P. Karimireddy *et al.*, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. ICML*, 2020.
[17] J. Lu, Q. Li, and H. Li, "FL-LLM: Leveraging large language models for efficient device selection in federated learning," SSRN preprint 5257764, 2025.
[18] Y. Zhou *et al.*, "Self-resource allocation in multi-agent LLM systems," arXiv:2504.02051, 2025.
[19] J. Huang *et al.*, "Leveraging large language models for efficient scheduling," *npj Advanced Manufacturing*, 2025.
[20] M. Zhou *et al.*, "Table meets LLM: Can large language models understand structured table data?" in *Proc. WSDM*, 2024.
[21] Z. Li *et al.*, "Exploring multimodal prompting strategies in LLMs," arXiv:2402.12424, 2024.
[22] Gemma Team, "Gemma 3 Technical Report," arXiv preprint arXiv:2503.19786 Mar. 2025.
[23] H. Li *et al.*, "An LLM-assisted decision-making framework of rule-based adaptive control," *Energy*, 2025.
[24] X. Zhang et al., "Energy-Efficient Federated Learning with Parameter Server over Wireless Networks," *IEEE Trans. Wireless Comm.*, 2024.