INTERACTION-AWARE CLIENT SELECTION IN FEDERATED LEARNING VIA LEARNED COALITION UTILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Client selection in federated learning is inherently combinatorial. The value of a training round depends on which clients train together, not on their individual contributions alone. Clients with overlapping data are redundant when combined, while those with complementary data reinforce one another, and which of these relationships matter shifts as the global model evolves. Methods that rank clients by individual importance, including Shapley-based selectors, and methods limited to pairwise correlations cannot capture this higher-order, evolving structure. We introduce NEXUSFL, which learns a surrogate of coalition utility directly and selects client subsets by their predicted joint utility. A tree-based surrogate captures client interactions up to a controllable order and is refit each round to track the evolving learning dynamics. The same surrogate proposes the candidate pool and chooses the aggregation set, and we analyse the induced selection regret. Because the fitted surrogate decomposes additively into interaction effects, each selection step scores a candidate by its individual effect and its interactions with the clients already selected, making the policy interpretable. Across twelve settings on nine datasets, NEXUSFL attains the highest trajectory-mean accuracy in every setting, by up to 12 points over the strongest baseline.

1 INTRODUCTION

Federated learning (FL) trains a shared model across decentralised data holders without requiring raw data to leave participating devices or organisations (McMahan et al., 2017; Kairouz & McMahan, 2021). It spans cross-device settings with large populations of edge devices and cross-silo settings in which organisations such as hospitals or financial institutions train collaboratively while retaining their data locally (Kairouz & McMahan, 2021). Communication, computation, availability, and energy constraints often make full participation impractical, particularly in cross-device deployments (Lai et al., 2021; Fu et al., 2023). Training therefore proceeds with only a subset of available clients in each round.

Which clients participate can substantially affect the resulting update. Repeatedly selecting redundant or uninformative participants wastes scarce participation opportunities, while selecting informative and complementary clients can improve training efficiency (Goetz et al., 2019; Balakrishnan et al., 2022). Client selection is therefore part of the learning problem itself (Cho et al., 2022). Yet federated client selection is usually treated as a problem of valuing clients (Sun et al., 2023; Singhal et al., 2024) even though the object that determines a training round is the coalition they form.

This distinction matters because client utility is not generally additive. Clients differ in their data distributions, dataset sizes, and label coverage (Fu et al., 2023), and the value of one client can depend on which others are selected alongside it (Tang et al., 2022; Huang et al., 2023). Figure 1 illustrates this difference. Two clients with similar individual values can form complementary or redundant pairs and therefore very different coalition utilities depending on their partners. A selector that scores each client individually keeps those scores fixed as the coalition is assembled, and therefore cannot prefer a complementary partner over a higher-scoring but redundant one.

Existing selectors differ in how much of this coalition structure they retain. Loss- and gradient-based selectors rank clients using individual training signals (Goetz et al., 2019; Cho et al., 2022). Shapley-based approaches account for coalition-dependent marginal contributions during valuation, but ultimately summarise them into a scalar value for each client before selection (Nagalapatti &

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Narayanam, 2021; Sun et al., 2023; Singhal et al., 2024). Such values can be useful for measuring contribution, but they do not preserve how a client's value changes with the particular coalition being assembled. Correlation-based methods go further by modelling pairwise client relationships (Tang et al., 2022), but their representation remains limited to second-order structure. These approaches motivate a more general question: *can we learn the utility of the coalition itself rather than reducing it to client-level scores or prescribed low-order relationships?*

We introduce NEXUSFL, which learns coalition utility directly. At each round, NEXUSFL evaluates subsets of a trained candidate pool and uses these observations to fit a tree-based surrogate from coalition membership to predicted utility. A persistent, recency-weighted reservoir lets the surrogate accumulate evidence across rounds while remaining responsive as the global model evolves. The learned utility map is then used both to construct the aggregation set and to propose the next candidate pool. Because the same surrogate admits an exact interaction decomposition, its selections can also be traced to the individual, pairwise, and higher-order client effects that drove them.

1.1 CONTRIBUTIONS

Our contributions can be summarized as follows: **(1) Coalition-utility surrogate.** We learn a coalition-utility surrogate that maps client subsets to predicted utility from evaluated subsets, and use it to guide client selection, with tree depth controlling the order of client interactions the surrogate can express. **(2) Persistent surrogate learning.** A single surrogate serves both pool proposal and final selection, determining which clients to train each round and which to aggregate. A persistent reservoir of past coalition evaluations enables cross-round learning, and recency-weighted refitting keeps the surrogate aligned with the evolving game. **(3) Interpretable selection policy.** The tree-based surrogate admits exact extraction of pairwise and higher-order interaction coefficients, providing per-round explanations of which client relationships drive selection. **(4) Empirical gains.** Across 12 settings on 9 datasets spanning vision, language, and tabular data, NEXUSFL attains the highest trajectory-mean accuracy in every setting, by up to 12 points over the strongest baseline.

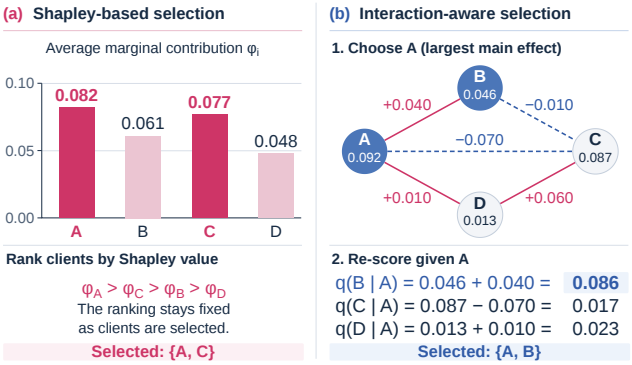


Figure 1: Clients with similar first-order scores can form complementary or redundant pairs. A ranking selector cannot tell them apart; an interaction-aware selector prefers the higher-utility pair.

2 CLIENT SELECTION IN FEDERATED LEARNING

Consider a federated learning system with client population $N = \{1, \ldots, n\}$. At round t, the server selects a subset $S_t \subseteq N$ of k clients. Starting from the current global model w_t, each selected client $i \in S_t$ computes a local update $g_i^{(t)}$, which the server aggregates to obtain

$$w_{t+1}(S_t) \;=\; w_t - \eta \sum_{i \in S_t} \alpha_i^{(t)} g_i^{(t)}, \tag{1}$$

where $\alpha_i^{(t)}$ is the aggregation weight assigned to client i. In FedAvg, participating clients are sampled at random and their updates are aggregated according to their local data weights (McMahan et al., 2017). Client-selection methods replace random participation with a policy that uses information about the clients to guide selection. A common approach is to assign each client a score and favour clients with larger scores. Active FL estimates the expected training benefit of individual clients (Goetz et al., 2019), Power-of-Choice favours clients with high local loss (Cho et al., 2022), and Oort combines statistical utility with system efficiency (Lai et al., 2021). Although their scoring and sampling rules differ, each method scores a client independently of which other clients are selected alongside it.

Other methods incorporate relationships between clients into the selection objective. DivFL promotes diversity through a submodular facility-location objective (Balakrishnan et al., 2022), FedCor models

pairwise correlations between client updates (Tang et al., 2022), and FilFL filters participants using a combinatorial objective (Fourati et al., 2024). These approaches move beyond independent client scoring, but encode how clients combine through particular forms of structure, such as diversity, pairwise correlation, or a specified combinatorial criterion.

Client Selection through Cooperative Game Theory. Cooperative game theory provides a general framework for assigning value to client subsets through a coalition utility function (Shapley et al., 1953). Data Shapley applies this framework to machine learning by treating training examples as players and attributing model performance to them through the Shapley value (Ghorbani & Zou, 2019). In FL, the same framework has been used for participant contribution evaluation and reward allocation (Wang et al., 2020; Song et al., 2019; Liu et al., 2022; Tastan et al., 2024; Liu et al., 2024; Guo et al., 2024).

Formally, at round t the clients are players and a subset $S \subseteq N$ is a coalition. Aggregating the updates from S through (1) yields the model $w_{t+1}(S)$. A characteristic function assigns a scalar utility to each coalition,

$$U_t : 2^N \to \mathbb{R}, \qquad S \longmapsto U_t(S). \tag{2}$$

For example, $U_t(S)$ may measure the change in accuracy on a server-side validation set (Nagalapatti & Narayanam, 2021; Sun et al., 2023), class-reweighted validation accuracy (Yang et al., 2025), the negative validation loss (Fourati et al., 2024), or the alignment of the aggregated update with a reference gradient (Xu et al., 2021). Under this utility, the size-k selection problem at round t is

$$S_t^\star \;=\; \arg \max_{S \subseteq N,\ |S|=k} U_t(S). \tag{3}$$

Shapley-based selectors assign each client a score equal to the weighted average of its marginal contributions $U_t(S \cup \{i\}) - U_t(S)$ over subsets $S \subseteq N \setminus \{i\}$. Their pipeline has three components: an *estimator* that approximates per-client Shapley values via Monte Carlo sampling (Nagalapatti & Narayanam, 2021) or other efficient variants (Singhal et al., 2024; Sun et al., 2023); an *updater* that carries estimates forward as running averages; and a *selector* that favours clients with high Shapley scores.

3 Interaction-Aware Client Selection

The Shapley value is sufficient for selection when U_t is additive, but its single score per client can discard how clients combine when utility contains interactions. The interaction decomposition of a set function (Grabisch & Roubens, 1999; Fujimoto et al., 2006) writes the utility in (2) as

$$U_t(S) = c_t + \sum_{i \in S} a_i^{(t)} + \sum_{\{i,j\} \subseteq S} b_{ij}^{(t)} + \cdots \tag{4}$$

where the sums collect first-order and pairwise effects, and the remaining terms are of higher order.

Figure 2 shows the share of variation in U_t attributable to each interaction order on a tractable 10-client CIFAR-10 partition over 100 rounds. Under lower-heterogeneity Dirichlet label skew with α=1, first-order effects still account for most of the variation. With α=0.1, their share falls to about 45%, while pairwise effects account for about 26% and a substantial share remains in higher-order terms. This pattern persists throughout training. Thus, label skew exposes utility structure that is not captured by scalar client values or pairwise relationships alone.

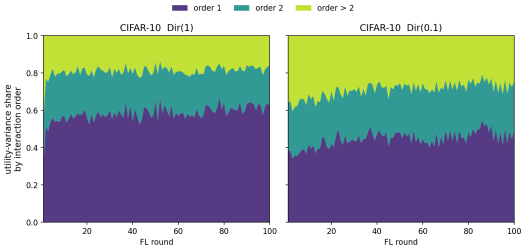


Figure 2: Share of the variation in U_t by interaction order on CIFAR-10. Stronger label skew moves more of the variation into pairwise and higher-order terms.

Under this structure, the marginal value of a client depends on which others are already selected (Figure 1). Extending the Shapley-based selection pipeline to interactions would stress each of its three components. The *estimator* now needs to cover $O(n^d)$ interaction quantities at order d, each

observed only sparsely under partial participation. The *updater* must normalise and refresh all of them as U_t shifts across rounds. The *selector* must then combine the estimated interactions to score candidate coalitions of size k.

3.1 NEXUSFL: LEARNING COALITION UTILITY DIRECTLY

Rather than estimating the coefficients of (4) individually, NEXUSFL learns the set function U_t as a whole and selects coalitions by their predicted utility. We represent each coalition by its indicator vector $\mathbf{1}[S] \in \{0,1\}^n$, whose i-th entry is 1 exactly when client i belongs to S, and the surrogate is therefore a function on the Boolean cube,

$$\hat{f}_t : \{0,1\}^n \to \mathbb{R}, \qquad \mathbf{1}[S] \longmapsto \hat{f}_t(\mathbf{1}[S]) \approx U_t(S). \tag{5}$$

Surrogate Training Data. Fitting $\hat{f}_t$ requires observations $(\mathbf{1}[S], U_t(S))$ from evaluated coalitions. Because a coalition can be evaluated only from the local updates of its members, a conventional round that trains only the k aggregated clients yields evaluations only among those k clients. This limited participation makes interactions particularly difficult to learn. Estimating a pairwise coefficient $b_{ij}^{(t)}$ in (4), for example, requires observations of coalitions containing both i and j.

NEXUSFL acquires additional observations by training a **candidate pool** $P_t \subseteq N$ of $p > k$ clients each round. A larger pool provides more interaction evidence by increasing the number of clients observed together. For fixed n, doubling p approximately quadruples the expected number of pairwise co-occurrences (Proposition 3). Because coalition evaluation can be expensive, NEXUSFL samples and evaluates B coalitions from P_t, rather than all 2^p subsets.

These B evaluations still lie entirely inside P_t, occupying a p-dimensional face of $\{0,1\}^n$ in the Boolean representation. They therefore give no training signal for coalitions that include clients outside P_t. Rather than training a larger pool, NEXUSFL accumulates coalition observations across rounds, allowing the surrogate to learn from a broader and more diverse set of client combinations. Since coalition utility shifts as the global model evolves, combining raw utility values over time can be problematic for surrogate training due to both value drift and scale changes. To mitigate this, NEXUSFL standardises each round's observed utilities before storage. For every coalition S in round t, it records $z_t(S) = (U_t(S) - \mu_t)/(\sigma_t + \epsilon_{\text{std}})$, where μ_t and σ_t are computed over this round's B evaluations and $\epsilon_{\text{std}} > 0$ ensures numerical stability. Each standardised observation $(\mathbf{1}[S], z_t(S), t)$ is stored in a **recency-weighted reservoir** where stale entries are more likely to be evicted, allowing the surrogate to adapt as utility evolves (see Section A.5 for details).

Learning the Surrogate. Any regression model over $\{0,1\}^n$ can serve as the surrogate class. We take $\mathcal{F}_d$ to be finite sums of regression trees of depth at most d, which are exactly the multilinear functions of degree at most d (Lemma 1) and therefore capture the interaction terms of (4) without estimating them individually. The same class also admits exact extraction of those coefficients (Section 3.3). In practice, we fit the ensemble with XGBoost (Chen & Guestrin, 2016), capping tree depth at d. Each reservoir entry collected at round t' is weighted by $\gamma^{\,t-t'}$ with $\gamma \in (0,1)$. Boosting fits $\hat{f}_t$ to the weighted least-squares objective

$$\min_{f \in \mathcal{F}_d} \sum_{(\mathbf{1}[S],\, z,\, t') \in \mathcal{R}} \gamma^{\,t-t'} \big(f(\mathbf{1}[S]) - z\big)^2. \tag{6}$$

Selection Policy. The fitted surrogate serves as a proxy for U_t in coalition search, and any search procedure that queries $\hat{f}_t$ can act as the selector. A query of $\hat{f}_t$ is a single pass through the trees, whereas an evaluation of $U_t(S)$ requires an aggregation step and a validation pass and becomes expensive for larger models (Section B). Exact maximisation over size-k subsets is nevertheless combinatorial and becomes costly for large n. NEXUSFL therefore constructs the coalition sequentially, spending one query per remaining candidate at each step. Given candidates Q, a partial coalition S, and a temperature $\tau > 0$, it scores every remaining client by $\hat{f}_t(\mathbf{1}[S \cup \{i\}])$ and samples from a softmax,

$$\Pr(i \mid S) \;\propto\; \exp\!\big(\hat{f}_t(\mathbf{1}[S \cup \{i\}])/\tau\big), \tag{7}$$

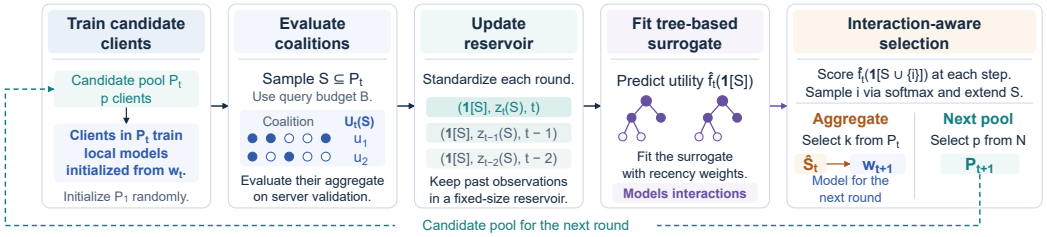


Figure 3: Round-level loop of NEXUSFL. Each round trains the candidate pool, evaluates and stores sampled coalitions, refits the surrogate, selects the aggregation set from the pool, and uses the same surrogate to propose the next pool.

repeating until the target size is reached. This policy is used twice, first to select the aggregation set $\hat{S}_t$ from P_t with temperature τ_{sel} and then to propose the next pool P_{t+1} from N with temperature τ_{pool}. In both uses, the temperature trades off exploitation of $\hat{f}_t$ against keeping lower-scoring candidates reachable. The full round-level loop is illustrated in Figure 3, and a formal algorithmic description is given in Appendix A.1. NEXUSFL is compatible with other surrogate models and selectors, and Sections F.10 and F.11 compare such alternatives in accuracy and runtime.

3.2 When Does Learned Coalition Utility Yield Good Selection?

Because the selector queries $\hat{f}_t$ rather than the true utility U_t, the quality of the selection depends on three factors: whether the candidate pool contains a good coalition, whether the surrogate ranks coalitions accurately inside the pool, and whether the practical selector finds a high-scoring subset. To separate those factors we introduce two intermediate coalitions between the unrestricted optimum $S_t^\star$ of (3) and the selected set $\hat{S}_t$. Let $S_{t,\text{pool}}^\star = \arg\max_{S \subseteq P_t,\, |S|=k} U_t(S)$ be the best coalition inside the realised pool, and $\tilde{S}_t = \arg\max_{S \subseteq P_t,\, |S|=k} \hat{f}_t(\mathbf{1}[S])$ the surrogate's in-pool maximiser. The per-round selection regret decomposes as

$$U_t(S_t^\star) - U_t(\hat{S}_t) = \underbrace{U_t(S_t^\star) - U_t(S_{t,\text{pool}}^\star)}_{\delta_t \text{ (pool gap)}} + \underbrace{U_t(S_{t,\text{pool}}^\star) - U_t(\tilde{S}_t)}_{\zeta_t \text{ (surrogate gap)}} + \underbrace{U_t(\tilde{S}_t) - U_t(\hat{S}_t)}_{\xi_t \text{ (selector gap)}}. \quad (8)$$

The three terms are the cost of restricting selection to P_t, of ranking in-pool coalitions with the surrogate rather than U_t, and of searching the surrogate by sequential softmax rather than exact maximisation. Because $\hat{f}_t$ is fit on standardised targets, we write those surrogate comparisons in utility units via $\tilde{U}_t = \mu_t + (\sigma_t + \epsilon_{\text{std}})\hat{f}_t$. The map is strictly increasing, and $\tilde{S}_t$ is unchanged.

The **surrogate gap** ζ_t is the error from ranking coalitions in P_t with the reservoir-fitted $\tilde{U}_t$ rather than U_t. If $\sup_{S \subseteq P_t,\, |S|=k} |\tilde{U}_t(S) - U_t(S)| \le \varepsilon_t$, then $\zeta_t \le 2\varepsilon_t$ (Proposition 2). The uniform error ε_t has three sources.

(1) *Representation.* A depth-d surrogate omits every interaction of order above d. Let $U_t^{(\le d)}$ be the order-d Möbius truncation of U_t (Grabisch & Roubens, 1999). The omitted mass on size-k coalitions is

$$\rho_{t,d} = \sup_{S \subseteq N,\, |S|=k} \left| U_t(S) - U_t^{(\le d)}(S) \right|. \quad (9)$$

(2) *Staleness.* The recency-weighted fit approximates $\bar{U}_t = \sum_{t' \le t} \alpha_{t,t'} U_{t'}$ rather than U_t, with $\alpha_{t,t'} \propto n_{t'} \gamma^{t-t'}$ and $n_{t'}$ the number of entries from round t' in $\mathcal{R}$. Its bias depends on how fast the game moves.

Assumption 1 (Temporal regularity)**.** *There is $L \ge 0$ such that $\sup_{S \subseteq N} |U_t(S) - U_{t'}(S)| \le L|t - t'|$ for all rounds t, t'.*

Proposition 1 (Staleness bias of recency weighting)**.** *Under Assumption 1, $\sup_S |\bar{U}_t(S) - U_t(S)| \le L A_t(\gamma)$, where $A_t(\gamma) = \sum_{t' \le t} \alpha_{t,t'}(t - t')$ is the weighted mean age of the reservoir.*

(3) *Estimation.* The remaining term $\varepsilon_t^{\text{est}}$ measures the weighted fit against the best depth-d model for $\bar{U}_t$. It also absorbs the mismatch introduced by per-round standardisation, since the stored targets

are rescaled by each round's own $(\mu_{t'}, \sigma_{t'})$. It depends on the supervision supplied by B, p, and the reservoir. Appendix C.3 decomposes the uniform error into these three sources,

$$\varepsilon_t \;\le\; \underbrace{\bar{\rho}_{t,d}}_{\text{representation}} \;+\; \underbrace{\varepsilon_t^{\text{est}}}_{\text{estimation}} \;+\; \underbrace{LA_t(\gamma)}_{\text{staleness}}, \tag{10}$$

where $\bar{\rho}_{t,d} = \max_{t'} \rho_{t',d}$ over the past games that still have observations in $\mathcal{R}$.

The **pool gap** δ_t is nonzero whenever the best size-k coalition lies outside the pool. It shrinks as p grows and as the selection policy places high-utility size-k coalitions in the pool. The **selector gap** ξ_t compares sequential softmax selection with exact surrogate maximisation. Combining (8) and (10) and Proposition 2 gives the cumulative result.

Theorem 1 (Cumulative selection regret)**.** *Under Assumption 1 and the condition of Proposition 2 in every round,*

$$\sum_{t=1}^{T}\big[U_t(S_t^\star) - U_t(\hat{S}_t)\big] \;\le\; \sum_{t=1}^{T}\delta_t + 2\sum_{t=1}^{T}\big(\bar{\rho}_{t,d} + \varepsilon_t^{\text{est}} + LA_t(\gamma)\big) + \sum_{t=1}^{T}\xi_t.$$

Section D measures the pool gap δ_t, the surrogate gap ζ_t, and the selector gap ξ_t exactly on a synthetic game with a large client population and known utility, and on a small CIFAR-10 population where every size-k coalition can be scored (Section D.2). The interaction spectrum (Figure 2) and the depth sweep (Section F.3) probe the representation term, Section F.5 tracks how the surrogate transfers across rounds as the game drifts, and Section 4.3 and Section F.8 vary B and the reservoir capacity, which control the estimation term.

Selection regret also enters an end-to-end convergence guarantee. When the utility is the reduction of a smooth server-side validation loss V, the per-round regret enters a standard FedAvg descent recursion as an additive bias, and the cumulative regret of Theorem 1 bounds the convergence of the iterates to a stationary neighbourhood (Theorem 2, Appendix C.6). Convergence on the training objective follows only insofar as the validation set is representative of training, through a bound Δ_{val} on the gradient mismatch along the iterates.

3.3 Interpretable Selection Policy

Because the surrogate is a tree ensemble over the coalition indicator, its interaction coefficients can be read off exactly. We obtain those coefficients by extracting the Walsh–Fourier expansion of the trees (Kushilevitz & Mansour, 1991; Gorji et al., 2026) and converting it in closed form (Butler et al., 2026) to a Faithful Banzhaf coefficient $I_{t,d}^{\text{FB}}(G)$ (FBII) (Tsai et al., 2023) for each client group G. This yields the exact representation

$$\hat{f}_t(\mathbf{1}[S]) \;=\; \sum_{G\in\mathcal{P}_{\le d}(S)} I_{t,d}^{\text{FB}}(G), \qquad \mathcal{P}_{\le d}(S) \;=\; \{G \subseteq S : |G| \le d\}. \tag{11}$$

The predicted utility of S is the sum of an intercept, the individual effects of clients in S, and their interactions of order at most d. For a partial selection S and client $i \notin S$, adding i activates exactly the terms that contain i, namely its individual effect and its interactions with clients already in S.

$$\Delta_t(i \mid S) \;:=\; \hat{f}_t(\mathbf{1}[S\cup\{i\}]) - \hat{f}_t(\mathbf{1}[S]) \;=\; \sum_{\substack{G\in\mathcal{P}_{\le d}(S\cup\{i\})\\ i\in G}} I_{t,d}^{\text{FB}}(G) \tag{12}$$

Since $\hat{f}_t(\mathbf{1}[S])$ is common to all candidates at a step, sampling by $\hat{f}_t(\mathbf{1}[S\cup\{i\}])$ in (7) is exactly sampling by $\Delta_t(i \mid S)$. Because tree traversal and summing the activated coefficients describe the same inference step, the extracted interactions explain the policy's selection decisions. Section E walks through the conversion and gives a worked example.

4 Experiments

Setup. We evaluate NexusFL across 9 datasets and 12 settings spanning vision, language, and tabular data. We report six settings here and a further six in Section F.1. We write $\mathrm{D}(\alpha)$ for a

Table 1: Main results across the benchmark suite. Final and trajectory-mean test accuracy (%) with standard deviations as subscripts over 5 seeds. Best per metric in **bold**.

	CIFAR-10 D(0.1)		SVHN D(0.1)		DBpedia D(0.1)		Covertype D(0.5)		CelebA D(0.5)		Sent140	
Method	Final	Mean	Final	Mean	Final	Mean	Final	Mean	Final	Mean	Final	Mean
FedAvg	$51.2_{\pm 3.0}$	$40.0_{\pm 1.2}$	$67.6_{\pm 5.9}$	$36.6_{\pm 4.3}$	$97.8_{\pm 0.4}$	$78.1_{\pm 2.9}$	$85.5_{\pm 1.9}$	$81.9_{\pm 1.2}$	$89.9_{\pm 1.4}$	$83.3_{\pm 1.2}$	$82.3_{\pm 2.2}$	$75.9_{\pm 1.6}$
S-FedAvg	$51.3_{\pm 5.3}$	$40.3_{\pm 1.1}$	$70.0_{\pm 1.2}$	$36.5_{\pm 3.9}$	$98.0_{\pm 0.2}$	$79.4_{\pm 0.3}$	$86.1_{\pm 1.3}$	$82.0_{\pm 1.2}$	$89.7_{\pm 2.1}$	$82.8_{\pm 0.9}$	$82.4_{\pm 2.9}$	$76.7_{\pm 1.6}$
AFedSV+	$52.7_{\pm 0.8}$	$43.0_{\pm 1.1}$	$50.6_{\pm 8.6}$	$24.3_{\pm 2.7}$	$97.0_{\pm 0.7}$	$75.4_{\pm 1.6}$	$86.4_{\pm 1.1}$	$82.7_{\pm 0.4}$	$91.2_{\pm 0.4}$	$86.2_{\pm 0.9}$	$84.4_{\pm 1.1}$	$78.7_{\pm 2.3}$
FedCor	$\mathbf{55.9}_{\pm 0.7}$	$45.6_{\pm 0.9}$	$70.0_{\pm 6.7}$	$37.6_{\pm 5.7}$	$98.1_{\pm 0.1}$	$78.3_{\pm 1.5}$	$88.5_{\pm 0.5}$	$84.3_{\pm 0.9}$	$90.9_{\pm 0.9}$	$83.7_{\pm 1.2}$	$78.4_{\pm 2.1}$	$76.6_{\pm 1.7}$
DivFL	$52.1_{\pm 4.4}$	$41.4_{\pm 0.7}$	$66.9_{\pm 4.1}$	$37.6_{\pm 4.6}$	$98.1_{\pm 0.2}$	$80.4_{\pm 3.4}$	$88.2_{\pm 1.6}$	$84.8_{\pm 0.3}$	$87.4_{\pm 3.3}$	$74.6_{\pm 4.3}$	$82.4_{\pm 3.4}$	$78.7_{\pm 2.1}$
FilFL	$50.5_{\pm 3.3}$	$40.3_{\pm 0.9}$	$32.9_{\pm 13.3}$	$16.1_{\pm 2.6}$	$97.6_{\pm 0.3}$	$78.6_{\pm 0.5}$	$87.3_{\pm 2.2}$	$84.5_{\pm 0.5}$	$91.2_{\pm 0.6}$	$85.7_{\pm 0.9}$	$84.2_{\pm 1.3}$	$78.8_{\pm 1.5}$
NEXUSFL	$55.5_{\pm 0.9}$	$\mathbf{48.0}_{\pm 1.1}$	$\mathbf{74.8}_{\pm 1.7}$	$\mathbf{49.9}_{\pm 4.7}$	$\mathbf{98.2}_{\pm 0.4}$	$\mathbf{88.7}_{\pm 0.5}$	$\mathbf{89.3}_{\pm 0.5}$	$\mathbf{86.0}_{\pm 0.2}$	$\mathbf{91.5}_{\pm 0.3}$	$\mathbf{87.6}_{\pm 0.6}$	$\mathbf{84.7}_{\pm 1.5}$	$\mathbf{81.1}_{\pm 0.8}$

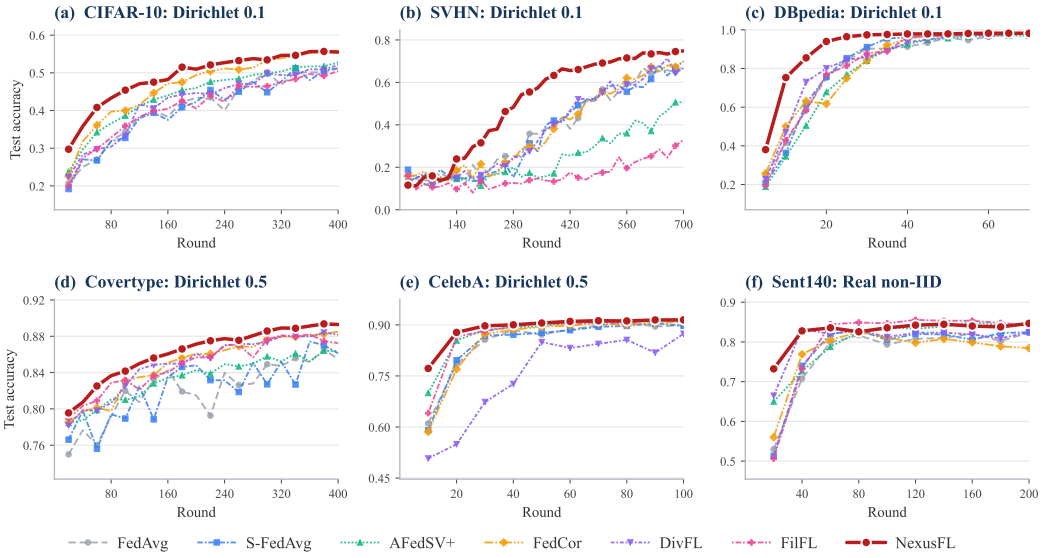


Figure 4: Learning curves across the benchmark suite. Panels show CIFAR-10, SVHN, and DBpedia under $\mathrm{D}(0.1)$, Covertype and CelebA under $\mathrm{D}(0.5)$, and Sent140 on its real-user partition.

Dirichlet label-skew partition with concentration α, where smaller α means stronger heterogeneity. CIFAR-10 and SVHN use a CNN under $\mathrm{D}(0.1)$ (Krizhevsky, 2009; Netzer et al., 2011). DBpedia uses DistilBERT under the same partition (Zhang et al., 2015). Covertype uses an MLP under $\mathrm{D}(0.5)$ (Blackard et al., 1998). CelebA uses a CNN on the smiling attribute under $\mathrm{D}(0.5)$ (Liu et al., 2015). Sent140 uses DistilBERT on its natural user partition (Go et al., 2009; Sanh et al., 2019). All experiments use $n = 100$ clients and select $k = 10$; NEXUSFL additionally trains a candidate pool of size $p = 20$. We report test accuracy as mean $\pm$ standard deviation over 5 seeds. We compare against **FedAvg** (McMahan et al., 2017) with uniform random selection, **S-FedAvg** (Nagalapatti & Narayanam, 2021) and **AFedSV+** (Sun et al., 2023) as Shapley-style baselines, correlation-based **FedCor** (Tang et al., 2022), diversity-based **DivFL** (Balakrishnan et al., 2022), and filtering-based **FilFL** (Fourati et al., 2024). For parity, all selection methods train a candidate pool of p=20 clients per round. S-FedAvg and AFedSV+ are additionally matched on the coalition budget, evaluating B=500 coalitions per round. Further details on implementation and experiments are in Appendix A.

Choice of Utility. The main experiments take U_t to be the change in accuracy on a server-side validation set. Section F.6 varies the size and label balance of this set and NEXUSFL remains robust to these changes. The same pipeline also finds high-value coalitions for other utilities, including gradient alignment (Xu et al., 2021), which needs no server validation set (Section F.9).

4.1 TRAJECTORY GAINS ACROSS BENCHMARKS

Table 1 reports both final accuracy and trajectory-mean accuracy, and Figure 4 shows the corresponding learning curves. Because client selection acts at every round, trajectory-mean accuracy captures

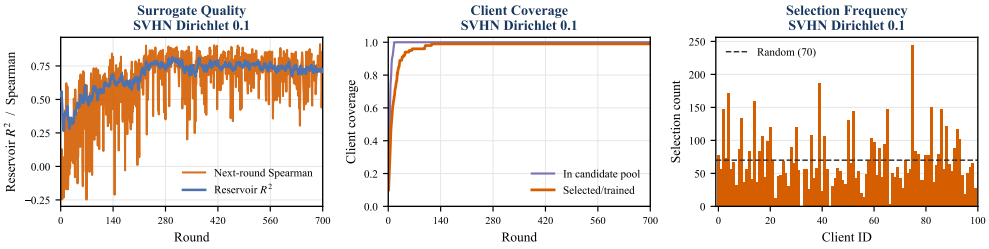


Figure 5: Surrogate quality, client coverage, and selection frequency on SVHN $\mathrm{D}(0.1)$. *Left:* reservoir R^2 and previous-round Spearman. *Centre:* cumulative fraction of clients that have entered the pool and the selected set. *Right:* per-client selection counts against the uniform expectation.

its effect throughout training. NEXUSFL has the highest trajectory mean on all 12 settings (Table 1 and Section F.1).

The gains are clearest in the more heterogeneous settings. On SVHN $\mathrm{D}(0.1)$ and DBpedia $\mathrm{D}(0.1)$, NEXUSFL reaches trajectory means of 49.9% and 88.7%, over 12 and 8 points above the strongest baseline. On the milder partitions the margins narrow but stay positive, consistent with Figure 2, where less of U_t sits in interaction terms as heterogeneity weakens. NEXUSFL also has the best final accuracy on SVHN, DBpedia, Covertype, CelebA, and Sent140, and is second on CIFAR-10 $\mathrm{D}(0.1)$.

The baseline ranking shifts across datasets. FedCor is strong on several tasks and takes the best final accuracy on CIFAR-10 $\mathrm{D}(0.1)$, while DivFL and FilFL are competitive elsewhere and Shapley-based baselines trail under strong heterogeneity. When more of U_t sits in interaction terms, there is more to gain from reasoning about the coalition than from ranking clients independently. NEXUSFL is the only method that turns this structure into a stronger training trajectory across modalities, heterogeneity regimes, and model classes.

4.2 SURROGATE QUALITY AND CLIENT COVERAGE

The regret bound of Theorem 1 depends on the surrogate tracking U_t across rounds. Figure 5 reports three diagnostics on SVHN $\mathrm{D}(0.1)$. The left panel checks how well the surrogate tracks U_t. Reservoir R^2 measures the fit on the reservoir coalitions, rising to about 0.75 by mid-training and remaining flat across 700 rounds. This in-sample fit is the observable counterpart of the estimation term $\varepsilon_t^{\mathrm{est}}$ in (10). The orange curve scores the previous round's surrogate on freshly sampled current-round coalitions. Because per-round standardisation changes with the global model, we report Spearman rank correlation here. Spearman averages about 0.63 (Table 14) with large early scatter and occasional dips. The previous-round surrogate therefore still carries a useful ranking signal for current-round coalitions, which is what pool proposal queries.

The centre panel shows the cumulative fraction of clients that have entered the candidate pool and the selected set. Both reach the full population after 120 rounds, as sequential softmax keeps lower-scoring clients reachable. The right panel shows per-client selection counts against the expected count of 70 under uniform selection. Selection is uneven, with some clients well above that line, an intended bias that reflects the surrogate preferring certain clients more than others. Section F.5 reports the same three diagnostics for every setting in the benchmark.

4.3 SCALING AND EFFICIENCY

Figure 6 reports two views of computational cost. Panel (a) breaks one round into its components on a CNN and on DistilBERT, and panel (b) plots final accuracy against per-round time as B varies on CIFAR-10 $\mathrm{D}(0.1)$ for NEXUSFL and the Shapley-based baselines.

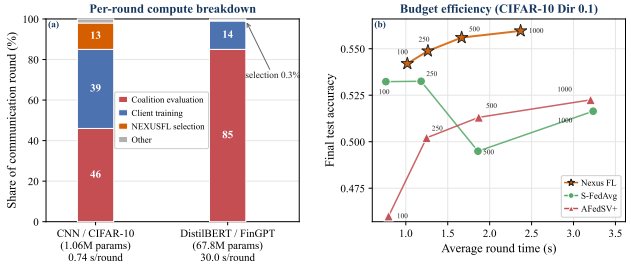


Figure 6: (a) Share of per-round time on a CNN and on DistilBERT. (b) Final accuracy versus per-round time as B varies on CIFAR-10 $\mathrm{D}(0.1)$.

Surrogate fitting, final selection, and pool proposal together (NEXUSFL) account for 13% of a round on CIFAR-10 (1.06M parameters) and only 0.3% on FinGPT (67.8M parameters). Client training and coalition

evaluation dominate the remainder. Since coalition evaluation scales with model size while surrogate fitting and inference do not, NEXUSFL's relative overhead decreases as models grow. Section B gives the runtime complexity of each stage and confirms this scaling.

Since coalition evaluation is the dominant cost and matters even more as models grow, panel (b) asks how effectively that coalition-evaluation budget is used. NEXUSFL is compared against S-FedAvg and AFedSV+, the Shapley-based baselines that spend their budget on the same coalition evaluations. NEXUSFL already exceeds both at $B{=}100$, and its accuracy keeps improving as B grows, whereas the Shapley baselines gain less from the same evaluations. The surrogate pipeline therefore converts additional coalition evaluations into better selections more reliably than the Shapley alternatives. Section F.2 further shows that NEXUSFL remains competitive when scaling to $n{=}1000$ clients.

4.4 POLICY INTERPRETABILITY

The pairwise FBII of the surrogate (Section 3.3) lets us interpret the selection policy each round in terms of which client pairs it treats as complementary or redundant. To validate that these coefficients capture meaningful structure, we construct a controlled MNIST experiment with known ground-truth complementarity where 20 clients each hold exactly 3 of 10 digit classes, arranged so that neighbours on a ring share most classes (redundant) while opposite nodes share none (complementary). Figure 7 visualises the pairwise FBII network at two training snapshots alongside the interaction gain of selected versus unselected candidates.

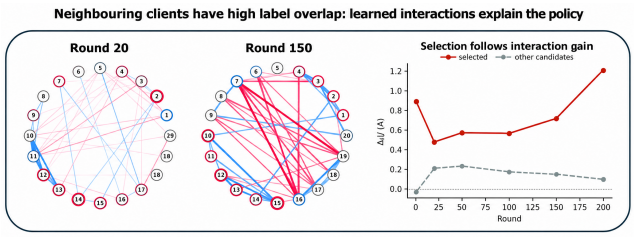


Figure 7: MNIST blocked-triplet partition (20 clients, $k{=}10$). Red/blue edges denote positive/negative pairwise FBII; thickness scales with magnitude. *Left:* round 20. *Centre:* round 150. *Right:* interaction gain of selected (red) vs. unselected (grey) candidates.

At round 20 (left) the interaction structure is diffuse and the surrogate has not yet differentiated redundant from complementary pairs. By round 150 (centre), thick red edges connect distant clients with disjoint classes and thick blue edges link nearby clients with overlapping classes, confirming that the surrogate has learned to reward complementarity and penalise redundancy. The right panel tracks the marginal interaction gain $\Delta f(j \mid A)$ that each candidate would add to the current partial selection. We see that selected clients (red) consistently score above unselected ones (grey), showing that the selector actively exploits the learned interaction signal. Section F.7 finds the same pattern on CIFAR-10 under Dirichlet skew where complementarity has no ground-truth labels. Pairwise coefficients stay positively correlated with how much the two clients' label distributions differ.

5 CONCLUSION

We introduced NEXUSFL, a method that reframes federated client selection as a coalition-level decision. By learning a coalition-utility surrogate from a persistent reservoir of subset evaluations, NEXUSFL captures complementarity and redundancy among clients without explicitly estimating interaction indices. The fitted surrogate decomposes into individual and interaction effects, and each selection step scores a candidate by those coefficients. Across 12 settings on 9 datasets spanning vision, language, and tabular data, NEXUSFL attains the highest trajectory-mean accuracy, with the largest margins under strong heterogeneity where per-client scoring methods degrade. The surrogate remains predictive across rounds and maintains broad client coverage, providing a practical route to interaction-aware federated learning.

Limitations and Future Work. Surrogate quality depends on the coalition budget B, since more evaluations improve the fit but increase validation cost. In this work, we keep B fixed. Adapting it across rounds would require reliable uncertainty estimates under drift (Kim & Choi, 2022). We leave this coupling to future work. Further, when a validation set is used to score utility, NEXUSFL is sensitive to whether that set covers the classes of interest (Section F.6).

Ethics Statement

This work uses only standard, publicly available benchmark datasets (CIFAR-10, SVHN, DBpedia, Covertype, CelebA, Sent140, FinGPT, Yelp, PathMNIST) in a simulated federated learning setting and does not involve human subjects, private user data, or the release of high-risk models. Client selection changes how often individual clients participate, and deployments should monitor participation balance and downstream fairness. We are aware of the ICLR Code of Ethics and believe this work conforms to it.

Reproducibility Statement

We provide the information needed to reproduce our results. The full method is specified in Algorithm 1. Datasets and federated partitions are described in Appendix A.2, baselines in Appendix A.3, and all training and surrogate hyperparameters in Appendix A.4 (Tables 2 to 4). Hardware and compute are reported in Appendix A.6. Complete assumptions and proofs for all theoretical results appear in Appendix C. All main results report mean and standard deviation over five seeds.

AI Use Statement

Generative AI tools were used to assist with writing and editing (polishing prose and LaTeX formatting) and with coding. They were not used as a component of the proposed method, nor for generating research ideas, data, or citations. All AI-assisted text and code were reviewed and verified by the authors, who take full responsibility for the final content of this paper.

References

Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022.

JA Blackard, DJ Dean, and CW Anderson. Covertype data set, uci machine learning repository, 1998.

Landon Butler, Abhineet Agarwal, Justin Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. Proxyspex: Inference-efficient interpretability via sparse feature interactions in llms. *Advances in Neural Information Processing Systems*, 38:72306–72340, 2026.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards Understanding Biased Client Selection in Federated Learning. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pp. 10351–10375. PMLR, 2022.

Xiaotie Deng and Christos H Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

Fares Fourati, Salma Kharrat, Vaneet Aggarwal, Mohamed-Slim Alouini, and Marco Canini. Filfl: Client filtering for optimized client participation in federated learning. In *ECAI 2024*, pp. 2460–2467. IOS Press, 2024.

Lei Fu, Huanle Zhang, Ge Gao, Mi Zhang, and Xin Liu. Client Selection in Federated Learning: Principles, Challenges, and Opportunities. *IEEE Internet of Things Journal*, 10(24):21811–21819, December 2023. ISSN 2327-4662, 2372-2541. doi: 10.1109/JIOT.2023.3299573.

Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1): 72–99, 2006.

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Shap-iq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 36:11515–11551, 2023.

Amirata Ghorbani and James Zou. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 2019.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. Active Federated Learning, September 2019.

Ali Gorji, Andisheh Amrollahi, and Andreas Krause. Shap values via sparse fourier representation. *Advances in Neural Information Processing Systems*, 38:18827–18860, 2026.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

Peng Guo, Yanqing Yang, Wei Guo, and Yanping Shen. A Fair Contribution Measurement Method for Federated Learning. *Sensors*, 24(15):4967, July 2024. ISSN 1424-8220. doi: 10.3390/s24154967.

Jiyue Huang, Chi Hong, Yang Liu, Lydia Y. Chen, and Stefanie Roos. Maverick Matters: Client Contribution and Selection in Federated Learning. volume 13936, pp. 269–282. Springer Nature Switzerland, Cham, 2023. ISBN 978-3-031-33376-7 978-3-031-33377-4. doi: 10.1007/978-3-031-33377-4_21.

Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Jungtaek Kim and Seungjin Choi. On uncertainty estimation by tree-based surrogate models in sequential model-based optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 4359–4375. PMLR, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pp. 455–464, 1991.

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient Federated Learning via Guided Participant Selection. In *15th USENIX Symposium on Operating Systems Design and Implementation*, 2021.

Yiqi Liu, Shan Chang, Ye Liu, Bo Li, and Cong Wang. FairFed: Improving Fairness and Efficiency of Contribution Evaluation in Federated Learning via Cooperative Shapley Value. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pp. 621–630, Vancouver, BC, Canada, May 2024. IEEE. ISBN 979-8-3503-8350-8. doi: 10.1109/INFOCOM52122.2024.10621438.

Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–21, August 2022. ISSN 2157-6904, 2157-6912. doi: 10.1145/3501811.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Velibor V Mišić. Optimization of tree ensembles. *Operations Research*, 68(5):1605–1624, 2020.

Lokesh Nagalapatti and Ramasuri Narayanam. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9046–9054, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Lloyd S Shapley et al. A value for n-person games. 1953.

Pranava Singhal, Shashi Raj Pandey, and Petar Popovski. Greedy Shapley Client Selection for Communication-Efficient Federated Learning. *IEEE Networking Letters*, 6(2):134–138, June 2024. ISSN 2576-3156. doi: 10.1109/LNET.2024.3363620.

Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit Allocation for Federated Learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2577–2586, Los Angeles, CA, USA, December 2019. IEEE. ISBN 978-1-7281-0858-2. doi: 10.1109/BigData47090.2019.9006327.

Qiheng Sun, Xiang Li, Jiayao Zhang, Li Xiong, Weiran Liu, Jinfei Liu, Zhan Qin, and Kui Ren. ShapleyFL: Robust Federated Learning Based on Shapley Value. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2096–2108, Long Beach CA USA, August 2023. ACM. ISBN 979-8-4007-0103-0. doi: 10.1145/3580305.3599500.

Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10092–10101, New Orleans, LA, USA, June 2022. IEEE. doi: 10.1109/cvpr52688.2022.00986.

Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horváth, and Karthik Nandakumar. Redefining Contributions: Shapley-Driven Federated Learning. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, pp. 5009–5017, Jeju, South Korea, August 2024. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/554.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023.

Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. In *Federated Learning: Privacy and Incentive*, pp. 153–167. Springer, 2020.

Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Advances in Neural Information Processing Systems*, 34:16104–16117, 2021.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*, 2021.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023a.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific data*, 10(1):41, 2023b.

Mengwei Yang, Baturalp Buyukates, and Athina Markopoulou. Rewarding the Rare: Maverick-Aware Shapley Valuation in Federated Learning. *Transactions on Machine Learning Research*, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

APPENDIX

This appendix provides implementation details for NEXUSFL, describes the datasets and experimental setup, reports computational complexity, gives proofs for the theoretical statements in the main text, measures the regret decomposition empirically, walks through interaction extraction, and collects additional results and qualitative diagnostics.

A Implementation and Experimental Details . . . 14
- A.1 Algorithm and Method Implementation . . . 14
- A.2 Datasets and Federated Partitions . . . 14
- A.3 Baselines . . . 15
- A.4 Hyperparameters . . . 15
- A.5 Reservoir Implementation . . . 16
- A.6 Hardware and Compute . . . 16

B Computational Complexity . . . 16

C Proofs . . . 17
- C.1 Proof of Proposition 2 (surrogate-to-selection transfer) . . . 17
- C.2 Proof of Lemma 1 (interaction capacity) . . . 18
- C.3 Proof of Proposition 1 and of (10) . . . 18
- C.4 Pair co-occurrence under uniform pool sampling . . . 19
- C.5 Proof of Theorem 1 (cumulative selection regret) . . . 19
- C.6 End-to-end convergence . . . 19

D Empirical Regret Decomposition . . . 21
- D.1 Exact decomposition on a synthetic game . . . 21
- D.2 Pool gap on a real dataset . . . 22

E Interaction Extraction and Worked Example . . . 22

F Additional Results and Ablations . . . 24
- F.1 Additional Benchmark Settings . . . 24
- F.2 Client-population scaling . . . 25
- F.3 Pool size and tree depth . . . 25
- F.4 Isolating pool proposal and final selection . . . 26
- F.5 Surrogate generalisation and non-stationarity . . . 27
- F.6 Robustness to the server validation set . . . 31
- F.7 Learned interactions on real data . . . 32
- F.8 Reservoir size ablation . . . 32
- F.9 Value function ablation . . . 33
- F.10 Selection mechanism ablation . . . 33
- F.11 Surrogate class . . . 33

A Implementation and Experimental Details

A.1 Algorithm and Method Implementation

The complete round-level procedure follows. The first-round pool P_1 is drawn uniformly from N; thereafter the fitted surrogate $\hat{f}_{t-1}$ proposes P_t.

Algorithm 1 NexusFL client selection at round t

Require: clients N, current model w_t, server step size η, aggregation weights $\alpha_i^{(t)}$, pool size p, selection size k, coalition budget B, reservoir $\mathcal{R}$, tree depth d, recency factor γ, eviction decay κ, selection temperature τ_{sel}, pool-proposal temperature τ_{pool}, stability constant ϵ_{std}, coalition utility U_t ((2))

Ensure: selected clients $\hat{S}_t$, next model w_{t+1}, and the fitted $\hat{f}_t$ and updated reservoir $\mathcal{R}$ carried to round $t+1$

$P_t \leftarrow$ SeqSoftmaxSelect(N, p, $\hat{f}_{t-1}$, τ_{pool}) ▷ propose candidate pool; round 1: sample P_1 uniformly
Train each client $i \in P_t$ locally from w_t and cache its update $g_i^{(t)}$.
Sample B coalitions $\mathcal{B}_t = \{S_t^{(b)}\}_{b=1}^{B}$ from P_t with balanced coverage across sizes.
for $S \in \mathcal{B}_t$ **do**
    Form the coalition model $w_{t+1}(S) = w_t - \eta \sum_{i \in S} \alpha_i^{(t)} g_i^{(t)}$ as in (1).
    Evaluate $U_t(S)$ for the coalition model $w_{t+1}(S)$.
end for
Standardise utilities within the round: $z_t(S) = (U_t(S) - \mu_t)/(\sigma_t + \epsilon_{\text{std}})$ for all $S \in \mathcal{B}_t$.
Add $\{(\mathbf{1}[S], z_t(S), t) : S \in \mathcal{B}_t\}$ to $\mathcal{R}$; if $\mathcal{R}$ overflows, evict entries without replacement with probability $\propto \exp(\kappa \cdot \text{age})$ (equal-age ties broken by the draw).
Fit the coalition-utility surrogate $\hat{f}_t \in \mathcal{F}_d$ by gradient boosting on the weighted objective $\sum_{(\mathbf{1}[S], z, t') \in \mathcal{R}} \gamma^{t-t'} (f(\mathbf{1}[S]) - z)^2$.
$\hat{S}_t \leftarrow$ SeqSoftmaxSelect(P_t, k, $\hat{f}_t$, τ_{sel}) ▷ final aggregation set
Aggregate selected cached updates: $w_{t+1} \leftarrow w_t - \eta \sum_{i \in \hat{S}_t} \alpha_i^{(t)} g_i^{(t)}$.
return $\hat{S}_t, w_{t+1}, \hat{f}_t, \mathcal{R}$ ▷ $\hat{f}_t$ proposes the next round's pool

function SeqSoftmaxSelect(candidates Q, $numToSelect$, surrogate $\hat{f}$, temperature τ)
    $S \leftarrow \emptyset$
    while $|S| < numToSelect$ **do**
        For each $i \in Q \setminus S$, score $q_i = \hat{f}(\mathbf{1}[S \cup \{i\}])$.
        Sample $i^\star \sim \text{softmax}(\{q_i/\tau : i \in Q \setminus S\})$.
        $S \leftarrow S \cup \{i^\star\}$.
    end while
    return S
end function

A.2 Datasets and Federated Partitions

We evaluate NexusFL on nine datasets that together cover vision, text, and tabular data, for twelve settings in total. As in the main text, $\mathrm{D}(\alpha)$ denotes a Dirichlet label-skew partition with concentration α, and smaller α means stronger heterogeneity.

Six settings appear in the main text. In vision, **CIFAR-10** (Krizhevsky, 2009) and **SVHN** (Netzer et al., 2011) use a CNN under $\mathrm{D}(0.1)$, and **CelebA** (Liu et al., 2015), where the task is to predict the smiling attribute, uses a CNN under $\mathrm{D}(0.5)$. For text, **DBpedia** (Zhang et al., 2015) is a topic-classification task under $\mathrm{D}(0.1)$, and **Sent140** (Go et al., 2009) is a Twitter sentiment task with its natural user partition from LEAF (Caldas et al., 2018). We fine-tune DistilBERT (Sanh et al., 2019) on both. For tabular data, **Covertype** (Blackard et al., 1998) forest-cover classification uses an MLP under $\mathrm{D}(0.5)$.

The appendix adds six more settings. CIFAR-10 appears again under $\mathrm{D}(0.5)$ and under long-tail imbalance with imbalance ratio $\mathrm{IR}{=}10$. **FinGPT** (Yang et al., 2023a) is a financial sentiment task under $\mathrm{D}(0.1)$ with a DistilBERT backbone. **Yelp** (Zhang et al., 2015) is a review-classification task under $\mathrm{D}(0.1)$ and $\mathrm{D}(0.5)$, also with DistilBERT. Finally, **PathMNIST** (Yang et al., 2023b) is a medical-imaging classification task under $\mathrm{D}(0.1)$ with a CNN.

Every setting has $n{=}100$ clients, and $k{=}10$ of them are aggregated in each round. NEXUSFL first trains a candidate pool of $p{=}20$ clients and then selects the aggregation set from it. For Sent140, we keep the natural user identities as clients and take the top-n users with enough local examples. Unless stated otherwise, accuracies are averaged over 5 seeds, and the result tables show standard deviations as subscripts.

A.3 BASELINES

We compare against six baselines: **FedAvg** (McMahan et al., 2017), **S-FedAvg** (Nagalapatti & Narayanam, 2021), **AFedSV+** (Sun et al., 2023), **FedCor** (Tang et al., 2022), **DivFL** (Balakrishnan et al., 2022), and **FilFL** (Fourati et al., 2024). A fair comparison requires that every method sees the same amount of client training, so that differences in accuracy reflect the selection rule and not access to more trained candidates. All baselines except FedAvg therefore train, on average, a pool of $p{=}20$ clients per round and select $k{=}10$ from within it. The two Shapley-based selectors, S-FedAvg and AFedSV+, spend their budget on the same resource as NEXUSFL, namely coalition evaluations, so we also match them on the coalition budget $B{=}500$. Table 2 lists their method-specific hyperparameters.

Matching Training and Communication Budgets. For most methods in Table 1, the match follows directly from the protocol: they train $p{=}20$ candidates and aggregate $k{=}10$, so client-side training and upload volume equal those of NEXUSFL. FilFL (Fourati et al., 2024) trains all 100 clients every ninth round (its filtering interval) and 10 clients otherwise, which averages to $(100 + 8 \cdot 10)/9 = 20$ client trainings per round. DivFL (Balakrishnan et al., 2022) runs in its practical variant, which probes only the 20 candidates; their gradients refresh the dissimilarity matrix, and 10 of them are aggregated. FedAvg is matched by setting D of Section F.4, which trains and aggregates 20 random clients per round. FedCor (Tang et al., 2022) is the one method that communicates more than the others. During warm-up and Gaussian-process data collection, its protocol periodically broadcasts candidate models to all clients and collects their local losses, in addition to the uploads of the selected clients.

Table 2: Hyperparameters for budget-matched baselines. All listed methods train $p{=}20$ candidates per round. S-FedAvg and AFedSV+ also share the coalition budget $B{=}500$.

Method	Parameter	Value
S-FedAvg	MC permutations per round	$\lfloor B/p \rfloor = 25$
	Credit EMA weights (old/new)	0.75 / 0.25
AFedSV+	CC-Shapley samples per round	$\lceil B/(2p{-}1) \rceil = 13$
	Shapley EMA decay γ_{sv}	0.5

A.4 HYPERPARAMETERS

All main settings share one configuration of NEXUSFL. Each round trains a candidate pool of $p{=}20$ clients and evaluates $B{=}500$ coalitions from it. The reservoir keeps up to $|\mathcal{R}|_{\max}{=}2500$ observations and evicts old entries with decay $\kappa{=}0.5$, and the surrogate down-weights past rounds with recency factor $\gamma{=}0.5$. Final selection samples with temperature $\tau_{\mathrm{sel}}{=}0.01$ and pool proposal with $\tau_{\mathrm{pool}}{=}0.05$, and utilities are standardised with $\epsilon_{\mathrm{std}}{=}10^{-8}$. The surrogate itself is an XGBoost ensemble of 200 trees of depth at most $d{=}4$, trained with learning rate 0.1, row and column subsampling of 0.8, and L2 regularisation $\lambda{=}5.0$.

The federated training setup differs by dataset and is listed in Table 3. Table 4 describes the four models. Every setting has $n{=}100$ clients, aggregates $k{=}10$ of them per round, and runs one local epoch per round. Text inputs are truncated to 64 tokens on Sent140 and FinGPT, 128 on DBpedia, and 256 on Yelp. On Yelp, each client holds 250 training examples, and on DBpedia, 5,040.

Table 3 gives the federated training configuration of every setting, and Table 4 gives the model architectures.

Table 3: Federated training configuration per dataset. The models are described in Table 4.

Dataset	Partition	Model	Rounds T	Optimiser	Learning rate
CIFAR-10	D(0.1), D(0.5), long-tail IR= 10	CNN	400	SGD, momentum 0.5	0.01
SVHN	D(0.1)	CNN	700	SGD, momentum 0.5	0.001
PathMNIST	D(0.1)	CNN	400	SGD, momentum 0.5	0.01
CelebA	D(0.5)	CNN-GN	100	SGD, momentum 0.5	0.01
Covertype	D(0.5)	MLP	400	SGD, momentum 0.5	0.05
Sent140	natural users (top 100)	DistilBERT-LoRA	200	AdamW	2×10^{-5}
FinGPT	D(0.1)	DistilBERT-LoRA	200	AdamW	10^{-4}
DBpedia-14	D(0.1)	DistilBERT-LoRA	70	AdamW	10^{-6}
Yelp Review Full	D(0.1), D(0.5)	DistilBERT-LoRA	100	AdamW	10^{-4}

Table 4: Model architectures. Convolutions are 3×3, pooling is 2×2 max-pooling, and both CNNs use dropout 0.2. C is the number of classes. For DistilBERT, the number in parentheses is the number of trainable parameters.

Model	Layers	Parameters
CNN	conv(16) $\rightarrow$ pool $\rightarrow$ conv(32) $\rightarrow$ pool $\rightarrow$ FC(512) $\rightarrow$ FC(C)	1.06M
CNN-GN	$3\times$[conv $\rightarrow$ GroupNorm(8) $\rightarrow$ pool] with 32, 64, 128 channels $\rightarrow$ avg-pool to 4×4 $\rightarrow$ FC(256) $\rightarrow$ FC(2)	619K
MLP	FC(128) $\rightarrow$ FC(64) $\rightarrow$ FC(2)	15.4K
DistilBERT-LoRA	frozen `distilbert-base-uncased`; LoRA (r=8, α=16, dropout 0.05) on the query, key, value, and output projections $\rightarrow$ linear head with C classes	67.0M (0.89M)

A.5 Reservoir Implementation

The reservoir $\mathcal{R}$ is the memory of the surrogate. It holds coalition observations from past rounds, so the surrogate can learn from far more client combinations than a single round provides. Those observations do not remain valid indefinitely, because the utility of a coalition drifts as the global model changes.

The reservoir therefore has a fixed capacity $|\mathcal{R}|_{\max}$. When new observations would overflow it, existing entries are evicted without replacement. Each entry is drawn for eviction with probability proportional to $\exp(\kappa \cdot \mathrm{age})$, where $\kappa > 0$ is a decay parameter and $\mathrm{age} = t - t'$ counts the rounds since the entry was stored. The draw itself breaks ties between entries of equal age. Old entries, whose utilities are the most likely to be stale, are thus the first to go, while recent evidence about the current game stays.

A.6 Hardware and Compute

All experiments ran on a single machine with an Intel Xeon Platinum 8480+ CPU (56 cores, 112 threads) and one NVIDIA H100 GPU (80 GB). We simulated federated training locally. Within each communication round, client updates ran one after another unless stated otherwise. Coalition evaluations reused the cached local updates of the candidate pool and ran in batches. The whole project, including preliminary runs, used a few GPU-weeks on this machine.

B Computational Complexity

This section derives the cost of each stage of NexusFL and confirms the analysis with wall-clock measurements. The analysis separates the cost of the surrogate from the cost of the coalition evaluations it learns from.

Notation. Let $|\theta|$ be the number of model parameters, $|\theta|_{\text{train}}$ the trainable ones, C_{fwd} the cost of one forward pass on one example, $\bar{s} = \mathbb{E}|S|$ the mean sampled-coalition size, and M the number of trees of depth d in the surrogate.

Cost of Each Stage. Coalition evaluation costs $O\big(B\,(\bar{s}\,|\theta|_{\text{train}} + |\mathcal{D}_{\text{val}}|\,C_{\text{fwd}})\big)$: for each of the B coalitions, this means averaging the cached updates of its members and then running one validation pass. This cost is linear in B and $|\mathcal{D}_{\text{val}}|$ and independent of n, because coalitions are formed inside the pool. Surrogate fitting costs $O(M\,d\,|\mathcal{R}|\,n)$ on n binary features. It depends on neither $|\theta|$ nor $|\mathcal{D}_{\text{val}}|$, and it stays roughly constant across rounds because $|\mathcal{R}|$ is capped. Final selection and pool proposal cost $O(k\,p\,M\,d)$ and $O(p\,n\,M\,d)$. Each step of the sequential softmax scores the remaining candidates with one pass through the tree-based surrogate model and does not use the global model.

Measured Cost. Table 5 gives the wall-clock breakdown over the last 100 rounds of the seed-1 runs with $|\mathcal{D}_{\text{val}}| = 1024$ and $B = 500$. Coalition evaluation takes 46% of the round for the CNN and 85% for DistilBERT. Coalition evaluation and client training both grow with model size, whereas surrogate fitting and both selection steps stay below 0.1 s on both models. Together, learning and querying the surrogate take about 11% of the CNN round and 0.3% of the DistilBERT round. The primary cost of the method is therefore its coalition evaluations. This is the same currency that the Shapley selectors spend, and Figure 6 shows what each method buys with it.

Table 5: Per-round wall-clock time by component (seconds; share of the full round in parentheses), mean over the last 100 rounds, seed 1. "Other" is test evaluation and bookkeeping.

Setting	$\lvert\theta\rvert$	Coalition eval.	Surrogate fit	Final selection	Pool proposal	Client training	Other	Full round
CNN / CIFAR-10	1.06M	0.34 (46%)	0.07 (10%)	0.004 (1%)	0.01 (2%)	0.29 (39%)	0.02 (2%)	0.74
DistilBERT / FinGPT	67.8M	25.61 (85%)	0.07 (0.24%)	0.004 (0.01%)	0.01 (0.04%)	4.11 (14%)	0.17 (0.71%)	30.0

C Proofs

The results of Section 3.2 fit together as follows. Proposition 2 turns a uniform surrogate error into a bound on the surrogate gap. Lemma 1 says what a depth-d tree ensemble can represent, which gives the representation term. Proposition 1 bounds the staleness term, and together with Lemma 1 it yields the three-way error decomposition (10). Theorem 1 then adds these per-round bounds over the whole run, and Theorem 2 turns the resulting regret into a convergence guarantee. Separately, Proposition 3 supports the pool-size argument of Section 3.1.

C.1 Proof of Proposition 2 (Surrogate-to-Selection Transfer)

Proposition 2 (Surrogate-to-selection transfer)**.** *Fix round t and pool P_t. If $\sup_{S\subseteq P_t,\,|S|=k} |\tilde{U}_t(S) - U_t(S)| \le \varepsilon_t$, then the surrogate gap satisfies $\zeta_t = U_t(S^\star_{t,\text{pool}}) - U_t(\tilde{S}_t) \le 2\varepsilon_t$.*

Proof. Let $\tilde{S}_t = \arg\max_{S\subseteq P_t,\,|S|=k} \hat{f}_t(S)$ be the surrogate's in-pool maximiser. Destandardisation is strictly increasing, and $\tilde{S}_t$ is therefore also the maximiser of the destandardised surrogate $\tilde{U}_t$. The uniform destandardised-surrogate error assumption gives, for every feasible $S \subseteq P_t$ with $|S| = k$,

$$U_t(S) \ge \tilde{U}_t(S) - \varepsilon_t \quad \text{and} \quad \tilde{U}_t(S) \ge U_t(S) - \varepsilon_t.$$

Applying the first inequality to $\tilde{S}_t$, using optimality of $\tilde{S}_t$ under $\tilde{U}_t$, and then applying the second inequality to $S^\star_{t,\text{pool}}$ yields

$$\begin{aligned} U_t(\tilde{S}_t) &\ge \tilde{U}_t(\tilde{S}_t) - \varepsilon_t \\ &\ge \tilde{U}_t(S^\star_{t,\text{pool}}) - \varepsilon_t \\ &\ge U_t(S^\star_{t,\text{pool}}) - 2\varepsilon_t. \end{aligned}$$

Rearranging gives $\zeta_t = U_t(S^\star_{t,\text{pool}}) - U_t(\tilde{S}_t) \le 2\varepsilon_t$, which bounds the surrogate-gap term in (8).

C.2 Proof of Lemma 1 (Interaction Capacity)

Lemma 1 (Interaction capacity of depth-bounded tree ensembles)**.** *Let $\mathcal{F}_d$ be the set of functions on $\{0,1\}^n$ representable as a finite sum of regression trees of depth at most d. Then every $f \in \mathcal{F}_d$ has the multilinear representation*

$$f(S) \;=\; \sum_{\substack{A \subseteq S \\ |A| \le d}} m(A),$$

and every multilinear function of degree at most d has such a tree-ensemble representation. In particular, $m(A) = 0$ for $|A| > d$.

Corollary 1. *Depth-one coalition-utility learning is equivalent to scalar client valuation. It represents U_t exactly only when the utility is additive; otherwise it omits every interaction term of order two or higher.*

Proof sketch. Both inclusions are standard. Write $x_i = \mathbf{1}[i \in S]$. A leaf of a depth-$d$ tree is reached when a conjunction of at most d literals in the x_i holds, and its indicator is a polynomial of degree at most d, multilinear because $x_i^2 = x_i$ on $\{0,1\}$. A finite sum of such trees is therefore a multilinear polynomial of degree at most d; this is the degree bound underlying the Fourier analysis of decision trees (Kushilevitz & Mansour, 1991). Conversely, each monomial $\prod_{i \in A} x_i$ with $|A| \le d$ is a single depth-$|A|$ path with leaf value 1, and a degree-d multilinear polynomial is a finite linear combination of such monomials. Finally, every set function has a unique Möbius representation $f(S) = \sum_{A \subseteq S} m(A)$ (Grabisch & Roubens, 1999), which coincides with the unique multilinear form on $\{0,1\}^n$; hence $m(A) = 0$ for $|A| > d$.

C.3 Proof of Proposition 1 and of the Decomposition (10)

Proof of Proposition 1. The weights $\alpha_{t,t'} \propto n_{t'} \gamma^{\,t-t'}$ are non-negative, vanish for rounds with no entries in $\mathcal{R}$, and sum to one over $t' \le t$. For any S,

$$\begin{aligned} |\bar{U}_t(S) - U_t(S)| &= \Big| \sum_{t' \le t} \alpha_{t,t'} \big(U_{t'}(S) - U_t(S) \big) \Big| \le \sum_{t' \le t} \alpha_{t,t'} \, |U_{t'}(S) - U_t(S)| \\ &\le L \sum_{t' \le t} \alpha_{t,t'} \, (t - t') = L \, A_t(\gamma), \end{aligned}$$

using Assumption 1 in the third step. Taking the supremum over S proves the claim.

Derivation of (10). Let $\| \cdot \|_{P_t}$ denote the supremum over $S \subseteq P_t$ with $|S| = k$, and let

$$f_t^\circ \in \arg\min_{f \in \mathcal{F}_d} \|f - \bar{U}_t\|_{P_t}$$

be the best in-class approximation of the target. Define $\varepsilon_t^{\text{est}} := \|\tilde{U}_t - f_t^\circ\|_{P_t}$, the error of the weighted least-squares fit relative to f_t°. Because the fit is trained on targets standardised by each round's own $(\mu_{t'}, \sigma_{t'})$, this term includes any mismatch between the standardised mixture and $\bar{U}_t$ in addition to finite-sample error. By the triangle inequality,

$$\varepsilon_t \;=\; \|\tilde{U}_t - U_t\|_{P_t} \;\le\; \|\tilde{U}_t - f_t^\circ\|_{P_t} + \|f_t^\circ - \bar{U}_t\|_{P_t} + \|\bar{U}_t - U_t\|_{P_t}.$$

The first term is $\varepsilon_t^{\text{est}}$ by definition and the third is at most $L\, A_t(\gamma)$ by Proposition 1. For the second, the Möbius transform is linear. The order-d truncation of the mixture is therefore the mixture of the truncations, $\bar{U}_t^{(\le d)} = \sum_{t'} \alpha_{t,t'} U_{t'}^{(\le d)}$, and $\bar{U}_t^{(\le d)} \in \mathcal{F}_d$ by Lemma 1. Hence

$$\|f_t^\circ - \bar{U}_t\|_{P_t} \;\le\; \|\bar{U}_t^{(\le d)} - \bar{U}_t\|_{P_t} \;\le\; \sum_{t'} \alpha_{t,t'} \, \|U_{t'}^{(\le d)} - U_{t'}\|_{P_t} \;\le\; \sum_{t'} \alpha_{t,t'} \, \rho_{t',d} \;\le\; \bar{\rho}_{t,d},$$

where the third inequality uses that the supremum over $S \subseteq P_t$ with $|S| = k$ is at most the supremum over all $S \subseteq N$ with $|S| = k$ in (9). Collecting the three bounds gives (10).

C.4 Pair Co-occurrence under Uniform Pool Sampling

Proposition 3 (Pair co-occurrence rate). *If pools of size p are drawn uniformly at random from N, then for any client pair $\{i, j\} \subseteq N$ and horizon $T \geq 1$, the expected number of rounds in which both i and j appear in the same pool is* $\mathbb{E}[\,\#\{t \leq T : \{i, j\} \subseteq P_t\}\,] = T \cdot p(p-1)/[n(n-1)]$.

Proof. Fix a pair $\{i, j\} \subseteq N$. In one round, the pool P_t is a size-p subset of N. The number of pools that contain both i and j is $\binom{n-2}{p-2}$, while the total number of possible pools is $\binom{n}{p}$. Hence

$$\Pr[\{i, j\} \subseteq P_t] = \frac{\binom{n-2}{p-2}}{\binom{n}{p}} = \frac{p(p-1)}{n(n-1)}.$$

Let $X_t = \mathbf{1}\{\{i, j\} \subseteq P_t\}$. Then

$$\mathbb{E}\left[\sum_{t=1}^{T} X_t\right] = \sum_{t=1}^{T} \mathbb{E}[X_t] = T \cdot \frac{p(p-1)}{n(n-1)},$$

by linearity of expectation.

For fixed n, doubling p roughly quadruples the number of pairs contained in a size-p pool.

C.5 Proof of Theorem 1 (Cumulative Selection Regret)

Proof. By (8), the per-round regret decomposes as $U_t(S_t^\star) - U_t(\hat{S}_t) = \delta_t + \zeta_t + \xi_t$. Proposition 2 gives $\zeta_t \leq 2\varepsilon_t$, and (10) gives $\varepsilon_t \leq \bar{\rho}_{t,d} + \varepsilon_t^{\text{est}} + L\, A_t(\gamma)$ under Assumption 1. Summing over rounds,

$$\sum_{t=1}^{T} \big[U_t(S_t^\star) - U_t(\hat{S}_t)\big] \leq \sum_{t=1}^{T} \delta_t + 2 \sum_{t=1}^{T} \big(\bar{\rho}_{t,d} + \varepsilon_t^{\text{est}} + L\, A_t(\gamma)\big) + \sum_{t=1}^{T} \xi_t.$$

C.6 End-to-End Convergence

Setting. Let V be a smooth server-side validation loss and F the federated training objective. Let $\mathfrak{F}_t$ denote the history up to and including w_t, before the round-t local training, and let $G_t(S) = \sum_{i \in S} \alpha_i^{(t)} g_i^{(t)}$ be the aggregated update of coalition S formed from the cached updates, as in (1). We define $w_{t+1}(S) = w_t - \eta_t G_t(S)$ and take the realised loss reduction $U_t(S) = V(w_t) - V(w_{t+1}(S))$ as the utility. The analysis allows a round-dependent server step η_t; the constant step η of (1) and Algorithm 1 is the special case $\eta_t \equiv \eta$. Algorithm 1 evaluates a generic U_t of $w_{t+1}(S)$. The descent argument below uses this loss-reduction special case. Given the cached updates, that utility is a deterministic set function. The selected coalition $\hat{S}_t$ may depend on these updates, and the argument below therefore takes expectations only after comparing coalitions pathwise.

Assumptions. (A1) V is L_V-smooth. (A2) $V \geq V_{\inf} > -\infty$. (A3) A reference size-$k$ coalition S_t^{ref}, chosen before the round-t local training and hence $\mathfrak{F}_t$-measurable, satisfies

$$\mathbb{E}[\langle \nabla V(w_t), G_t(S_t^{\text{ref}}) \rangle \mid \mathfrak{F}_t] \geq a \|\nabla V(w_t)\|^2, \qquad \mathbb{E}[\|G_t(S_t^{\text{ref}})\|^2 \mid \mathfrak{F}_t] \leq b \|\nabla V(w_t)\|^2 + \sigma_{\text{ref}}^2.$$

(A4) The in-pool surrogate error is bounded by ε_t as in Proposition 2. (A5) The validation and training gradients satisfy $\|\nabla F(w) - \nabla V(w)\| \leq \Delta_{\text{val}}$ at every iterate. Conditions of the form (A3) are standard for FedAvg-type updates under nonconvex partial participation when stated for the training objective F (Karimireddy et al., 2020; Yang et al., 2021). We impose the condition on the validation loss V, which additionally requires the client updates to be aligned with the validation gradient. Both (A3) and (A5) therefore encode the representativeness of the validation set.

Theorem 2 (End-to-end convergence). *Under (A1)–(A5), with* $H_T = \sum_{t=1}^{T} \eta_t$,

$$\frac{1}{H_T} \sum_{t=1}^{T} \eta_t\, \mathbb{E}\|\nabla F(w_t)\|^2 \;\leq\; O\Big(\frac{1}{H_T} + \frac{\sum_t \eta_t^2}{H_T} + \frac{1}{H_T} \sum_{t=1}^{T} \mathbb{E}[\delta_t + 2\varepsilon_t + \xi_t]\Big) + 2\Delta_{\text{val}}^2.$$

Precise Statement. Under (A1)–(A5), if $\eta_t \le a/(L_V b)$ and $H_T = \sum_{t=1}^{T} \eta_t$, the NEXUSFL iterates satisfy

$$\frac{1}{H_T}\sum_{t=1}^{T} \eta_t \,\mathbb{E}\|\nabla V(w_t)\|^2 \;\le\; \frac{2(V(w_1)-V_{\inf})}{aH_T} + \frac{L_V \sigma_{\mathrm{ref}}^2}{aH_T}\sum_{t=1}^{T}\eta_t^2 + \frac{2}{aH_T}\sum_{t=1}^{T}\mathbb{E}[\delta_t + 2\varepsilon_t + \xi_t],$$

and the same average of $\mathbb{E}\|\nabla F(w_t)\|^2$ is bounded by twice the right-hand side plus $2\Delta_{\mathrm{val}}^2$. This is the form of Theorem 2 with all constants explicit.

Proof. Reference descent. By L_V-smoothness (A1), for any feasible S,

$$V(w_t) - V\big(w_t - \eta_t G_t(S)\big) \;\ge\; \eta_t \,\langle \nabla V(w_t), G_t(S)\rangle - \frac{L_V \eta_t^2}{2}\,\|G_t(S)\|^2.$$

Applying this to the reference coalition S_t^{ref}, taking $\mathbb{E}[\cdot \mid \mathfrak{F}_t]$, which is legitimate because S_t^{ref} is $\mathfrak{F}_t$-measurable, and using the two bounds in (A3),

$$\mathbb{E}\big[U_t(S_t^{\mathrm{ref}}) \mid \mathfrak{F}_t\big] \;\ge\; \eta_t\Big(a - \frac{L_V b\,\eta_t}{2}\Big)\|\nabla V(w_t)\|^2 - \frac{L_V\eta_t^2}{2}\,\sigma_{\mathrm{ref}}^2.$$

Since $\eta_t \le a/(L_V b)$ implies $a - L_V b\,\eta_t/2 \ge a/2$,

$$\mathbb{E}\big[U_t(S_t^{\mathrm{ref}}) \mid \mathfrak{F}_t\big] \;\ge\; \frac{a\,\eta_t}{2}\,\|\nabla V(w_t)\|^2 - \frac{L_V\eta_t^2}{2}\,\sigma_{\mathrm{ref}}^2. \tag{13}$$

Optimal-coalition dominance. Given the cached updates, $S_t^\star$ maximises U_t over size-k coalitions, and therefore $U_t(S_t^\star) \ge U_t(S_t^{\mathrm{ref}})$ pathwise. Taking $\mathbb{E}[\cdot \mid \mathfrak{F}_t]$ and using (13),

$$\mathbb{E}\big[U_t(S_t^\star) \mid \mathfrak{F}_t\big] \;\ge\; \frac{a\,\eta_t}{2}\,\|\nabla V(w_t)\|^2 - \frac{L_V\eta_t^2}{2}\,\sigma_{\mathrm{ref}}^2. \tag{14}$$

Selection regret. By (8), $U_t(\hat S_t) = U_t(S_t^\star) - \delta_t - \zeta_t - \xi_t$ holds pathwise, and Proposition 2 with (A4) gives $\zeta_t \le 2\varepsilon_t$. Hence

$$U_t(\hat S_t) \;\ge\; U_t(S_t^\star) - \delta_t - 2\varepsilon_t - \xi_t. \tag{15}$$

Combining. Because $w_{t+1} = w_{t+1}(\hat S_t)$, the identity $U_t(\hat S_t) = V(w_t) - V(w_{t+1})$ holds pathwise, whatever the dependence of $\hat S_t$ on the cached updates. Taking $\mathbb{E}[\cdot \mid \mathfrak{F}_t]$ in (15), inserting (14), and taking total expectation:

$$\mathbb{E}[V(w_{t+1})] \;\le\; \mathbb{E}[V(w_t)] - \frac{a\,\eta_t}{2}\,\mathbb{E}\|\nabla V(w_t)\|^2 + \frac{L_V\eta_t^2}{2}\,\sigma_{\mathrm{ref}}^2 + \mathbb{E}[\delta_t + 2\varepsilon_t + \xi_t].$$

Summing over $t = 1, \dots, T$ telescopes the $\mathbb{E}[V(w_t)]$ terms; using $V(w_{T+1}) \ge V_{\inf}$ (A2) and rearranging,

$$\frac{a}{2}\sum_{t=1}^{T}\eta_t\,\mathbb{E}\|\nabla V(w_t)\|^2 \;\le\; V(w_1) - V_{\inf} + \frac{L_V\sigma_{\mathrm{ref}}^2}{2}\sum_{t=1}^{T}\eta_t^2 + \sum_{t=1}^{T}\mathbb{E}[\delta_t + 2\varepsilon_t + \xi_t],$$

and dividing by $aH_T/2$ gives the bound on the weighted average of $\mathbb{E}\|\nabla V(w_t)\|^2$.

Training objective. By (A5), $\|\nabla F(w_t)\|^2 \le 2\|\nabla V(w_t)\|^2 + 2\Delta_{\mathrm{val}}^2$. Multiplying by η_t, taking expectations, summing, and dividing by H_T,

$$\frac{1}{H_T}\sum_{t=1}^{T}\eta_t\,\mathbb{E}\|\nabla F(w_t)\|^2 \;\le\; \frac{2}{H_T}\sum_{t=1}^{T}\eta_t\,\mathbb{E}\|\nabla V(w_t)\|^2 + 2\Delta_{\mathrm{val}}^2,$$

and substituting the first bound gives twice its right-hand side plus $2\Delta_{\mathrm{val}}^2$.

Corollary 2 (Asymptotics). *If $H_T \to \infty$, $\sum_t \eta_t^2/H_T \to 0$, and $\frac{1}{H_T}\sum_t \mathbb{E}[\delta_t + 2\varepsilon_t + \xi_t] \to 0$, then $\frac{1}{H_T}\sum_t \eta_t\,\mathbb{E}\|\nabla V(w_t)\|^2 \to 0$. If in addition $\Delta_{\mathrm{val}} = 0$, the same holds for F; for fixed $\Delta_{\mathrm{val}} > 0$ the limit superior for F is at most $2\Delta_{\mathrm{val}}^2$.*

D EMPIRICAL REGRET DECOMPOSITION

Theorem 1 bounds selection regret by three terms, the pool gap, the surrogate gap, and the selector term, and each design decision of NEXUSFL targets one of them. On a large real FL task none of the three is observable, because the optimal coalition $S_t^\star$ is unknown. This appendix measures the decomposition exactly on a synthetic game with a large client population and known utility, and on a small CIFAR-10 population where every size-k coalition can be scored.

D.1 EXACT DECOMPOSITION ON A SYNTHETIC GAME

Exact measurement needs a known optimum on a large population. Enumeration can recover the optimum on a small problem, while a pool is most useful at large n, where interaction estimation is arguably harder.

We therefore use a Sum of Unanimity Models (SOUM) game (Deng & Papadimitriou, 1994; Fumagalli et al., 2023) with n=192 clients, whose utility is known in closed form:

$$U_t(S) \;=\; \sum_m c_m(t)\,\mathbf{1}[T_m \subseteq S].$$

Each T_m is a fixed client subset of size 1, 2, or 3, and a coalition S earns the bonus $c_m(t)$ only if it contains every client in T_m. Singletons, pairs, and triples therefore encode first-, second-, and third-order effects of known strength. To make the game non-stationary in the sense of Assumption 1, the coefficients drift as $c_m(t) = c_m\,(1 + \eta\, w_m \sin(2\pi t/T_{\text{per}} + \phi_m))$, with amplitude η=0.4, period T_{per}=40, and random per-term scales w_m and phases ϕ_m.

We select k=5 clients, evaluate B=128 coalitions per round for 50 rounds, and average over three seeds. The main setting uses p=10, which keeps the ratio p/k=2 used in the rest of the paper; we then increase only p, to 20 and 40. Because U_t is known, we can compute $S_t^\star$ and the best in-pool coalition $S_{t,\text{pool}}^\star$ exactly, and decompose regret as in (8) into the pool gap δ_t, the surrogate gap ζ_t, and the signed selector term ξ_t (Table 6). The selector term can be negative, so we report raw values rather than shares of total regret.

Table 6: Exact regret decomposition on the SOUM game (n=192, k=5, B=128; three-seed average). Negative ξ_t means the sequential softmax returned a coalition with higher true utility than the surrogate maximiser.

Pool size p	**Regret** $\downarrow$	**Pool gap** δ_t	**Surrogate gap**	**Selector term** ξ_t
10	2.26	2.17	0.04	0.05
20	1.11	0.91	0.23	−0.03
40	1.09	0.48	0.71	−0.10

Three things follow. First, at the paper's ratio p/k=2, almost all regret comes from the pool. The pool gap accounts for 2.17 of the total 2.26, and the surrogate gap is negligible at 0.04. Once the right clients are in the pool, the surrogate finds the right coalition among them; the binding constraint is which clients get in.

Second, enlarging the pool moves regret from one term to the other. From p=10 to p=20, the pool gap falls by 58% and total regret by 51%. From p=20 to p=40, the pool gap keeps falling, but the number of size-5 coalitions inside the pool grows from 15,504 to 658,008, while the surrogate still sees only B=128 observations per round. The surrogate gap rises to 0.71 and becomes the larger term, and total regret barely moves. This is the coverage-versus-estimation trade-off that Theorem 1 predicts at a fixed coalition budget, and Section F.3 shows the same saturation on CIFAR-10.

Third, the selector term is small for every p and negative for $p \geq 20$. The stochastic softmax draw (Section 3.2) costs nothing in regret, and it sometimes finds a better coalition than the surrogate's own maximiser.

Table 7 compares selection rules on the same game. NEXUSFL incurs less than a third of the regret of the two Shapley-value selectors, which on this game do worse than uniform random selection. The reason is structural: averaging a client's marginal contribution over random contexts discards exactly the pair and triple structure that determines U_t. In the oracle-pool row, the pool always contains $S_t^\star$,

and selection within the pool is nearly exact. The remaining regret of NEXUSFL is therefore the pool gap.

Table 7: Regret of selection methods on the SOUM game (p=10, k=5, B=128; mean $\pm$ std over 3 seeds).

Method	Selection protocol	Regret ↓
NEXUSFL	surrogate-proposed pool of 10; sequential softmax selects 5	2.261 ± 0.861
S-FedAvg	Shapley-value-based selection of 5	8.107 ± 0.697
AFedSV+	Shapley-value-based inclusion probabilities for 5	7.881 ± 0.958
Random	uniform selection of 5	6.449 ± 0.918
Oracle pool	pool contains $S_t^\star$; NEXUSFL selects within it	0.016 ± 0.025

D.2 POOL GAP ON A REAL DATASET

Enumeration can recover $S_t^\star$ on a small population. We therefore run CIFAR-10 $\mathrm{D}(0.1)$ with n=10 clients and k=4, so that all $\binom{10}{4}$=210 size-4 coalitions can be scored from cached updates. Every client is trained each round. That full ranking is used only to form the table. The surrogate is fit only on B=32 evaluations drawn inside the pool.

The comparison is a 40-round mean, with tree depth 4, reservoir capacity 2500. The first-round pool is uniform. Later pools are proposed by the previous surrogate at temperature τ_{pool}=0.05, and the size-k set is drawn by sequential softmax at τ_{sel}=0.01. Aggregation weights clients by training-set size. Local training is one epoch with batch size 32, learning rate 0.01, and momentum 0.5. Utility is the change in accuracy of the size-4 aggregate on a fixed 1024-example validation subset, drawn from a class-balanced 2000-example hold-out of the CIFAR-10 training set.

Regret is $U_t(S_t^\star) - U_t(\hat{S}_t)$ and splits as $\delta_t + \zeta_t + \xi_t$, as in (8). The uniform error ε_t is the largest absolute error of the surrogate on the size-4 coalitions inside the pool. The number of those coalitions is $\binom{p}{4}$: 70, 15, 5, and 1 at $p \in \{8, 6, 5, 4\}$. On every round of every pool size, the surrogate gap satisfies $\zeta_t \le 2\varepsilon_t$, which is Proposition 2.

Table 8: Exact regret decomposition on CIFAR-10 $\mathrm{D}(0.1)$ with n=10, k=4, and B=32 (40-round mean, one seed). Gaps are differences in validation accuracy.

Pool size p	Regret ↓	Pool gap δ_t	Surrogate gap ζ_t	Selector gap ξ_t	ε_t
8	0.042	0.008	0.014	0.020	0.086
6	0.027	0.013	0.003	0.010	0.048
5	0.025	0.022	0.001	0.002	0.012
4	0.047	0.047	0	0	0.003

At p=4 the pool is the only feasible coalition, so there is nothing for the surrogate or the selector to choose. Both gaps are zero and the regret equals the pool gap. The remaining ε_t=0.003 is the error in the predicted value of that one coalition. With a single feasible set it does not move the selection.

Enlarging the pool reduces the pool gap, from 0.047 at p=4 to 0.008 at p=8, and raises the uniform error, from 0.003 to 0.086, as the in-pool space grows from 1 to 70 coalitions under a fixed budget B=32. Total regret is smallest at p=5 (0.025) and p=6 (0.027). At p=8 the pool gap is the smallest of the four, but the surrogate and selector gaps together exceed it, and regret rises to 0.042. This is the same coverage-versus-estimation trade-off as on the SOUM game.

E INTERACTION EXTRACTION AND WORKED EXAMPLE

This section shows how the interaction coefficients of Section 3.3 are read off the trained surrogate, and then works through a two-client example by hand.

Write $\hat{f}_t(S)$ for $\hat{f}_t(\mathbf{1}[S])$. The surrogate is a tree ensemble whose input is the binary indicator of a coalition $S \subseteq N$, and this structure is what lets us read off interaction coefficients. Each split on a client contributes that client's effect on predicted utility. When that effect depends on another client tested deeper along the same path, the two clients interact: the value of adding one depends on

whether the other is already present. Splits nested more deeply produce higher-order interactions, up to the tree depth d. Collecting these contributions across the ensemble gives, for each client group G, a single Faithful Banzhaf coefficient $I^{\mathrm{FB}}_{t,d}(G)$ that summarises the joint effect of the clients in G.

Formally, $\hat{f}_t$ admits the Walsh–Fourier expansion

$$\hat{f}_t(S) \;=\; \sum_{J \subseteq N} \psi_t(J)\,(-1)^{|J \cap S|} \tag{16}$$

(Kushilevitz & Mansour, 1991), whose coefficients are extracted from the trees (Gorji et al., 2026) and converted in closed form (Butler et al., 2026) to the order-d Faithful Banzhaf coefficients (Tsai et al., 2023),

$$I^{\mathrm{FB}}_{t,d}(G) \;=\; (-2)^{|G|} \sum_{\substack{J \supseteq G \\ |J| \le d}} \psi_t(J). \tag{17}$$

These yield the exact representation (11). The predicted utility of S is the sum of an intercept, individual client effects, and interaction effects among the clients in S. Intermediate steps are given in the cited works.

For a partial selection S and client $i \notin S$, adding i activates exactly the terms that contain i, namely its individual effect and its interactions with clients already in S, which is (12). NEXUSFL scores each remaining candidate by $\hat{f}_t(S \cup \{i\})$. Since $\hat{f}_t(S)$ is common to all clients considered at the same step, sampling by $\hat{f}_t(S \cup \{i\})$ in (7) is exactly equivalent to using $\Delta_t(i \mid S)$. Tree traversal and summing the activated interaction coefficients are two exact descriptions of the same inference step. The extracted interactions therefore explain the scores that determine the policy's selection probabilities rather than serving as a post-hoc interpretation.

Worked Example. Consider two clients and a depth-two surrogate with

$$\hat{f}_t(\emptyset) = 0, \qquad \hat{f}_t(\{1\}) = 1, \qquad \hat{f}_t(\{2\}) = 0, \qquad \hat{f}_t(\{1,2\}) = 3.$$

The Walsh–Fourier coefficients $\psi_t(J)$ and the order-two Faithful Banzhaf coefficients $I^{\mathrm{FB}}_{t,2}(G) = (-2)^{|G|} \sum_{J \supseteq G} \psi_t(J)$ are

G	$\psi_t(G)$	$I^{\mathrm{FB}}_{t,2}(G)$
$\emptyset$	1	0
$\{1\}$	-1	1
$\{2\}$	$-\frac{1}{2}$	0
$\{1,2\}$	$\frac{1}{2}$	2

One checks that $\hat{f}_t(S) = \sum_J \psi_t(J)(-1)^{|J \cap S|}$ reproduces the four values above. The exact representation gives, for the joint coalition,

$$\hat{f}_t(\{1,2\}) = I^{\mathrm{FB}}_{t,2}(\emptyset) + I^{\mathrm{FB}}_{t,2}(\{1\}) + I^{\mathrm{FB}}_{t,2}(\{2\}) + I^{\mathrm{FB}}_{t,2}(\{1,2\}) = 0 + 1 + 0 + 2 = 3.$$

At the first selection step the selector compares $\Delta_t(1 \mid \emptyset) = I^{\mathrm{FB}}_{t,2}(\{1\}) = 1$ with $\Delta_t(2 \mid \emptyset) = I^{\mathrm{FB}}_{t,2}(\{2\}) = 0$ and prefers client 1. With client 1 selected, adding client 2 gives

$$\Delta_t(2 \mid \{1\}) = I^{\mathrm{FB}}_{t,2}(\{2\}) + I^{\mathrm{FB}}_{t,2}(\{1,2\}) = 0 + 2 = 2,$$

so that $\hat{f}_t(\{1,2\}) = \hat{f}_t(\{1\}) + \Delta_t(2 \mid \{1\}) = 1 + 2 = 3$ matches the tree. The positive pair coefficient $I^{\mathrm{FB}}_{t,2}(\{1,2\}) = 2$ records complementarity: the joint value exceeds the sum of the individual effects. Had the surrogate instead learned $\hat{f}_t(\{1,2\}) = 1$, the pair coefficient would be 0 and client 2 would add nothing given client 1; a value below 1 would give a negative coefficient and mark the two clients as redundant.

Table 9: Additional benchmark settings. Final and trajectory-mean test accuracy (%) with standard deviations as subscripts over 5 seeds. Best per metric in **bold**.

	CIFAR-10 LT		FinGPT		Yelp D(0.5)		PathMNIST D(0.1)		CIFAR-10 D(0.5)		Yelp D(0.1)	
Method	Final	Mean	Final	Mean	Final	Mean	Final	Mean	Final	Mean	Final	Mean
FedAvg	$41.4_{\pm 1.1}$	$30.4_{\pm 0.7}$	$72.3_{\pm 16.2}$	$74.4_{\pm 2.7}$	$53.2_{\pm 1.8}$	$49.6_{\pm 0.4}$	$64.9_{\pm 3.8}$	$53.3_{\pm 1.7}$	$59.4_{\pm 1.3}$	$48.3_{\pm 0.2}$	$48.1_{\pm 1.9}$	$42.5_{\pm 0.2}$
S-FedAvg	$41.5_{\pm 1.3}$	$30.4_{\pm 0.7}$	$75.6_{\pm 21.9}$	$75.5_{\pm 1.5}$	$55.8_{\pm 0.8}$	$50.0_{\pm 0.3}$	$63.5_{\pm 8.3}$	$53.6_{\pm 1.4}$	$59.7_{\pm 1.9}$	$48.3_{\pm 0.1}$	$49.3_{\pm 4.7}$	$42.3_{\pm 0.5}$
AFedSV+	$42.5_{\pm 1.1}$	$31.7_{\pm 0.7}$	$86.9_{\pm 0.3}$	$82.2_{\pm 1.0}$	$55.3_{\pm 1.4}$	$50.4_{\pm 0.2}$	$68.6_{\pm 2.1}$	$59.4_{\pm 1.5}$	$59.1_{\pm 1.2}$	$49.1_{\pm 0.4}$	$48.9_{\pm 3.2}$	$42.8_{\pm 0.7}$
FedCor	$41.0_{\pm 1.2}$	$30.4_{\pm 0.6}$	$\mathbf{89.6}_{\pm 5.0}$	$85.3_{\pm 2.6}$	$56.1_{\pm 2.6}$	$51.3_{\pm 0.3}$	$69.4_{\pm 2.3}$	$57.8_{\pm 1.5}$	$60.0_{\pm 0.6}$	$49.3_{\pm 0.2}$	$51.0_{\pm 3.8}$	$44.7_{\pm 0.6}$
DivFL	$41.5_{\pm 0.5}$	$30.4_{\pm 0.7}$	$88.0_{\pm 5.0}$	$74.7_{\pm 0.8}$	$55.7_{\pm 0.6}$	$51.1_{\pm 0.2}$	$66.9_{\pm 3.6}$	$56.3_{\pm 1.8}$	$\mathbf{60.3}_{\pm 0.2}$	$48.5_{\pm 0.2}$	$50.6_{\pm 2.7}$	$46.7_{\pm 1.1}$
FilFL	$42.7_{\pm 1.0}$	$31.9_{\pm 0.7}$	$82.5_{\pm 14.7}$	$80.8_{\pm 6.5}$	$57.1_{\pm 0.9}$	$52.0_{\pm 0.3}$	$65.8_{\pm 4.5}$	$55.5_{\pm 1.9}$	$59.4_{\pm 0.5}$	$48.2_{\pm 0.5}$	$49.5_{\pm 4.2}$	$46.3_{\pm 0.7}$
NexusFL	$\mathbf{43.4}_{\pm 0.9}$	$\mathbf{32.5}_{\pm 0.6}$	$87.9_{\pm 3.9}$	$\mathbf{87.4}_{\pm 2.4}$	$\mathbf{57.2}_{\pm 0.5}$	$\mathbf{53.6}_{\pm 0.05}$	$\mathbf{71.6}_{\pm 2.6}$	$\mathbf{60.6}_{\pm 2.2}$	$60.1_{\pm 1.5}$	$\mathbf{50.5}_{\pm 0.7}$	$\mathbf{52.1}_{\pm 1.8}$	$\mathbf{48.4}_{\pm 0.5}$

F Additional Results and Ablations

This appendix reports further empirical results and ablates the hyperparameters of NexusFL. We first extend the benchmark to six further settings (Section F.1) and to a population of 1000 clients (Section F.2). We then examine the two parameters that the theory ties to regret, pool size and tree depth (Section F.3), and ask where the gains come from: from pool proposal or from final selection (Section F.4). Next, we measure how well the surrogate transfers across a game that keeps moving (Section F.5), stress the server validation set (Section F.6), and check that the learned interactions mean something on real data (Section F.7). The last four subsections vary the reservoir size, the value function, the selection rule, and the surrogate class (Sections F.8 to F.11).

Unless stated otherwise, CIFAR-10 experiments use n=100 clients, p=20, k=10, B=500, and the hyperparameters of Section A.4. "Mean" denotes trajectory-mean test accuracy over all rounds, and "Final" the test accuracy at the last round.

F.1 Additional Benchmark Settings

Table 9 and Figure 8 extend the benchmark of Section 4.1 to six further settings. NexusFL has the highest trajectory mean in all six. Its gains are again clearest in the more heterogeneous settings. On FinGPT and Yelp D(0.1), it reaches trajectory means of 87.4% and 48.4%, which is 2.1 and 1.7 points above the strongest baseline. On the milder CIFAR-10 D(0.5) partition, the mean margin is 1.2 points and stays positive. On final accuracy, NexusFL is best on CIFAR-10 LT, Yelp D(0.5), PathMNIST D(0.1), and Yelp D(0.1), second on CIFAR-10 D(0.5), and third on FinGPT.

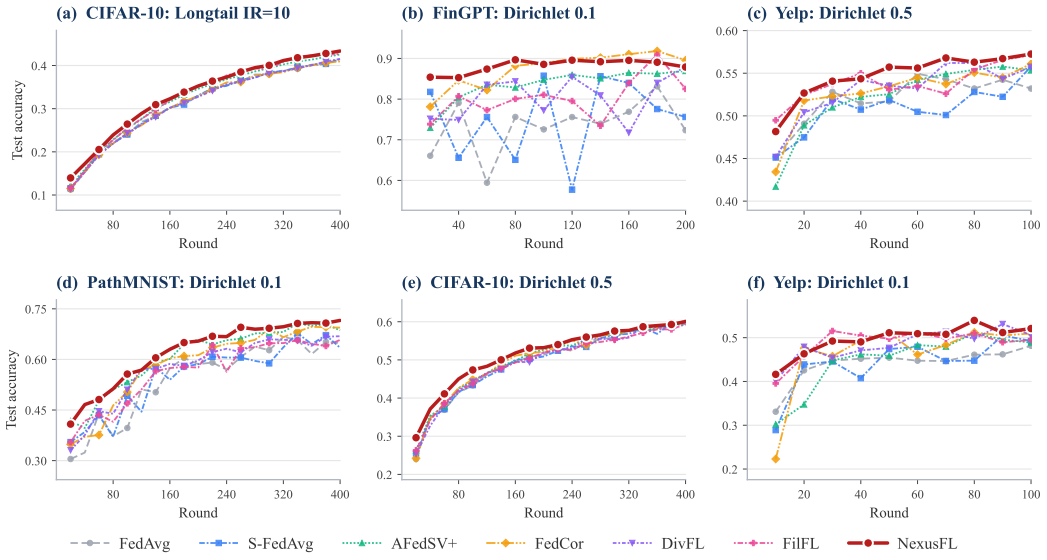


Figure 8: Additional learning curves. Panels show CIFAR-10 long-tail, FinGPT under D(0.1), Yelp under D(0.5), PathMNIST under D(0.1), CIFAR-10 under D(0.5), and Yelp under D(0.1).

F.2 Client-Population Scaling

The main experiments use 100 clients. To test whether NexusFL remains effective at a larger scale, we increase the CIFAR-10 client population to n=1000 under Dirichlet label skew with α=1.0. The larger population comes with a larger candidate pool (p=50), coalition budget (B=2500), and replay reservoir ($|\mathcal{R}|_{\max}$=25000). Results are averaged over five random seeds. Figure 9 shows that NexusFL leads throughout training and finishes just above 50%.

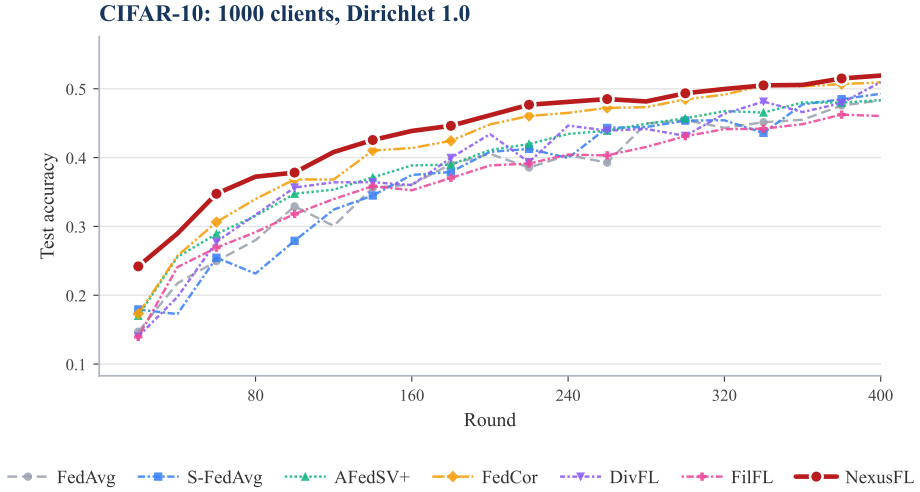


Figure 9: Client-population scaling on CIFAR-10 D(1.0) with n=1000 clients, p=50, and B=2500. NexusFL leads throughout and finishes just above 50%, with FedCor close behind.

F.3 Pool Size and Tree Depth

The pool size p and the tree depth d are the two parameters that Theorem 1 ties to specific regret terms: p to the pool gap δ_t and d to the truncation error $\bar{\rho}_{t,d}$. We vary each on CIFAR-10 with everything else fixed.

Pool Size. We vary $p \in \{10, 20, 50, 100\}$ on CIFAR-10 D(0.1) with B, the reservoir capacity, and k=10 held fixed (Table 10). At p=10 the pool coincides with the aggregation set, so there is no in-pool selection and the surrogate acts as a direct client ranker; this is the worst setting by a wide margin, 11 pp of Final accuracy below p=20. Beyond p=20 the returns diminish, and p=100 is below p=50. With B fixed, a larger pool means the reservoir covers a thinner fraction of the in-pool coalition space, so the surrogate's estimate degrades while training and communication cost grow linearly in p. The pool must be larger than the selection set, and moderately so.

Table 10: Pool-size ablation on CIFAR-10 D(0.1) with B, reservoir capacity, and k=10 fixed.

Pool size p	Mean / Final	Seconds per round
10	42.9 / 43.2	0.85
20	46.5 / 54.3	1.45
50	47.6 / 58.5	2.17
100	46.1 / 55.6	2.88

Tree Depth. By Lemma 1, d is the highest interaction order the surrogate can express, and d=1 is an additive model, that is, a per-client score. We sweep $d \in \{1, 2, 4, 8\}$ on CIFAR-10 with Dirichlet $\alpha \in \{0.1, 0.5, 1.0\}$ over three seeds, keeping B, the reservoir capacity, p, and k fixed (Table 11). The additive surrogate is the worst in all three regimes, which is the empirical counterpart of Corollary 1: its approximation error is the interaction mass that Figure 2 shows to be present at every α. Going from d=2 to d=4 helps at every α and most at α=0.1, where the spectrum places the most mass beyond pairwise terms. Depth d=8 is the best under the mildest skew (α=1.0) and falls below d=4 at α=0.1, where coalitions vary more and interactions among specific groups of eight clients

are supported by too few observations to estimate. Tree depth d still limits the highest interaction order the surrogate can represent, but because fitting is inexpensive relative to coalition evaluation (Sections B and 4.3), d can be selected by standard hyperparameter tuning.

Table 11: Tree-depth ablation on CIFAR-10 (Final accuracy, mean $\pm$ std over 3 seeds). Bold marks the best depth per column.

Depth d	$\boldsymbol{\alpha}$**=0.1**	$\boldsymbol{\alpha}$**=0.5**	$\boldsymbol{\alpha}$**=1.0**
1	54.6 ± 0.4	57.1 ± 0.8	58.3 ± 0.4
2	54.9 ± 0.9	59.3 ± 0.2	60.2 ± 0.4
4	$\mathbf{56.6} \pm 1.0$	$\mathbf{59.8} \pm 0.9$	60.9 ± 0.3
8	55.7 ± 0.8	59.1 ± 1.3	$\mathbf{61.1} \pm 0.4$

F.4 Isolating Pool Proposal and Final Selection

The surrogate acts twice per round: it proposes the p clients that train, and it selects the k of them that are aggregated. Theorem 1 assigns each stage its own regret term, the pool gap δ_t to proposal and the surrogate and selector terms to selection, so the two contributions can be separated empirically. We compare four settings that all train p=20 clients per round: (A) surrogate-proposed pool, aggregate all 20; (B) surrogate-proposed pool, surrogate-select and aggregate 10, which is NexusFL; (C) surrogate-proposed pool, randomly select and aggregate 10; (D) uniformly random pool, aggregate all 20, which is FedAvg with the same training budget. Table 12 reports three-seed results on CIFAR-10.

Table 12: Pool proposal versus final selection on CIFAR-10 (Mean / Final, mean $\pm$ standard error over 3 seeds). Bold marks the best Mean and Final per column.

	Setting	$\mathbf{D}(\mathbf{0.1})$	**Long-tail**
A	surrogate pool / aggregate all 20	47.40±0.31 / **56.00**±1.81	31.26±0.40 / 41.87±0.69
B	surrogate pool / surrogate-select 10 (NexusFL)	**47.98**±0.82 / 55.05±0.41	**32.28**±0.42 / **43.20**±0.67
C	surrogate pool / random 10	43.35±0.33 / 54.88±1.41	31.13±0.38 / 42.49±0.66
D	random pool / aggregate all 20 (FedAvg-20)	45.17±0.65 / 53.12±0.63	30.21±0.41 / 41.41±0.75

Both stages contribute, and their relative weight depends on the heterogeneity. Under Dirichlet skew, final selection is the larger effect: replacing random in-pool selection by the surrogate (B−C) adds 4.63 pp Mean, whereas replacing the random pool by the surrogate-proposed pool (A−D) adds 2.23 pp. Under long-tail imbalance the two are comparable, 1.15 and 1.05 pp. Selecting 10 clients from the surrogate's own pool also beats aggregating all 20 of them (B−A, +0.58 / +1.02 pp Mean), so the gain does not come from averaging more updates; it comes from choosing which updates to average. Setting D is the budget-matched FedAvg reference and trails NexusFL on both metrics in both settings.

Setting C also allows a within-round view of the same question. Each round we score the current utility $U_t(S)$ of three coalitions from the same surrogate-proposed pool: the surrogate's preferred 10, the random 10 that was aggregated, and all 20 (Table 13). The surrogate-selected coalition has the highest utility in both settings. On the long-tail partition its utility is about eight times that of the full pool.

Table 13: Per-round utility $U_t(S)$ of three coalitions from the same surrogate-proposed pool (setting C), averaged over 400 rounds and 3 seeds.

Aggregated coalition	$\mathbf{D}(\mathbf{0.1})$	**Long-tail**
Surrogate-selected 10	0.0536	0.0132
Random 10	0.0014	0.0010
All 20	0.0409	0.0017

Figure 10 shows the pool-proposal effect over training on SVHN. Both variants use the surrogate selector inside the pool and differ only in how the next pool is chosen. The surrogate-proposed pool pulls ahead through the middle of training, and the two curves finish close.

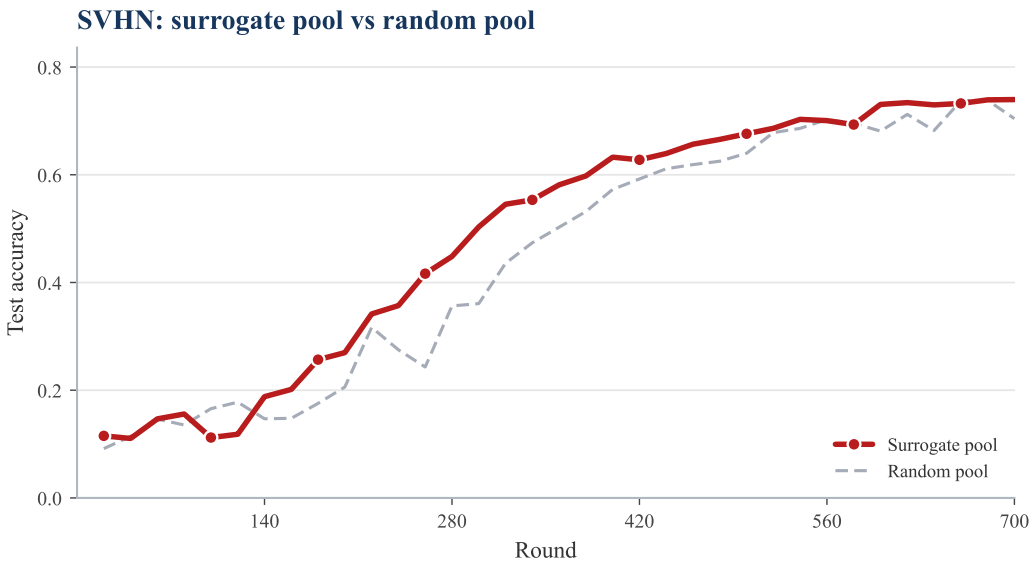


Figure 10: Surrogate-proposed versus uniform random pools on SVHN $\mathrm{D}(0.1)$. Both variants use the surrogate selector inside the pool.

F.5 Surrogate Generalisation and Non-Stationarity

Pool proposal scores clients across all of N with the surrogate of the previous round, and the in-round R^2 of Section 4.2 does not measure how well that surrogate transfers to the next game. This subsection measures that transfer and then how the learned interactions and the selected coalitions change over training.

Transfer to the Next Round. In each round, we score the previous round's surrogate on freshly sampled coalitions of the current round. Per-round standardisation changes with the global model, so we report two scale-invariant measures, Spearman and Pearson correlation. Both stay positive in every setting (Table 14), so the previous surrogate is a usable signal for pool proposal.

Table 14: Surrogate generalisation to the next round. The previous round's surrogate is scored on fresh current-round coalitions.

	CIFAR D0.1	CIFAR D0.5	CIFAR LT	SVHN	FinGPT	Sent140
Previous-round Spearman ↑	.809	.761	.442	.631	.234	.339
Previous-round Pearson ↑	.796	.754	.529	.630	.340	.433

How the Game Moves. On CIFAR-10 $\mathrm{D}(0.1)$, we extract the pairwise interaction coefficients $I^{\mathrm{FB}}_{t,2}(\{i,j\})$ of Section 3.3 at rounds 25, 50, 100, 200, and 400. For each pair of consecutive snapshots, we keep the client pairs that are supported by at least 50 reservoir coalitions in both rounds, and we count how often a coefficient changes sign (Table 15). Between 36% and 48% of these well-supported pairs flip, and the rate does not decay: between rounds 200 and 400 it is still 36%, as it is between rounds 50 and 100. Which clients complement each other, and which duplicate each other, keeps changing after test accuracy has plateaued.

We next look at the selected sets themselves. We divide each run into four equal stages and report the mean Jaccard overlap between the selected sets of consecutive rounds (Table 16). The overlap sits around 0.2 in every setting and does not rise over training. The policy does not converge to a fixed coalition; it keeps tracking a moving target.

Table 15: Sign flips of pairwise interaction coefficients between snapshot rounds on CIFAR-10 $\mathrm{D}(0.1)$, restricted to pairs with reservoir support ≥ 50 in both rounds.

Rounds compared	Sign flips (%)	Pairs
25 vs. 50	47.5	111
50 vs. 100	36.1	221
100 vs. 200	42.2	149
200 vs. 400	36.3	213

Table 16: Mean Jaccard overlap between consecutive selected sets, by quarter of training.

Setting	Q1	Q2	Q3	Q4
CIFAR D0.1	.23	.23	.24	.24
CIFAR D0.5	.19	.17	.20	.21
CIFAR LT	.26	.27	.24	.28
SVHN	.20	.26	.23	.22
FinGPT	.17	.20	.16	.17
Sent140	.27	.37	.26	.21

Per-Setting Diagnostics. Figures 11 and 12 repeat the three diagnostics of Figure 5 for all twelve settings. Each row shows one setting in three panels: reservoir R^2 and next-round Spearman, cumulative pool and selection coverage, and per-client selection counts against the uniform expectation.

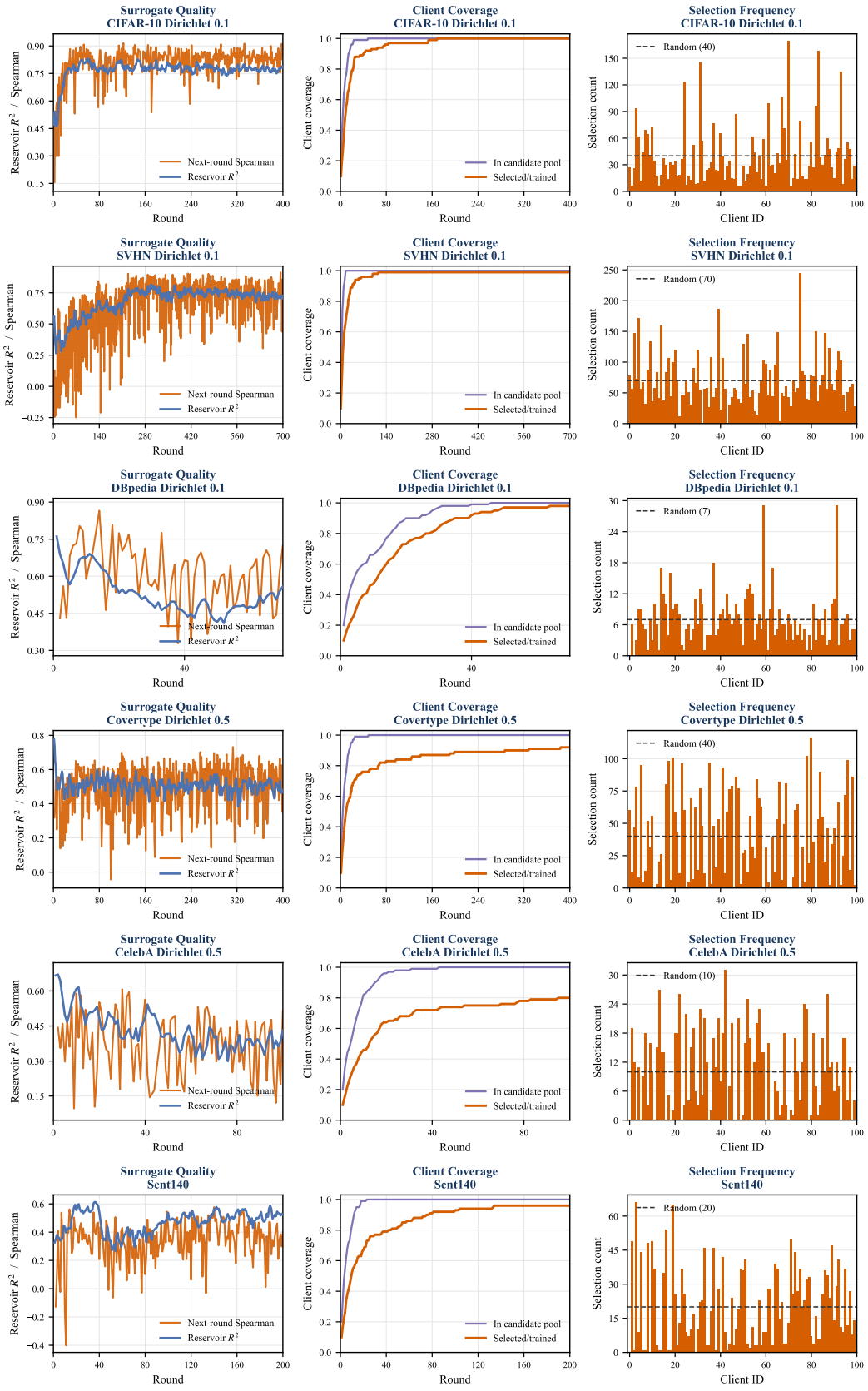


Figure 11: Surrogate quality, client coverage, and selection frequency. Rows are CIFAR-10 $\mathrm{D}(0.1)$, SVHN $\mathrm{D}(0.1)$, DBpedia $\mathrm{D}(0.1)$, Covertype $\mathrm{D}(0.5)$, CelebA $\mathrm{D}(0.5)$, and Sent140.

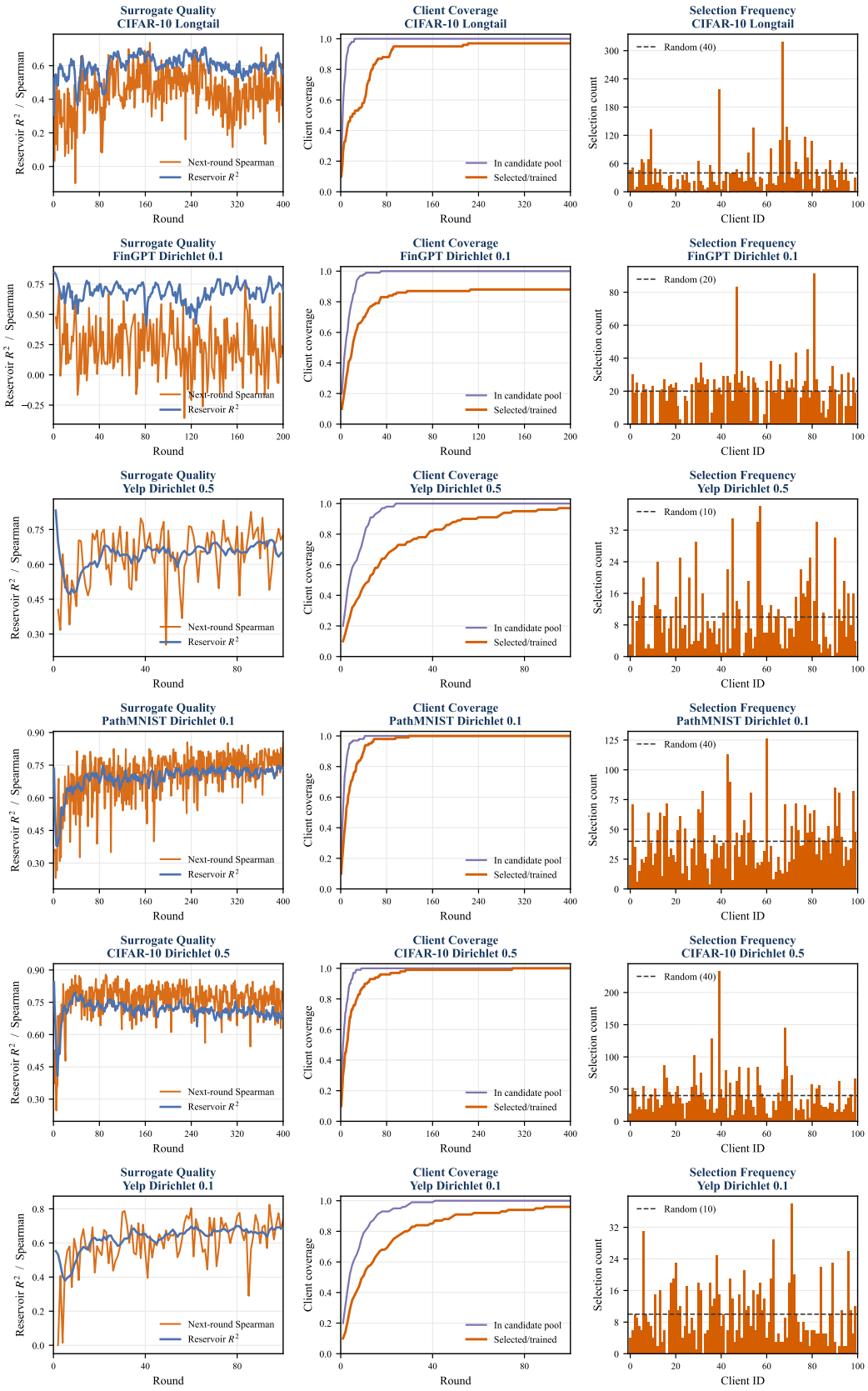


Figure 12: Surrogate quality, client coverage, and selection frequency. Rows are CIFAR-10 long-tail, FinGPT $\mathrm{D}(0.1)$, Yelp $\mathrm{D}(0.5)$, PathMNIST $\mathrm{D}(0.1)$, CIFAR-10 $\mathrm{D}(0.5)$, and Yelp $\mathrm{D}(0.1)$.

F.6 Robustness to the Server Validation Set

The representativeness of a server-side validation set may not be realistic, and in many cases it need not reflect the distribution of the FL system. We stress-test NexusFL by degrading $\mathcal{D}_{\text{val}}$ in size, label balance, and class coverage, each in turn, on CIFAR-10 $\mathrm{D}(0.1)$. We compare with the three baselines that also select on server validation data, S-FedAvg, AFedSV+, and FilFL, so that the comparison isolates the selection rule from the shared dependence on the set. Only the validation set changes; client data and the clean test set are fixed. Results are Mean / Final over 400 rounds for a single seed.

Size. With $e \in \{200, 500, 1024\}$ validation examples (Table 17), NexusFL's Mean varies within 2 pp and stays ahead of every baseline at every size. Coalition utilities estimated from 200 examples are noisy, but the surrogate averages that noise over the B coalitions it is fit on, and the ranking it induces survives.

Table 17: Validation-set size on CIFAR-10 $\mathrm{D}(0.1)$ (Mean / Final, seed 1). Bold marks the best Mean and Final per column.

Method	e=200	e=500	e=1024
NexusFL	**46.21** / 50.99	**48.25 / 54.99**	**46.56 / 55.59**
S-FedAvg	42.00 / 49.86	41.95 / 49.39	41.91 / 50.29
AFedSV+	43.21 / 51.39	43.19 / 52.19	43.59 / 52.94
FilFL	42.10 / **53.25**	42.11 / 51.22	41.22 / 44.72

Label Skew. We draw the validation label distribution from a Dirichlet with $\alpha \in \{0.05, 0.1, 0.5\}$ and set m_{val} to balanced accuracy (mean per-class recall), since plain accuracy on a skewed set would reward the majority classes alone, in line with the class-aware valuation of Yang et al. (2025). NexusFL leads on both metrics at every skew level (Table 18), and its Mean is flat across α. Two things follow: label imbalance in $\mathcal{D}_{\text{val}}$ is harmless once the metric accounts for it, and the same pipeline optimises whatever scalar m_{val} returns, since nothing in the surrogate or the selector is specific to accuracy.

Table 18: Validation label skew on CIFAR-10 $\mathrm{D}(0.1)$; metric is balanced accuracy (Mean / Final, seed 1). Bold marks the best per column.

Method	α=0.05	α=0.1	α=0.5
NexusFL	**47.24 / 55.20**	**46.79 / 55.82**	**47.74 / 56.23**
S-FedAvg	39.66 / 54.58	39.59 / 48.97	39.57 / 54.61
AFedSV+	43.57 / 54.04	43.28 / 52.49	42.68 / 53.75
FilFL	40.00 / 52.20	40.05 / 51.21	39.70 / 50.15

Class Coverage. Finally, we remove $m \in \{0, 1, 2, 3, 5\}$ classes from the validation set entirely, keeping the remaining classes balanced and the size at about 1024 examples (Table 19). With up to two of the ten classes absent, NexusFL keeps the best Mean and Final. At m=3, its Mean falls by 8 pp and below both Shapley selectors, and at m=5 it is close to FilFL. The Shapley selectors lose little across the whole range.

The asymmetry is consistent with how closely each method follows the objective it is given. NexusFL learns which coalitions score highest on m_{val}. If the validation set has no examples of a class, coalitions that neglect this class are not penalised, and the surrogate can learn to prefer them. The Shapley selectors average each client's marginal contribution over random contexts, which makes them less sensitive to the exact objective and thus to the missing classes. Class coverage of the server set is therefore the operative requirement for NexusFL, and its size and balance are not.

Table 19: Missing validation classes on CIFAR-10 $\mathrm{D}(0.1)$ (Mean / Final, seed 1). Bold marks the best per row.

Missing m	NEXUSFL	S-FedAvg	AFedSV+	FilFL
0	**46.56 / 55.59**	41.91 / 50.29	43.59 / 52.94	41.22 / 44.72
1	**46.76 / 53.38**	42.04 / 50.22	43.14 / 51.56	40.75 / 49.38
2	**43.00 / 51.69**	41.93 / 50.15	41.63 / 51.00	37.48 / 46.73
3	38.17 / 43.02	**41.83** / 49.92	40.66 / **51.15**	34.71 / 46.51
5	36.23 / 43.52	**41.63 / 49.27**	37.74 / 47.16	33.99 / 44.62

F.7 LEARNED INTERACTIONS ON REAL DATA

Section 4.4 checks the extracted interaction coefficients against a partition in which complementarity is built in. On a real benchmark there is no ground truth for which client pairs complement each other, so the question becomes whether the coefficients line up with a data-level notion of complementarity. We use the Jensen–Shannon (JS) divergence between the empirical label distributions of each client pair as that notion: a large divergence means the two clients hold different classes, which is complementarity in the sense of Figure 1. JS is fixed for the run, and we track the Spearman correlation between the pairwise coefficients $I^{\mathrm{FB}}_{t,2}(\{i,j\})$ of the fitted surrogate and JS_{ij} over snapshot rounds on CIFAR-10 $\mathrm{D}(0.1)$ (Table 20).

Table 20: Spearman correlation between pairwise interaction coefficients and label-distribution JS divergence on CIFAR-10 $\mathrm{D}(0.1)$, by snapshot round.

Round	25	50	100	200	400
$\rho(I^{\mathrm{FB}}_{t,2}(\{i,j\}),\ \mathrm{JS}_{ij})$	.23	.25	.20	.25	.29

The correlation is positive at every snapshot and highest at the end of training, so the surrogate assigns positive pair coefficients to clients with different label mixtures and negative ones to clients with similar mixtures, and this alignment strengthens as the surrogate accumulates evidence. The magnitude $|I^{\mathrm{FB}}_{t,2}(\{i,j\})|$ correlates near zero or negatively with JS: the sign of a coefficient says whether two clients complement or duplicate each other, its size does not say by how much their labels differ. The correlation is moderate, as it should be, since label overlap is one source of complementarity among several (feature distribution, data quality, local optimisation noise) and the coefficients also change over training (Table 15). Table 13 shows that this structure is acted on: in the same setting, the coalition the surrogate selects has far higher validation utility than a random coalition from the same pool.

F.8 RESERVOIR SIZE ABLATION

We vary the reservoir capacity $|\mathcal{R}|_{\max}$ on SVHN $\mathrm{D}(0.1)$ with p=20 and B=500 fixed (Table 21). With B=500 new observations per round, the capacity sets how many past rounds the reservoir can hold, from about 2 rounds at 1000 to about 20 at 10,000.

Table 21: Reservoir capacity on SVHN $\mathrm{D}(0.1)$ with p=20 and B=500 (Mean / Final test accuracy). Bold marks the best Mean and Final.

Capacity $\lvert\mathcal{R}\rvert_{\max}$	Mean	Final
1000	46.22	73.31
2500	47.67	73.96
5000	**49.03**	**74.05**
10,000	47.07	73.65

A larger reservoir lets the surrogate learn from observations collected over more rounds and therefore from more client combinations. This helps up to a capacity of 5000, where Mean rises by 2.81 pp over capacity 1000. At 10,000 Mean falls back by 1.96 pp. At a fixed budget B, the extra capacity is filled with observations from older rounds, whose utilities are more likely to be stale, which is consistent with the drop. Final accuracy moves less than Mean, within 0.74 pp across all four capacities.

F.9 Value Function Ablation

The surrogate is trained on the scalar utility it is given. The same selection rule can therefore target utilities other than the change in validation accuracy used in the main experiments. Table 22 reports the mean utility of the selected coalition over 100 rounds on CIFAR-10 $\mathrm{D}(0.1)$, for a single seed. Higher is better on every utility. The four utilities are the negative validation loss (Fourati et al., 2024), the change in validation loss, the maverick score of class-reweighted validation accuracy (Yang et al., 2025), and the alignment of the aggregated update with a reference gradient (Xu et al., 2021). The reference is an oracle gradient formed from the cached updates of all n clients. Gradient alignment therefore uses no server validation set.

Table 22: Mean utility of the selected coalition over 100 rounds on CIFAR-10 $\mathrm{D}(0.1)$, single seed. Higher is better on every utility.

Method	Negative loss	Loss change	Maverick score	Gradient alignment
NexusFL	$\mathbf{-1.965}$	$\mathbf{0.009}$	$\mathbf{0.305}$	$\mathbf{0.890}$
S-FedAvg	-2.091	0.007	0.250	0.682
FilFL	-2.029	0.005	0.237	0.592
AFedSV+	-2.122	-0.075	0.247	0.549

NexusFL attains the highest mean on all four utilities. The surrogate therefore tracks the utility it is trained on, including gradient alignment, which does not require a server validation set.

F.10 Selection Mechanism Ablation

NexusFL builds the coalition by sequential softmax over the surrogate. Table 23 compares that rule with greedy selection with swaps, multi-start swap search, and exact maximization of the fitted tree ensemble (Mišić, 2020) on CIFAR-10 $\mathrm{D}(0.1)$ and on the long-tail partition.

Table 23: Selection rules on CIFAR-10 (Mean / Final test accuracy, and seconds per round). Bold marks the best Mean and Final in each partition.

	$\mathbf{D(0.1)}$		**Long-tail**	
Selection rule	**Mean / Final**	**s/round**	**Mean / Final**	**s/round**
Sequential softmax	$\mathbf{46.58 / 55.48}$	1.56	$\mathbf{32.74 / 43.43}$	1.36
Greedy selection with swaps	$43.13 / 50.03$	1.44	$32.56 / 43.10$	1.29
Multi-start swap search	$44.44 / 53.07$	1.70	$32.42 / 43.03$	1.44
Exact tree optimization	$45.92 / 52.50$	4.17	$32.58 / 43.13$	3.18

Sequential softmax has the highest Mean and Final on both partitions. On $\mathrm{D}(0.1)$ it leads exact tree optimization by 0.66 pp of Mean and 2.98 pp of Final, and greedy swaps by 3.45 and 5.45 pp. On the long-tail partition the four rules lie within 0.32 pp of Mean and 0.40 pp of Final. Exact maximization of the ensemble takes 4.17 and 3.18 seconds per round, against 1.56 and 1.36 for sequential softmax. The softmax policy is more accurate on both partitions and runs in less than half the time.

F.11 Surrogate Class

The guarantees of Section 3.2 are stated for any surrogate in the class $\mathcal{F}_d$ of depth-d tree ensembles. Algorithm 1 fits this class with gradient boosting. This ablation asks how much the choice of surrogate matters. We replace XGBoost with three alternatives that differ in what they can represent and in how they are fit. A depth-4 random forest lies in the same class $\mathcal{F}_4$ but averages independently grown trees instead of fitting residuals. A zero-mean Gaussian process with a fixed RBF kernel on the coalition indicators can model interactions, but it does not admit the exact interaction extraction of Section 3.3. Ridge regression is linear in the indicators, which makes it an additive, per-client model and the $d{=}1$ case of Corollary 1. All four surrogates are fit on the same reservoir and queried by the same sequential softmax. The comparison is a single seed on CIFAR-10 $\mathrm{D}(0.1)$ for 400 rounds.

Table 24: Surrogate class on CIFAR-10 $\mathrm{D}(0.1)$, 1 seed, 400 rounds (Mean / Final test accuracy, and mean seconds per round). The tree models use depth 4. Bold marks the best Mean and Final.

Surrogate	Mean / Final	s/round
XGBoost, depth 4	**48.05 / 57.48**	1.56
Random forest, depth 4	46.77 / 54.22	3.00
Gaussian process	46.81 / 56.14	2.72
Ridge regression	44.89 / 51.89	1.50

XGBoost has the highest Mean and Final. The depth-4 forest can represent the same functions, yet it trails by 1.28 pp of Mean and 3.26 pp of Final. The fitting procedure therefore matters, not only the class. Boosting fits each tree to the residual of the ensemble so far and reduces the bias of shallow trees, whereas averaging mainly reduces variance and leaves the bias of each depth-4 tree in place. With at most 2500 reservoir entries, bias reduction appears to matter more than variance reduction. The Gaussian process comes closest on Final (1.34 pp behind), but it adds 1.16 seconds per round and gives up exact interaction extraction. Ridge regression is the weakest, 3.16 pp behind on Mean and 5.59 pp on Final. An additive surrogate cannot prefer a complementary partner over a redundant one, which is consistent with $d{=}1$ being the worst depth in Table 11. Coalition evaluation and client training are shared across the four variants, and the differences in seconds per round therefore come mainly from fitting and querying the surrogate.