BYZANTINE-TOLERANT SGD

FOR DISTRIBUTED SYNCHRONOUS SGD

REFERENCE

- Xie, Cong, Oluwasanmi Koyejo, and Indranil Gupta. "Generalized byzantine-tolerant sgd." arXiv preprint arXiv:1802.10116 (2018).

ABSTRACT

ABSTRACT

- 새로운 강건한(robust) 동기 SGD 통합 규칙을 제시
  - 비잔틴(Byzantine)으로부터
  - 서버와 통신하는 데이터에 대해
  - 임의의 조작이 가능한 상황

ABSTRACT

- 새로운 강건한(robust) 동기 SGD 통합 규칙을 제시
    - 비잔틴 저항성을 증명하고
    - 현재의 접근방식과 분석 및 비교함

INTRODUCTION

INTRODUCTION

- 분산 머신러닝은 여러 종류의 공격에 취약
  - 실패/공격
  - 붕괴
  - 연산 에러 등
- 공격에 대한 탄력성/저항성은 갈수록 더 중요해지고 있음

INTRODUCTION

- 일반화된 실패 모델인 Byzantine failure를 고려
  - 통신(전송)하는 값이 임의의 값으로 위/변조될 있는 상황
  - 어떠한 실패나 공격의 제약이 없는 상황

INTRODUCTION

- 분산 학습 프레임워크는 파라미터 서버(Parameter Server, PS)를 가정
  - 클라이언트-서버 구조
  - 서버는 모델의 글로벌 복사본을 저장, gradients를 통합, 전파
  - 클라이언트는 서버로부터 최신 모델을 받아 개인 데이터로 학습, 전파

INTRODUCTION

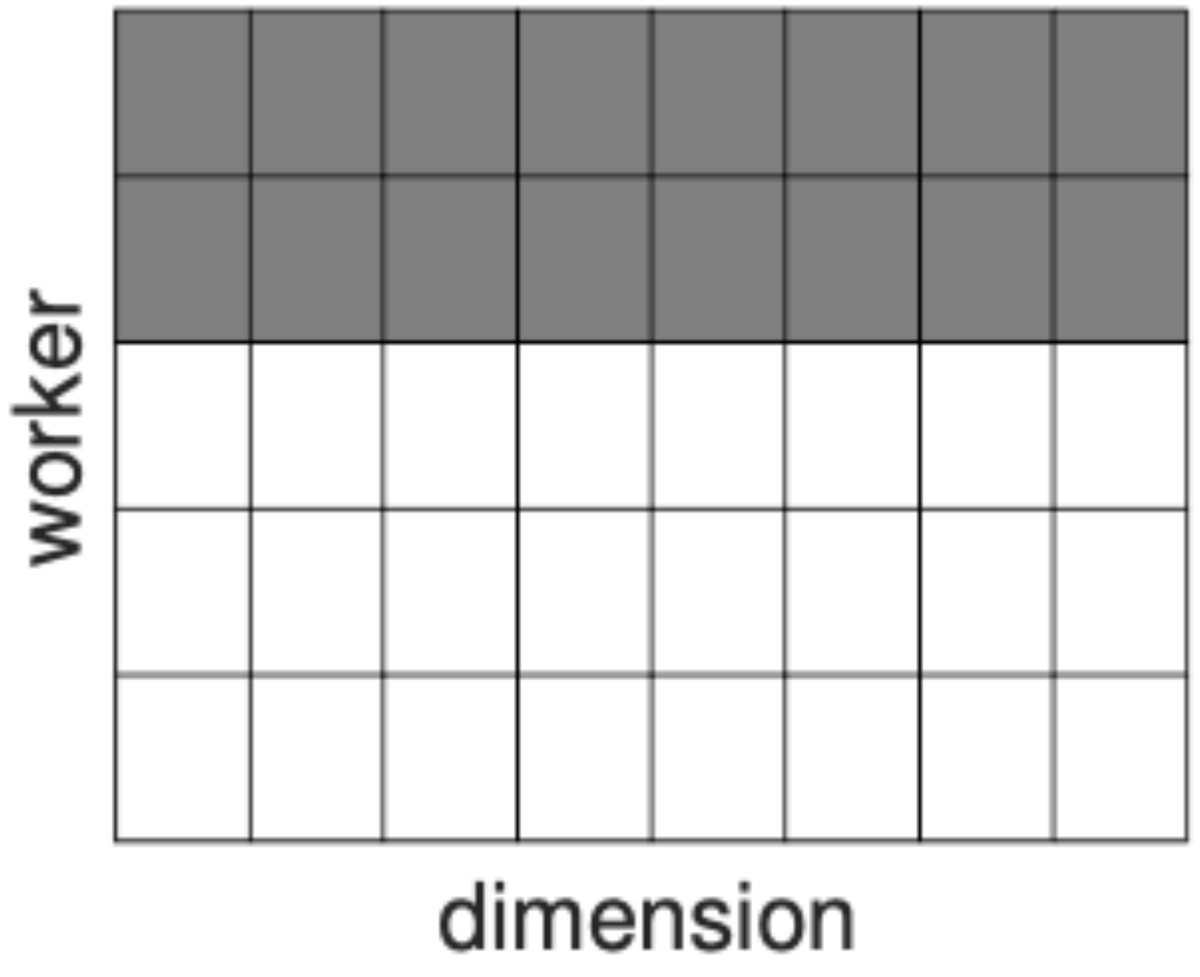
- Synchronous Stochastic Gradient Descent의 비잔틴 연구를 수행
  - PS 구조에서는 널리 쓰이는 알고리즘
  - Gradient를 모아서 동기적으로 다음 반복으로 넘어감

INTRODUCTION

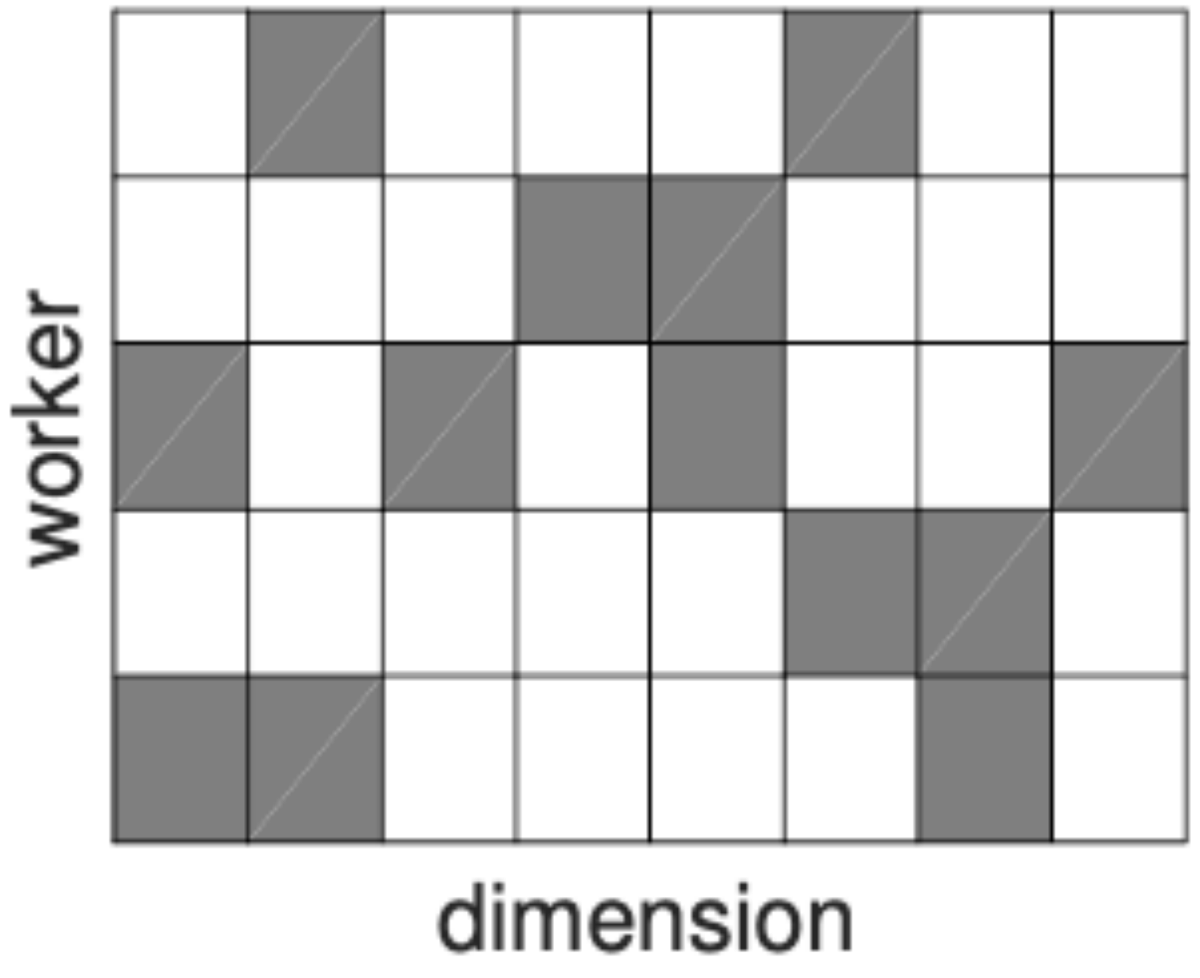
- 실패 모델은 $n \times d$ 행렬로 표현 가능
  - d-차원의 gradients
  - n명의 워커(workers, 클라이언트)

INTRODUCTION

- 실패 모델은 $n \times d$ 행렬로 표현 가능
  - 5명의 워커, 8차원의 그레디언트
  - (a)는 (b)의 특별한 경우에 해당



(a) Classic Byzantine



(b) Generalized Byzantine

INTRODUCTION

- 여러 종류의 공격 유형이 있음
- 일반적으로 공격자는 모델 학습을 방해하고자 함
  - SGD 수렴을 느리게 만들거나
  - 나쁜 솔루션을 향하게 만듦

INTRODUCTION

- 가능한 공격 유형을 3개로 분류
- Gamber
  - 공격자가 데이터를 무작위 선출해
  - 악의적으로 수정
  - 서버가 받은 데이터가 임의로 변경 되었을 수 있음
- 전통적인 비잔틴

INTRODUCTION

- 가능한 공격 유형을 3개로 분류
- Omniscient
  - 모든 워커들이 전송한 그레디언트를 알고 있음
  - 그레디언트의 총합에 매우 큰 음수를 곱함
  - 목표는 SGD가 원래의 반대 방향으로 크게 이동하도록 하는 것
- Dimensional 비잔틴

INTRODUCTION

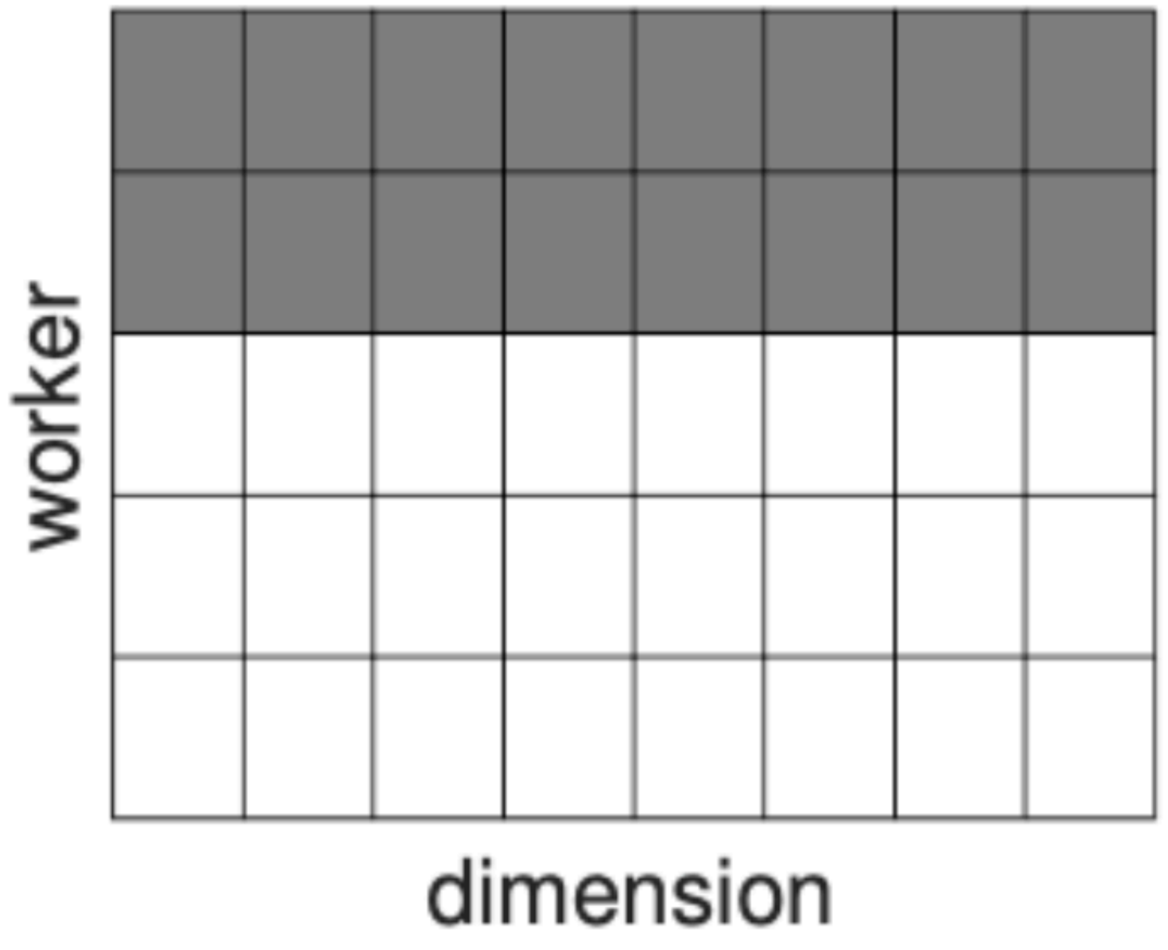
- 가능한 공격 유형을 3개로 분류
- Gaussian attack
  - 일부 그레디언트 벡터가 무작위 벡터로 대체
  - 가우시안 분포를 따르는 무작위 벡터
- Dimensional 비잔틴

INTRODUCTION

- 각 차원에 대해 비잔틴 값의 수는 절반 미만으로 가정
    - 흔한 가정
    - Dimensional Byzantine 탄력성이라 칭함

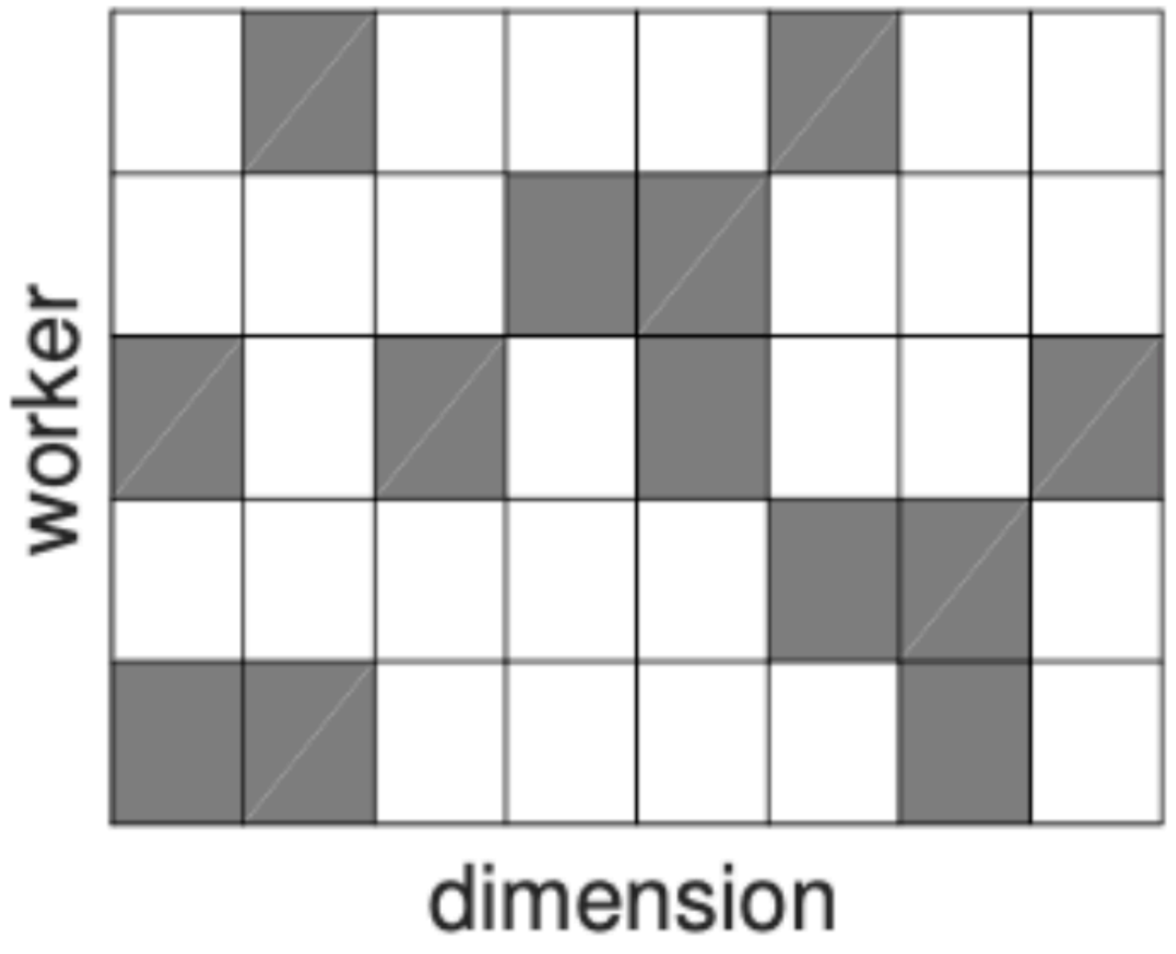
BYZANTINE MODEL

BYZANTINE MODEL

- 전통적인 비잔틴 모델:
- $$\tilde{v}_i = \begin{cases} v_i, \textit{if the ith worker is correct,} \\ arbitrary, \textit{if the ith worker is Byzantine.} \end{cases}$$
- 

(a) Classic Byzantine

BYZANTINE MODEL

- 일반화된 비잔틴 모델:
- $$(\tilde{v}_i)_j = \begin{cases} (v_i)_j, \textit{if the the } j\textit{th dimension of } v_i \textit{ is correct,} \\ arbitrary, otherwise, \end{cases}$$
- 

(b) Generalized Byzantine

MEDIAN-BASED AGGREGATION

MEDIAN-BASED AGGREGATION

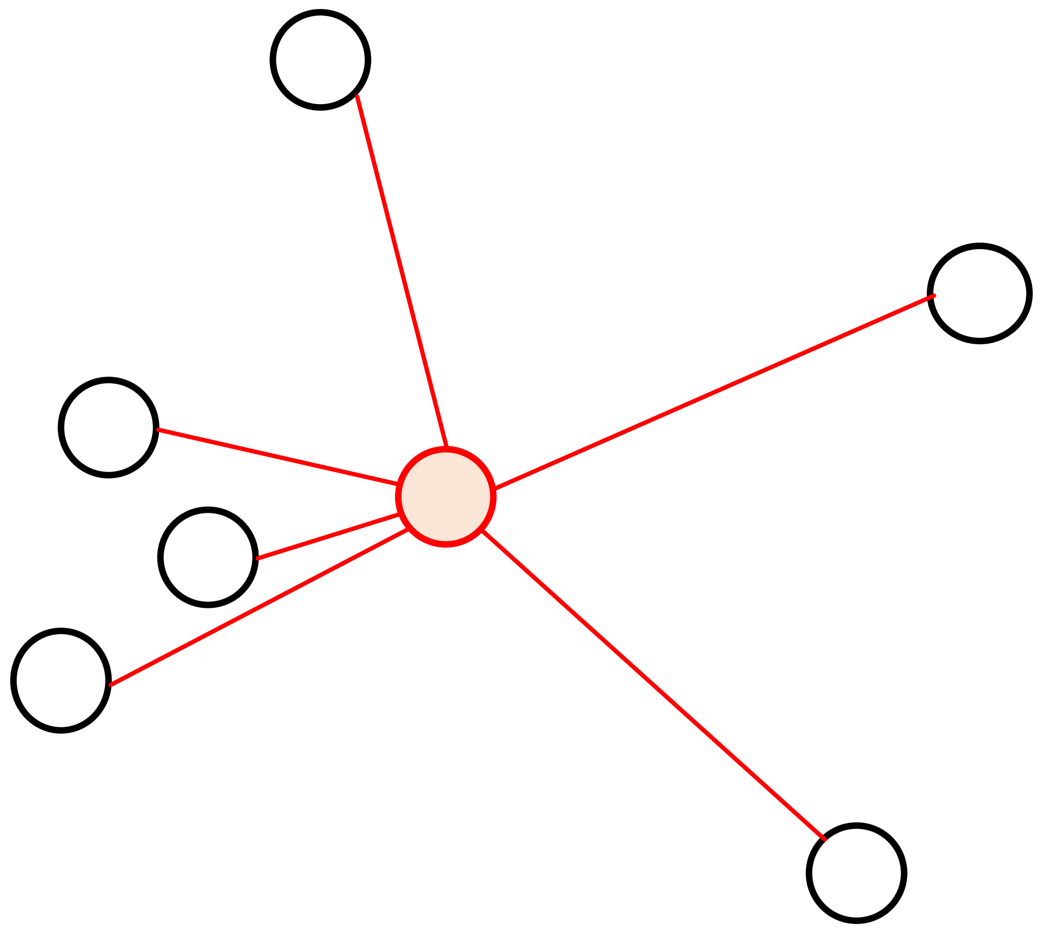
- 3개의 중앙값 기반 통합 규칙을 제안
  - Geometric Median
  - Marginal Median
  - Beyond Median

MEDIAN-BASED AGGREGATION

- Geometric Median (기하중앙값)
- 평균에 대한 강건한 추정량으로 사용
  - 최대, 데이터의 절반이 부정해도
  - 부정하지 않은 데이터에 대한 추정을 제공

MEDIAN-BASED AGGREGATION

- Geometric Median (기하중앙값)
- $$\lambda = GeoMed(\{\tilde{v}_i : i \in [n]\}) = \underset{v \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} \|v - \tilde{v}_i\|$$
  - 거리의 합의 최소



MEDIAN-BASED AGGREGATION

- Marginal Median
- $\mu = MarMed(\{\tilde{v}_i : i \in [n]\})$
- μ의 j번째 차원은 $\mu_j = median(\{(\tilde{v}_1)_j, \ldots, (\tilde{v}_n)_j\})$
  - $median(\cdot)$은 1차원 중앙값

MEDIAN-BASED AGGREGATION

- Marginal Median
- μ의 j번째 차원은 $\mu_j = median(\{(\tilde{v}_1)_j, \ldots, (\tilde{v}_n)_j\})$

worker

1	3	1	5	9	2	5
4	5	1	4	5	1	9
6	4	3	2	2	3	1
9	8	5	1	5	2	2
1	6	4	8	4	2	3

dimension

Sorting

1	3	1	1	2	1	1
1	4	1	2	4	2	2
4	5	3	4	5	2	3
6	6	4	5	5	2	5
9	8	5	8	9	3	9

MEDIAN-BASED AGGREGATION

- Beyond Median
- 비잔틴의 수 q를 쉽게 추정할 수 있다면
- 중앙값에 가까운 $n - q$ 개의 값의 평균을 활용할 수 있을 것
  - “mean around median”

MEDIAN-BASED AGGREGATION

- Beyond Median
- $\rho = MeaMed(\{\tilde{v}_i : i \in [n]\})$
- ρ의 j번째 차원은 $\rho_j = \frac{1}{n-q}\Sigma_{\mu_j \to i}(\tilde{v}_i)j$
- $\mu_j \to i$는 중앙값 μ_j에 가장 가까운 top-$(n-q)$ 값들

TIME COMPLEXITY

- Geometric Median
  - Closed-form 해법은 없음
  - $(1+\epsilon)$ 추정을 통하면 $O(dnlog^3\frac{1}{\epsilon})$에 가능
  - $O(dn)$과 유사

TIME COMPLEXITY

- Marginal Median
  - 각 차원에 대한 정렬 알고리즘이 필요하므로 $O(dn \log n)$
- 각 중앙값 선출을 위해 Selection algorithm을 사용해
  - 평균 시간 복잡도가 $O(n)$
  - 최악의 경우 $O(n^2)$이 되도록 할 수도 있음
- 따라서 평균적으로 $O(dn)$

TIME COMPLEXITY

- Beyond Median
  - 시간 복잡도는 Marginal median과 동일

EXPERIMENTS

EXPERIMENT

- 수렴성과 비잔틴 탄력성을 평가
  - 제안한 방법들에 대해

EXPERIMENT

- 두 종의 이미지 분류 tasks를 고려
- MNIST
  - 은닉층 두 개의 MLP (multi-layer perceptron)
- 물체 인식
  - 5개의 컨볼루션 레이어, 2개의 전연결층으로 구성된 CNN

EXPERIMENT

- 20명의 워커
- 10번 실험하고 평균을 구함
- 랜덤 시드는 고정해둠
- 평가에는 top-1 또는 top-3 정확도를 사용

EXPERIMENT

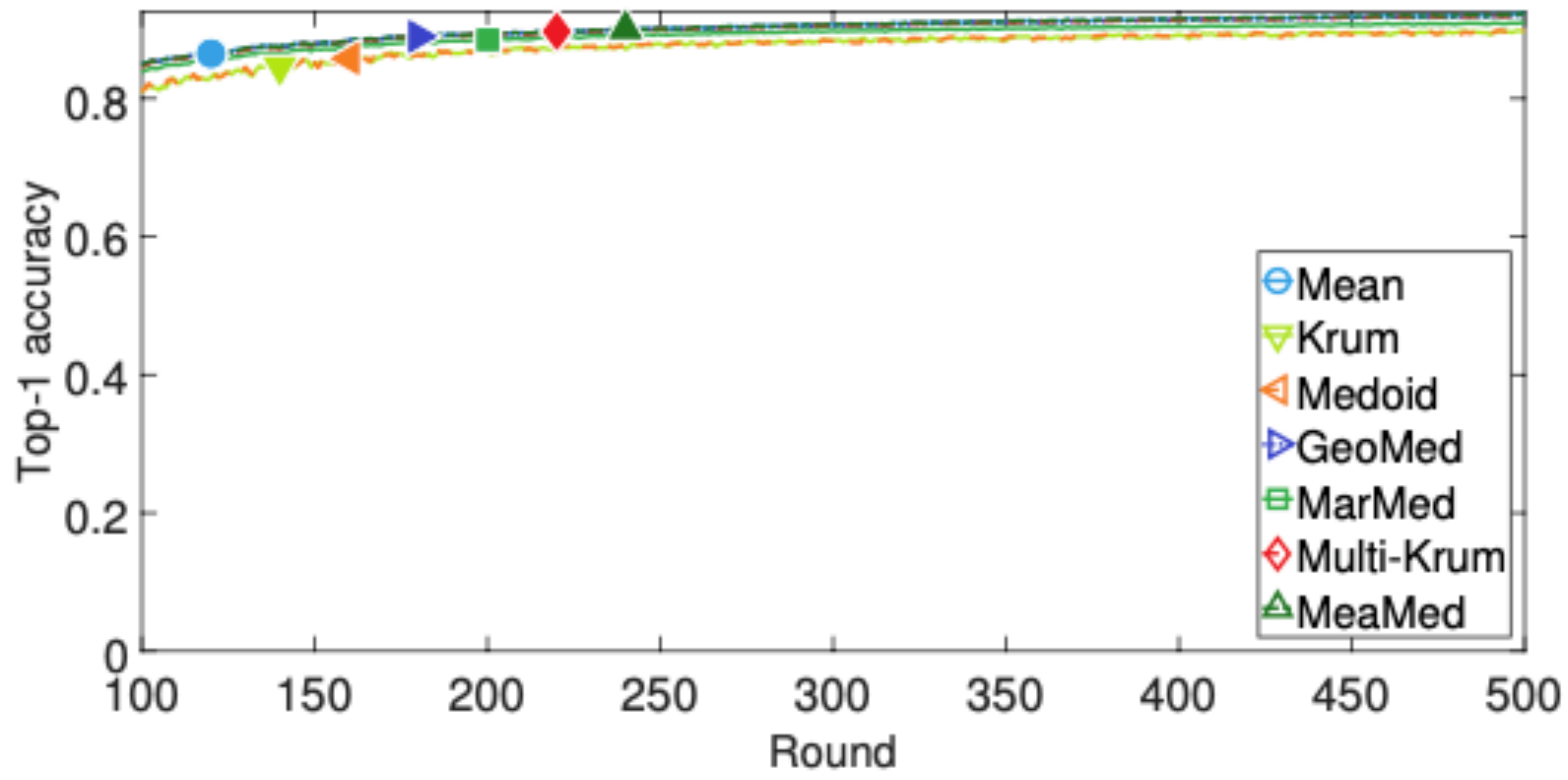
- 요약:

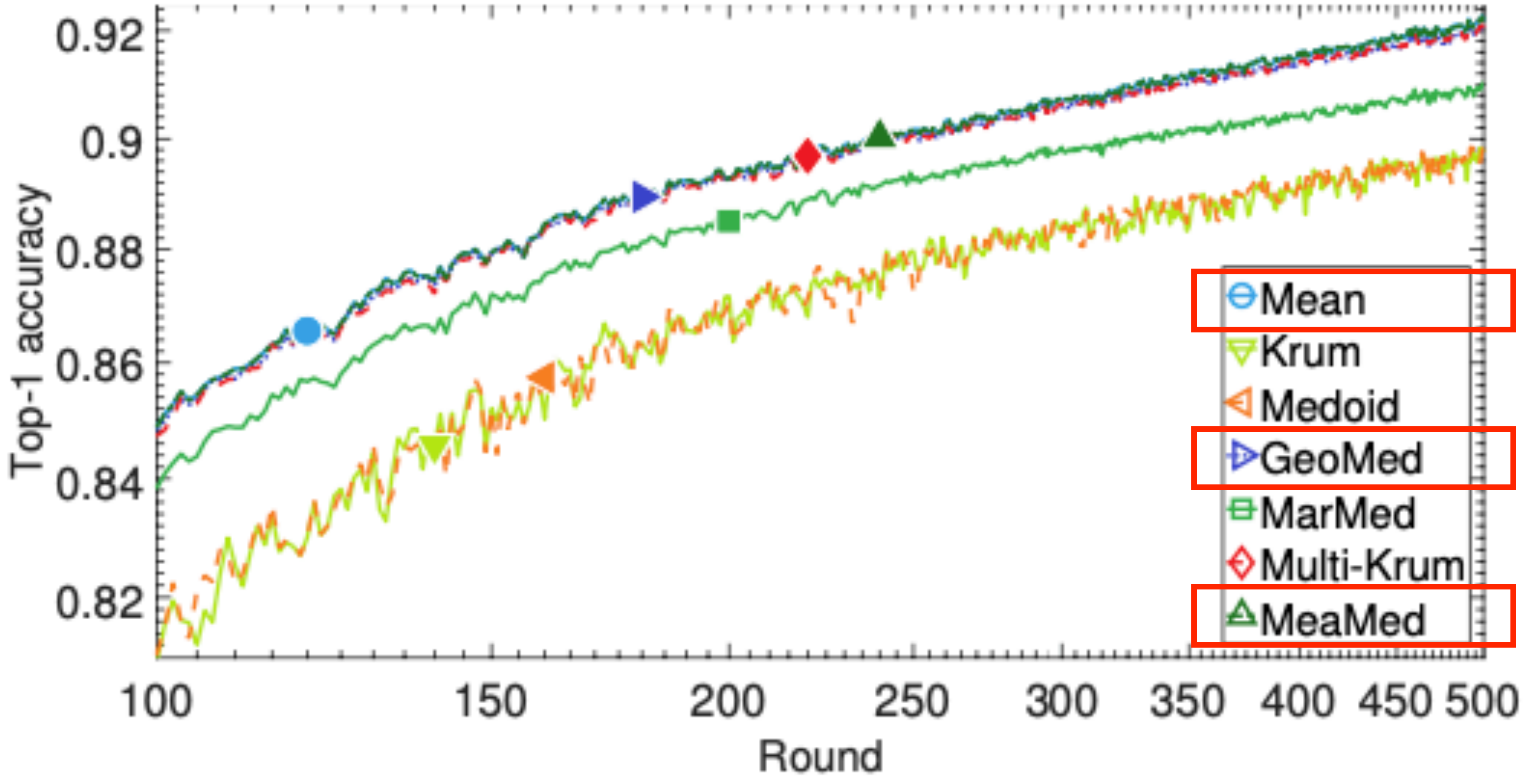
- *Table 1.* Experiment Summary

Dataset	# train	# test	γ	# rounds	Batchsize	Evaluation metric
MNIST (Loosli et al., 2007)	60k	10k	0.1	500	32	top-1 accuracy
CIFAR10 (Krizhevsky & Hinton, 2009)	50k	10k	5e-4	4000	128	top-3 accuracy

EXPERIMENT

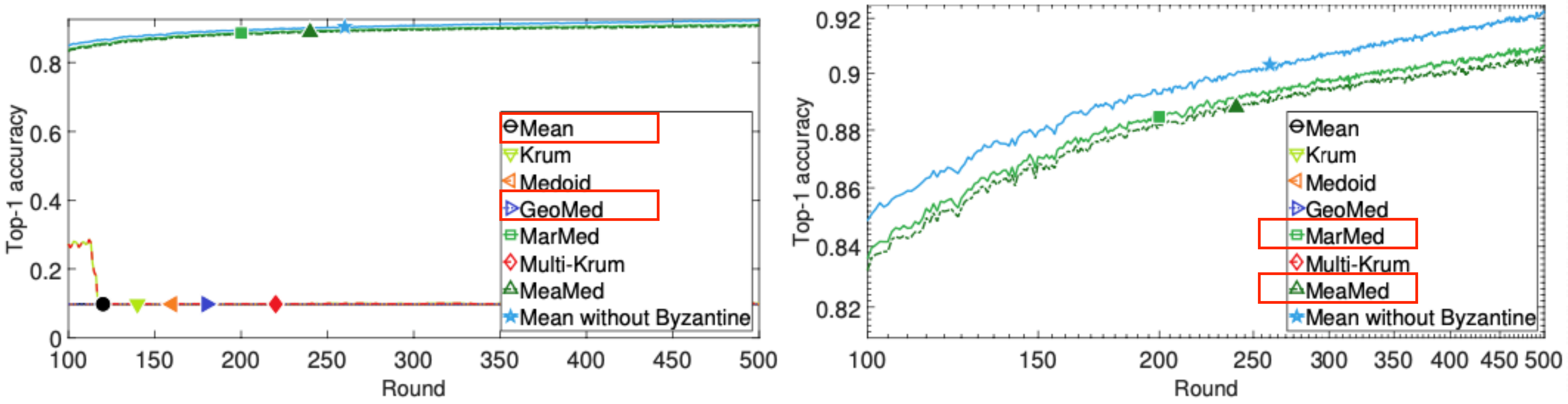
- MNIST, 비잔틴이 없을 때의 Top-1 정확도





EXPERIMENT

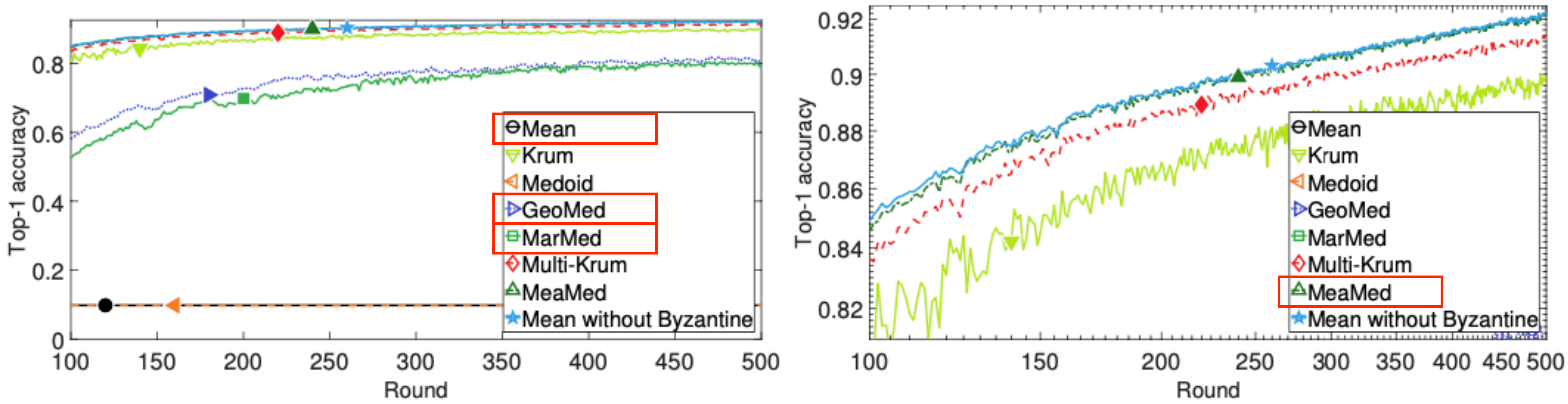
- MNIST, Gambler 공격이 있을 때의 Top-1 정확도



- 파라미터를 20등분, 하나에 대해 0.05%의 확률로 $-1e20$이 곱해짐

EXPERIMENT

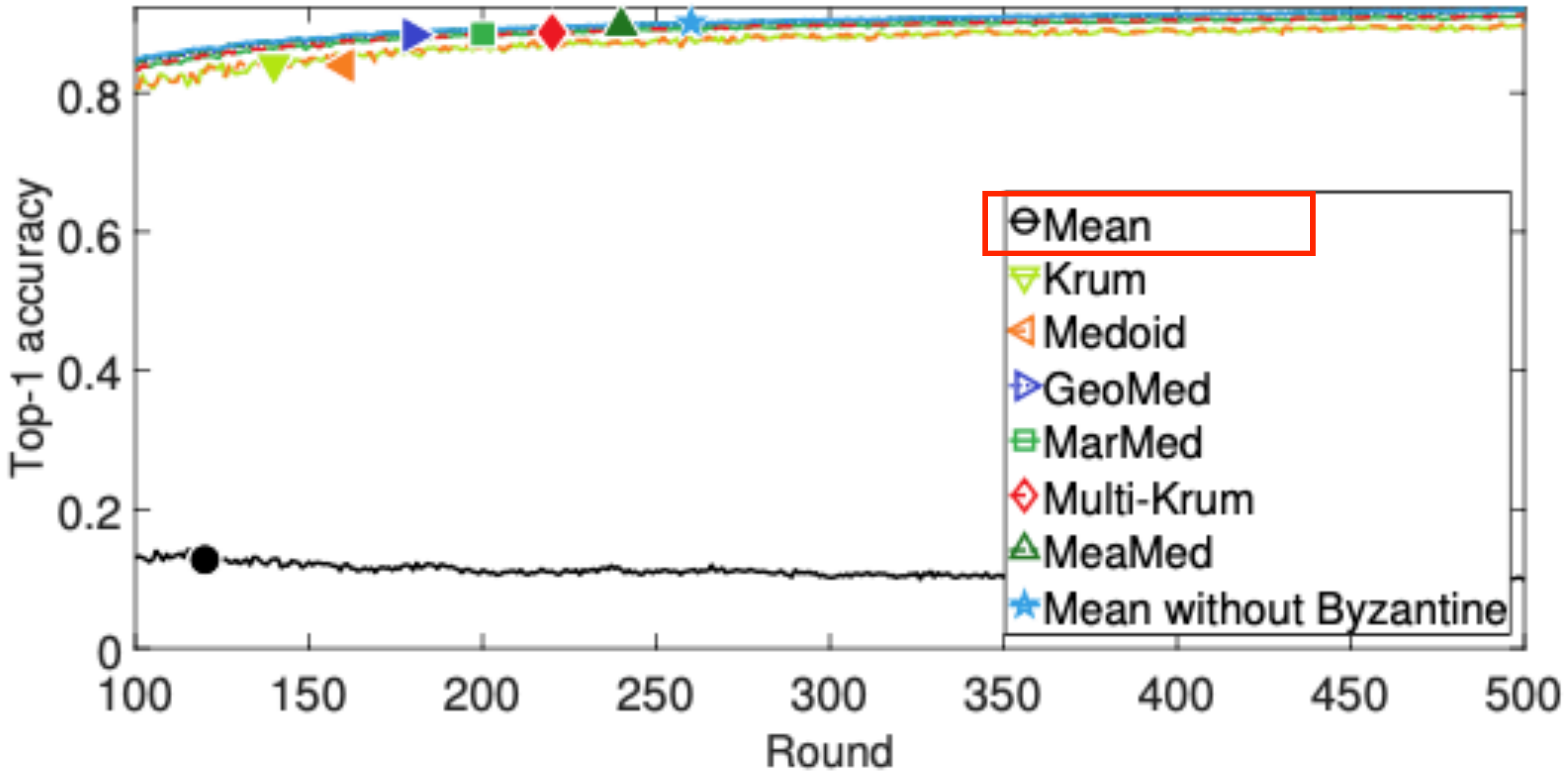
- MNIST, Omniscient 공격이 있을 때의 Top-1 정확도

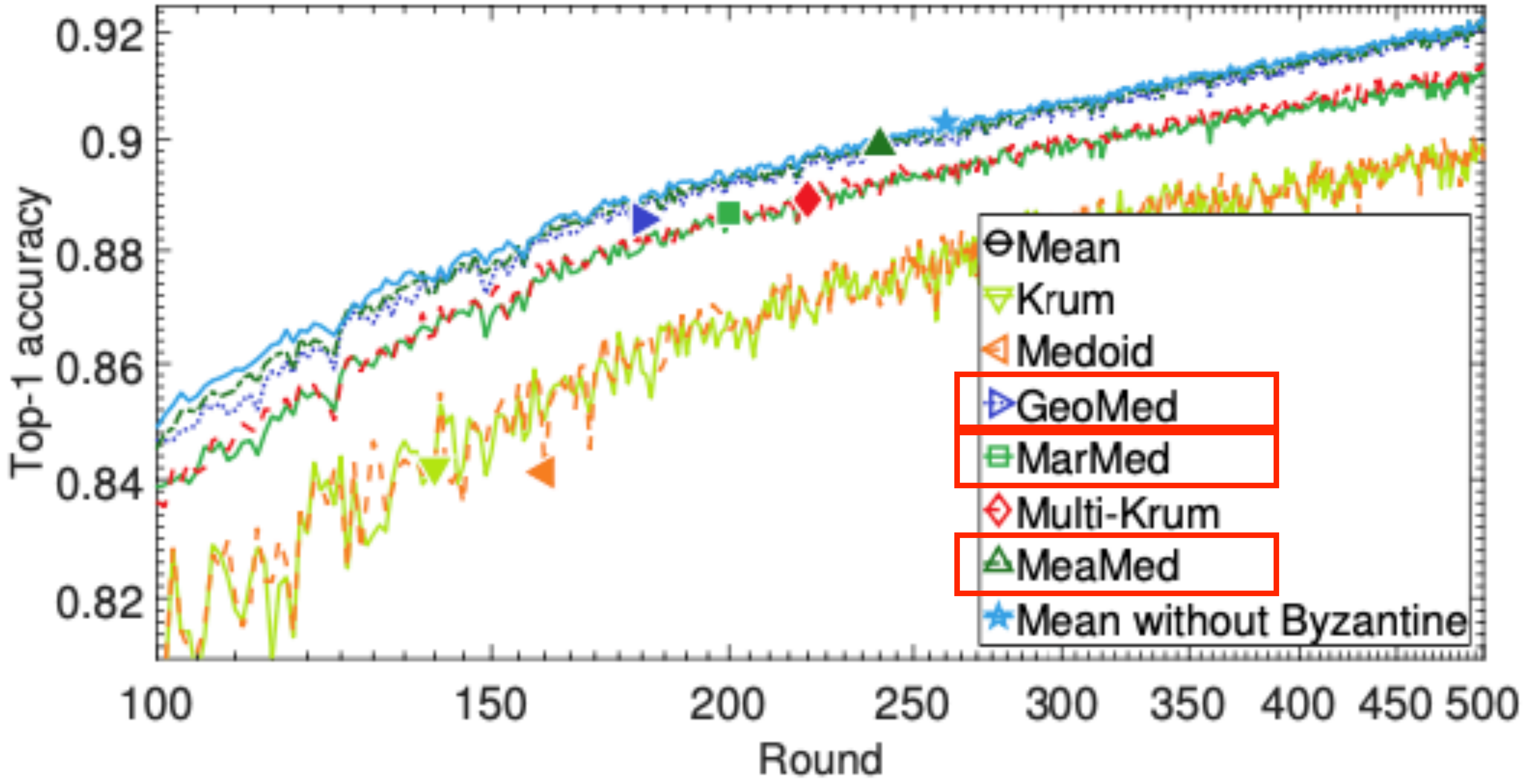


- 20개의 벡터 중 비잔틴으로부터 6개가 교체됨

EXPERIMENT

- MNIST, Gaussian 공격이 있을 때의 Top-1 정확도





- 20개의 벡터 중 비잔틴으로부터 6개가 교체됨

DISCUSSION

- 예상대로, 평균(mean) 방법은 비잔틴 탄력성이 없음
- GeoMed
  - 전통적인 비잔틴 탄력성은 있으나
  - Dimensional 비잔틴 탄력성은 없음
- MarMed와 MeaMed는 Dimensional 비잔틴 탄력성이 있음
  - 그러나 Omniscient 공격에서 MarMed는 수렴이 늦음

CONCLUSION

CONCLUSION

- PS 구조에서의 일반화된 비잔틴 탄력성을 소개
- 동기 SGD를 위한 3가지의 중앙값 기반 통합 규칙을 제안
- 이 방법들은 낮은 시간 복잡도를 가짐
- 실제로 좋은 성능을 보임

BYZANTINE-TOLERANT SGD

FOR DISTRIBUTED SYNCHRONOUS SGD