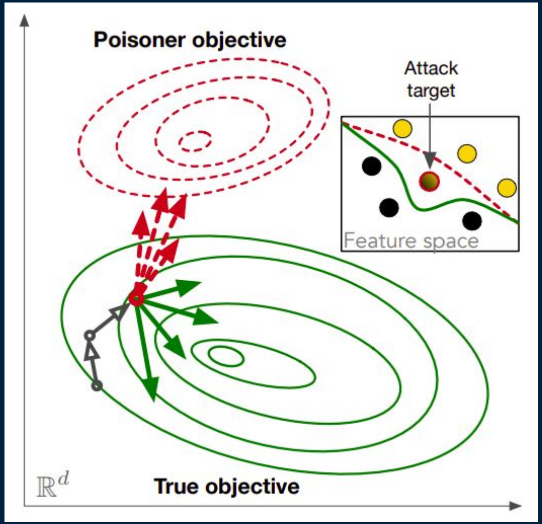
The Limitations of Federated Learning in Sybil Settings

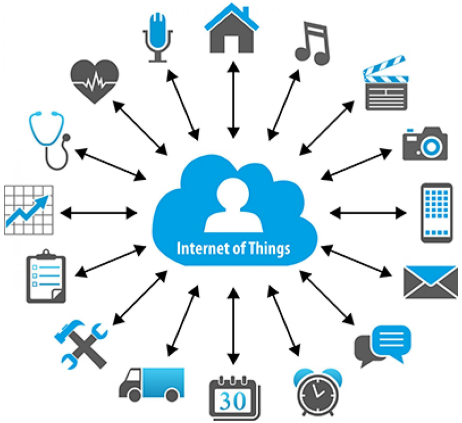
Clement Fung*, Chris J.M. Yoon+, Ivan Beschastnikh+
** Carnegie Mellon University* *+ University of British Columbia*

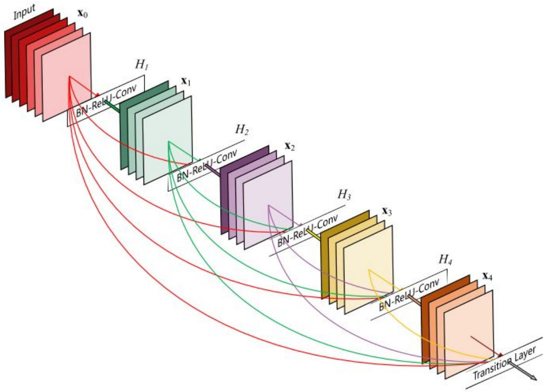


The evolution of machine learning at scale

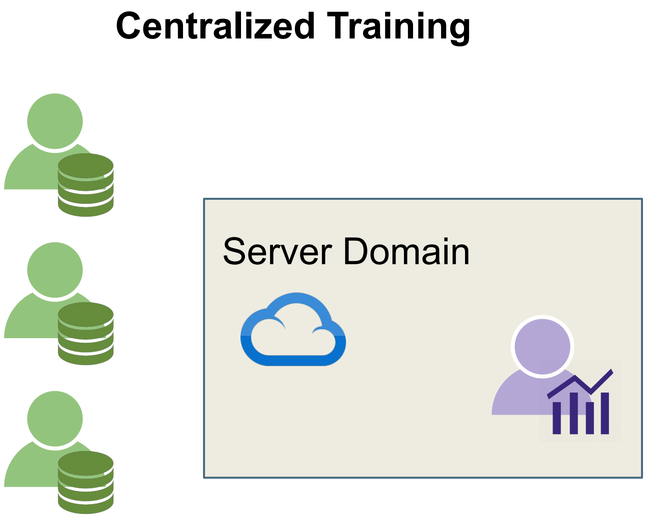
- Machine learning (ML) is a data hungry application
  - Large volumes of data
  - Diverse data
  - Time-sensitive data





The evolution of machine learning at scale

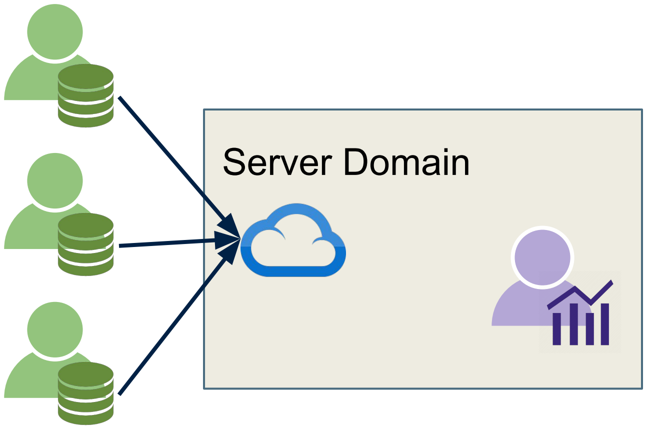
1. **Centralized training** of ML model



The evolution of machine learning at scale

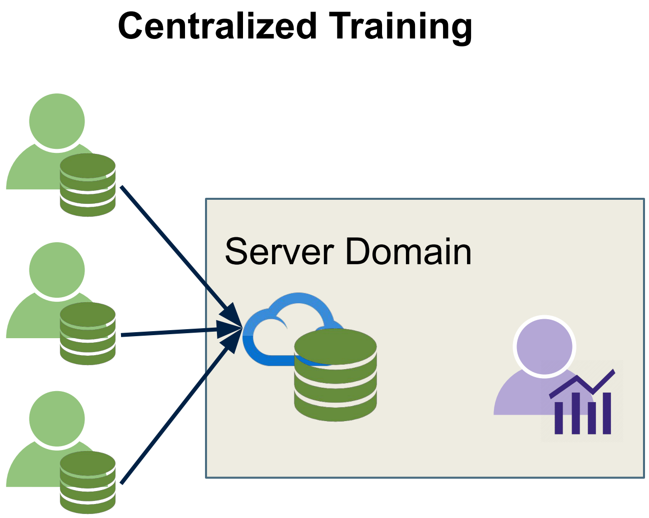
1. **Centralized training** of ML model

Centralized Training



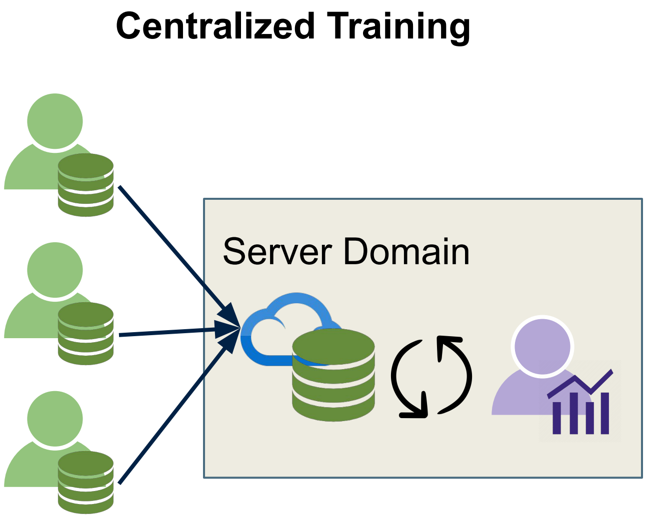
The evolution of machine learning at scale

1. **Centralized training** of ML model



The evolution of machine learning at scale

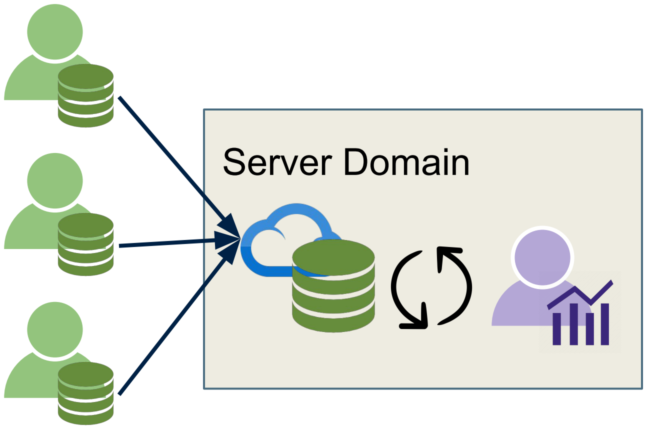
1. **Centralized training** of ML model



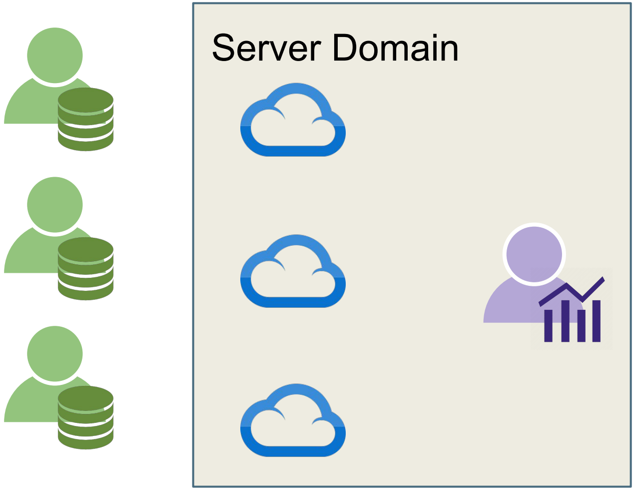
The evolution of machine learning at scale

1. Centralized training of ML model
2. **Distributed training** over sharded dataset and workers

Centralized Training

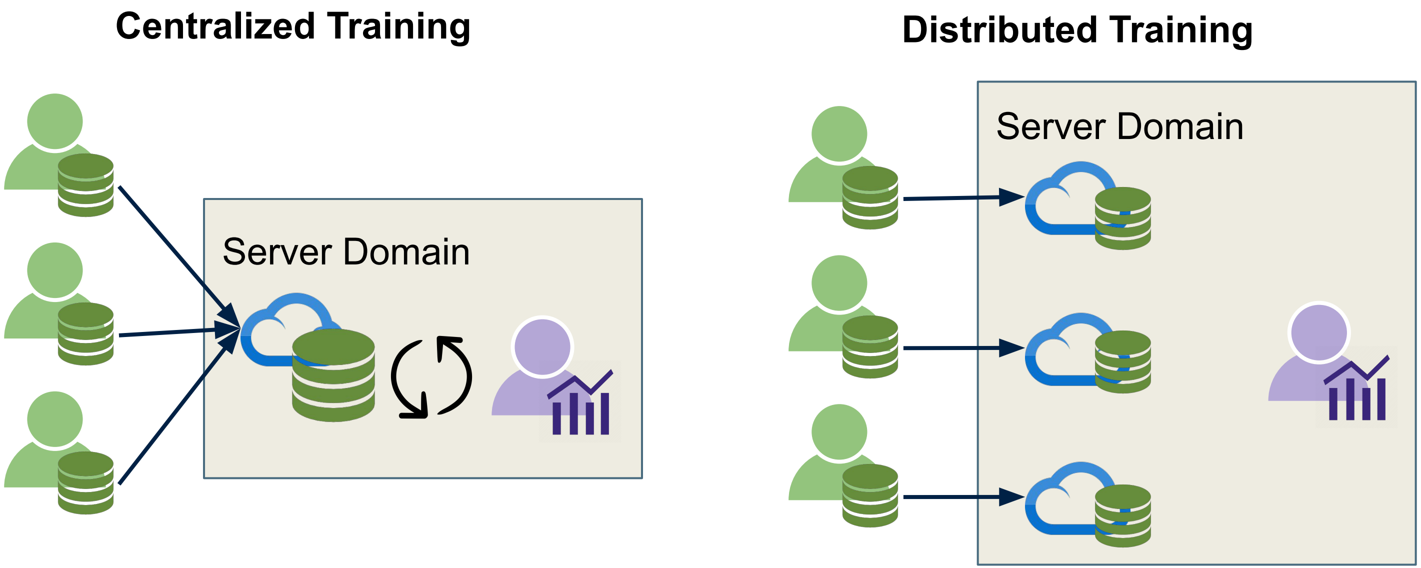


Distributed Training



The evolution of machine learning at scale

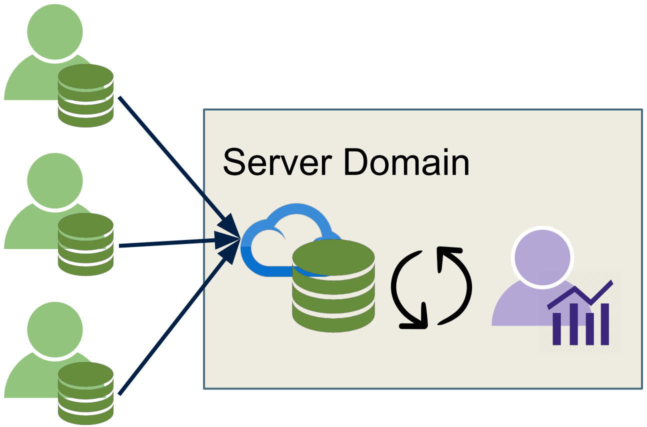
1. Centralized training of ML model
2. **Distributed training** over sharded dataset and workers



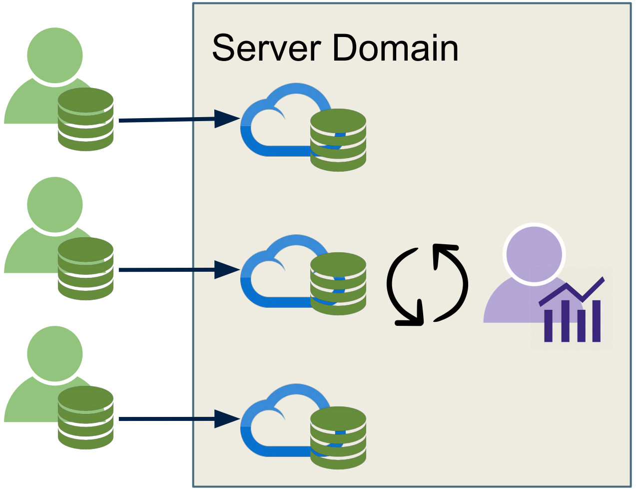
The evolution of machine learning at scale

1. Centralized training of ML model
2. **Distributed training** over sharded dataset and workers

Centralized Training

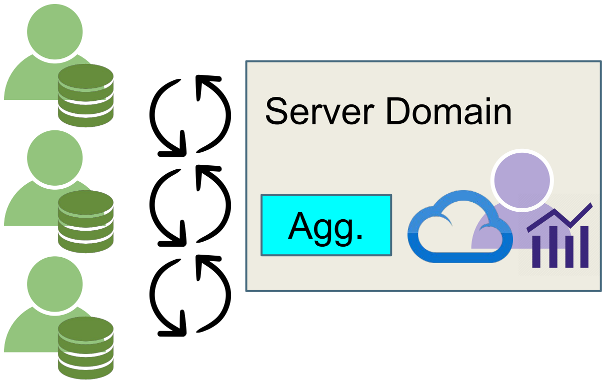


Distributed Training



Federated learning (FL)

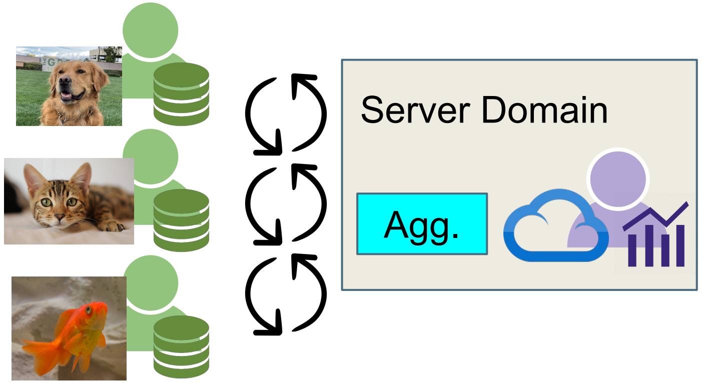
- Train ML models **over network**
  - Less network cost, no data transfer [1]
  - Server aggregates updates across clients
- Enables privacy-preserving alternatives
  - Differentially private federated learning [2]
  - Secure aggregation [3]



[1] McMahan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017
[2] Geyer et al. Differentially Private Federated Learning: A Client Level Perspective. NIPS 2017
[3] Bonawitz et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning. CCS 2017.

Federated learning (FL)

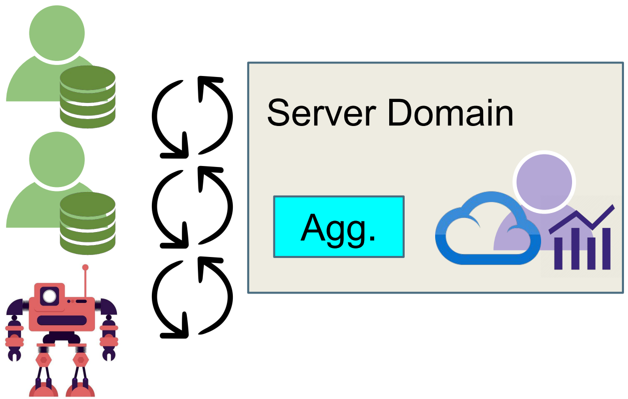
- Train ML models **over network**
  - Less network cost, no data transfer [1]
  - Server aggregates updates across clients
- Enables privacy-preserving alternatives
  - Differentially private federated learning [2]
  - Secure aggregation [3]
- Enables training over **non i.i.d. data settings**
  - Users with disjoint data types
  - Mobile, internet of things, etc.



[1] McMahan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017
[2] Geyer et al. Differentially Private Federated Learning: A Client Level Perspective. NIPS 2017
[3] Bonawitz et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning. CCS 2017.

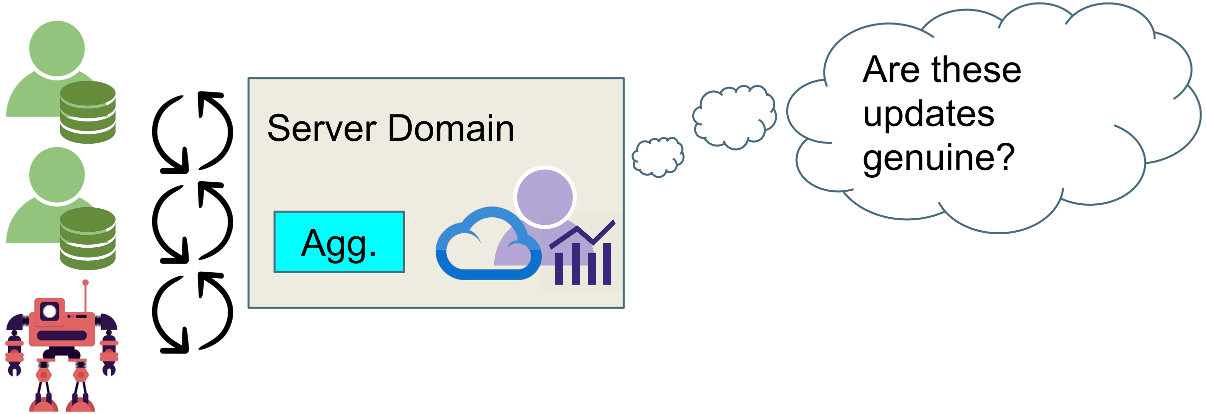
Federated learning: new threat model

- The role of the client has changed significantly!
  - Previously: passive data providers
  - Now: perform **arbitrary compute**



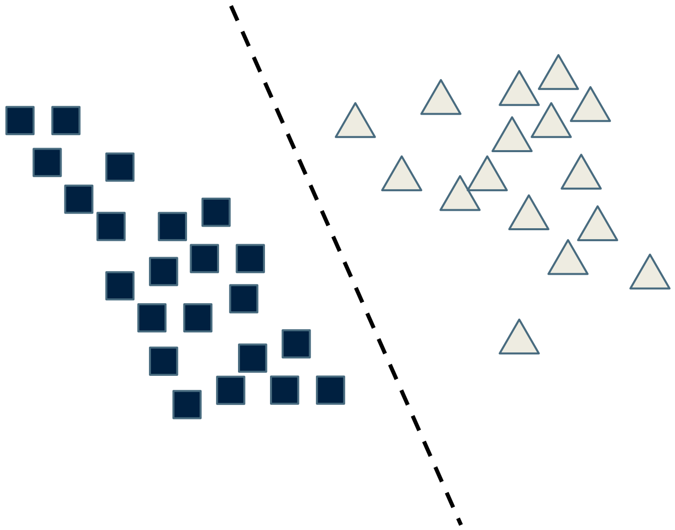
Federated learning: new threat model

- The role of the client has changed significantly!
  - Previously: passive data providers
  - Now: perform **arbitrary compute**
- Aggregator never sees client datasets, compute outside domain
  - Difficult to validate clients in "diverse data" setting



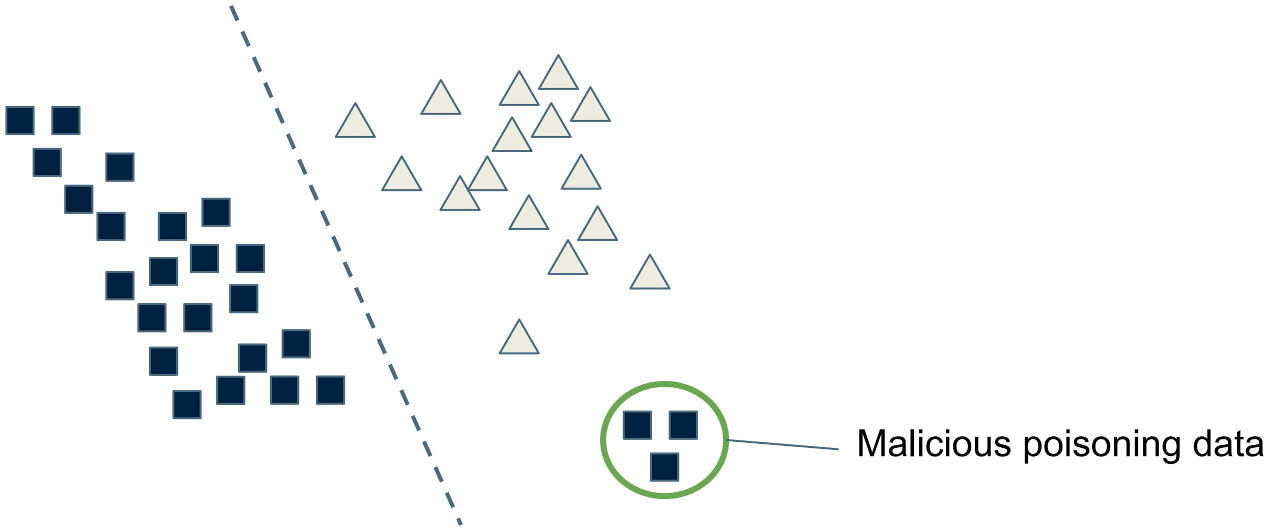
Poisoning attacks

- Traditional poisoning attack: malicious training data
    - Manipulate behavior of final trained model



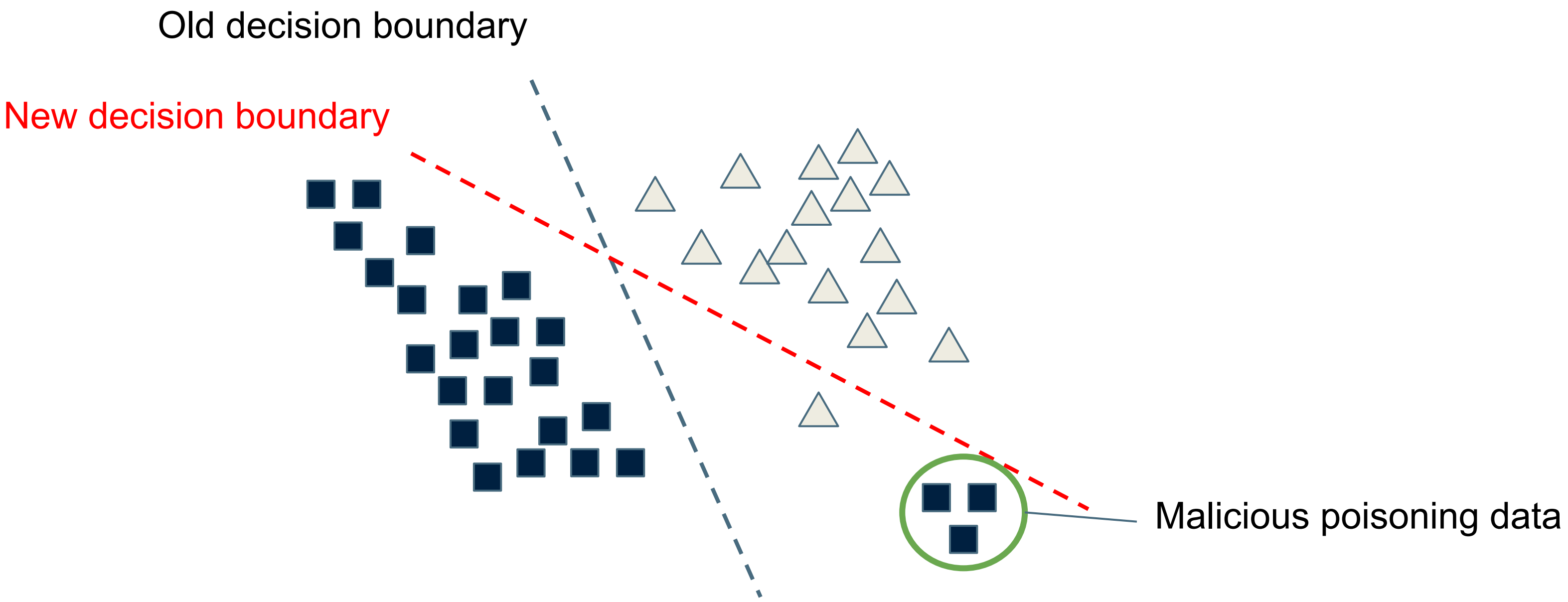
Poisoning attacks

- Traditional poisoning attack: malicious training data
  - Manipulate behavior of final trained model



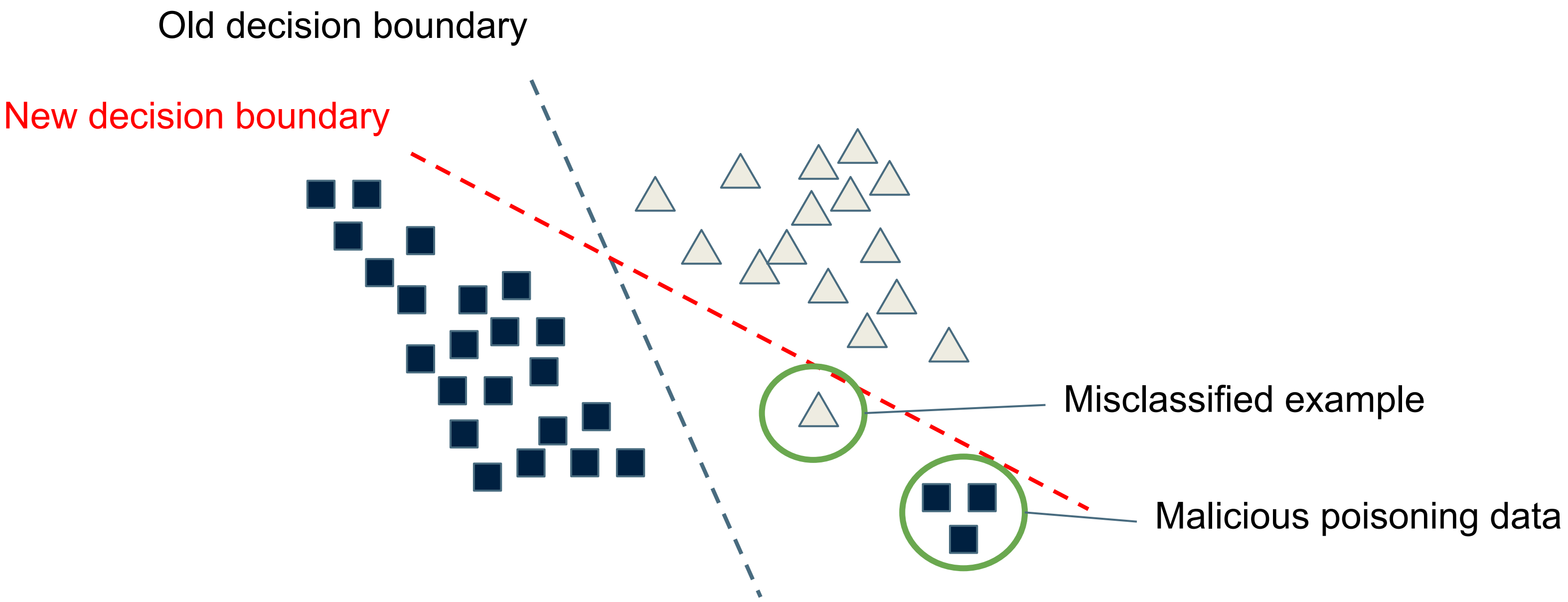
Poisoning attacks

- Traditional poisoning attack: malicious training data
    - Manipulate behavior of final trained model



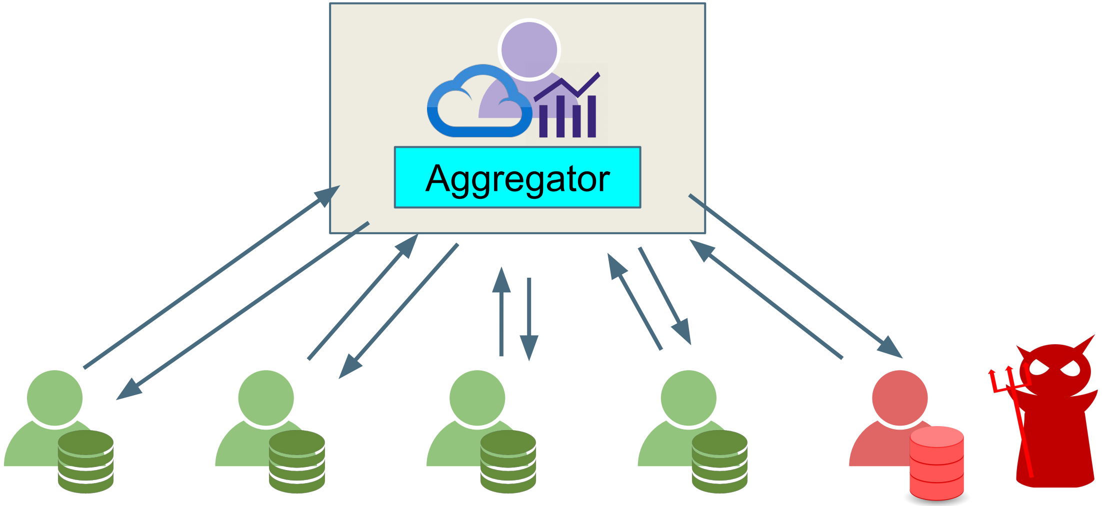
Poisoning attacks

- Traditional poisoning attack: malicious training data
  - Manipulate behavior of final trained model



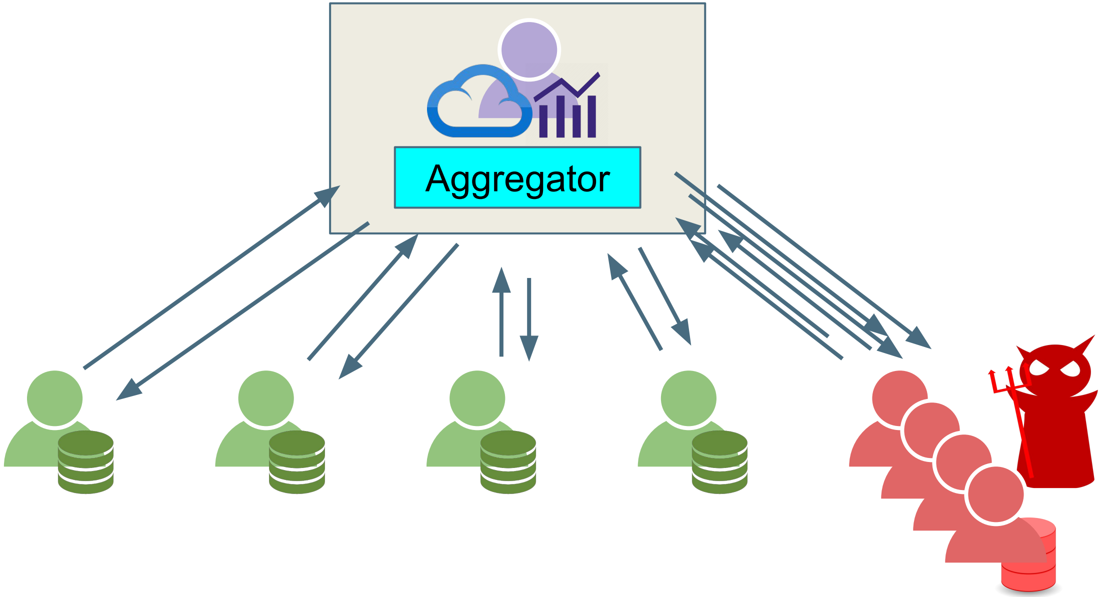
Sybil-based poisoning attacks

- In federated learning: provide malicious model updates



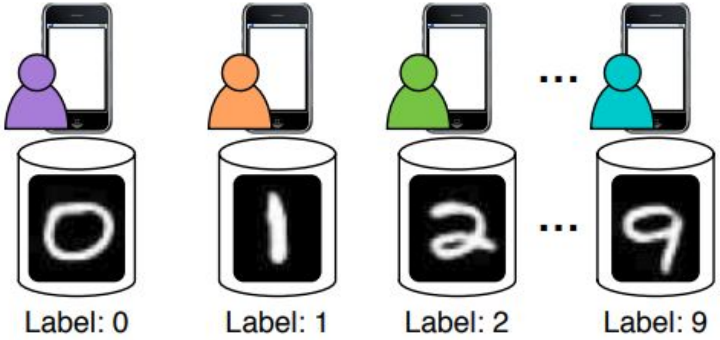
Sybil-based poisoning attacks

- In federated learning: provide malicious model updates
- With **sybils**: each account increases influence in system
  - Made worse in non-i.i.d setting



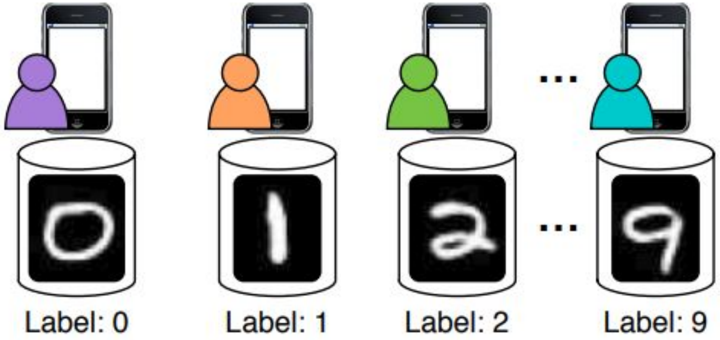
E.g. Sybil-based poisoning attacks

- A 10 client, non-i.i.d MNIST setting



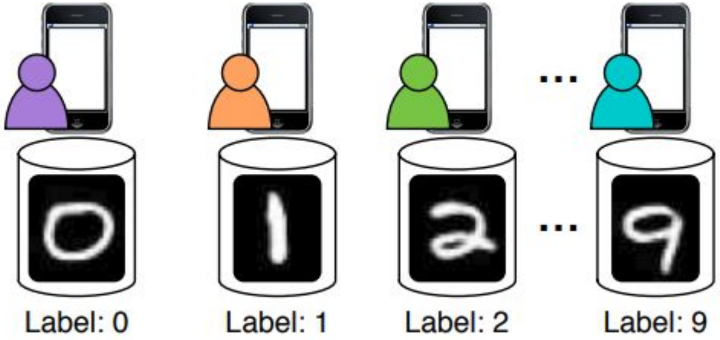
E.g. Sybil-based poisoning attacks

- A 10 client, non-i.i.d MNIST setting
- Sybil attackers with mislabeled “1-7” data
  - Need at least 10 sybils?



E.g. Sybil-based poisoning attacks

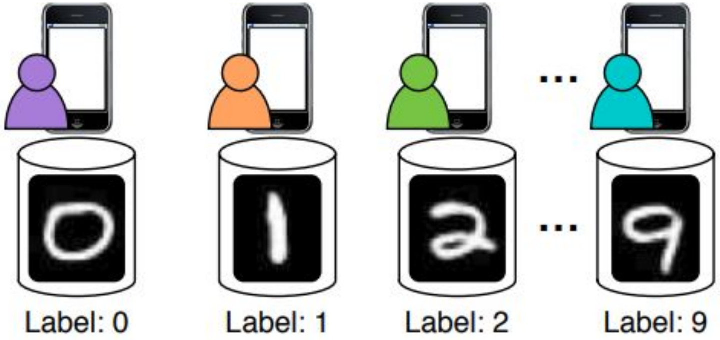
- A 10 client, non-i.i.d MNIST setting
- Sybil attackers with mislabeled “1-7” data
- At only 2 sybils:
    - 96.2% of 1s are misclassified as 7s
    - Minimal impact on accuracy of other digits



	Baseline	Attack 1	Attack 2
# honest clients	10	10	10
# malicious sybils	0	1	2
Accuracy (digits: 0, 2-9)	90.2%	89.4%	88.8%
Accuracy (digit: 1)	96.5%	60.7%	0.0%
Attack success rate	0.0%	35.9%	96.2%

E.g. Sybil-based poisoning attacks

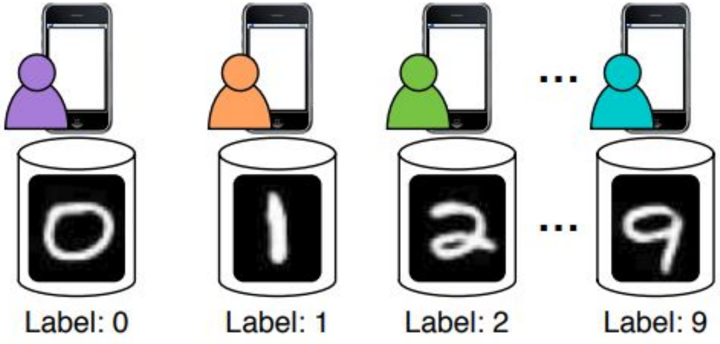
- A 10 client, non-i.i.d MNIST setting
- Sybil attackers with mislabeled “1-7” data
- At only 2 sybils:
    - 96.2% of 1s are misclassified as 7s
    - Minimal impact on accuracy of other digits



	Baseline	Attack 1	Attack 2
# honest clients	10	10	10
# malicious sybils	0	1	2
Accuracy (digits: 0, 2-9)	90.2%	89.4%	88.8%
Accuracy (digit: 1)	96.5%	60.7%	0.0%
Attack success rate	0.0%	35.9%	96.2%

E.g. Sybil-based poisoning attacks

- A 10 client, non-i.i.d MNIST setting
- Sybil attackers with mislabeled "1-7" data
- At only 2 sybils:
    - 96.2% of 1s are misclassified as 7s
    - Minimal impact on accuracy of other digits



	Baseline	Attack 1	Attack 2
# honest clients	10	10	10
# malicious sybils	0	1	2
Accuracy (digits: 0, 2-9)	90.2%	89.4%	88.8%
Accuracy (digit: 1)	96.5%	60.7%	0.0%
Attack success rate	0.0%	35.9%	96.2%

Our contributions

- Identify **gap in existing FL defenses**
  - No prior work has studied sybils in FL
- Categorize sybil attacks on FL along two dimensions:
  - Sybil objectives/targets
  - Sybil capabilities
- FoolsGold: a defense against sybil-based poisoning attacks on FL
  - Addresses targeted poisoning attacks
  - Preserves benign FL performance
  - Prevents poisoning from 99% sybil adversary

Federated learning: sybil attacks, defenses and new opportunities

Types of attacks on FL

- **Model quality**: modify the performance of the trained model
    - Poisoning attacks [1], backdoor attacks [2]
- **Privacy**: attack the datasets of honest clients
    - Inference attacks [3]
- **Utility**: receive an unfair payout from the system
    - Free-riding attacks [4]
- **Training inflation**: inflate the resources required (new!)
    - Time taken, network bandwidth, GPU usage

[1] Fang et al. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. Usenix Security 2020.
[2] Bagdasaryan et al. How To Backdoor Federated Learning. AISTATS 2020.
[3] Melis et al. Exploiting Unintended Feature Leakage in Collaborative Learning. S&P 2019.
[4] Lin et al. Free-riders in Federated Learning: Attacks and Defenses. arXiv 2019.

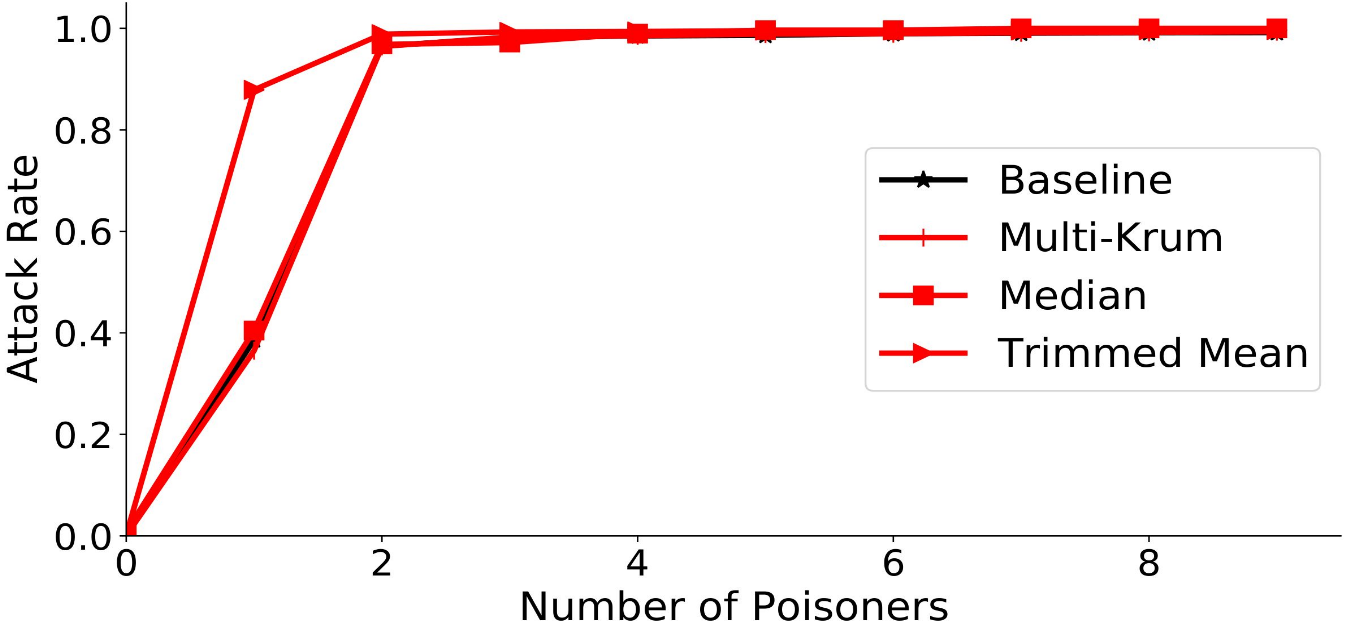
Existing defenses for FL are limited

- Existing defenses are aggregation statistics:
  - Multi-Krum [1]
  - Bulyan [2]
  - Trimmed Mean/Median [3]
- Require a bounded number of attackers
  - Do not handle sybil attacks
- Focus on poisoning attacks (model quality)
  - Do not handle other attacks (e.g., training inflation)

[1] Blanchard et al. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NIPS 2017
[2] El Mhamdi et al. The Hidden Vulnerability of Distributed Learning in Byzantium. ICML 2018.
[3] Yin et al. Byzantine-robust distributed learning: Towards optimal statistical rates. ICML 2018.

Existing defenses for FL

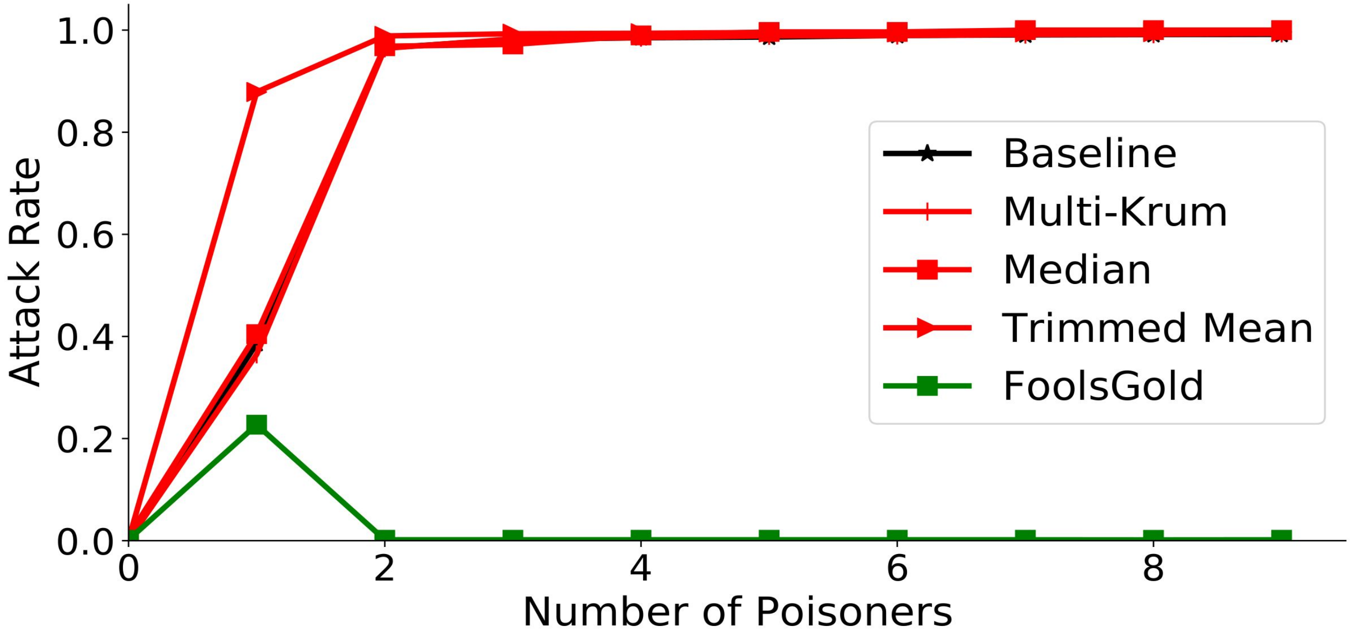
- Cannot defend against an increasing number of poisoners



[1] Blanchard et al. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NIPS 2017
[2] El Mhamdi et al. The Hidden Vulnerability of Distributed Learning in Byzantium. ICML 2018.
[3] Yin et al. Byzantine-robust distributed learning: Towards optimal statistical rates. ICML 2018.

Existing defenses for FL

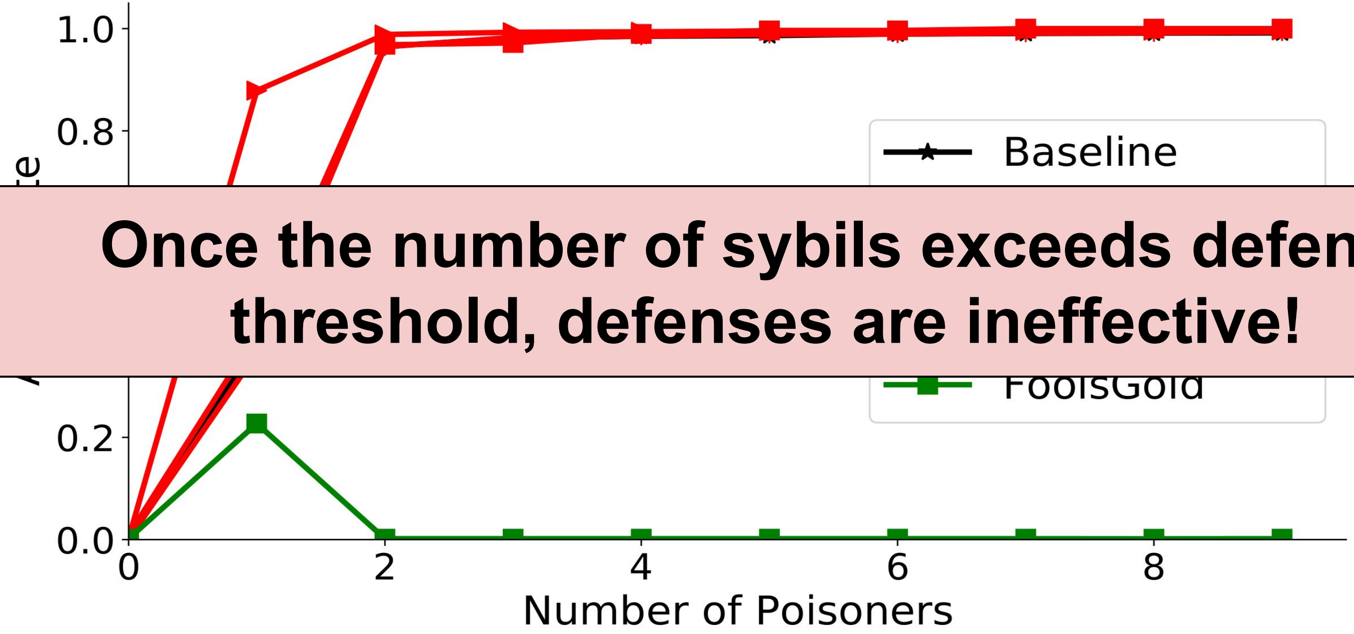
- FoolsGold is robust to an increasing number of poisoners



[1] Blanchard et al. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NIPS 2017
[2] El Mhamdi et al. The Hidden Vulnerability of Distributed Learning in Byzantium. ICML 2018.
[3] Yin et al. Byzantine-robust distributed learning: Towards optimal statistical rates. ICML 2018.

Existing defenses for FL

- FoolsGold is robust to an increasing number of poisoners

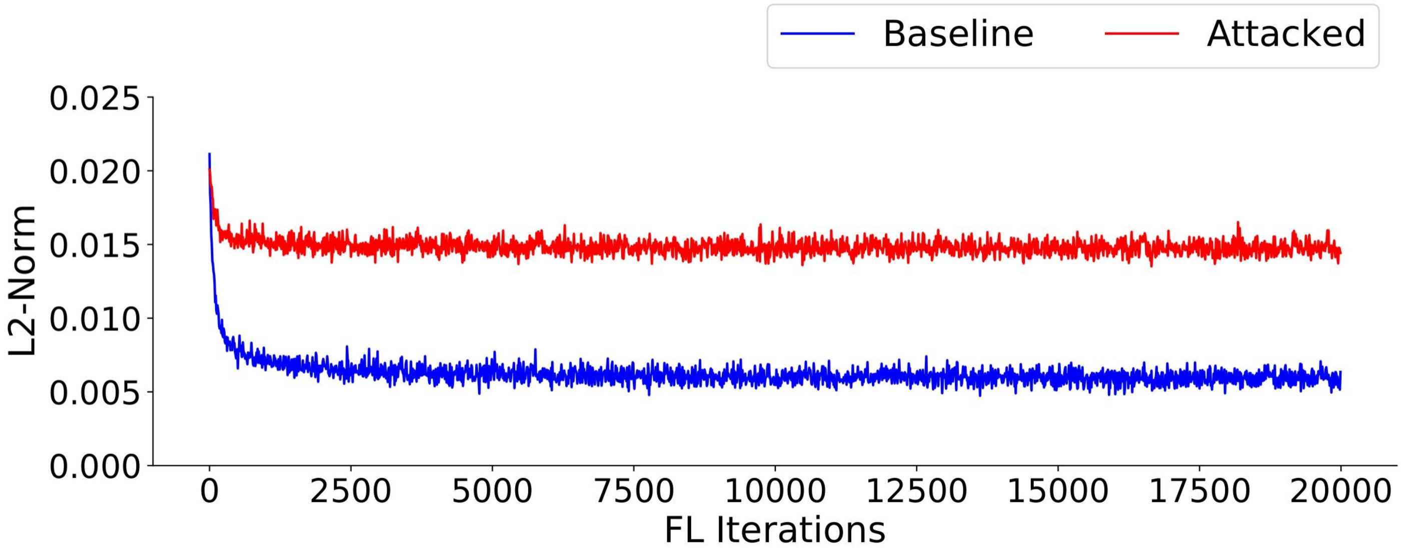


Once the number of sybils exceeds defense threshold, defenses are ineffective!

[1] Blanchard et al. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NIPS 2017
[2] El Mhamdi et al. The Hidden Vulnerability of Distributed Learning in Byzantium. ICML 2018.
[3] Yin et al. Byzantine-robust distributed learning: Towards optimal statistical rates. ICML 2018.

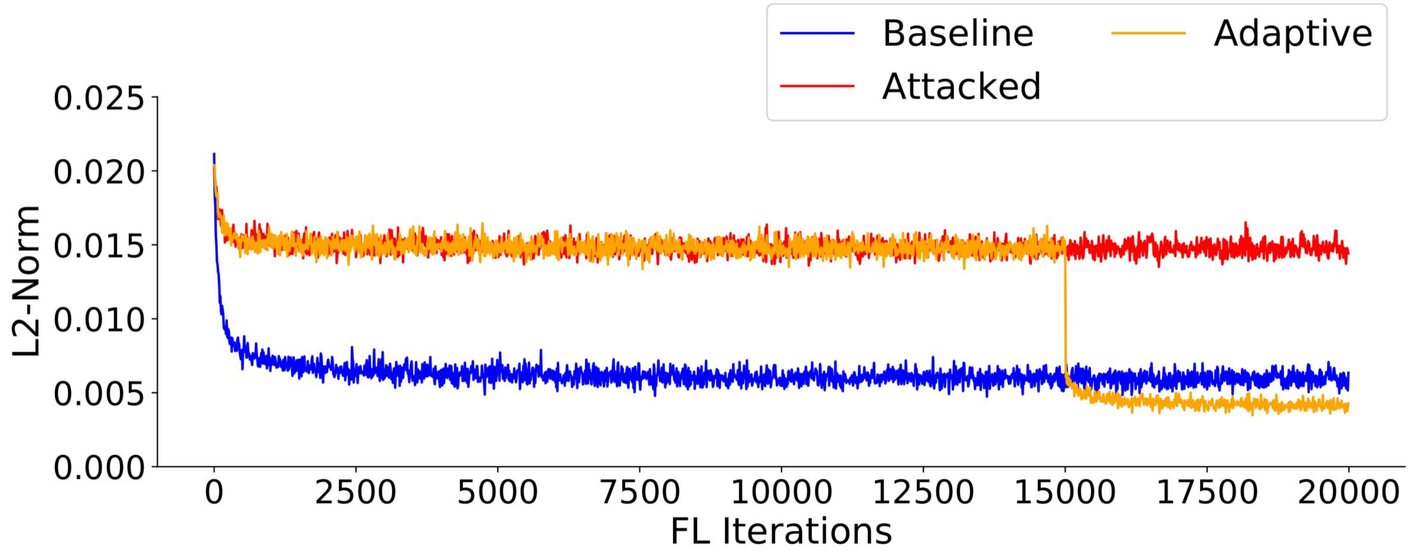
Training inflation on FL

- Manipulate ML stopping criteria to **ensure maximum time/usage**:
  - Validation error, size of gradient norm
  - Coordinated attacks can be **direct,**



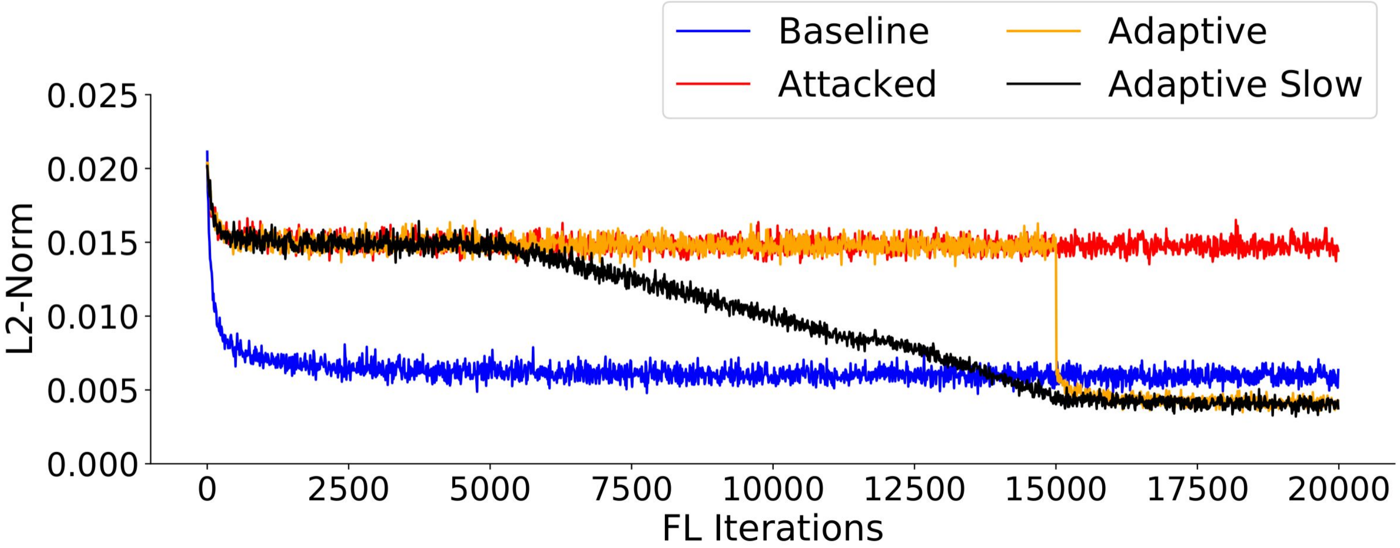
Training inflation on FL

- Manipulate ML stopping criteria to **ensure maximum time/usage**:
  - Validation error, size of gradient norm
  - Coordinated attacks can be **direct, timed,**



Training inflation on FL

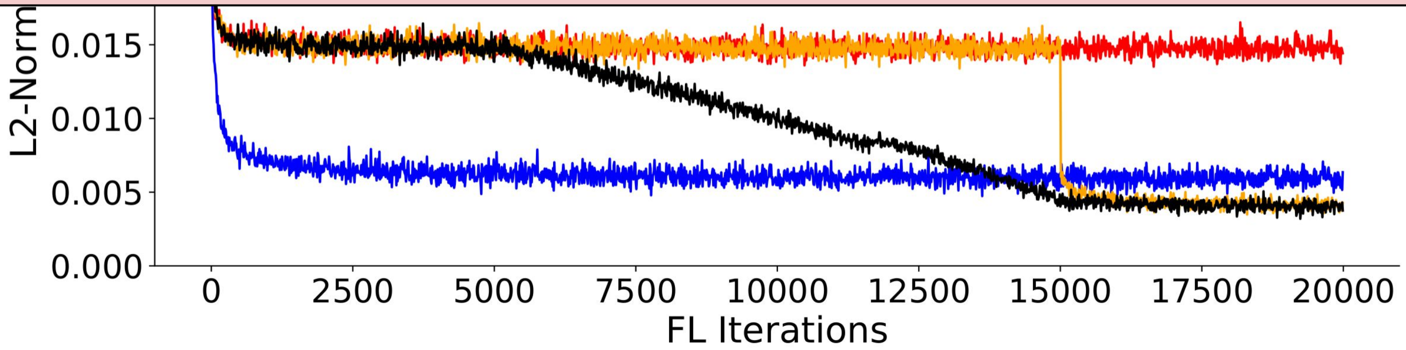
- Manipulate ML stopping criteria to **ensure maximum time/usage**:
  - Validation error, size of gradient norm
  - Coordinated attacks can be **direct, timed, or stealthy**



Training inflation on FL

- Manipulate ML stopping criteria to **ensure maximum time/usage**:
    - Validation error, size of gradient norm
    - Coordinated attacks can be **direct, timed, or stealthy**

Coordinated adversary can arbitrarily manipulate the length of federated learning process!



Sybil strategies when attacking FL

- **Attack data diversity**:
  - How common is the strategy used between sybils?
  - Identical datasets? Diverse datasets?
- **Coordination**:
  - How much state do sybils share?
  - How often do sybils communicate?
- **Churn**:
  - Do sybils benefit when joining/leaving system during the attack?

Sybil strategies when attacking FL

- We categorize existing FL attacks based on these criteria
  - Many can be **categorized by their sybil strategies**
  - See discussion and table in the paper

Table 2 *Sybil strategies*			Table 1 *Attack types*					
Churn	**Data**	**Coordination**	**U.Poison**	**T.Poison**	**D.Inversion**	**M.Infer**	**M.Free**	**T.Inflate**
Remainers	Clones	Uncoordinated		FoolsGold **§6**		[41]		
		Swarm						
		Puppets						
	Act-alikes	Uncoordinated		[58]				
		Swarm						
		Puppets						
	Clowns	Uncoordinated		[3,6,36]	[26,48,56]	[41,42,54]	[34]	**§5.2**
		Swarm						
		Puppets	[19,59]	**§7.3, §7.4**				**§5.2**
Churners	*All*	*All*	*Unexplored*					

Sybil strategies when attacking FL

- We categorize existing FL attacks based on these criteria
    - Many can be **categorized by their sybil strategies**
    - See discussion and table in the paper

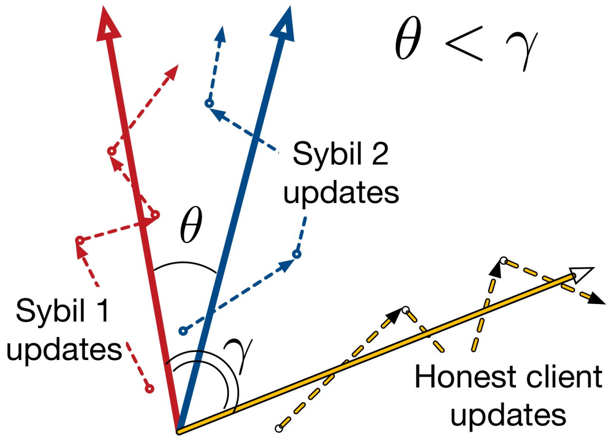
Table 2 *Sybil strategies*			Table 1 *Attack types*					
Churn	**Data**	**Coordination**	**U.Poison**	**T.Poison**	**D.Inversion**	**M.Infer**	**M.Free**	**T.Inflate**
Remainers	Clones	Uncoordinated		FoolsGold **§6**		[41]		
		Swarm						
		Puppets						
	Act-alikes	Uncoordinated		[58]				
		Swarm						
		Puppets						
	Clowns	Uncoordinated		[3,6,36]	[26,48,56]	[41,42,54]	[34]	**§5.2**
		Swarm						
		Puppets	[19,59]	**§7.3, §7.4**				**§5.2**
Churners	*All*	*All*	*Unexplored*					

FoolsGold: Defending against sybil-based targeted poisoning attacks

FoolsGold threat model and assumptions

- Addresses one section within table
  - Targeted poisoning attacks
  - Sybils with similar datasets
- Assume:
  - Non i.i.d federated learning setting
  - At least one honest client in FL system
  - Server can observe all model updates
    - No secure aggregation

FoolsGold algorithm



1. Collect **model update history** from each client
2. Compute **feature significance**
3. Pairwise **cosine similarity** between clients
4. Normalize through the inverse logit function
    - Ensures all weights are spread across 0-1 range
5. **Reduce learning rate** of contributions that are highly similar

Effect: **highly similar clients** will be **penalized over time**

Evaluating FoolsGold

- Attack scenario:
  - Defined source and target class attacks
  - Sybils join FL system and execute targeted poisoning
    - Uncoordinated attack with same poisoned dataset
    - Single attacker, N attackers, 99% attackers, etc.
- Datasets/models:
  - MNIST - softmax (image data)
  - VGGFace2 - Squeezenet DNN (multi-channel image data)
- See paper for more datasets and attack variants!

Baseline results

- FoolsGold does not interfere with benign setting

	Test Accuracy	Attack Rate
MNIST No Attack	0.92 (0.91 on FL)	n/a
VGGFace2 No attack	0.78 (0.75 on FL)	n/a

Baseline results

- FoolsGold does not interfere with benign setting
- FoolsGold defends against increasing number of sybils

	Test Accuracy	**Attack Rate**
MNIST No Attack	0.92 (0.91 on FL)	n/a
MNIST 5 sybils (33%)	0.91	0.001
VGGFace2 No attack	0.78 (0.75 on FL)	n/a
VGGFace2 5 sybils (33%)	0.78	0.001

Baseline results

- FoolsGold does not interfere with benign setting
- FoolsGold defends against increasing number of sybils

	Test Accuracy	Attack Rate
MNIST No Attack	0.92 (0.91 on FL)	n/a
MNIST 5 sybils (33%)	0.91	0.001
MNIST 990 sybils (99%)	0.91	0.001
VGGFace2 No attack	0.78 (0.75 on FL)	n/a
VGGFace2 5 sybils (33%)	0.78	0.001

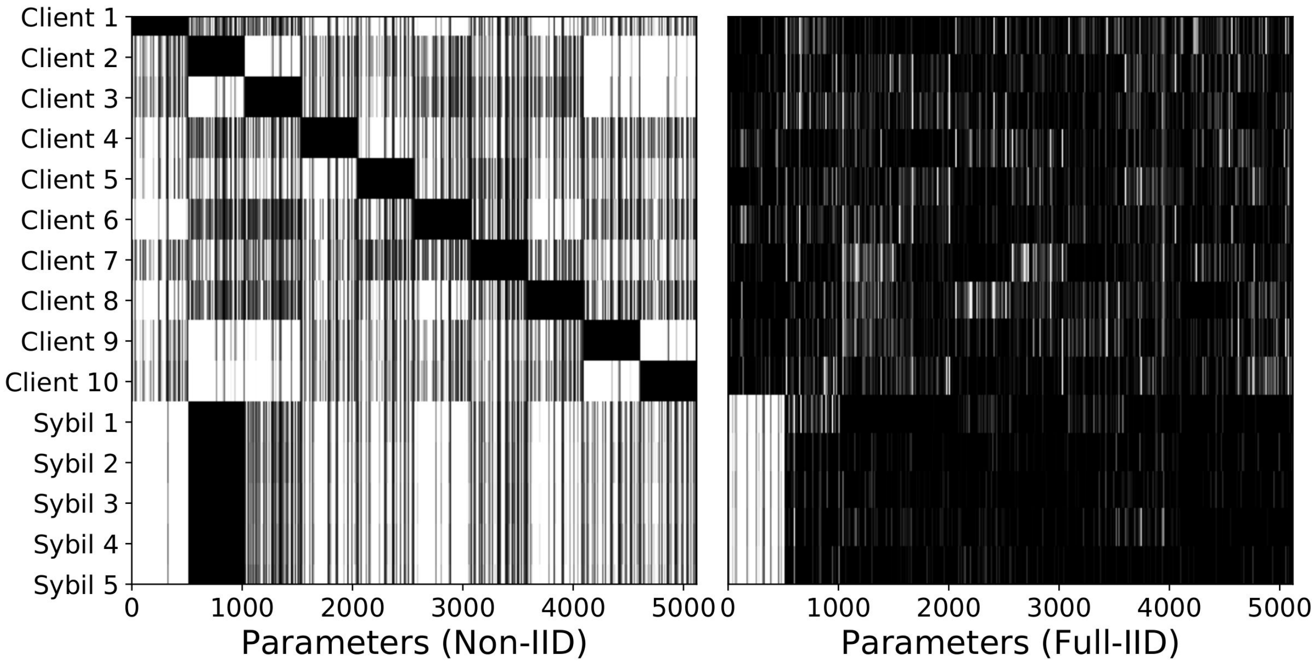
Baseline results

- FoolsGold does not interfere with benign setting
- FoolsGold defends against increasing number of sybils
- Performance against single attacker is worst

	Test Accuracy	**Attack Rate**
MNIST No Attack	0.92 (0.91 on FL)	n/a
MNIST 5 sybils (33%)	0.91	0.001
MNIST 990 sybils (99%)	0.91	0.001
MNIST 1 sybil	**0.74**	**0.23**
VGGFace2 No attack	0.78 (0.75 on FL)	n/a
VGGFace2 5 sybils (33%)	0.78	0.001
VGGFace2 1 sybil	**0.62**	**0.44**

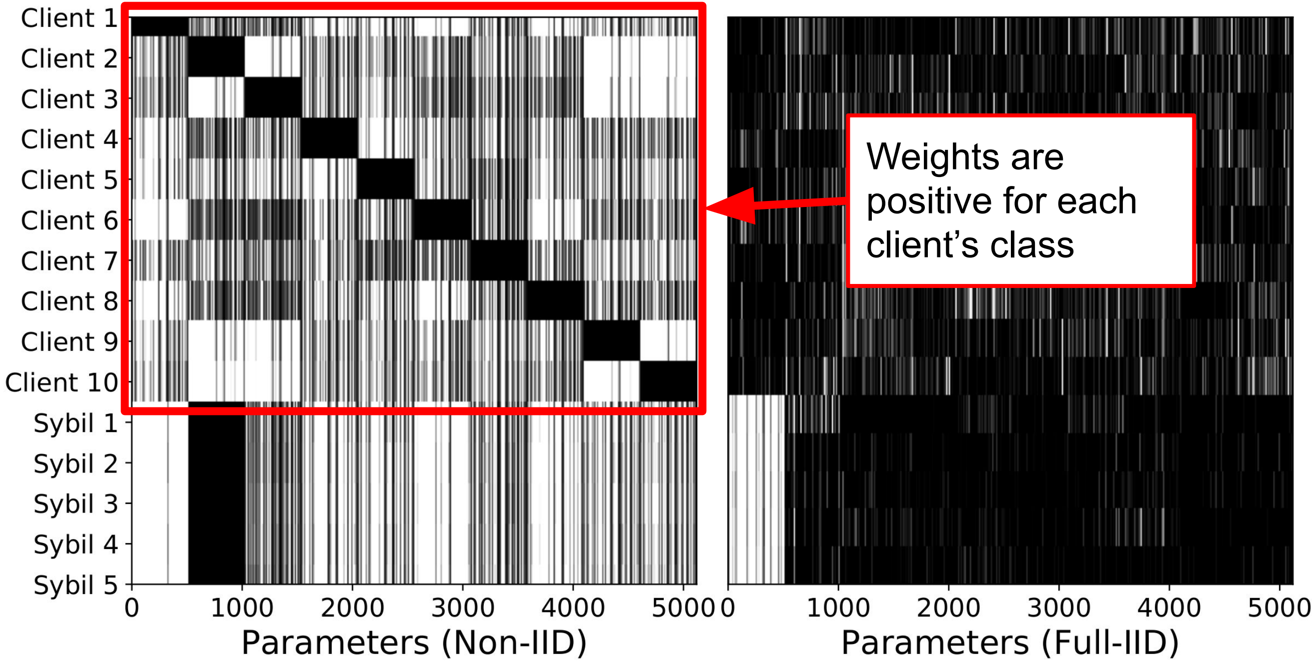
FoolsGold performs well even when i.i.d.

- How similar are model updates over VGGFace2 training process?
  - For each client/sybil, plot weights of final update



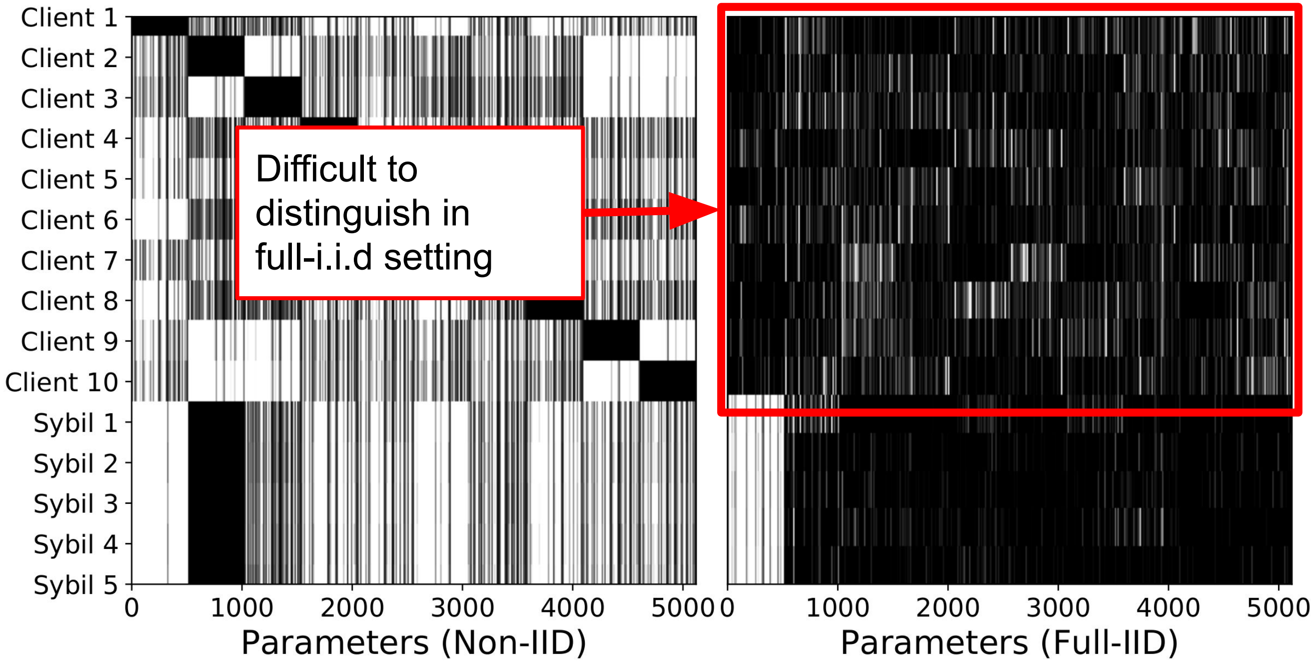
FoolsGold performs well even when i.i.d.

- How similar are model updates over VGGFace2 training process?
  - For each client/sybil, plot weights of final update



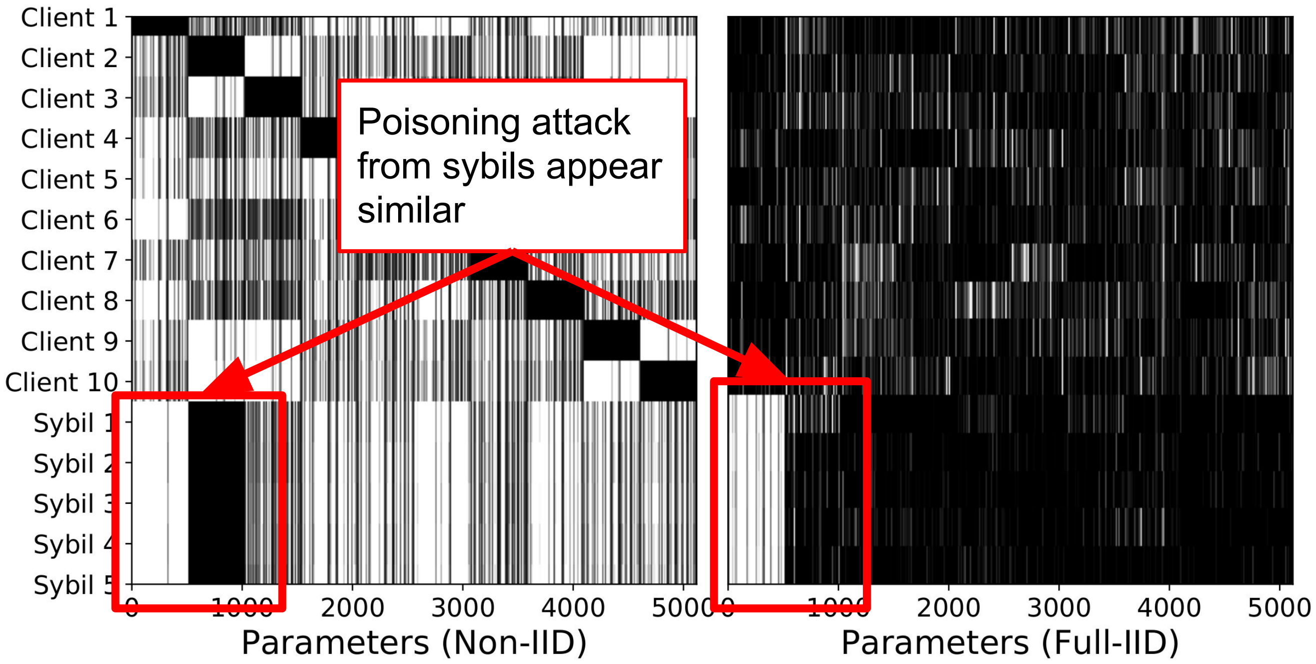
FoolsGold performs well even when i.i.d.

- How similar are model updates over VGGFace2 training process?
  - For each client/sybil, plot weights of final update



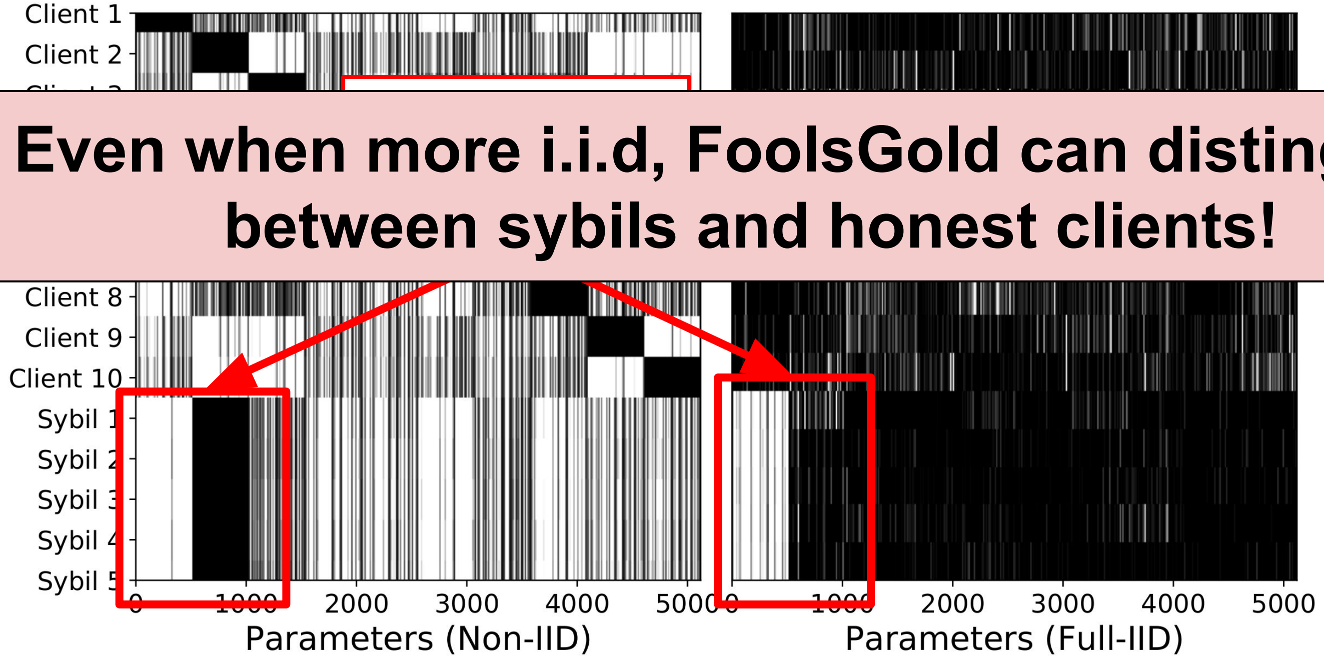
FoolsGold performs well even when i.i.d.

- How similar are model updates over VGGFace2 training process?
  - For each client/sybil, plot weights of final update



FoolsGold performs well even when i.i.d.

- How similar are model updates over VGGFace2 training process?
  - For each client/sybil, plot weights of final update



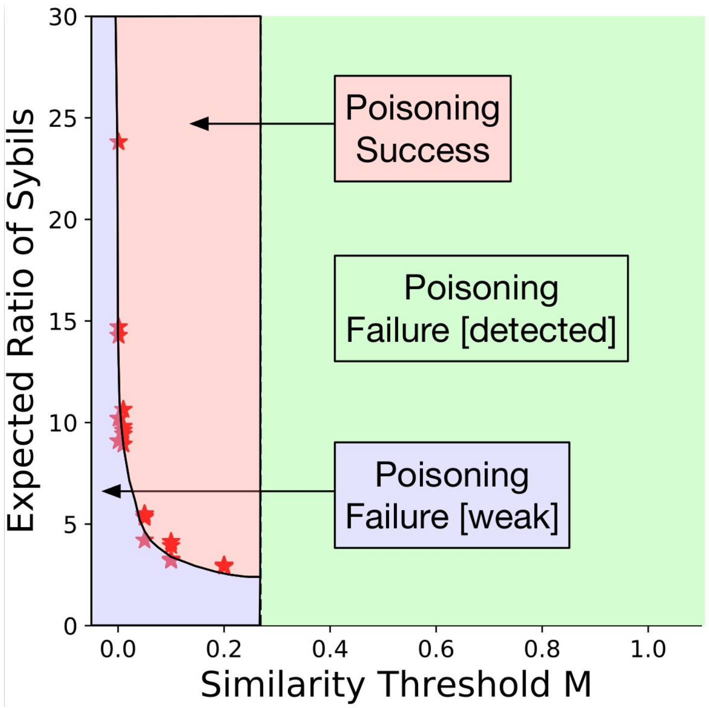
Even when more i.i.d, FoolsGold can distinguish between sybils and honest clients!

Can an intelligent attacker defeat FoolsGold?

- What if the attacker is stronger?
    - They know the FoolsGold algorithm
    - They can **coordinate at each iteration**
- Bypass FoolsGold by increasing dissimilarity amongst sybils
    - Modify model updates with orthogonal perturbations
    - Withhold poisoning attacks to avoid detection

Coordinated sybils can bypass FoolsGold

- Limiting malicious model update frequency
  - Monitor FoolsGold similarity
  - Only poison when similarity is below M
- Too often: Detected by FoolsGold (M>0.25)
- Too infrequent: Cannot overpower honest clients in system
- With lower M, success requires more sybils
  - Also requires estimate of honest client data distribution



The bigger picture

- FoolsGold can be defeated by increasing coordinated attackers
- Attacks extend beyond model quality attacks
- As future defenses are designed for federated learning:
  - Consider sybil capabilities when defining attacker

Table 2 *Sybil strategies*			Table 1 *Attack types*					
Churn	**Data**	**Coordination**	**U.Poison**	**T.Poison**	**D.Inversion**	**M.Infer**	**M.Free**	**T.Inflate**
Remainers	Clones	Uncoordinated		FoolsGold **§6**		[41]		
		Swarm						
		Puppets						
	Act-alikes	Uncoordinated		[58]				
		Swarm						
		Puppets						
	Clowns	Uncoordinated		[3,6,36]	[26,48,56]	[41,42,54]	[34]	**§5.2**
		Swarm						
		Puppets	[19,59]	**§7.3, §7.4**				**§5.2**
Churners	*All*	*All*	*Unexplored*					

Contributions

- Federated learning: new threat model
  - Adversaries perform **arbitrary compute**
- New attacks are possible/stronger with sybils
  - Categorize sybil strategies/capabilities
  - New training inflation attacks on FL
- FoolsGold: defending against sybil-based poisoning attacks
  - Detect sybils based on **client similarity**

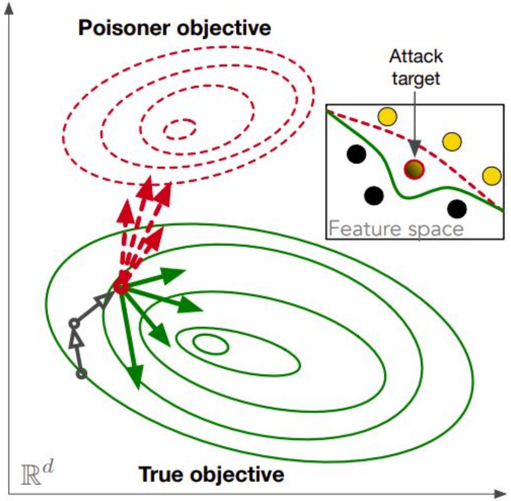


Table 2 *Sybil strategies*			Table 1 *Attack types*					
Churn	**Data**	**Coordination**	**U.Poison**	**T.Poison**	**D.Inversion**	**M.Infer**	**M.Free**	**T.Inflate**
Remainers	Clones	Uncoordinated		FoolsGold §6		[41]		
		Swarm						
		Puppets						
	Act-alikes	Uncoordinated		[58]				
		Swarm						
		Puppets						
	Clowns	Uncoordinated		[3,6,36]	[26,48,56]	[41,42,54]	[34]	§5.2
		Swarm						
		Puppets	[19,59]	§7.3, §7.4				§5.2
Churners	*All*	*All*	*Unexplored*					

Contact: clementf@andrew.cmu.edu
Our code can be found at:
https://github.com/DistributedML/FoolsGold