Single-Round Scalable Analytic Federated Learning

Alan T. L. Bacellar[1], Mustafa Munir[1], Felipe M. G. França[2],
Priscila M. V. Lima[3], Radu Marculescu[1], Lizy K. John[1]
[1]University of Texas at Austin [2]Google
[3]Federal University of Rio de Janeiro

alanbacellar@utexas.edu

Abstract

Federated Learning (FL) is plagued by two key challenges: high communication overhead and performance collapse on heterogeneous (non-IID) data. Analytic FL (AFL) provides a single-round, data distribution invariant solution, but is limited to linear models. Subsequent non-linear approaches, like DeepAFL, regain accuracy but sacrifice the single-round benefit. In this work, we break this trade-off. We propose SAFLe, a framework that achieves scalable non-linear expressivity by introducing a structured head of bucketed features and sparse, grouped embeddings. We prove this non-linear architecture is mathematically equivalent to a high-dimensional linear regression. This key equivalence allows SAFLe to be solved with AFL's single-shot, invariant aggregation law. Empirically, SAFLe establishes a new state-of-the-art for analytic FL, significantly outperforming both linear AFL and multi-round DeepAFL in accuracy across all benchmarks, demonstrating a highly efficient and scalable solution for federated vision.

1. Introduction

Federated Learning (FL) enables multiple clients or devices to collaboratively train a shared model without exposing their private data. Instead of centralizing data, clients perform local updates and periodically communicate model parameters to a server, which aggregates them into a global model [16]. While conceptually appealing, conventional FL frameworks require many communication rounds—often hundreds or thousands—for a model to converge. In practical deployments, clients can operate at different speeds, disconnect intermittently, or fail mid-training, creating stragglers and asynchronous updates. Such instability causes training to progress unevenly, and the global model may take days or weeks to reach convergence, severely limiting FL's real-world scalability.

Beyond communication inefficiency, a deeper issue lies in **statistical heterogeneity** across clients. In real FL systems, local data distributions often differ sharply—for instance, users capture different visual styles, hospitals record different patient populations, or sensors observe non-overlapping environments. This non-IID nature of the data means that each client's gradient direction diverges from the global optimum, degrading performance and convergence stability. Existing methods attempt to address this through various regularizers, dynamic aggregation schemes, and using pre-trained models for initialization and distilation [1, 13, 14, 18], but these stuggles with non-IID settings.

To overcome these limitations, recent work proposed *Analytic Federated Learning* (AFL) [7], which formulates the FL problem in closed form. AFL leverages a pre-trained backbone to extract embeddings on each client, and trains a linear regression head analytically in only one communication round. Its analytic aggregation law guarantees invariance to both data partitioning and client count, enabling the global solution to remain identical to centralized training regardless of heterogeneity. As a result, AFL achieves higher accuracy than conventional iterative FL methods under highly non-IID conditions, while requiring only a single communication round instead of hundreds. Despite these appealing properties, AFL remains constrained by its linear model structure, which limits representational capacity and the ability to capture nonlinear feature interactions.

More recently, DeepAFL [3] proposed a layer-wise analytic training scheme that extends AFL into deeper architectures. DeepAFL retains AFL's invariance property, but trades communication efficiency for greater accuracy. Each analytic layer requires a separate aggregation round, increasing synchronization overhead and deviating from AFL's single-pass analytic design. Consequently, DeepAFL achieves higher accuracy than AFL on non-IID data but at the cost of multiple communication rounds.

In this work, we propose **SAFLe** — *Sparse Analytic Federated Learning with nonlinear embeddings* — a framework that retains AFL's single-round analytic formulation

while significantly enhancing model expressivity. SAFLe introduces a deterministic nonlinear transformation pipeline composed of three stages: *feature bucketing*, *shuffling and grouping* and *sparse embeddings*. We prove that this nonlinear transformation pipeline can be reformulated as an equivalent analytic regression problem, preserving AFL's closed-form training and invariance properties.

This design allows SAFLe to scale model capacity by simply increasing the number of sparse embeddings, without altering the analytic formulation or introducing extra communication rounds. Empirically, SAFLe achieves higher accuracy than both AFL and DeepAFL across all datasets, including highly non-IID and large-client settings, while maintaining a single-round communication regime.

Our main contributions are summarized as follows:

- **A Novel Non-Linear Analytic Framework (SAFLe):** We propose SAFLe, a framework that uses a sparse, multi-embedding architecture to learn non-linear feature interactions, dramatically increasing model expressivity over previous analytic methods.
- **Proof of Linear Equivalence:** We prove that this complex non-linear model is mathematically equivalent to a high-dimensional linear regression. This is the key theoretical insight that makes it analytically solvable.
- **Single-Round and Invariant Aggregation:** By leveraging this equivalence, SAFLe is the first non-linear framework to inherit AFL's two key properties: it converges in a single communication round and its solution is mathematically invariant to statistical data heterogeneity.
- **State-of-the-Art Analytic Performance:** SAFLe establishes a new state-of-the-art for analytic federated learning, significantly outperforming prior single-round (AFL) and multi-round (DeepAFL) methods.

2. Background and Preliminaries

2.1. Federated Learning

Conventional iterative FL frameworks, pioneered by FedAvg [16], train a global model through multiple rounds of communication. In this paradigm, clients train local models, and a central server periodically averages their weights. While effective, this iterative approach faces significant challenges with non-independently and identically distributed (non-IID) data. A large body of research has been proposed to address this. Methods like FedProx [14] add a proximal term to the local objective to restrict updates, while FedNova [18] normalizes local updates before averaging. Other major approaches include Model-Contrastive Federated Learning (MOON) [13], which utilizes contrastive learning; FedDyn [1], which uses dynamic regularization; and various knowledge distillation techniques like FedNTD [12], FedGen [21], and FedDisco [20]. Despite their different strategies, all these methods are fundamentally gradient-based and iterative, incurring substantial communication rounds.

To address the high communication overhead, One-Shot Federated Learning (OFL) methods have been proposed [2]. These methods aim to train a final model with only a single communication round, where clients train models locally and send them to the server once. Common OFL strategies involve ensembling the local models on the server [2], often using knowledge distillation (KD) to create a single, unified model [4, 19]. Methods like FedCVAE-Ens use generative models to facilitate this one-shot distillation [8]. However, this efficiency comes at a well-documented cost: OFL methods typically suffer a significant drop in accuracy compared to their iterative counterparts, especially in settings with high data heterogeneity [2].

2.2. Analytic Federated Learning (AFL)

Analytic Federated Learning (AFL) [7] was proposed to solve both problems. It is a gradient-free, single-round framework that uses a pre-trained frozen backbone. On each client k, a linear classification head is trained by solving a simple least-squares (LS) problem:

$$\min_{W_k} \mathcal{L}(W_k) = ||Y_k - X_k W_k||_F^2 \tag{1}$$

where X_k is the matrix of backbone embeddings and Y_k is the one-hot label matrix.

AFL's core contribution is the **Absolute Aggregation (AA) law**. This law provides an analytic formula for the server to perfectly reconstruct the centralized global model W (the model that would have been trained on all data at once) from the local models (W_u, W_v, etc.) in a single round. For two clients u and v, the aggregation is:

$$W = \mathcal{W}_u W_u + \mathcal{W}_v W_v \tag{2}$$

The weighting matrices, $\mathcal{W}_u$ and $\mathcal{W}_v$, are not simple averages; they are computed analytically from the clients' data covariance matrices ($C_u = X_u^T X_u$, $C_v = X_v^T X_v$). Because this formula is an exact, closed-form solution, the final model W is mathematically invariant to how the data is partitioned, making it robust to any degree of non-IID data or client count. To handle rank-deficient data, a **Regularization Intermediary (RI)** process is used to add and then analytically remove regularization, preserving the exact solution.

AFL is thus extremely fast and robust, but its primary limitation is its constrained representational power, as it can only train a single *linear* layer.

2.3. Deep Analytic Federated Learning (DeepAFL)

DeepAFL [3] was introduced to address AFL's linearity constraint by building a deeper, non-linear analytic model

inspired by ResNet, that achieves non-linearity by using layers with random weights.

However, DeepAFL sacrifices AFL's single-shot efficiency for this added expressivity. It uses a sequential, layer-wise training protocol that requires multiple communication rounds per layer. For each layer t in the model, the server must: 1) Perform a full aggregation round (like AFL) to solve for the layer's classifier, W_t. 2) Perform a *second*, separate aggregation round to collect components needed to solve for a new feature transformation, Ω_{t+1}, which uses random projections to build the input for the next layer.

This multi-round process must be repeated for every layer added to the model. While DeepAFL achieves higher accuracy than AFL, it re-introduces significant synchronization overhead. This creates a clear trade-off: AFL is single-round but linear, while DeepAFL is non-linear but multi-round.

3. Methodology

3.1. Intuition

The primary limitation of AFL is its linear model, which restricts representational capacity. The key challenge is to introduce non-linearity while preserving the single-round, analytic solution. DeepAFL, the first attempt to solve this, derives its non-linearity from random projections. This stochastic approach is inefficient; it relies on the chance that stacking enough random matrices and activations will eventually approximate the complex, non-linear landscape of the data. Consequently, it requires many layers to achieve high accuracy, which in turn re-introduces the multi-round communication burden that AFL was designed to eliminate.

We propose a different, more structured approach. Instead of relying on random weights, our core idea is to deterministically partition the continuous feature space using bucketing. This creates a set of discrete "regions" in the data landscape. We can then use embedding layers to directly learn the optimal output (e.g., logits) for inputs that fall into specific combinations of these regions. This directly models the non-linear function. A naive implementation, however, would be an intractable lookup table that would severely overfit. We solve this by introducing a sparse, multi-embedding architecture. The feature-buckets are shuffled and partitioned into many small, independent groups (E), and each embedding layer learns from only one of these "subviews." The final prediction is the sum of all these "expert" embeddings. See Figure 1. This forces generalization, as the model learns to make predictions from diverse, partial feature-combinations. This method is fundamentally more structured than DeepAFL's, and as we will prove, it remains fully analytic and solvable in a single round.

This architecture is conceptually similar to a Mixture of Experts (MoE) model [9], as it employs a set of "experts" (our E embedding layers) to learn specialized functions over different parts of the input space. However, SAFLe differs in two crucial ways: its routing mechanism and its activation pattern. First, instead of a *learned* gating network that dynamically routes an input, SAFLe uses our *deterministic* bucketing-and-shuffling pipeline as a fixed pre-router. Second, rather than the sparse activation of a standard MoE, SAFLe uses dense activation: *all* E experts are activated for *every* input.

This design is the key to both its expressivity and its analytic nature. Each expert is forced to learn a specialized, non-linear function based on only a small, fixed "subview" of the total input features. The model's full non-linear capacity comes from summing the contributions of all these parallel "subview experts." Crucially, this fixed, deterministic routing—unlike a standard MoE's learned gate—allows us to reformulate the entire non-linear architecture as an equivalent high-dimensional linear regression. This reformulation is precisely what makes our model compatible with AFL's Absolute Aggregation (AA) law, enabling a single-round, gradient-free solution.

3.2. SAFLe

Our objective is to break AFL's linearity barrier while retaining its single-round, gradient-free properties. We replace AFL's simple linear regressor with a deterministic, non-linear transformation pipeline, as illustrated in Figure 1.

Let the output of the pre-trained backbone be a feature vector $x \in \mathbb{R}^{d_b}$, where d_b is the dimension of the backbone's feature output. Our non-linear head, $f_{NL}(x)$, transforms this vector into the final class logits $\hat{y} \in \mathbb{R}^C$. This transformation consists of three stages:

1. Pre-Non-Linearity Bucketing: First, we apply L different bucketing functions $B_l(\cdot)$ to each feature x_i in the backbone output. Each function $B_l : \mathbb{R} \to \{0, \dots, k-1\}$ quantizes the continuous feature into one of k discrete bins. This transforms the original vector $x \in \mathbb{R}^{d_b}$ into a new quantized integer vector $b \in \mathbb{Z}^{d_q}$, where the new dimension $d_q = d_b \times L$.

$$b = [B_1(x_1), \dots, B_L(x_1), \dots, B_1(x_{d_b}), \dots, B_L(x_{d_b})] \tag{3}$$

The choice of bucketing strategy is discussed in our ablations (Section 4).

2. Shuffling and Grouping: To break feature locality and create diverse "subviews" for our experts, the integer vector b is shuffled using a fixed, deterministic permutation P. This shuffled vector $b' = P(b)$ is then partitioned into E groups (our "experts"). Each group j contains G integer indices, $g_j = [b'_{j,1}, \dots, b'_{j,G}]$, such that $E \times G = d_q$.

3. Sparse Embedding and Summation: For each of the E groups, we compute a single composite index idx_j by

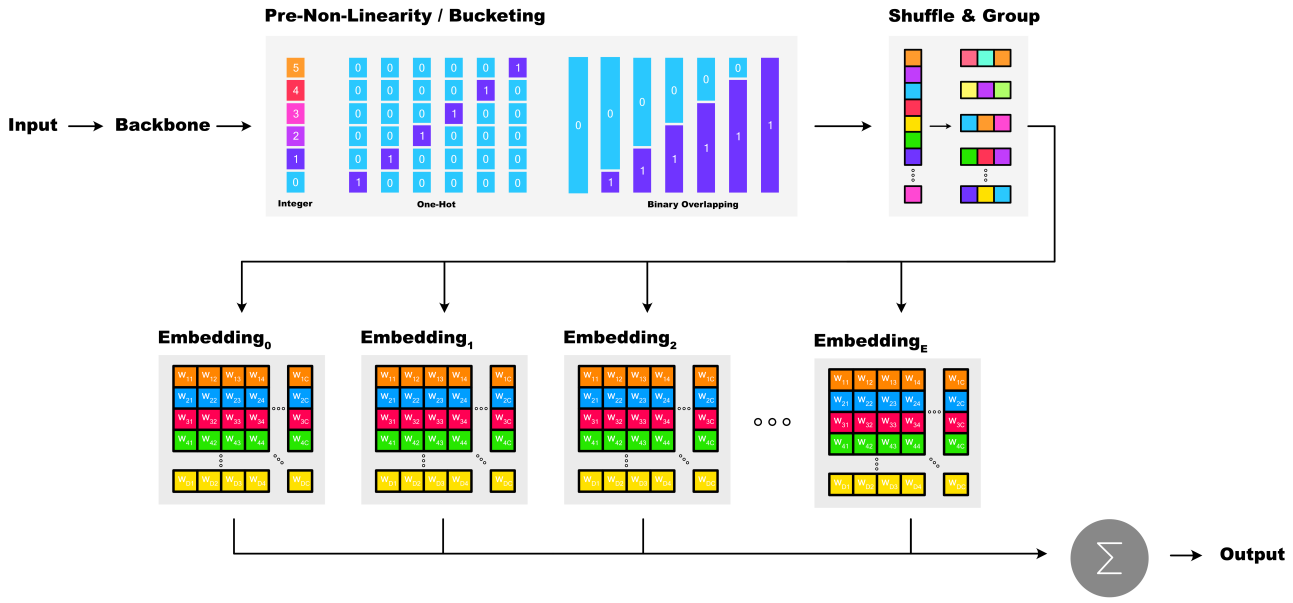


Figure 1. Illustration of the proposed SAFLe model. Input images are first processed by a pre-trained backbone network to extract features. A Pre-Non-Linearity by Bucketing transformation is then applied to each feature using one of three methods: Integer, One-Hot, or Binary Overlapping bucketing. The resulting bucketed features are shuffled and divided into E groups, each of which is passed through a corresponding learnable embedding. The outputs of all embeddings are summed to produce the final model output. The embedding parameters admit an analytical solution and can be optimized through a federated learning algorithm that remains invariant to both data distribution and the number of participating clients.

treating the G integers as digits in a base-k number system:

$$idx_j = \sum_{i=1}^{G} g_{j,i} \cdot k^{i-1} \tag{4}$$

This composite index has a maximum value of $V = k^G$, which defines the "vocabulary size" (i.e., number of rows) for each of our E embedding matrices. Each embedding matrix $W_j \in \mathbb{R}^{V \times C}$ thus maps its specific group of G bucketed features to a C-dimensional output vector.

The final output logit $\hat{y}$ is the dense activation and sum of the outputs from all E embedding-experts:

$$\hat{y} = f_{NL}(x) = \sum_{j=1}^{E} W_j[idx_j, :] \tag{5}$$

As we show in our ablations, using many small, independent embeddings (high E, small G and V) is critical for generalization.

3.3. Analytic Formulation and Aggregation

We now prove that the proposed non-linear architecture f_{NL} can be solved with a closed-form analytic solution. We first formalize the linear equivalence.

Lemma 1. ***(SAFLe Linear Equivalence).*** *The non-linear model $f_{NL}(x)$, which learns feature interactions via grouped embedding lookups, is equivalent to a linear regressor W_{global} in a high-dimensional sparse feature space $\Phi(x)$.*

Proof. The non-linearity stems from the grouped indexing (Eq. 2), where a group of G features $\{g_{j,1}, \ldots, g_{j,G}\}$ is combined to form a single index idx_j. The model's output (Eq. 3) is the sum of E such lookups.

To solve for the weights W_j analytically, we reformulate this lookup. An embedding lookup $W_j[idx_j, :]$ is a linear row-selection operation, which can be expressed using a one-hot vector $\phi_j(x) \in \{0, 1\}^V$ where the only non-zero entry is at position idx_j:

$$W_j[idx_j, :] = \phi_j(x)^T W_j \tag{6}$$

The total output $\hat{y}$ is the sum of all E lookups:

$$\hat{y} = \sum_{j=1}^{E} \phi_j(x)^T W_j \tag{7}$$

We now define two global structures. First, a single, high-dimensional sparse feature vector $\Phi(x)$ by horizontally concatenating all E one-hot vectors:

$$\Phi(x) = [\phi_1(x)^T | \phi_2(x)^T | \ldots | \phi_E(x)^T]^T \in \mathbb{R}^{D_e} \tag{8}$$

where the total embedding dimension is $D_e = E \times V$. Second, a global weight matrix W_{global} by vertically stacking all E embedding matrices:

$$W_{global} = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_E \end{pmatrix} \in \mathbb{R}^{D_e \times C} \tag{9}$$

With these structures, the output of our non-linear model $f_{NL}(x)$ is perfectly represented as a standard linear model:

$$\hat{y} = \Phi(x)^T W_{global} \tag{10}$$

For an entire batch of data X_k on client k, we can construct the full linearized feature matrix Φ_k (of size $N_k \times D_e$). The global objective over K clients is to find W_{global} by minimizing the least-squares loss:

$$\mathcal{L}(W_{global}) = \sum_{k=1}^{K} ||Y_k - \Phi_k W_{global}||_F^2 = ||Y - \Phi W_{global}||_F^2 \tag{11}$$

This objective is mathematically identical to the one solved by AFL, simply replacing their feature matrix X_k with our high-dimensional sparse feature matrix Φ_k. □

This equivalence allows us to apply AFL's federated aggregation framework. The optimal global model W_{global} is the solution to the normal equations:

$$W_{global} = (\Phi^T \Phi)^\dagger (\Phi^T Y) \tag{12}$$

where the global components are simple sums of the local components:

$$\Phi^T \Phi = \sum_{k=1}^{K} \Phi_k^T \Phi_k \quad \text{and} \quad \Phi^T Y = \sum_{k=1}^{K} \Phi_k^T Y_k \tag{13}$$

This structure is identical to AFL and can be solved in a single round using the Regularization Intermediary (RI) process.

Theorem 1. ***(Federated Analytic Solution).*** *By extending the RI-AA Law from [7], the global model W_{global} can be solved analytically in a single communication round.*

Proof. Following the RI process [7], each client k computes and transmits two matrices based on its local high-dimensional features Φ_k:

$$C_k^r = \Phi_k^T \Phi_k + \gamma I \in \mathbb{R}^{D_e \times D_e} \tag{14}$$

$$M_k = \Phi_k^T Y_k \in \mathbb{R}^{D_e \times C} \tag{15}$$

The server aggregates these components in a single step:

$$C_{agg}^r = \sum_{k=1}^{K} C_k^r \quad \text{and} \quad M_{agg} = \sum_{k=1}^{K} M_k \tag{16}$$

The server first computes the aggregated regularized solution:

$$W_{global}^r = (C_{agg}^r)^{-1} M_{agg} \tag{17}$$

Finally, the server analytically removes the regularization to recover the exact, unregularized global solution W_{global} using the recovery formula from AFL:

$$W_{global} = (C_{agg}^r - K\gamma I)^\dagger M_{agg} \tag{18}$$

Thus, we have derived a closed-form, single-round analytic solution for our non-linear model. The computation is tractable as $D_e = E \times V$ is a controllable hyperparameter. □

4. Experiments

4.1. Experimental Setup

Datasets & Settings. We conduct our experiments on three prominent federated learning benchmark datasets: CIFAR-10 [10], CIFAR-100 [10], and Tiny-ImageNet [11]. To simulate statistical heterogeneity, we follow the standard setup used in AFL [7] and DeepAFL [3] and employ two common non-IID partitioning strategies [15]: Non-IID-1 (LDA), where data is partitioned using a Latent Dirichlet Allocation (LDA) process controlled by the parameter α, and Non-IID-2 (Sharding), where data is sorted by label and divided into shards, with each client receiving s shards. In both settings, smaller values of α or s indicate a higher degree of data heterogeneity. For CIFAR-1.0, we test with $\alpha \in \{0.1, 0.05\}$ and $s \in \{2, 4\}$. For the more complex CIFAR-100 and Tiny-ImageNet datasets, we use $\alpha \in \{0.1, 0.01\}$ and $s \in \{5, 10\}$. Unless otherwise specified, all experiments use a total of $K = 100$ clients.

Baselines & Metrics. We compare our proposed SAFLe against two categories of federated learning methods. The first category, Iterative Baselines, includes five widely-used gradient-based methods: FedAvg [16], FedProx [14], MOON [13], FedDyn [1], and FedNTD [12]. The second, Analytic Baselines, includes the two most relevant analytic methods: AFL [7], the single-round linear analytic model, and DeepAFL [3], the state-of-the-art multi-round analytic model. For a fair and direct comparison, all methods (both iterative and analytic) are initialized with the same ResNet-18 backbone pre-trained on ImageNet. The results for all baseline methods are directly obtained from the established benchmark provided in [3, 7]. We evaluate all approaches on Top-1 Accuracy (%) to measure model performance, Communication Rounds to measure synchronization overhead, and Communication Cost (MB) to measure total data transmission per client.

4.2. Experimental Results

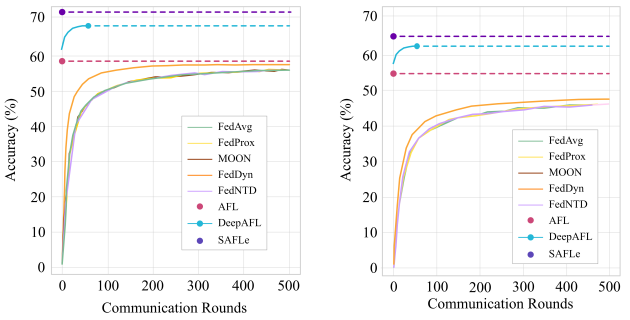
Accuracy Comparisons The main accuracy results, presented in Table 1, show that SAFLe consistently and sig-

nificantly outperforms all iterative and analytic baselines across all datasets and non-IID settings. On CIFAR-10, SAFLe achieves 90.73% accuracy, surpassing the next-best analytic method, DeepAFL [3] (86.43%), by over 4.3% and the original AFL [7] (80.75%) by a 10% margin. On CIFAR-100, SAFLe (70.61%) again leads DeepAFL (66.98%) and AFL (58.56%), and on Tiny-ImageNet, our method (64.58%) continues to outperform DeepAFL (62.35%) and AFL (54.67%). This demonstrates the superior representational power of our proposed non-linear analytic architecture, which achieves higher accuracy than even the multi-round DeepAFL while maintaining single-round efficiency.

Invariance Analysis Table 1 empirically shows the core theoretical benefit of our analytic formulation: invariance to data heterogeneity. While all iterative gradient-based methods suffer severe performance degradation as the data becomes more non-IID (e.g., $\alpha = 0.01$ or $s = 5$), our method does not. For instance, on CIFAR-100, FedAvg's accuracy plummets from 56.62% ($\alpha = 0.1$) to 32.99% ($\alpha = 0.01$), a drop of over 23.6%. In stark contrast, all analytic methods are completely unaffected by the data distribution. Our proposed SAFLe achieves an identical 70.61% accuracy across all four non-IID settings on CIFAR-100, just as AFL (58.56%) and DeepAFL (66.98%) maintain their respective scores. This invariance also extends to the number of clients, as shown in Figure 5. While FedAvg's performance clearly declines as K increases from 100 to 1000, SAFLe's accuracy remains constant, matching the invariance of AFL. This experimentally showcases SAFLe successful extension of the mathematical invariance of AFL to a more expressive non-linear model.

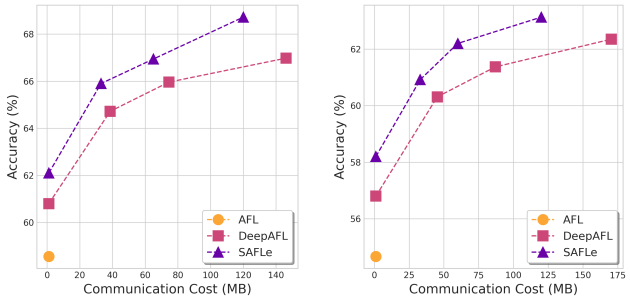
Communication Rounds Efficiency DeepAFL [3] scales AFL by adding *depth* (i.e., more layers), but at the direct trade-off of more communication rounds. As seen in Figures 4 and 2, DeepAFL must introduce two new communication rounds for every additional layer, requiring 41 rounds for a $T = 20$ model. Our method, SAFLe, scales by adding *width* (i.e., more experts E), allowing us to scale the model's expressivity while remaining a **single-shot** solution. SAFLe's expressivity is decoupled from its number of communication rounds, which is always one round.

Communication Cost Efficiency While SAFLe operates in a single round, this design choice increases the size of the payload for that single transmission. However, as shown in Figure 3, when comparing the total communication cost (MB) required to achieve a target accuracy, SAFLe is significantly more efficient than DeepAFL. For example, to reach $\approx 67\%$ accuracy on CIFAR-100, DeepAFL requires 146MB of total communication. Our method achieves this same accuracy using only 70MB, a reduction of over 50%. This efficiency gain is even more pronounced on Tiny-ImageNet: DeepAFL needs 170MB to achieve 62.3% accuracy, while SAFLe requires only 60MB, an almost 3x reduction. This low communication cost is possible due to the sparse nature of the embeddings, as discussed in the methodology, and experimentally shown in the next paragraph.



(a) CIFAR-100 (b) Tiny-ImageNet

Figure 2. Accuracy vs. Communication Rounds on (a) CIFAR-100 and (b) Tiny-ImageNet. Analytic methods SAFLe and AFL achieve final accuracy in a single round.



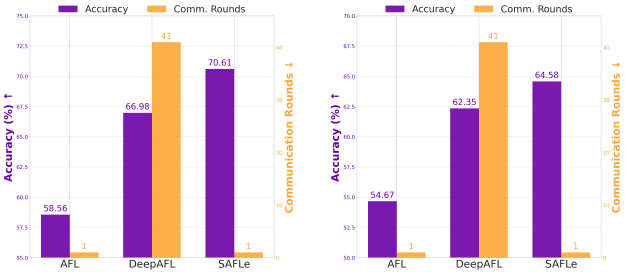
(a) CIFAR-100 (b) Tiny-ImageNet

Figure 3. Accuracy vs. Total Communication Cost (MB) per client on (a) CIFAR-100 and (b) Tiny-ImageNet. Our single-round method, SAFLe, achieves a given accuracy (e.g., $\approx 62\%$ on Tiny-ImageNet) for a fraction of the total communication cost required by the multi-round DeepAFL.

Model Size Comparison. We note that the additional parameters introduced by our SAFLe architecture—the sparse embedding layers—are negligible in size compared to the backbone. In all configurations, the total size of these embeddings is less than 0.3MB, which is insignificant compared to the $\approx$ 44MB ResNet-18. This ensures a fair comparison, as it demonstrates that our method's significant accuracy improvements are not attributable to an unfair advantage in total model parameters. While SAFLe's non-linear design *does* scale model capacity, which is the source of its improved expressivity, it does so with a fractional parameter increase so small that it does not to provide any meaningful unfair advantage to the standard iterative, gradient-based baselines. As a reference, this additional size is equivalent to adding approximately 146 neurons to

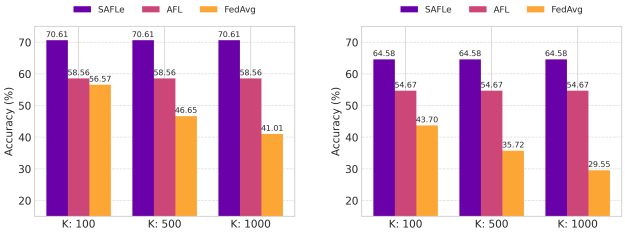
Table 1. The top-1 accuracy (%) of compared methods under two non-IID settings. Settings controlled by α and s are NIID-1 and NIID-2, respectively. The data is reported as average and standard deviation after multiple runs. Results in bold are the best within the compared methods in the same setting.

Dataset	Setting	FedAvg	FedProx	MOON	FedDyn	FedNTD	AFL	DeepAFL	SAFLe
CIFAR-10	$\alpha = 0.1$	$64.02_{\pm 0.18}$	$64.07_{\pm 0.08}$	$63.84_{\pm 0.03}$	$64.77_{\pm 0.11}$	$64.64_{\pm 0.02}$	$80.75_{\pm 0.00}$	$86.43_{\pm 0.07}$	$\mathbf{90.73_{\pm 0.07}}$
	$\alpha = 0.05$	$60.52_{\pm 0.39}$	$60.39_{\pm 0.09}$	$60.28_{\pm 0.17}$	$60.35_{\pm 0.54}$	$61.16_{\pm 0.33}$	$80.75_{\pm 0.00}$	$86.43_{\pm 0.07}$	$\mathbf{90.73_{\pm 0.07}}$
	$s = 4$	$68.47_{\pm 0.13}$	$68.46_{\pm 0.08}$	$68.47_{\pm 0.15}$	$73.50_{\pm 0.11}$	$70.24_{\pm 0.11}$	$80.75_{\pm 0.00}$	$86.43_{\pm 0.07}$	$\mathbf{90.73_{\pm 0.07}}$
	$s = 2$	$57.81_{\pm 0.03}$	$57.61_{\pm 0.12}$	$57.72_{\pm 0.15}$	$64.07_{\pm 0.09}$	$58.77_{\pm 0.18}$	$80.75_{\pm 0.00}$	$86.43_{\pm 0.07}$	$\mathbf{90.73_{\pm 0.07}}$
CIFAR-100	$\alpha = 0.1$	$56.62_{\pm 0.12}$	$56.45_{\pm 0.22}$	$56.58_{\pm 0.02}$	$57.55_{\pm 0.08}$	$56.60_{\pm 0.14}$	$58.56_{\pm 0.00}$	$66.98_{\pm 0.04}$	$\mathbf{70.61_{\pm 0.14}}$
	$\alpha = 0.01$	$32.99_{\pm 0.20}$	$33.37_{\pm 0.09}$	$33.34_{\pm 0.11}$	$36.12_{\pm 0.08}$	$32.59_{\pm 0.21}$	$58.56_{\pm 0.00}$	$66.98_{\pm 0.04}$	$\mathbf{70.61_{\pm 0.14}}$
	$s = 10$	$55.76_{\pm 0.13}$	$55.80_{\pm 0.16}$	$55.70_{\pm 0.25}$	$61.09_{\pm 0.09}$	$54.69_{\pm 0.15}$	$58.56_{\pm 0.00}$	$66.98_{\pm 0.04}$	$\mathbf{70.61_{\pm 0.14}}$
	$s = 5$	$48.33_{\pm 0.15}$	$48.29_{\pm 0.14}$	$48.34_{\pm 0.19}$	$59.34_{\pm 0.11}$	$47.00_{\pm 0.19}$	$58.56_{\pm 0.00}$	$66.98_{\pm 0.04}$	$\mathbf{70.61_{\pm 0.14}}$
Tiny ImageNet	$\alpha = 0.1$	$46.04_{\pm 0.27}$	$46.47_{\pm 0.23}$	$46.21_{\pm 0.14}$	$47.72_{\pm 0.22}$	$46.17_{\pm 0.16}$	$54.67_{\pm 0.00}$	$62.35_{\pm 0.01}$	$\mathbf{64.58_{\pm 0.23}}$
	$\alpha = 0.01$	$32.63_{\pm 0.19}$	$32.26_{\pm 0.14}$	$32.38_{\pm 0.20}$	$35.19_{\pm 0.06}$	$31.86_{\pm 0.44}$	$54.67_{\pm 0.00}$	$62.35_{\pm 0.01}$	$\mathbf{64.58_{\pm 0.23}}$
	$s = 10$	$39.06_{\pm 0.26}$	$38.97_{\pm 0.23}$	$38.79_{\pm 0.14}$	$41.36_{\pm 0.06}$	$37.55_{\pm 0.09}$	$54.67_{\pm 0.00}$	$62.35_{\pm 0.01}$	$\mathbf{64.58_{\pm 0.23}}$
	$s = 5$	$29.66_{\pm 0.19}$	$29.17_{\pm 0.16}$	$29.24_{\pm 0.30}$	$35.18_{\pm 0.18}$	$29.01_{\pm 0.14}$	$54.67_{\pm 0.00}$	$62.35_{\pm 0.01}$	$\mathbf{64.58_{\pm 0.23}}$



(a) CIFAR-100 (b) Tiny-ImageNet

Figure 4. Comparison of accuracy and communication rounds for analytic federated methods. SAFLe (ours) achieves the highest accuracy with the lowest rounds.



(a) CIFAR-100 (b) Tiny-ImageNet

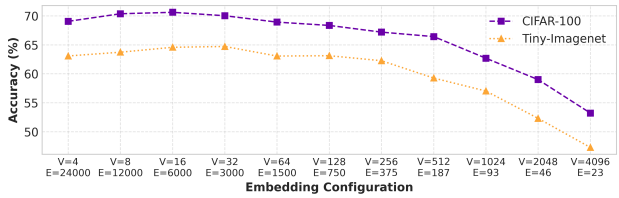
Figure 5. Accuracy over increasing numbers of clients (K). The proposed SAFLe is mathematically invariant to the number of clients, similar to AFL, whereas FedAvg's performance declines as K increases.

the final hidden layer head of the other SGD baseline methods, which produces no meaningful change in accuracy.

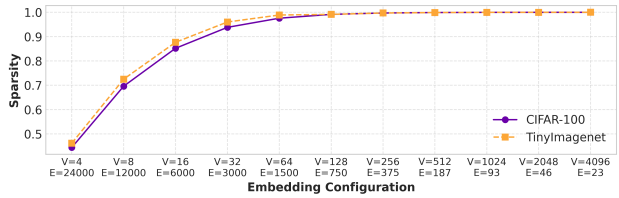
Training Time Comparison. All experiments were run on an NVIDIA RTX 4090 GPU, matching the hardware used in the prior analytic works [3, 7]. DeepAFL (Figure 3 in [3]) reports total training times of 130.7s, 145.7s, and 190.0s for its $T = 5, 10, 20$ models on Tiny-ImageNet. To reach the same accuracy levels (60.3%, 61.4%, 62.4%), our SAFLe models are computationally faster, taking 114s, 128s, and 143s, respectively.

Ablation: Embedding Sparsity Our communication cost efficiency is enabled by a trade-off between accuracy and the sparsity of the communicated correlation matrix. Figure 6 demonstrates this through an ablation where, for a fixed model size, we vary the embedding layer configuration, that is, the number of embeddings E and vocabulary size V. Configurations with a large group of small embeddings (high E, low V) achieve the best accuracy, which drops as we shift to fewer, larger embeddings (low E, high V). In contrast, the sparsity of the correlation matrix is inversely proportional. A layer with many small embeddings tends to be more dense, and the matrix quickly becomes very sparse as the embedding vocabulary size V grows. The ideal trade-off is the point that balances high accuracy with high sparsity (low communication cost), which we found to be around a vocabulary size of $V \in [32, 64]$.

Ablation: Bucketing Strategy The choice of bucketing function, which discretizes the continuous backbone features, is critical. We compare three strategies in Table 2: Standard (Integer), One-Hot, and Binary Overlapping. The results show that Binary Overlapping is the best-performing method, while standard Integer quantization is by far the worst. We attribute this to how each method handles local smoothness. Integer quantization creates a "cliff effect": a small change in an input feature that crosses a bin boundary results in a completely different index, leading to a drastic change in the looked-up embeddings. This prevents the model from generalizing to similar, nearby inputs. One-Hot binning is slightly better but still maps similar inputs to disjoint representations. The Binary Overlapping method excels because, by design, close input values share similar bit patterns. This allows the model to index and learn simi-



(a) Embedding Configuration vs Accuracy



(b) Embedding Configuration vs Sparsity

Figure 6. Ablation study of embedding layer configurations (x-axis in $\log_2$ scale). We vary the number of embeddings (E) and their vocabulary size (V) while keeping total model size fixed. (a) Many small embeddings (high E, low V) generalize best, whereas fewer large embeddings (low E, high V) overfit and yield lower accuracy. (b) Many small embeddings produce dense correlation matrices with high communication cost, while fewer large embeddings result in sparser, more communication-efficient matrices.

Table 2. Ablation study on bucketing strategies. Accuracy is reported across different numbers of buckets (B_n). Top accuracy in each setting is shown in green, and for each dataset in bold.

Dataset	B_n	Integer	OneHot	Binary Overlap
CIFAR-10	2	$87.16_{\pm 0.00}$	$87.54_{\pm 0.27}$	$88.43_{\pm 0.12}$
	8	$78.15_{\pm 0.00}$	$81.56_{\pm 0.21}$	$\mathbf{90.73_{\pm 0.08}}$
	14	$67.53_{\pm 0.00}$	$74.43_{\pm 0.17}$	$90.03_{\pm 0.07}$
	20	$59.75_{\pm 0.00}$	$68.27_{\pm 0.15}$	$89.12_{\pm 0.07}$
CIFAR-100	2	$67.72_{\pm 0.00}$	$68.22_{\pm 0.04}$	$68.82_{\pm 0.16}$
	8	$58.28_{\pm 0.00}$	$61.66_{\pm 0.04}$	$\mathbf{70.61_{\pm 0.12}}$
	14	$47.75_{\pm 0.00}$	$54.65_{\pm 0.04}$	$70.12_{\pm 0.14}$
	20	$40.09_{\pm 0.00}$	$48.77_{\pm 0.04}$	$69.35_{\pm 0.23}$
Tiny ImageNet	2	$61.38_{\pm 0.00}$	$61.78_{\pm 0.01}$	$62.43_{\pm 0.27}$
	8	$52.10_{\pm 0.00}$	$55.70_{\pm 0.01}$	$\mathbf{64.58_{\pm 0.23}}$
	14	$41.89_{\pm 0.00}$	$48.66_{\pm 0.01}$	$64.12_{\pm 0.15}$
	20	$34.70_{\pm 0.00}$	$43.22_{\pm 0.01}$	$63.93_{\pm 0.21}$

lar patterns on the embeddings for similar inputs, leading to significantly better generalization.

Ablation: Different Backbones. To demonstrate that our method's advantages are general and not specific to one backbone, we follow the ablation protocol from the AFL paper [7] and compare SAFLe against AFL using the same three pre-trained and frozen backbones: ResNet-18 [6], VGG11 [17], and ViT-B-16 [5]. The DeepAFL paper [3] does not include results for this specific experiment, and their code is not yet publicly available. The results in Table 3 show that SAFLe consistently outperforms AFL across all backbones and datasets. While a stronger backbone like ViT-B-16 significantly increases the performance of both methods [7], our non-linear analytic head still provides a clear and consistent accuracy gain over AFL's linear head. For instance, on CIFAR-100, SAFLe improves AFL's accuracy from 58.56% to 70.61% with ResNet-18, and from 75.45% to 77.83% with ViT-B-16. This confirms that the expressive power of our non-linear architecture provides a robust benefit, independent of the initial feature quality.

Table 3. Ablation study of Top-1 accuracy (%) using different backbones.

Dataset	Method	ResNet-18	VGG11	ViT-B-16
CIFAR-10	AFL	80.75	82.72	93.92
	SAFLe	**90.73**	**87.50**	**95.31**
CIFAR-100	AFL	58.56	60.43	75.45
	SAFLe	**70.61**	**62.68**	**77.83**
Tiny-ImageNet	AFL	54.67	54.73	82.02
	SAFLe	**64.58**	**58.34**	**82.71**

5. Conclusion

In this work, we addressed the fundamental trade-off in analytic federated learning between single-round efficiency and non-linear expressive power. We proposed SAFLe, a framework that breaks this trade-off, achieving high model expressivity while retaining the single-shot, gradient-free properties of AFL.

Our method introduces a structured non-linear head using feature bucketing and sparse, grouped embeddings. We proved that this complex architecture is mathematically equivalent to a high-dimensional linear regression. This key theoretical insight allows SAFLe to be the first non-linear framework to be solved using AFL's single-round Absolute Aggregation law, ensuring mathematical invariance to data heterogeneity.

Our experiments demonstrate that SAFLe establishes a new state-of-the-art for analytic FL, outperforming both the linear AFL and the multi-round DeepAFL in accuracy across all benchmark datasets. By delivering high accuracy, true single-shot efficiency, and robustness to non-IID data, SAFLe provides a practical, efficient and scalable solution for real-world federated vision tasks.

Acknowledgements

This research was supported by Semiconductor Research Corporation (SRC) Task 3148.001, National Science Foundation (NSF) Grants #2326894, #2425655 (supported in part by the federal agency and Intel, Micron, Samsung, and Ericsson through the FuSe2 program), NVIDIA Applied Research Accelerator Program, and compute resources on

the Vista GPU cluster through CGAI & TACC at UT Austin. This work is supported in part by the NSF grant CCF-2531882 and a UT Cockrell School of Engineering Doctoral Fellowship. Any opinions, findings, conclusions, or recommendations are those of the authors and not of the funding agencies.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 5

[2] Youssef Allouah, Akash Dhasade, Rachid Guerraoui, Nirupam Gupta, Anne-Marie Kermarrec, Rafael Pinot, Rafael Pires, and Rishi Sharma. Revisiting Ensembling in One-Shot Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2G024. 2

[3] Anonymous Authors. DeepAFL: Deep Analytic Federated Learning. In *Submitted to the International Conference on Learning Representations (ICLR)*, 2025. https://openreview.net/pdf?id=ve3EzAvMGe. 1, 2, 5, 6, 7, 8

[4] Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing One-Shot Federated Learning Through Data and Ensemble Co-Boosting. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2G016. 8

[7] Run He, Kai Tong, Di Fang, Han Sun, Ziqian Zeng, Haoran Li, Tianyi Chen, and Huiping Zhuang. AFL: A Single-Round Analytic Approach for Federated Learning with Pre-trained Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 5, 6, 7, 8

[8] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-Free One-Shot Federated Learning Under Very High Statistical Heterogeneity. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2

[9] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. 3

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[11] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[12] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the Global Knowledge by Not-True Distillation in Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 38461–38474, 2022. 2, 5

[13] Qinbin Li, Bingsheng He, and Dawn Song. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10713–10722, 2021. 1, 2, 5

[14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems (MLSys)*, pages 429–450, 2020. 1, 2, 5

[15] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, pages 2351–2363. Curran Associates, Inc., 2020. 5

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017. 1, 2, 5

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

[18] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7611–7623, 2020. 1, 2

[19] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature Communications*, 13(1): 2032, 2022. 2

[20] Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. FedDisco: Federated Learning with Discrepancy-Aware Collaboration. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 39879–39902. PMLR, 2023. 2

[21] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 12878–12889. PMLR, 2021. 2