Sketchguard

Scaling Byzantine-Robust Decentralized Federated Learning via Sketch-Based Screening

Murtaza Rangwala, Farag Azzedin, Richard O. Sinnott and Rajkumar Buyya

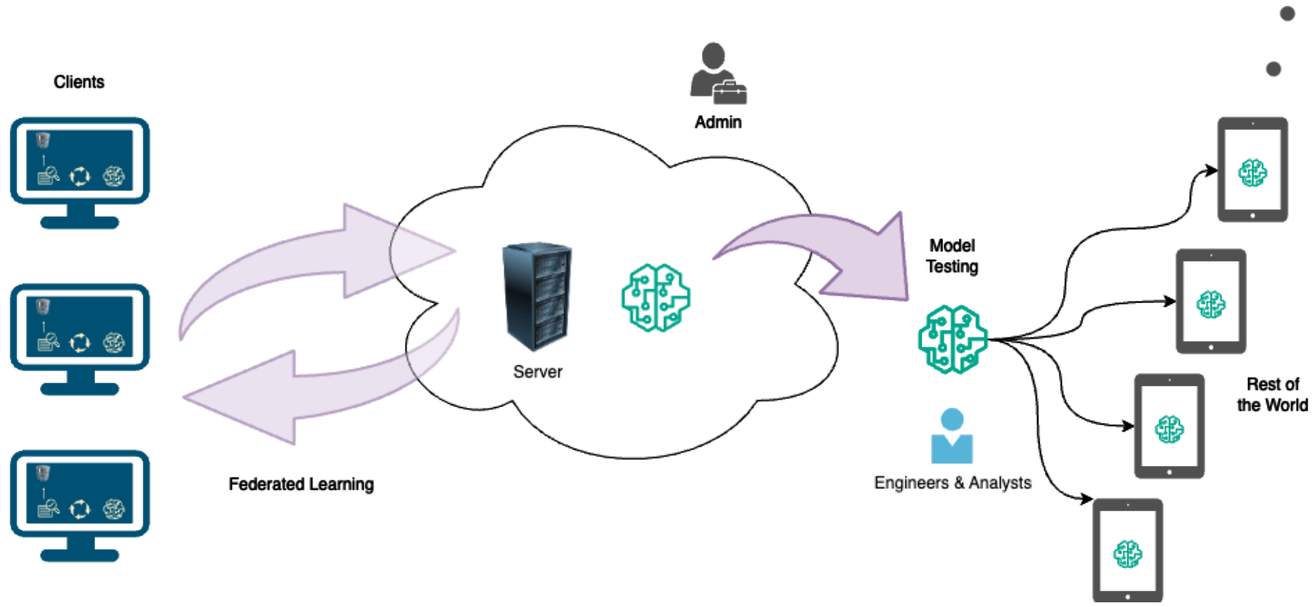
School of Computing and Information Systems
The University of Melbourne

1. Introduction to Federated Learning

Federated learning is a machine learning setting where **many clients** collaboratively **train a model** under the **orchestration of a central server**, while keeping the training **data decentralized**.

Iterative Protocol:

1. Eligible clients are selected by the server
2. Server sends the current model to selected clients
3. Clients train model on their private data
4. Server aggregates updates from each client
5. Global model is updated



2. Problem with Standard Federated Learning

Reliance on a Centralized Entity

Single Point of Failure

- System-wide disruption if coordinator fails
- Critical bottleneck for scaling
- Performance bounded by central node

Trust Requirements

- Participants must trust the central entity
- Coordinator has privileged position
- Potential for manipulation of global model

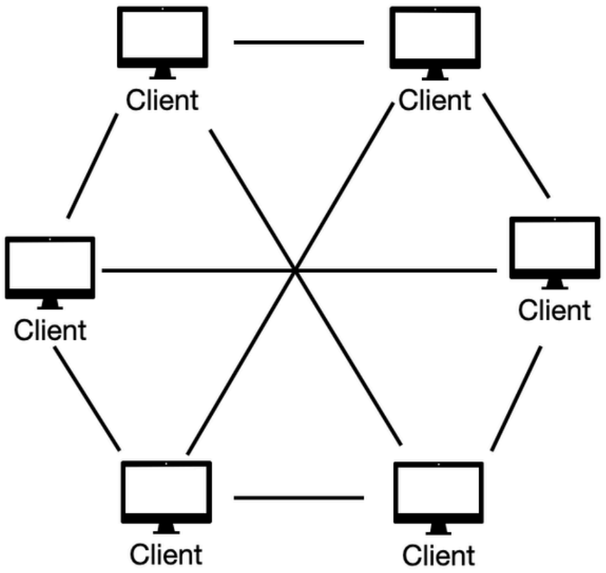
Regulatory Challenges

- Cross-border data governance issues
- Compliance with regional regulations
- Questions of model ownership

3. Decentralized or P2P Learning

Decentralized learning is a machine learning paradigm where **multiple independent nodes** train and share model updates in a peer-to-peer network **without any central coordinator or server.**

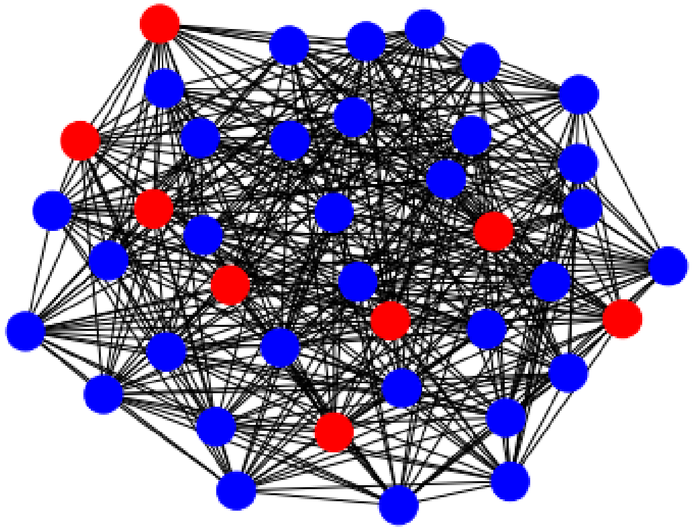
- **Communication topology:** connected graph with clients as nodes, edges as channels
- **Rounds:** local updates + neighbor information exchange
- **Updates:** gradient steps; **communication:** parameter averaging
- No global model state, but local models converge to global solution
- Central authority may still define the learning task



4. Core Challenges in Decentralization

Trust Establishment	• Establishing trust among distributed untrusting nodes
Resource Heterogeneity	• Dealing with nodes that have different computational power levels
Communication Efficiency	• Managing limited bandwidth for model updates
Model Convergence	• Achieving model consensus across distributed training nodes
Data Privacy	• Protecting node-specific data while enabling collaborative learning
Regulatory Compliance	• Navigating varied legal requirements across jurisdictions

5. The Trust Problem in DFL



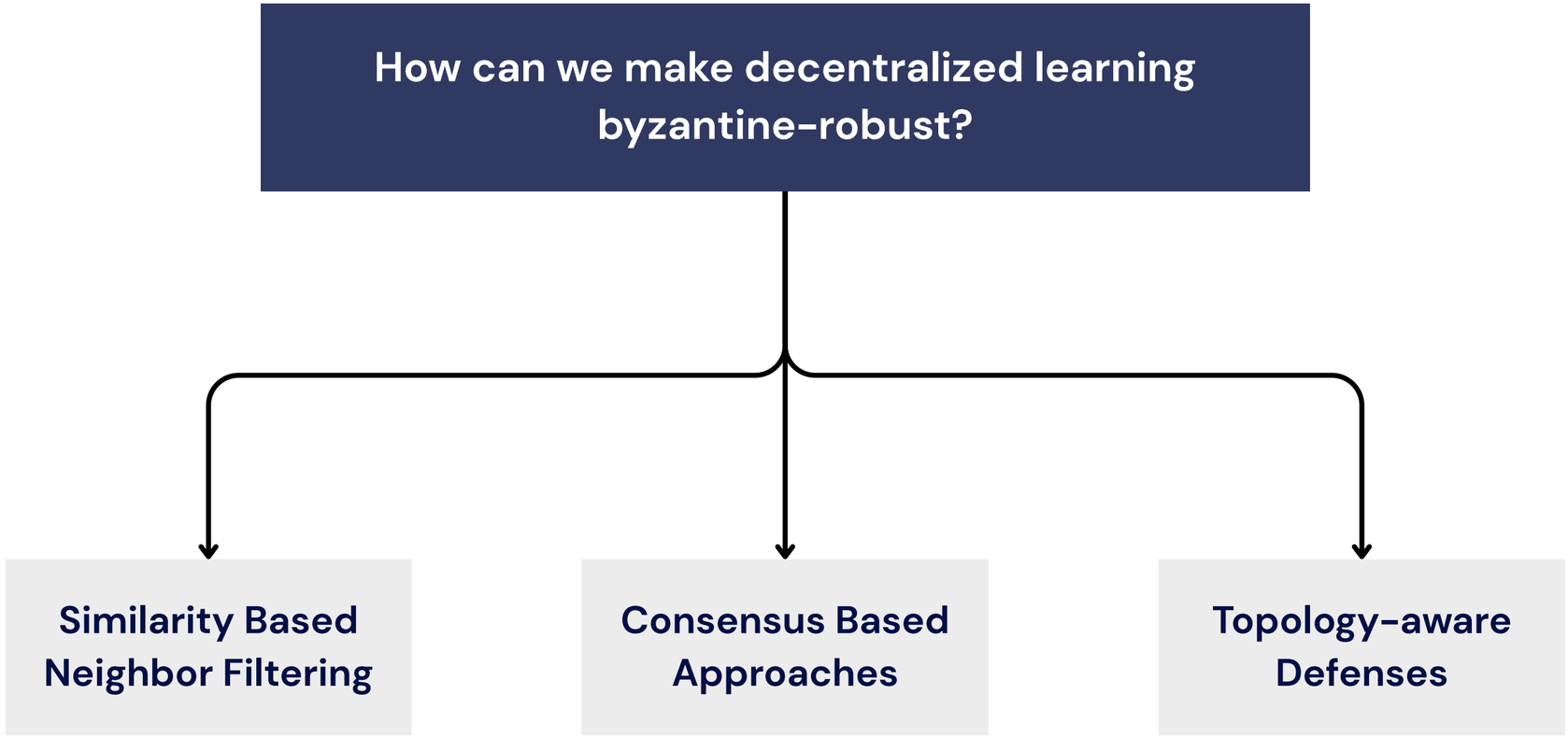
Untrustworthy nodes may be part of a DFL network

Each node needs to make a decision on which neighbors to trust

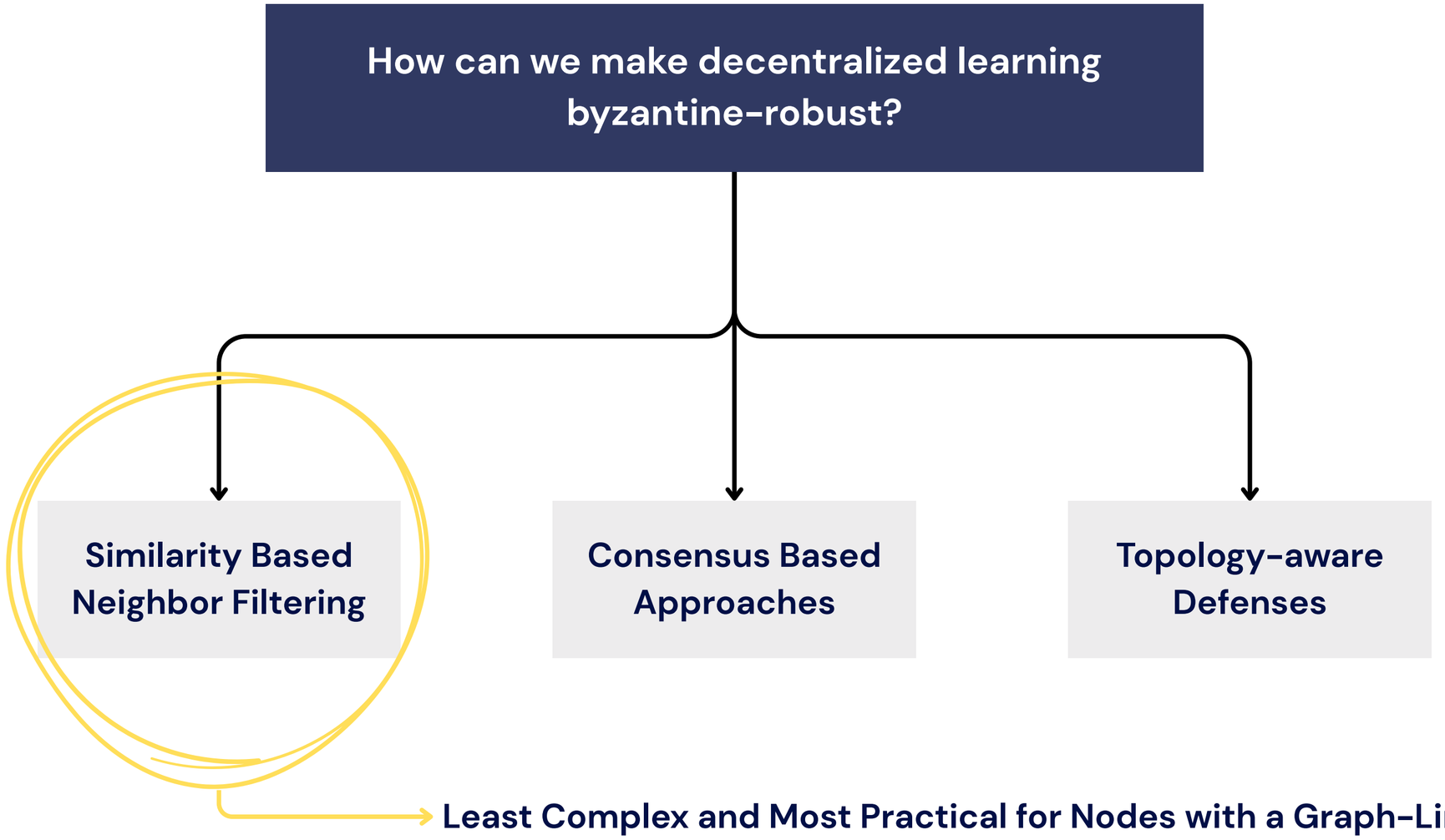
Wrong decisions can propogate bad updates across the network

In sparse network topologies, ignoring a malicious node can inadvertently isolate an honest subset of the network.

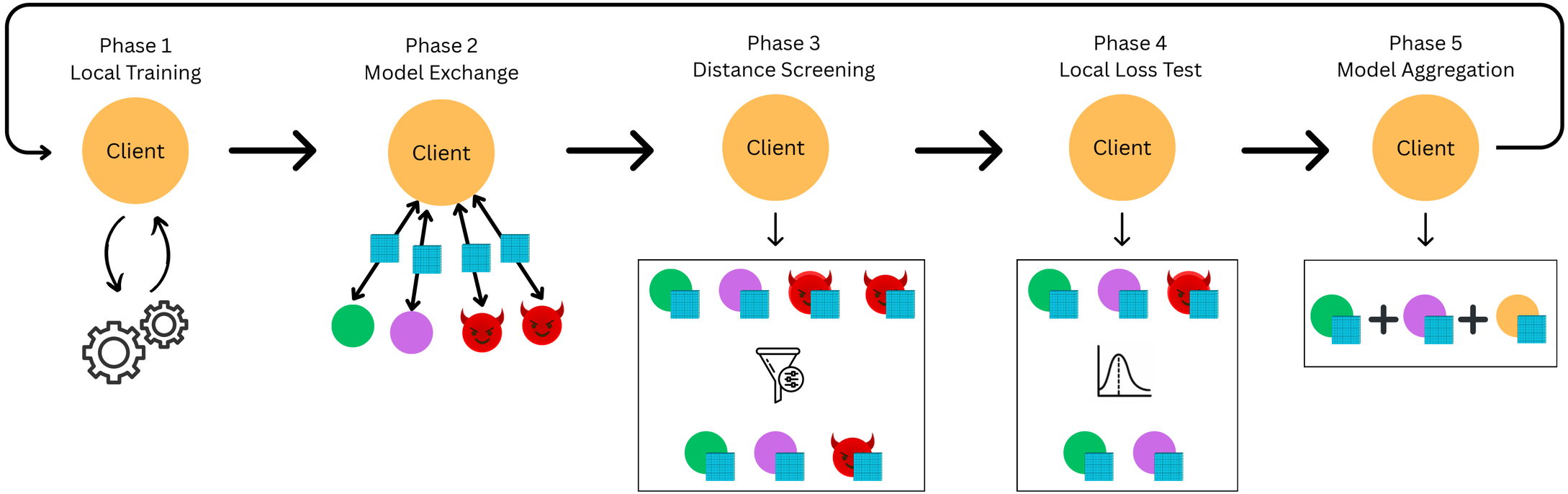
6. Existing Literature



6. Existing Literature



7. State of the Art – UBAR



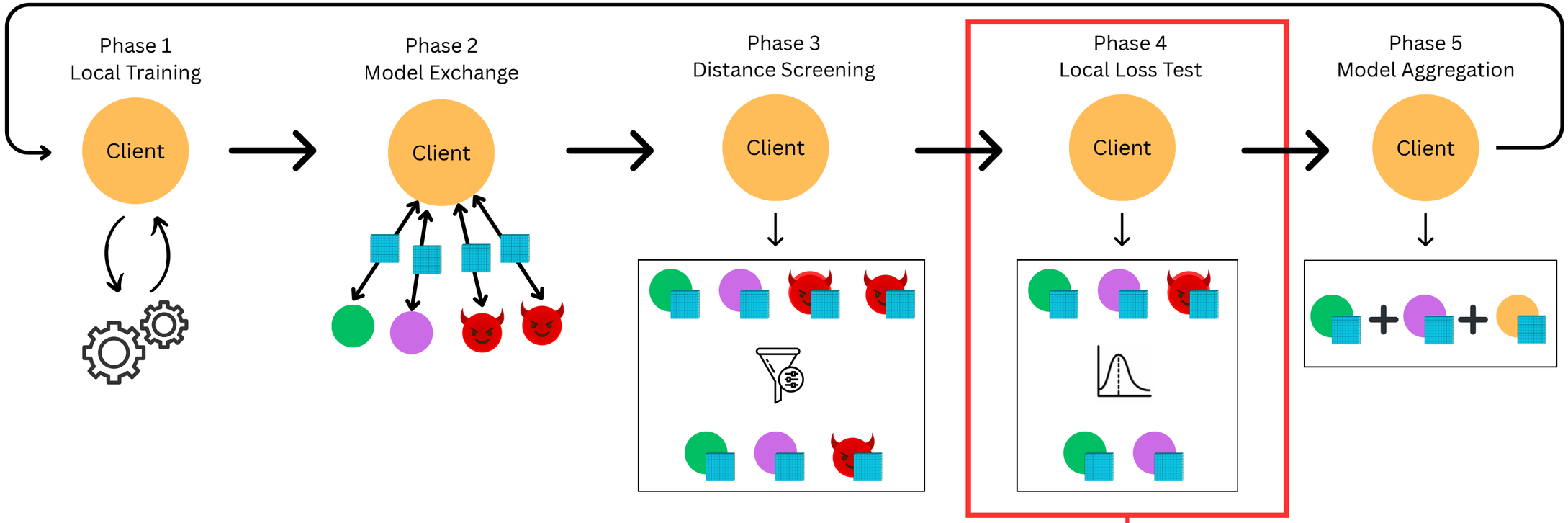
Distance Screening

Each client compares its model with neighbors and keeps only the closest subset. This fast, low-cost step removes distant or inconsistent models, cutting off obvious Byzantine outliers before deeper checks.

Local Loss Test

The shortlisted neighbors are re-evaluated using the client's own mini-batch. Only those whose models perform no worse than the client's local model are trusted for aggregation, ensuring robustness against subtle, data-aware attacks.

7. State of the Art – UBAR



Computationally Heavy and Slow

8. State of the Art – BALANCE

Excludes the **Local Loss Test**

Introduces a **dynamic thresholding mechanism** with **a decay factor**

$$\|w_i^{t+\frac{1}{2}} - w_j^{t+\frac{1}{2}}\| \leq \gamma \cdot \exp(-\kappa \cdot \lambda(t)) \|w_i^{t+\frac{1}{2}}\|$$

Upper limit for accepting a model as benign

Determines the rate at which the exponential function decreases; a larger κ results in a faster decay, while a smaller κ leads to a slower decay

A monotonically increasing and non-negative function associated with the training round index

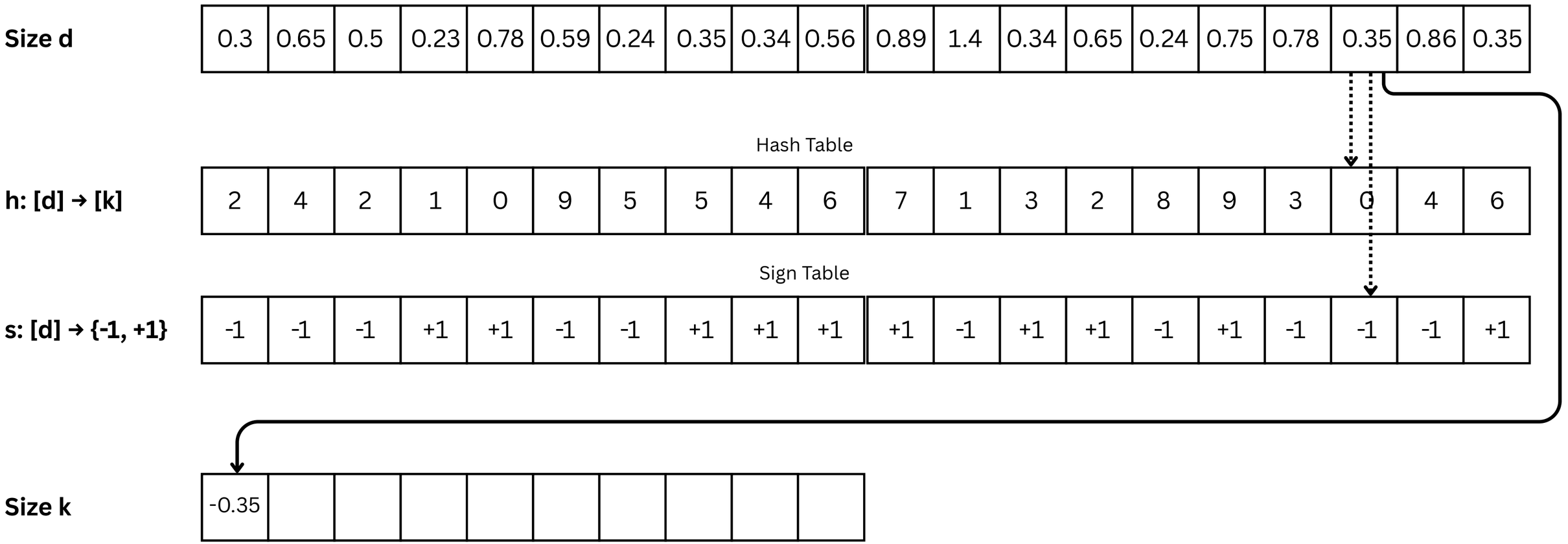
9. Scalability Bottleneck with SOTA

Fundamental Scalability Bottleneck

Clients exchange full model vectors with all neighbors in every round, even though many will later be discarded as malicious or irrelevant. This causes redundant communication and computation overhead.

10. An Introduction to Count Sketch

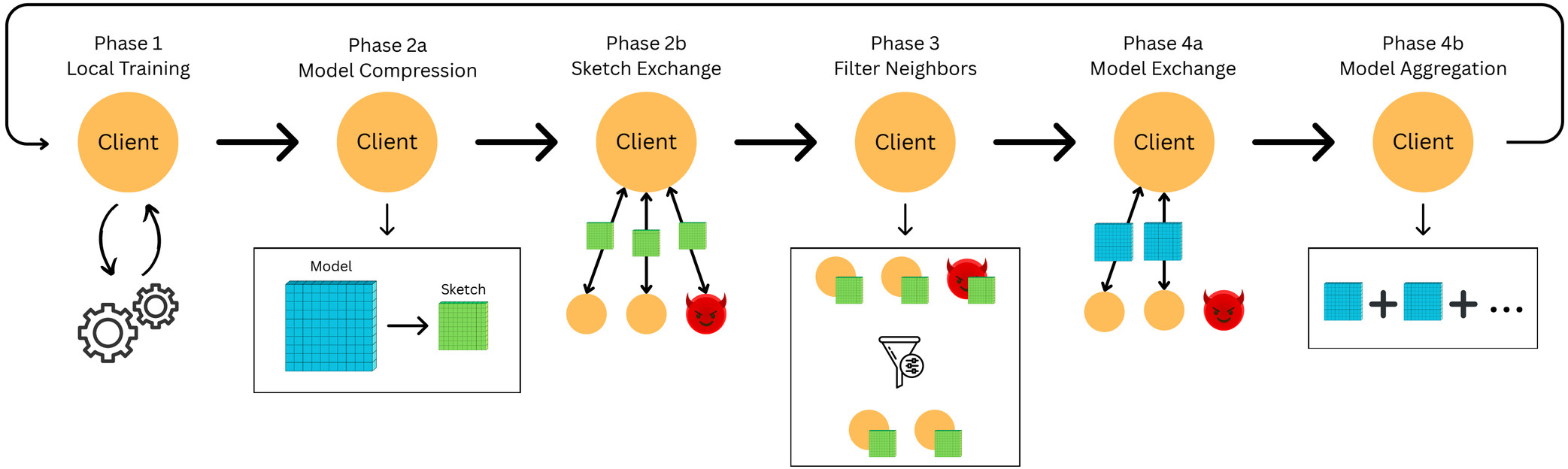
Count Sketch is a Dimensionality Reduction Technique



10. An Introduction to Count Sketch

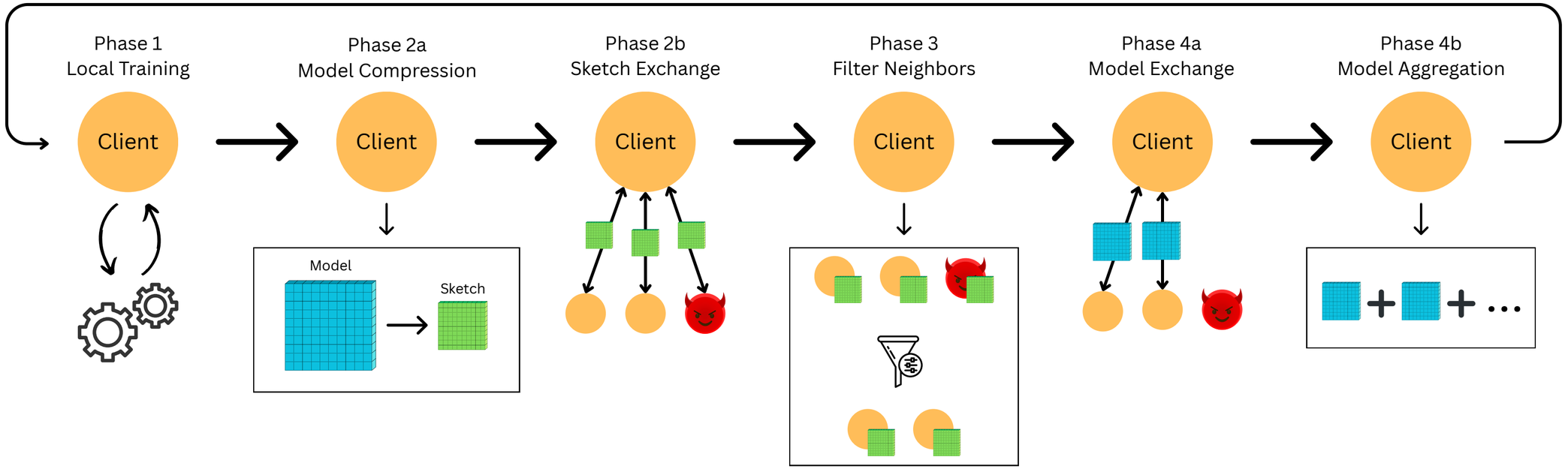
Properties of Count Sketch
They are Linear (CS(x + y) = CS(x) + CS(y))
Give unbiased estimates of true values or inner products
Works well even if vectors that are being compressed are sparse (many zeros)
Most Importantly, they preserve the Euclidean Distance of the original vector

11. Sketchguard



Key Insight: Similarity-based byzantine filtering can operate on compressed representations, while the final aggregation requires full precision models only for accepted neighbors.

11. Sketchguard



Phase 1: Client i performs local stochastic gradient descent

Phase 2: The updated model is compressed using count sketch, and these k-dimensional sketches are exchanged with the immediate neighbors

Phase 3: Neighbor j is accepted if their sketch distance satisfies: $\|\mathbf{s}_i^{t+1/2} - \mathbf{s}_j^{t+1/2}\| \leq \gamma \exp(-\kappa t/T)\|\mathbf{s}_i^{t+1/2}\|$

Phase 4: Full models are fetched from accepted neighbors. Before aggregation, each received model is verified by recomputing its sketch and comparing with the originally received sketch . Any neighbor whose model fails verification is removed from. The verified models are then aggregated: $\mathbf{w}_i^{t+1} = \alpha \mathbf{w}_i^{t+1/2} + \frac{(1-\alpha)}{|\mathcal{S}_i^t|} \sum_{j \in \mathcal{S}_i^t} \mathbf{w}_j^{t+1/2}$

12. Sketchguard Complexity vs SOTA

Training Phase	SOTA	SketchGuard
Local Training	$O(d)$	$O(d)$
Sketch Generation	–	$O(d)$
Neighbor Screening	$O(d \cdot \lvert\mathcal{N}_i\rvert)$	$O(k \cdot \lvert\mathcal{N}_i\rvert)$
Model Verification & Aggregation	$O(d \cdot \lvert\mathcal{S}_i^t\rvert)$	$O(d \cdot \lvert\mathcal{S}_i^t\rvert)$
Total Per Round	$O(d \cdot \lvert\mathcal{N}_i\rvert)$	$O(d+k \cdot \lvert\mathcal{N}_i\rvert+d \cdot \lvert\mathcal{S}_i^t\rvert)$

d Model Dimension (Size)

k Sketch Dimension (Size)

$|\mathcal{N}_i|$ Number of Neighbors

$|\mathcal{S}_i^t|$ Accepted neighbors in round t

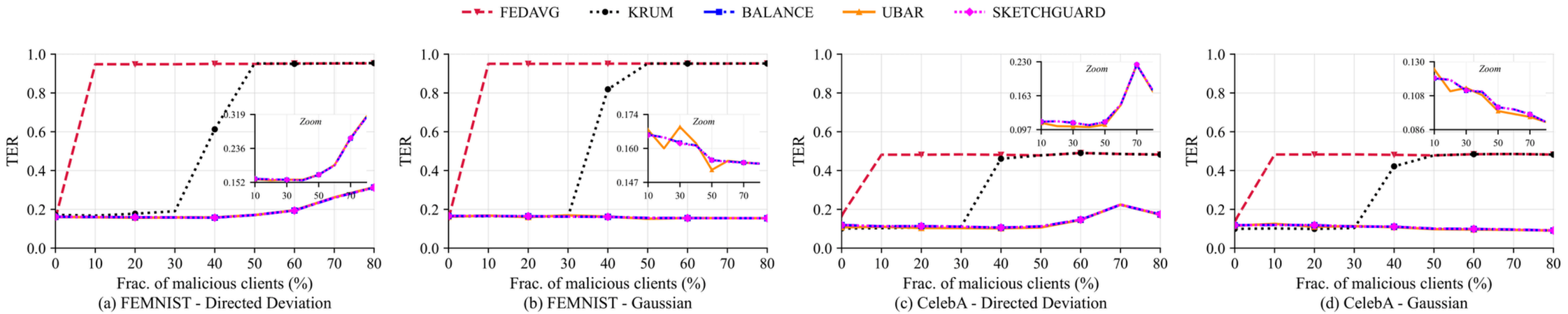
13. Sketchguard Robustness vs SOTA

Directed Deviation Attack

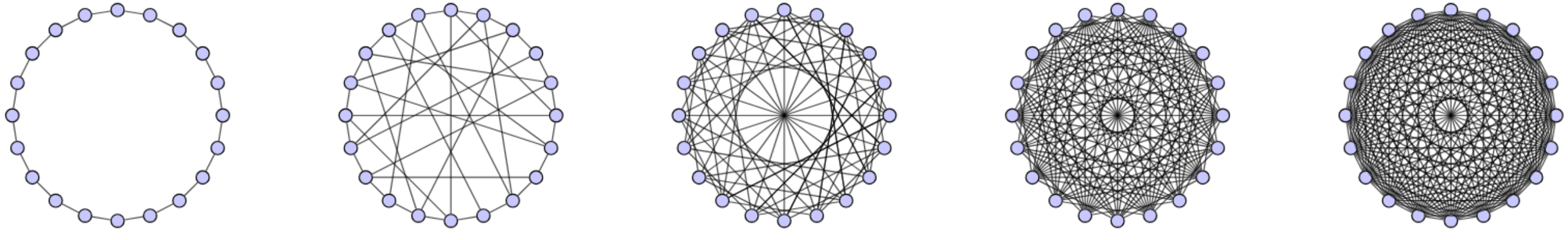
- These attacks employ an optimization-based approach that crafts malicious updates in the direction opposite to honest gradient descent.
- This adaptive attack strategy represents a strong adversary that actively attempts to subvert robust aggregation mechanisms.

Gaussian Attack

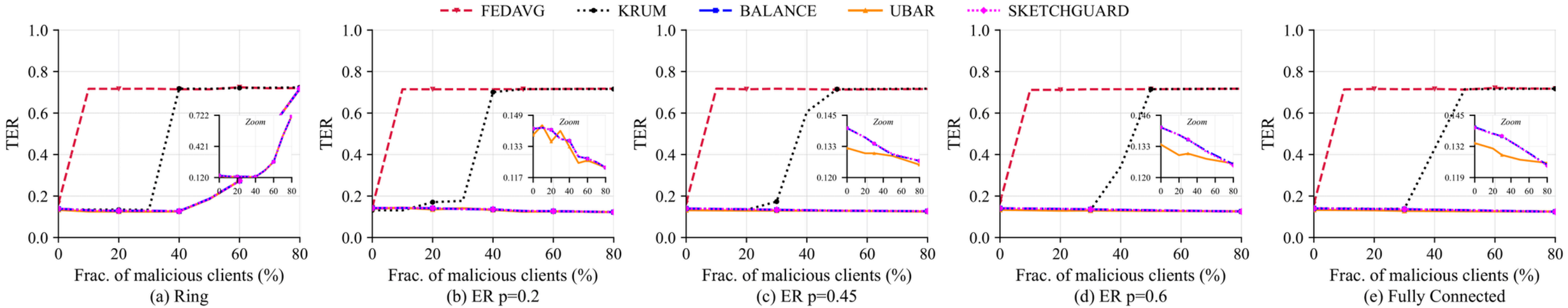
- These attacks inject random noise, representing less sophisticated but realistic adversaries that disrupt convergence through stochastic perturbations.
- These attacks model practical scenarios where attackers lack complete knowledge of the aggregation mechanism.



13. Sketchguard Robustness vs SOTA



(a) Ring (b) ER Sparse $p = 0.2$ (c) ER Medium $p = 0.45$ (d) ER Dense $p = 0.6$ (e) Fully Connected



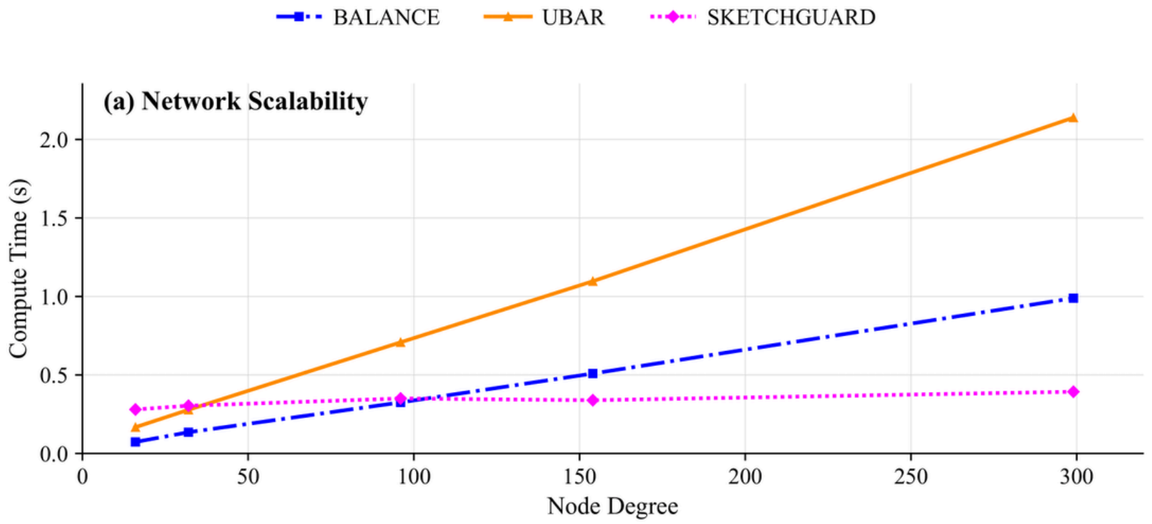
14. Sketchguard Scalability vs SOTA

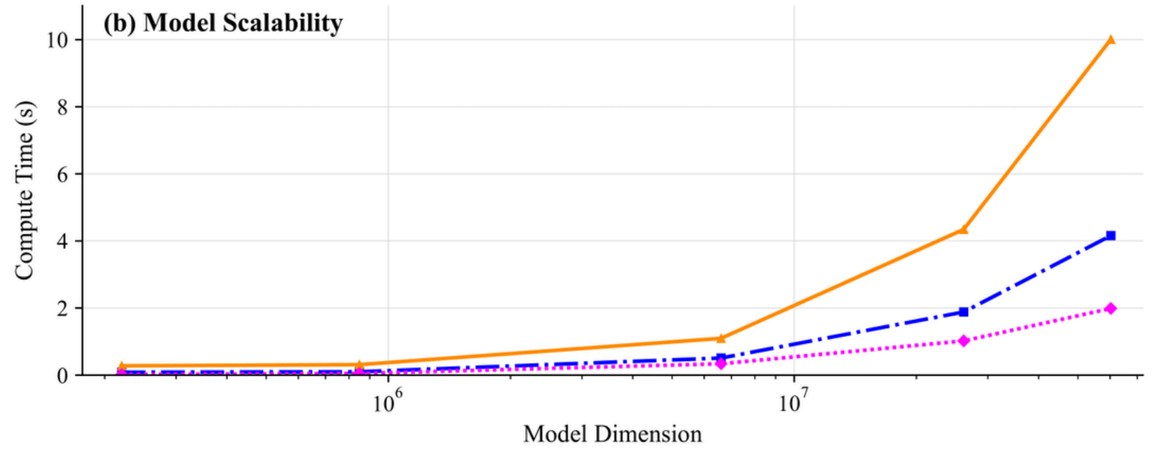
Network Scaling Experiment Configurations

Node Degree	Network Size	Attack Percentage
16	20	50%
32	35	50%
96	100	50%
154	155	50%
299	300	50%

Model Scaling Experiment Configurations

Variant	Architecture Modifications	Parameters
Tiny	Reduced filter counts and hidden units	220,318
Small	Standard configuration	848,382
Standard	Baseline FEMNIST architecture	6,603,710
Large	Increased filter counts and hidden units	26,154,814
XLarge	Further increased dimensions	60,271,678





14. Sketchguard Scalability vs SOTA

Communication efficiency gains of Sketchguard are directly proportional to the number of filtered neighbors (size of malicious neighborhood)

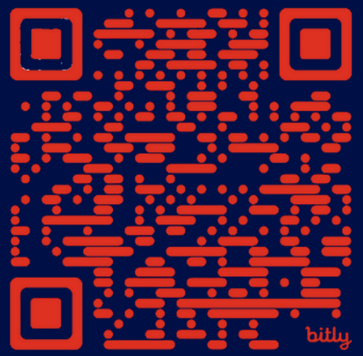
Since Sketchguard can support compression ratios greater than 6300:1, the sketch transmission overhead becomes negligible.

Table 3: Communication Overhead Comparison

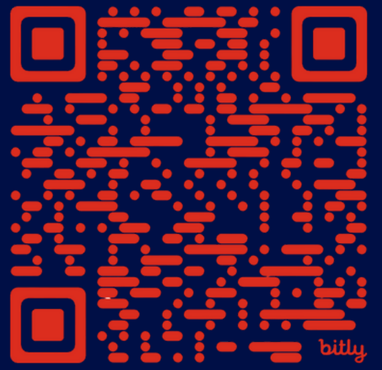
Scenario	Accepted Neighbors	Reduction
Benign (no filtering)	$\lvert\mathcal{S}_i\rvert = \lvert\mathcal{N}_i\rvert$	<0.02% penalty
50% Byzantine filtering	$\lvert\mathcal{S}_i\rvert \approx 0.5\lvert\mathcal{N}_i\rvert$	~50%
70% Byzantine filtering	$\lvert\mathcal{S}_i\rvert \approx 0.3\lvert\mathcal{N}_i\rvert$	~70%

Q&A

Paper Preprint



LinkedIn



THE UNIVERSITY OF

MELBOURNE

Presentation Link