Pre-Deployment Complexity Estimation for Federated Perception Systems

KMA Solaiman, Shafkat Islam, Ruy de Oliveira, Bharat Bhargava,

University of Maryland Baltimore County, Purdue University

ksolaima@umbc.edu

Lack of Pre-Deployment Task Difficulty Estimation in Federated Learning

Federated Learning for Perception Systems

- Distributed data across clients
- FedAvg aggregates models

?

Unknown before training

- Achievable Accuracy?
- Communication Cost?

Final Result of Poor Estimates

- More rounds
- Higher cost
- Lower accuracy

Difficulty depends on both data properties AND client participation

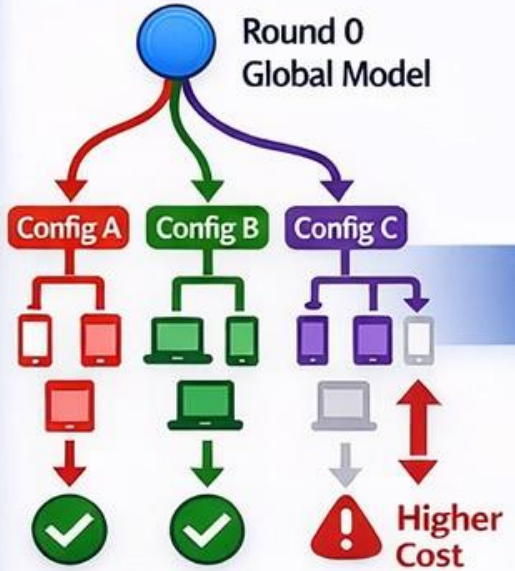
We need a pre-deployment metric for task difficulty — without breaking privacy

Gaps in Current Work vs Our Contribution

What Existing Approaches Lack

- **Centralized complexity measures only**
- **Classifier-dependent** *(need model training/loss)*
- **Post-training only** *(e.g., needs divergence, drops)*
- **Focus on optimization, not pre-training task difficulty estimation** [45]

No joint pre-deployment metric for data quality + participation trajectories [123]



Our Solution

$$F(d, X) = f(X) + f(d)$$

- First classifier-agnostic, pre-deployment metric
- Jointly captures intrinsic data + distributed client trajectories
- Additive, interpretable, predicts difficulty & cost upfront — No FedAvg run needed!

Our work gives the server the missing notion of combined distributed domain complexity

Pre-Deployment Complexity in Federated Learning

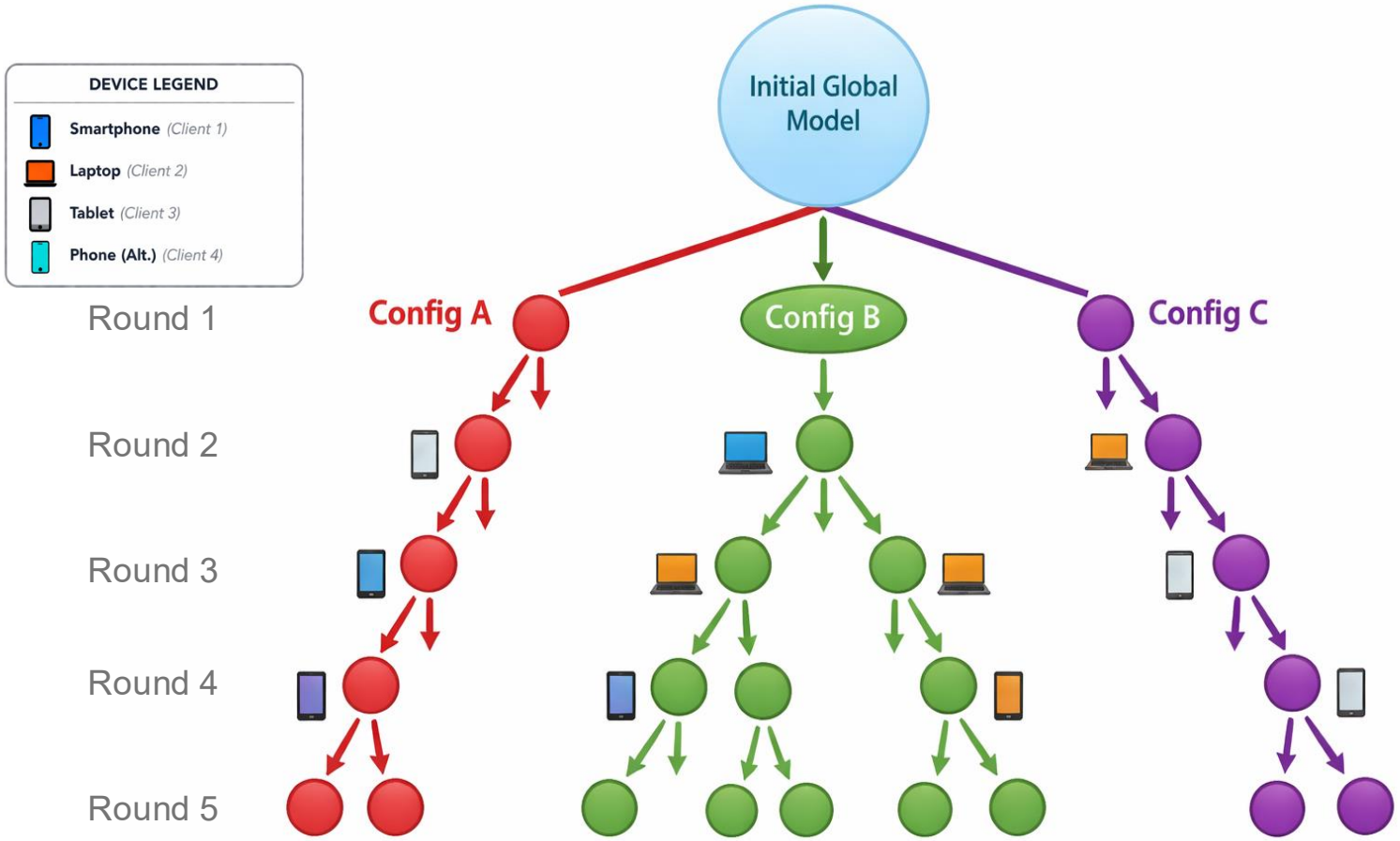


Fig. 1. Each path represents a distinct client participation sequence (training trajectory) → different expected difficulty

FL difficulty** comes from two sources: **data itself** and how it is **distributed

$f(X)$: **Intrinsic** Complexity
→ Data properties
→ Computable locally

$f(d)$: **Distributed** Complexity
→ Client participation across rounds
→ Captures system-level difficulty

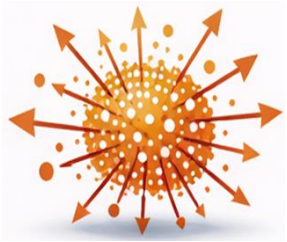
➔ **Unified pre-deployment** metric

- **Additive → interpretable diagnostic of FL difficulty**

Intrinsic Complexity $f(X)$

Intrinsic component of $F(d, X)$

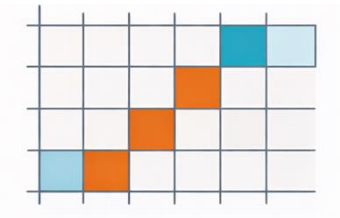
Dimensionality



Intrinsic Dimension (ID) Estimation

Higher ID → More complex manifold → harder to learn

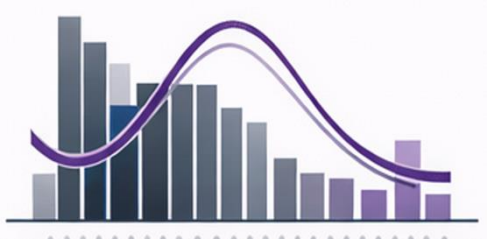
Sparsity



#PCA components for 80% variance

Fewer components → more **compressible** → easier task

Heterogeneity



Shannon Entropy (pixel distribution)

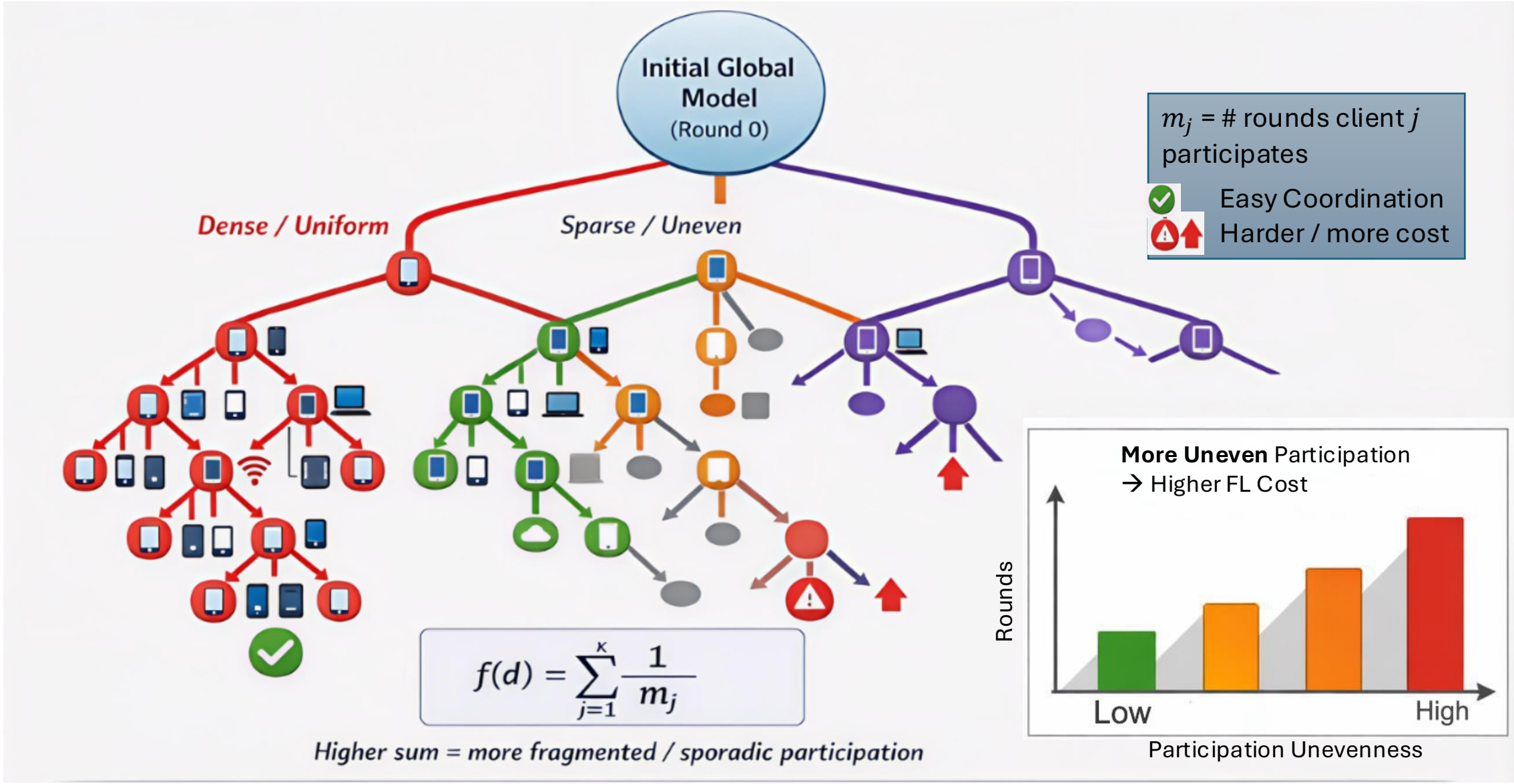
Higher entropy → more chaotic data → higher complexity

$$f(X) = \beta \|X\|_2$$

- Classifier-agnostic
- Computed directly from raw data
- No training required

Distributed Complexity $f(d)$

Extrinsic component of $F(d, X)$



Diagnostic: Predicts FL difficulty & cost upfront — NO FedAvg run needed!

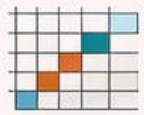
Towards Federated Complexity

- Harmonic form penalizes sporadic clients aggressively → captures coordination fragmentation
- Client participation estimated from:
  - historical participation logs (past FL jobs)
  - simulated dropout / availability patterns
  - planned client selection policy
- Diagnostic for $f(d)$: Predicts FL difficulty & cost upfront — NO FedAvg run needed!
- When environment is fixed → $f(X)$ dominates
- If data X is fixed but training paths vary → $f(d)$ dominated

The Unified Metric: $F(d, X)$

Intrinsic Complexity $f(X)$

Dimensionality



Sparsity

Heterogeneity

$\beta \cdot ||X||_2$

Distributed Complexity $f(d)$

Client participation sum = $\sum 1/m_j$

$$F(d, X) = f(X) + f(d)$$

$f(X)$: data complexity. $f(d)$: participation complexity

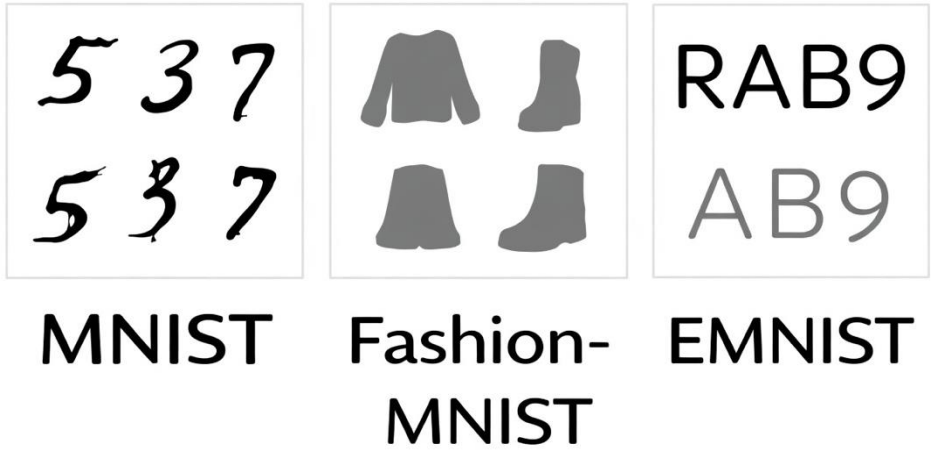
***Additive form**: orthogonal factors + interpretable diagnostic*

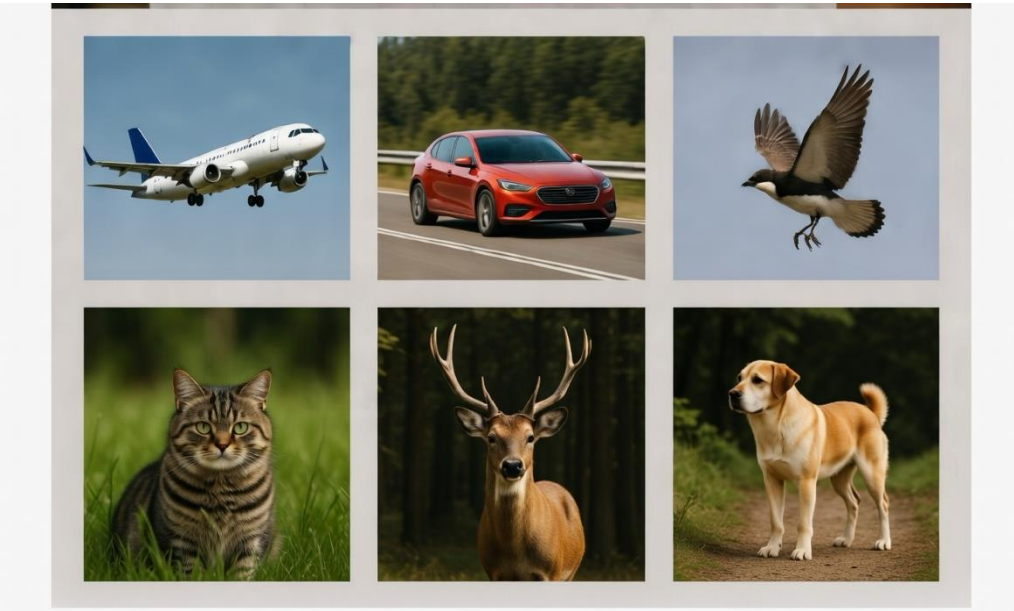
Low F → Easy FL task | **Medium F** → Moderate challenge | **High** → Hard FL task

Simple addition balances data difficulty with participation messiness → *powerful pre-deployment predictor*

Experimental Setup

- **MNIST Family**
  - Handwritten-MNIST, Fashion-MNIST, EMNIST
- **CIFAR-10**
  - converted to grayscale and resized to 28×28
- Dirichlet-based non-IID client partitioning *
- Federated Averaging (FedAvg), 100 communication rounds
- Shallow CNN (LeNet-style architecture)





FL Protocol & Metrics

- Metrics: Accuracy, Communication-Rounds-to-Threshold (Effort)
- FedAvg: Client participation varies per round (controlled via probabilistic participation)
- $f(X) = \|ID + Sparsity + heterogeneity\|_2$
- $f(d)$ computed from realized client participation during training
- Each participating client performs one full local epoch per round, following FedAvg protocol *

Q. Is complexity the primary driver of FL performance variation?

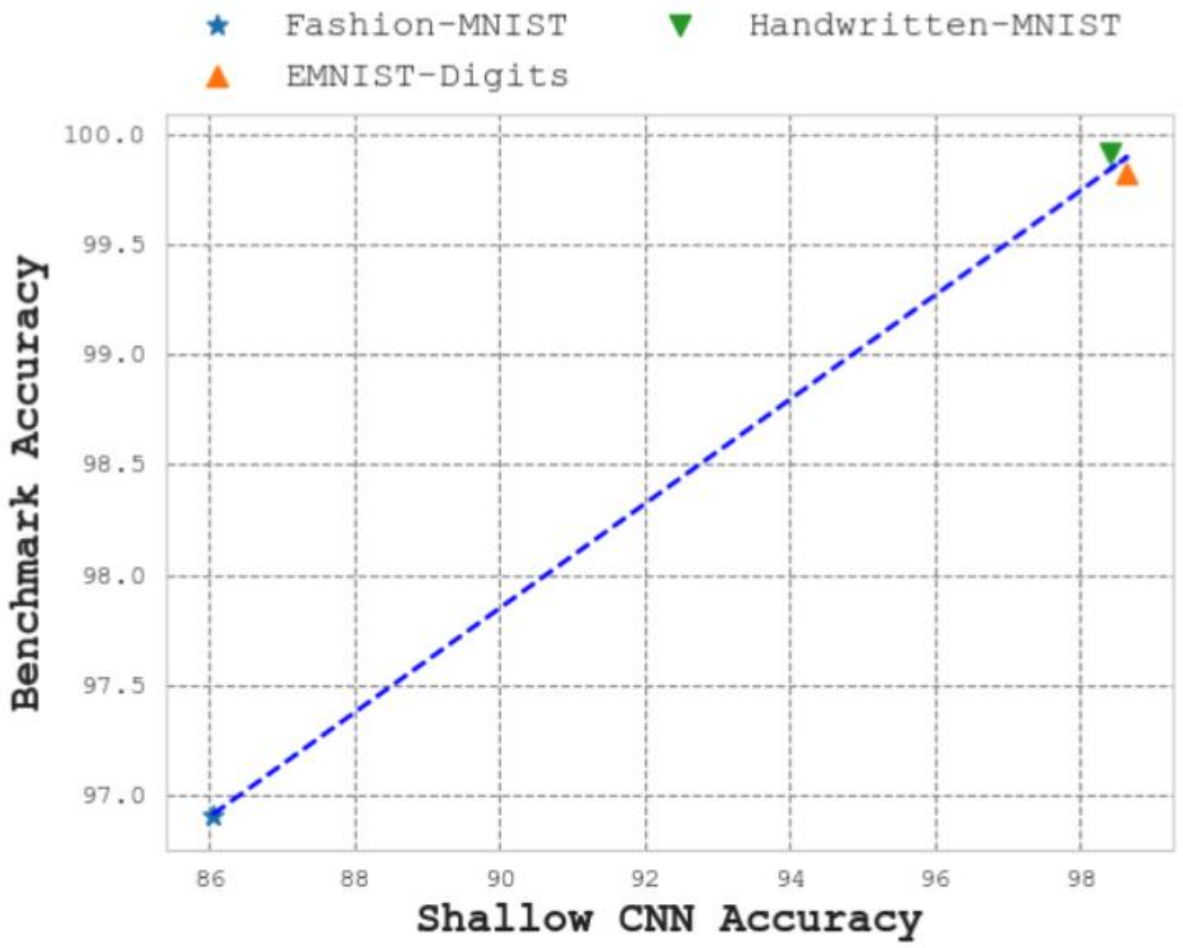
Enables prediction of FL difficulty BEFORE training

Why a Shallow Model is Sufficient

- Shallow CNN closely matches benchmark performance across datasets

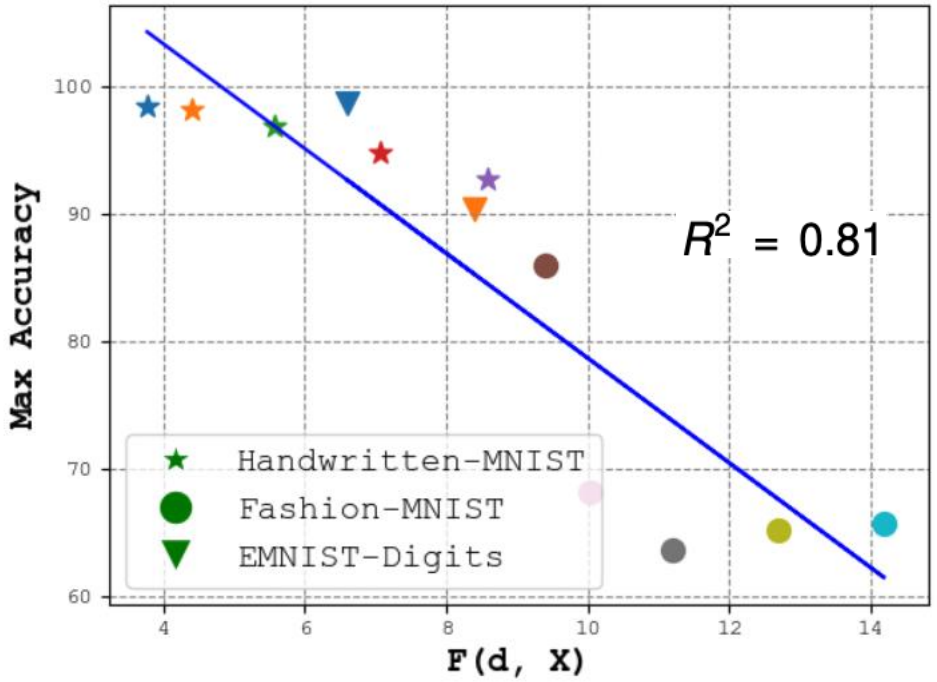
- Preserves dataset difficulty ordering (relative performance trends remain consistent)

- Allows isolating complexity effects without model confounds

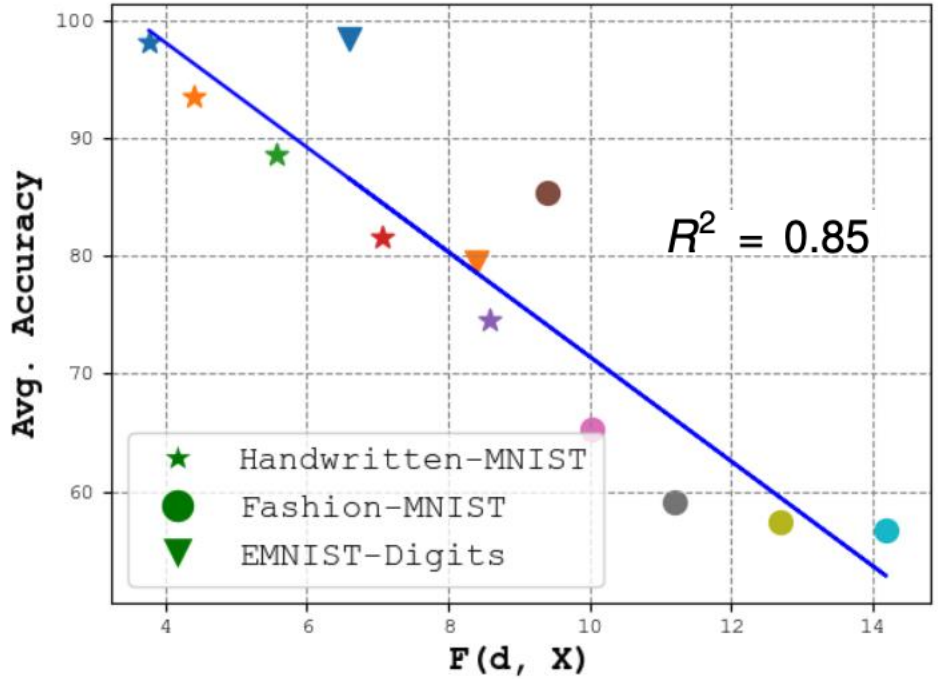


MNIST Validation:
Complexity Metric Predicts FL Performance

Strong negative correlation between F(d,X) and performance $R^2 \approx 0.81$, $R^2 \approx 0.85$ (Max & Avg. Acc.)



Low → high complexity

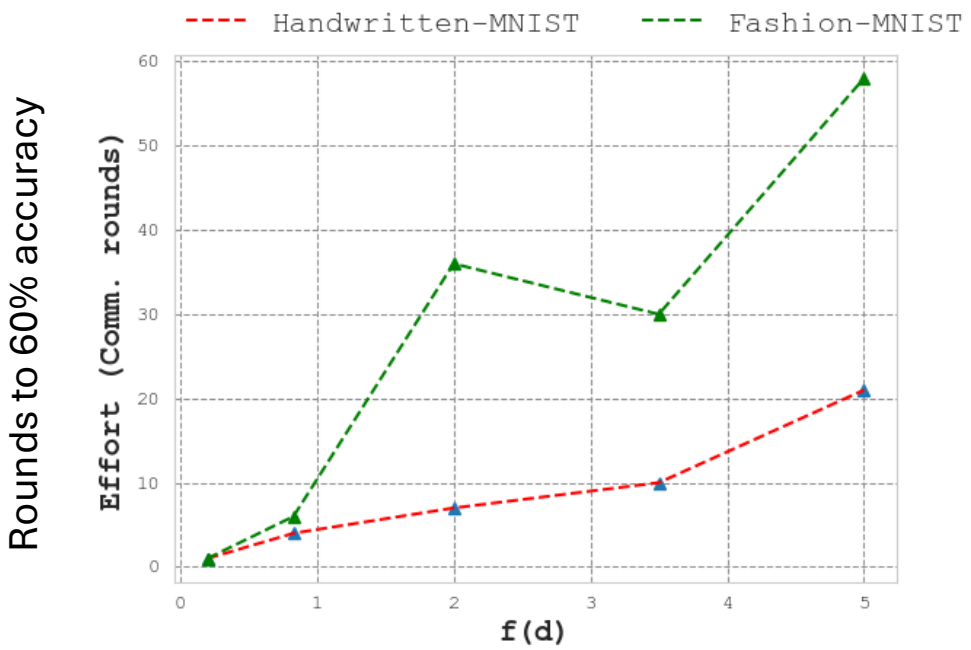


Low → high complexity

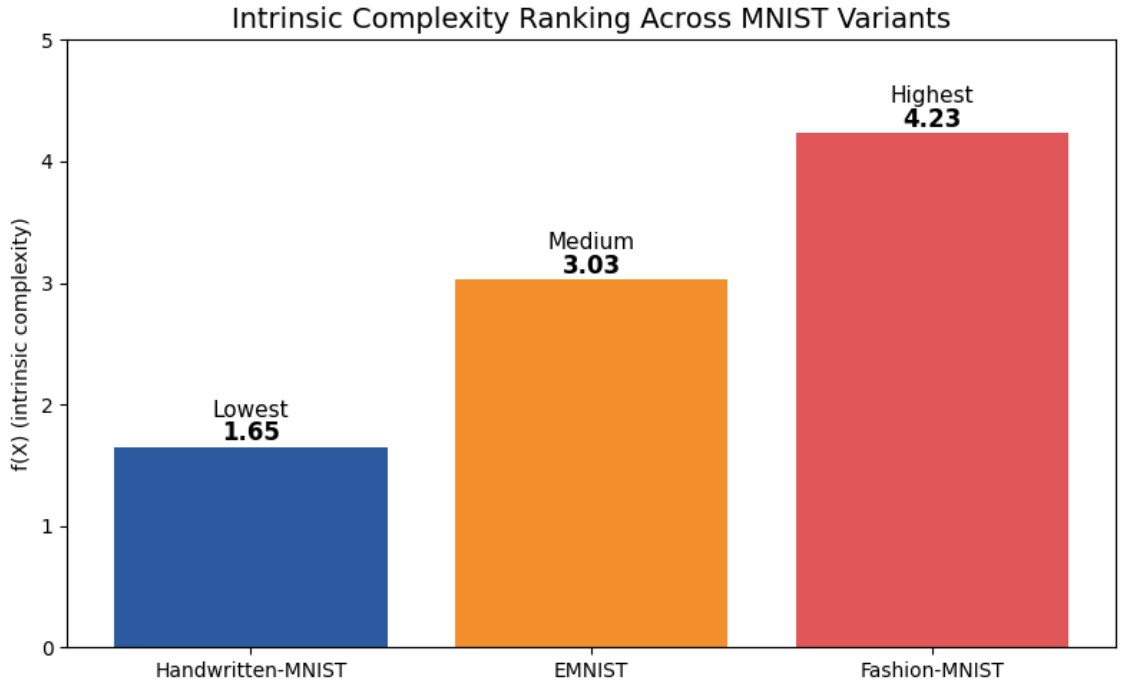
Higher Complexity → Lower Accuracy

Decomposing F(d, X): Distributed vs Intrinsic Effects

$$F\,(d, x)$$



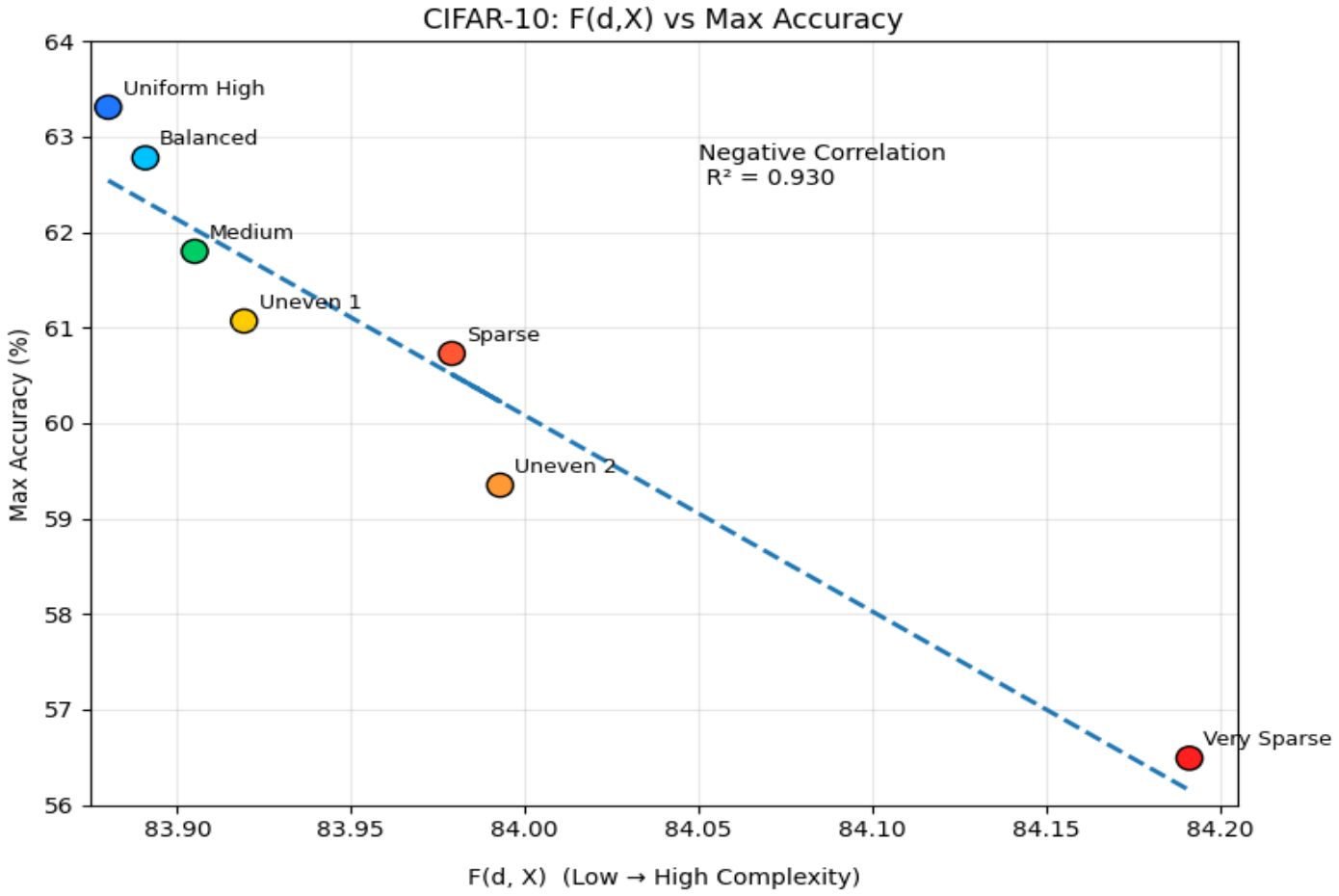
Higher participation imbalance + increasing number of clients → slower convergence

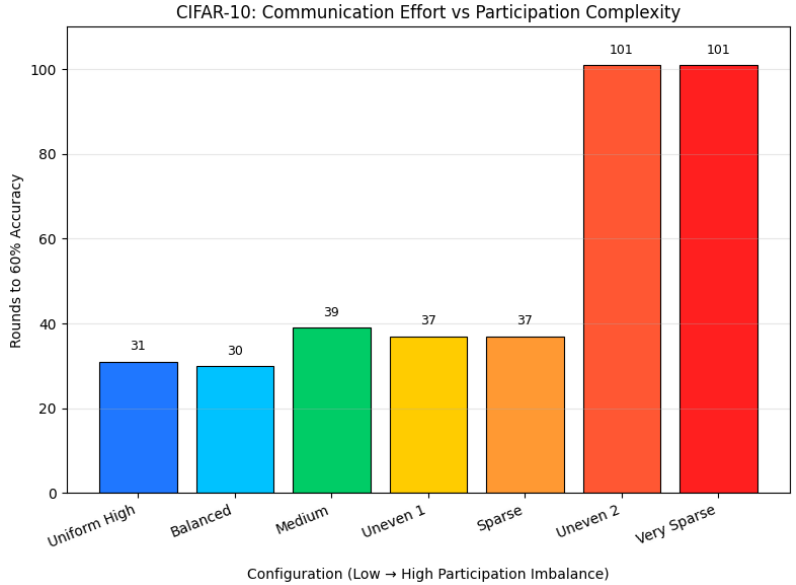


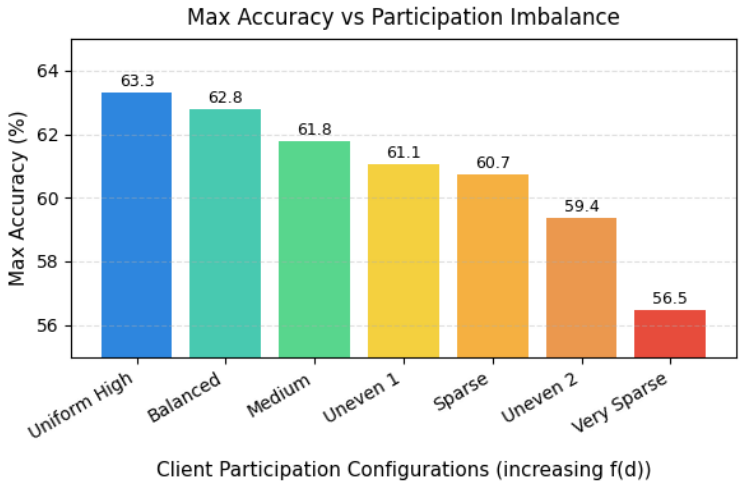
Intrinsic complexity ranking aligns with observed federated performance

f(X) alone predicts dataset difficulty ordering

CIFAR-10 Extended Validation: More complex dataset and varied distributed environment



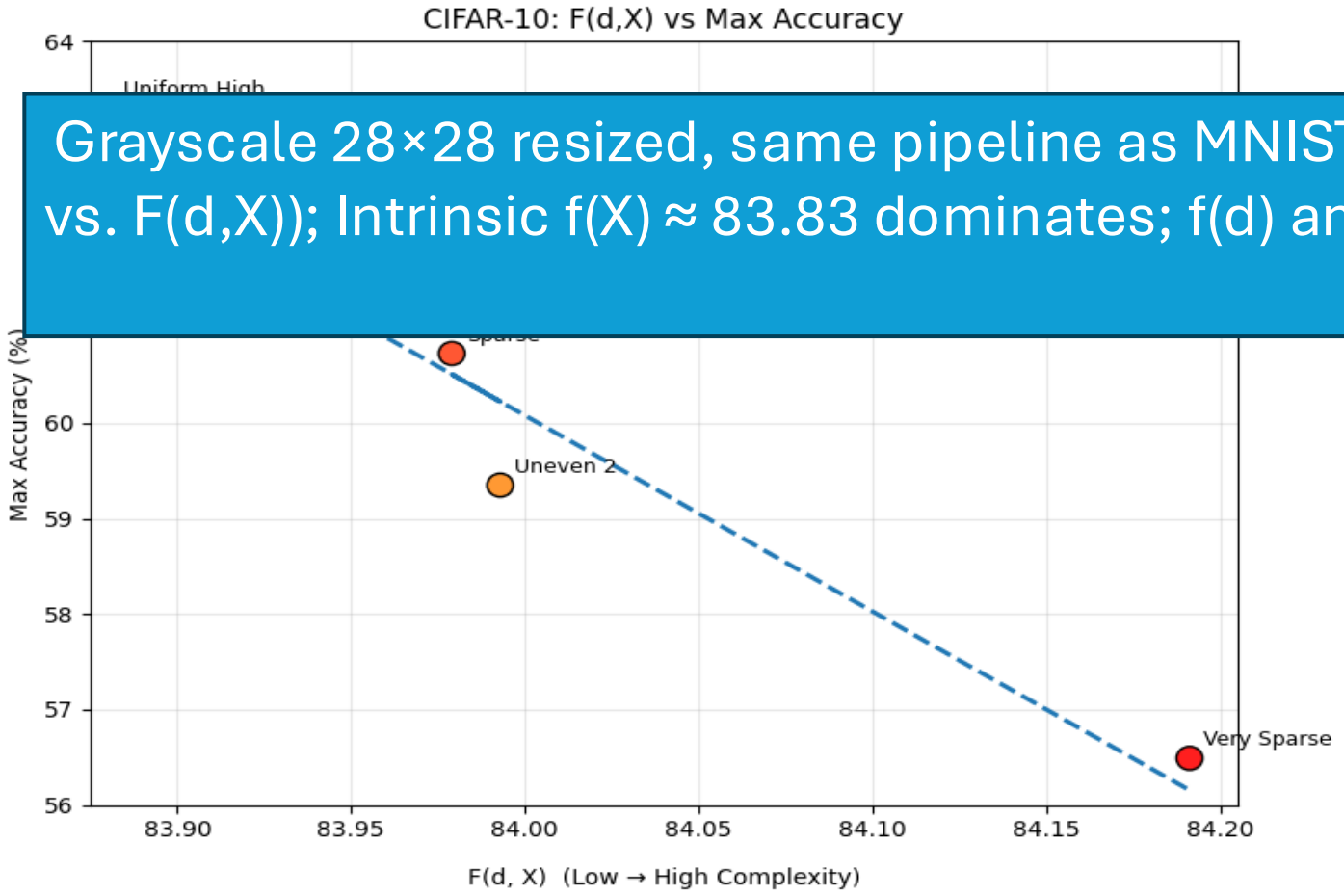


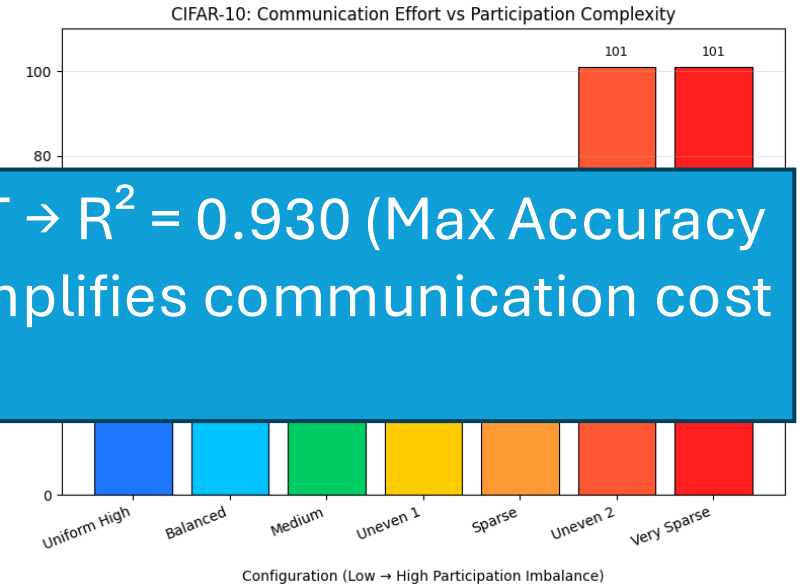


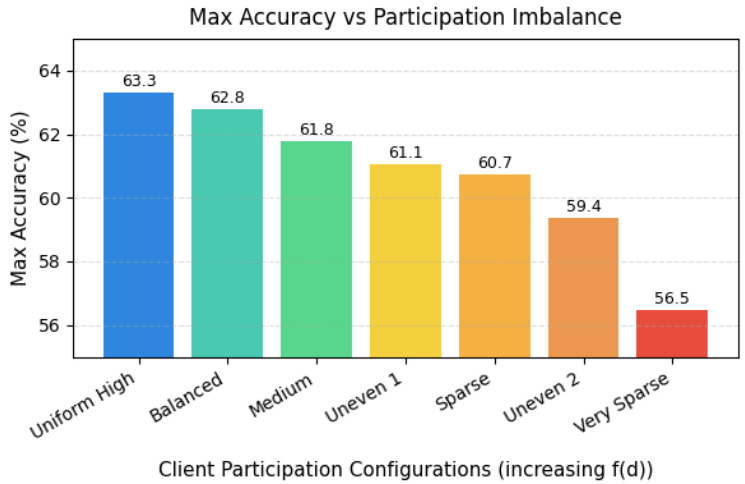
Higher complexity → lower accuracy & higher comms effort

CIFAR-10 Extended Validation: More complex dataset and varied distributed environment

Grayscale 28×28 resized, same pipeline as MNIST → $R^2 = 0.930$ (Max Accuracy vs. F(d,X)); Intrinsic f(X) ≈ 83.83 dominates; f(d) amplifies communication cost





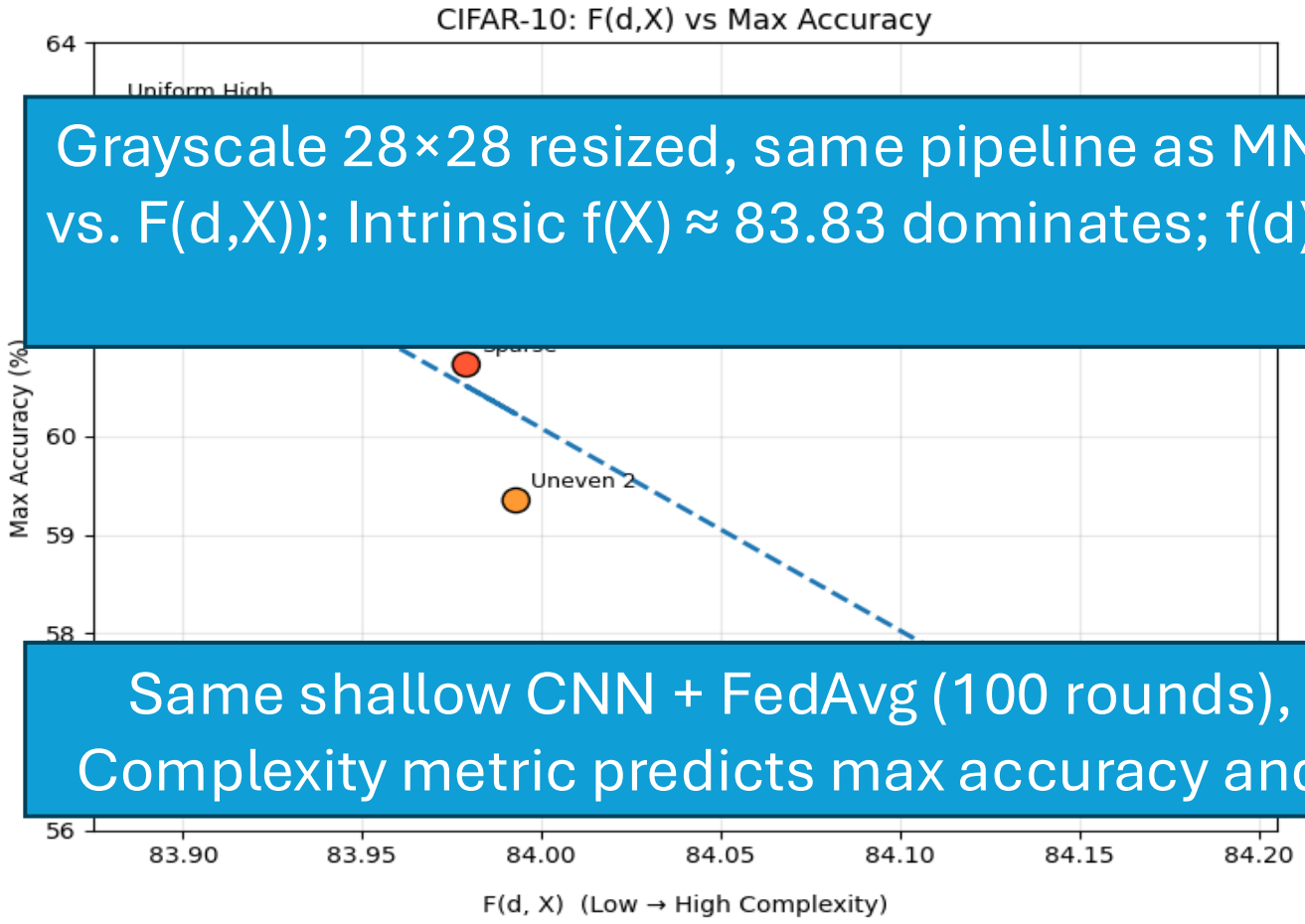


Higher complexity → lower accuracy & higher comms effort

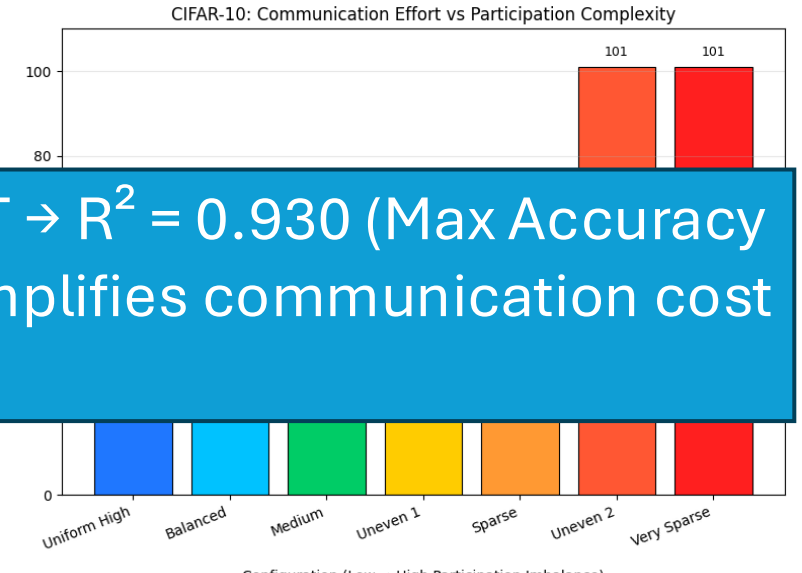
CIFAR-10 Extended Validation: More complex dataset and varied distributed environment

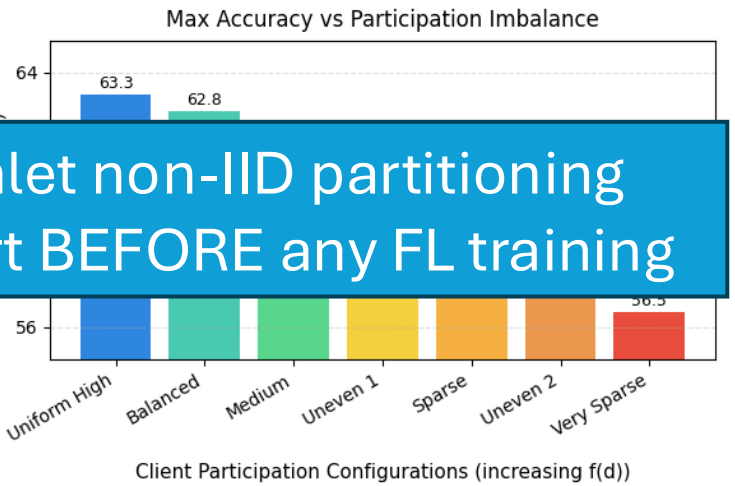
Grayscale 28×28 resized, same pipeline as MNIST → R^2 = 0.930 (Max Accuracy vs. F(d,X)); Intrinsic f(X) ≈ 83.83 dominates; f(d) amplifies communication cost

Same shallow CNN + FedAvg (100 rounds), Dirichlet non-IID partitioning
Complexity metric predicts max accuracy and effort BEFORE any FL training



Higher complexity → lower accuracy & higher comms effort





Conclusion and Future Directions

- **Predicts FL difficulty upfront:** resource-efficient planning in federated perception systems
- **Validated on perception tasks:** Strong correlations on MNIST-family ($R^2 \approx 0.81–0.85$) and CIFAR-10 ($R^2 = 0.930$) as complexity increases
- **Diagnostic tool, not an optimizer:**
  - Quantifies why a setup struggles
  - Guides better design choices before deployment

- Limited to low-to-medium complexity perception tasks (extend to real sensor streams, video perception)
- Explore full system heterogeneity (e.g., device compute/memory differences, dynamic dropouts, real noisy channels)
- Real-world FL deployment (asynchronous, partial participation beyond Dirichlet)
- Extend to prescriptive mode (for client selection, partitioning, or early stopping)

Thank You

- From trial-and-error training → predictable, pre-deployment AI.

- Strong diagnostic power for edge AI & federated perception pipelines

Questions?

H.A.R.M.O.N.I. Lab @ UMBC
https://ksolaiman.github.io/harmoni-lab/

References

1. Zhu, H., et al. (2021). Federated learning on non-IID data: A survey. *Neurocomputing*, 465, 1–21.
2. Kairouz, P., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1–210.
3. Li, T., et al. (2024). A comprehensive survey on client selection strategies in federated learning. *arXiv preprint arXiv:2405.12345*.
4. **Liu et al. (2024)** – “Recent advances on federated learning: A systematic survey” (Neurocomputing)
5. **Yang et al. (2024)** – “Review of Mathematical Optimization in Federated Learning” (arXiv:2412.01630)

Calculating F(x)

- These align better (both “stricter” / more realistic)
  - Sparsity → use **r^2 = 95%**
  - EC → use **EC_upper (v_θ = 90)**
- Use **min-max normalization for Sparsity and EC**
  - Even without this the relative ordering holds
- $\boldsymbol{\beta} = \mathbf{1}$

Experiment Results and Findings

- RQ1: Is there a complexity order for the MNIST datasets in terms of the proposed metrices in singular environment?

Table 7.1. Heterogeneity, Sparsity, Environment Complexity, and Intrinsic Dimensionality Measurement.

Dataset	Heterogeneity	Sparsity ($r^2 = 80\%$)	Sparsity ($r^2 = 95\%$)	$EC_{upper}(v_\theta = 0)$	$EC_{upper}(v_\theta = 90)$	ID
Handwritten-MNIST	1.60	740	629	717	530	13.368
EMNIST-digits	2.86	751	685	697	557	14.095
Fashion-MNIST	4.11	760	594	784	745	14.547

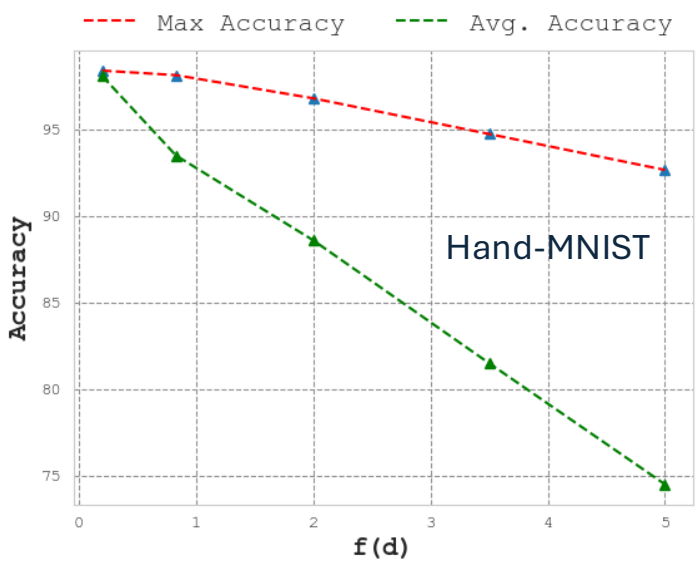
Findings:

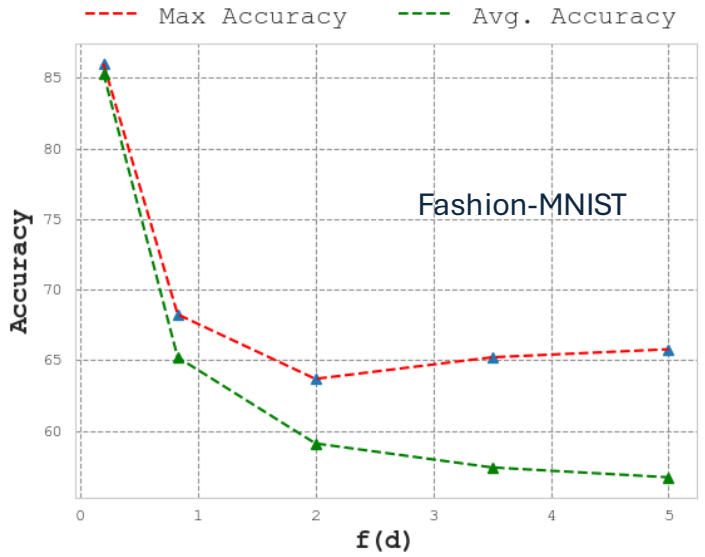
1. Handwritten- MNIST is the least complex and Fashion-MNIST is the most, seen by their variation, and EMNIST-digits is between both.
2. Sparsity at r^2 = 95% implies that Fashion-MNIST requires a lot of sparse components for explaining variances in between 80% and 95%.
3. At variance threshold = 0, Fashion-MNIST doesn't include many zeros over the data set, whereas if we consider pixel value 90 as threshold, it still has the largest environment complexity.

Experiment Results and Findings

- RQ2,3: How does Federated Environment Complexity ***f(d)*** effects the overall distributed learning complexity?

Five values of *d*: 1 – 5





Findings:

1. For easier handwritten-MNIST, accuracy is higher, and effort is lower than fashion-MNIST.
2. As the value of *f(d)* increases, accuracy decreases, in line with the benchmark classifier.
3. As *f(d)* increases, the effort increases, while the intrinsic features (at 60% var. thres.) [kept fixed] remains identical.