Perspectives on Algorithmic, Structural, and Pragmatic Acceleration Techniques in Machine Learning and Quantum Computing

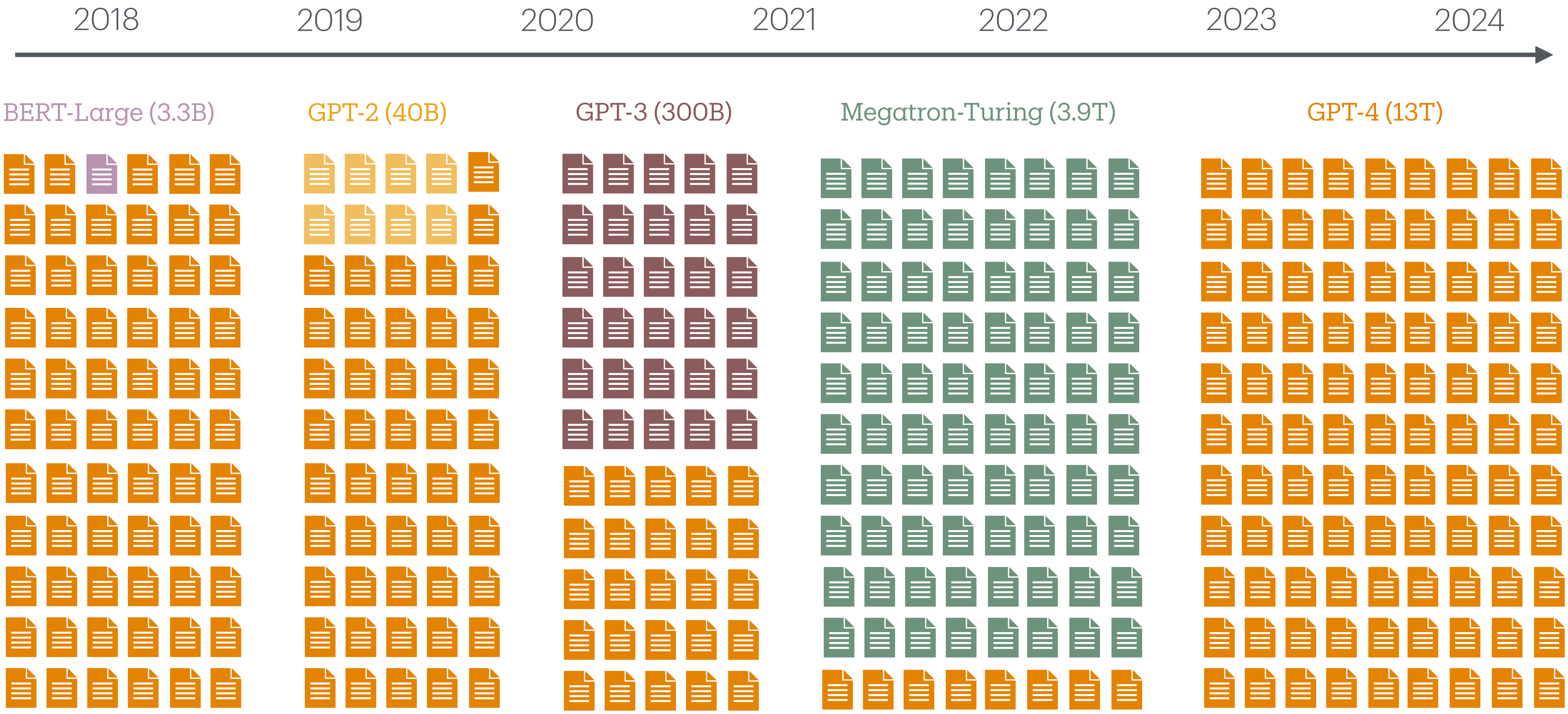
Ph.D. Defense, 23 April 2024

Junhyung Lyle Kim
김준형 🇰🇷

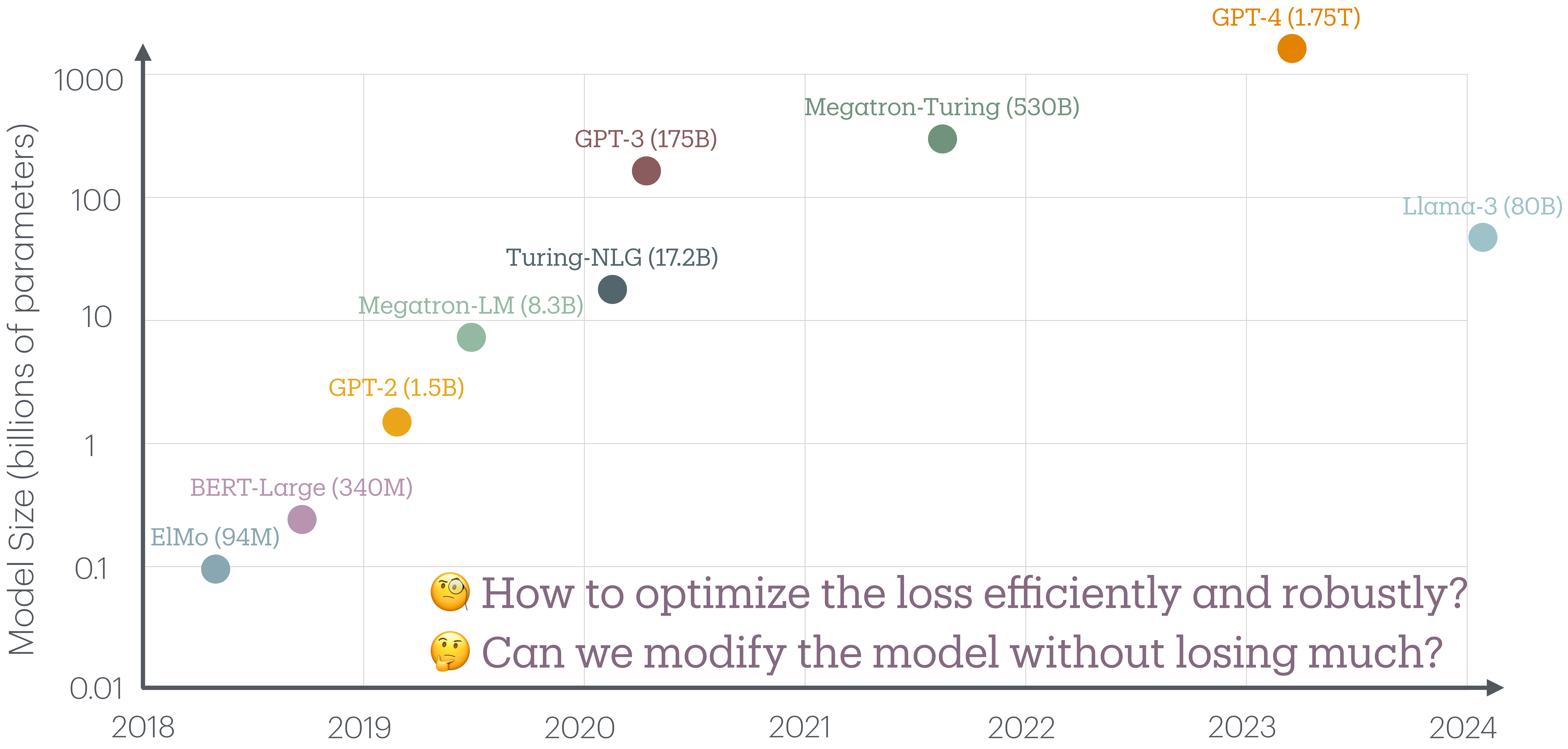
Committee:
Anastasios Kyrillidis (Rice CS, Chair)
César A. Uribe (Rice ECE)
Nai-Hui Chia (Rice CS)

Computer Science Department RICE

What is the trend of the dataset size in modern ML / AI ?



What about the size of the model? (# of parameters)



🧐 How to optimize the loss efficiently and robustly?

🤔 Can we modify the model without losing much?

What about Quantum Computing?

qubits ⚛ ⚛⚛ ⚛⚛⚛ ⚛⚛⚛⚛ ⚛⚛⚛⚛⚛ ⚛ × 6

⚛ × 100 ⇒

1.26765×10^{30}

complex numbers required to describe the system classically!

🧐 How to process meaningfully in the current era?

🤔 How to even store this amount of data?

Exploding Data: Stochastic Gradient Descent (SGD)

Number of data

Size of model

$$\min_{X \in \mathbb{R}^{p \times p}} f(X) = \frac{1}{n} \sum_{i=1}^{n} f_i(X)$$

$f(\cdot)$ evaluated at X with i-th data point

Gradient Descent: $X_{t+1} = X_t - \eta \nabla f\left(X_t\right)$

$$\nabla f(X_t) = \cancel{\frac{1}{n} \sum_{i=1}^{n}} \nabla f_i(X_t)$$

Stochastic Gradient Descent: $X_{t+1} = X_t - \eta \nabla f_{i_t}\left(X_t\right)$

$$\approx \nabla f_{i_t}(X_t)$$

BUT:

- η too small: SGD takes a long time to converge
- η too large: SGD numerically unstable / diverge

- Proximal/implicit update
- Adaptive step-size

Exploding model parameter: Factored GD

Number of data

Size of model

$$\min_{X \in \mathbb{R}^{p \times p}} f(X) = \frac{1}{n} \sum_{i=1}^{n} f_i(X)$$

$f(\cdot)$ evaluated at X with i-th data point

Gradient Descent: $X_{t+1} = X_t - \eta \nabla f\left(X_t\right)$

Low-rank Matrix Factorization:

Represent $X = UU^\top$

$$\min_{U \in \mathbb{R}^{p \times r}} f(UU^\top) = \frac{1}{n} \sum_{i=1}^{n} f_i(UU^\top)$$

Factored Gradient Descent: $U_{t+1} = U_t - \eta \nabla f\left(U_t U_t^\top\right) \cdot U_t$

- Momentum
- Stochasticity
- Distributed

Road Map: Algorithmic, Structural, and Pragmatic Acceleration

Algorithmic Pragmatic

Stochastic Proximal Point Method with Momentum

Keyword: implicit method, acceleration, stability

[**J. L. Kim,** P. Toulis, A. Kyrillidis. "Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum," **L4DC 2022**]

“Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum”

J. L. Kim (Rice CS)
P. Toulis (UChicago Booth)
A. Kyrillidis (Rice CS)

Published in “Learning for Dynamics and Control Conference” (L4DC), 2022.

Empirical risk minimization and SGD/SGDM

$$x_{t+1} = x_t - \eta \nabla f_{i_t}\left(x_t\right)$$

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\nabla f(x_t) = \cancel{\frac{1}{n}\sum_{i=1}^{n}} \nabla f_i(x_t) \approx \nabla f_{i_t}(x_t)$$

BUT:

1. SGD can take long to converge
2. SGD can be numerically unstable if step size is misspecified

GD: $f\left(x_t\right) - f^\star = O\left(\frac{1}{t}\right)$

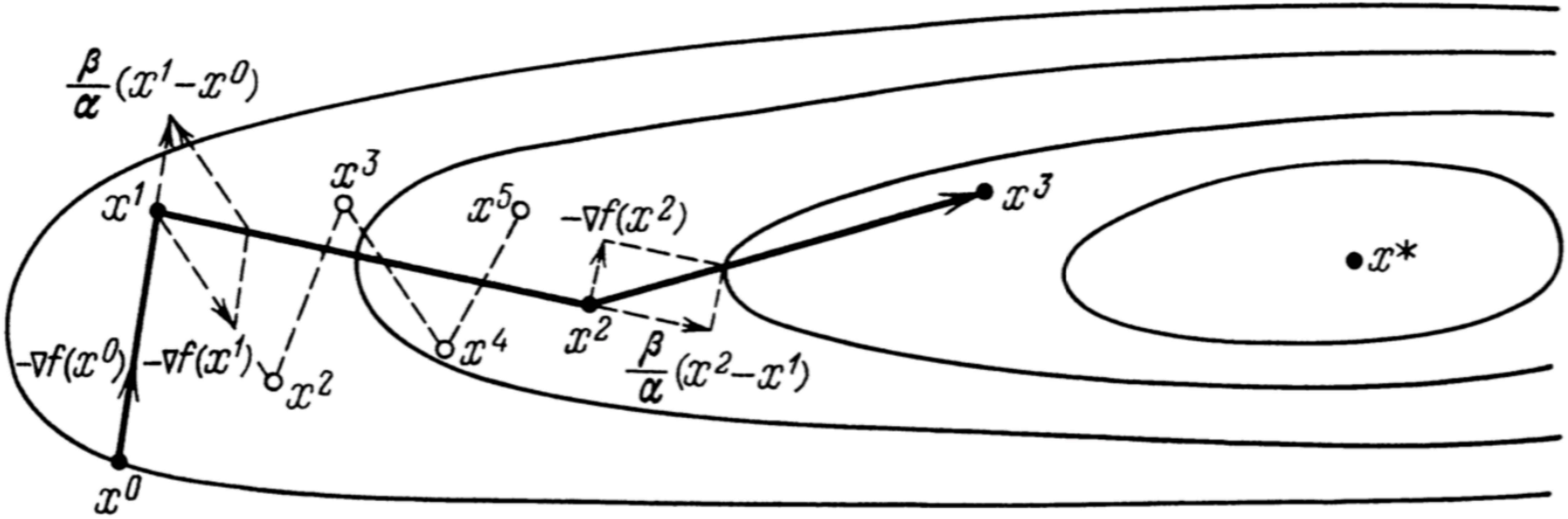
SGD: $f\left(x_t\right) - f^\star = O\left(\frac{1}{\sqrt{t}}\right)$

$$\mathbb{E}\|x_t - x^\star\|_2^2 \leq 2\exp(4L^2\eta_1^2\log(t))\|x_0 - x^\star\|_2^2 \cdots$$

[Bach and Moulines (2011). "Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning"]

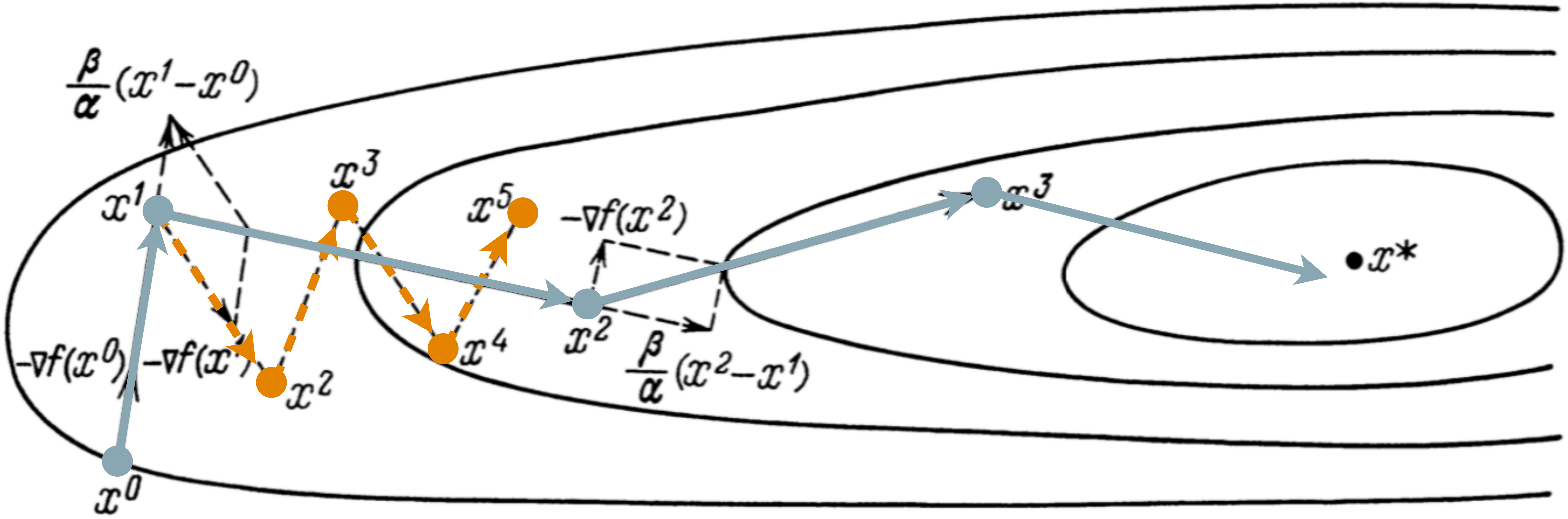
Empirical risk minimization and SGD/SGDM

$$x_{t+1} = x_t - \eta \nabla f_{i_t}(x_t) + \beta(x_t - x_{t-1})$$



Geometric intuition of momentum

- - - -	Gradient Descent	$x_{t+1} = x_t - \eta \nabla f\left(x_t\right)$
——	Gradient Descent with Momentum	$x_{t+1} = x_t - \eta \nabla f\left(x_t\right) + \beta(x_t - x_{t-1})$

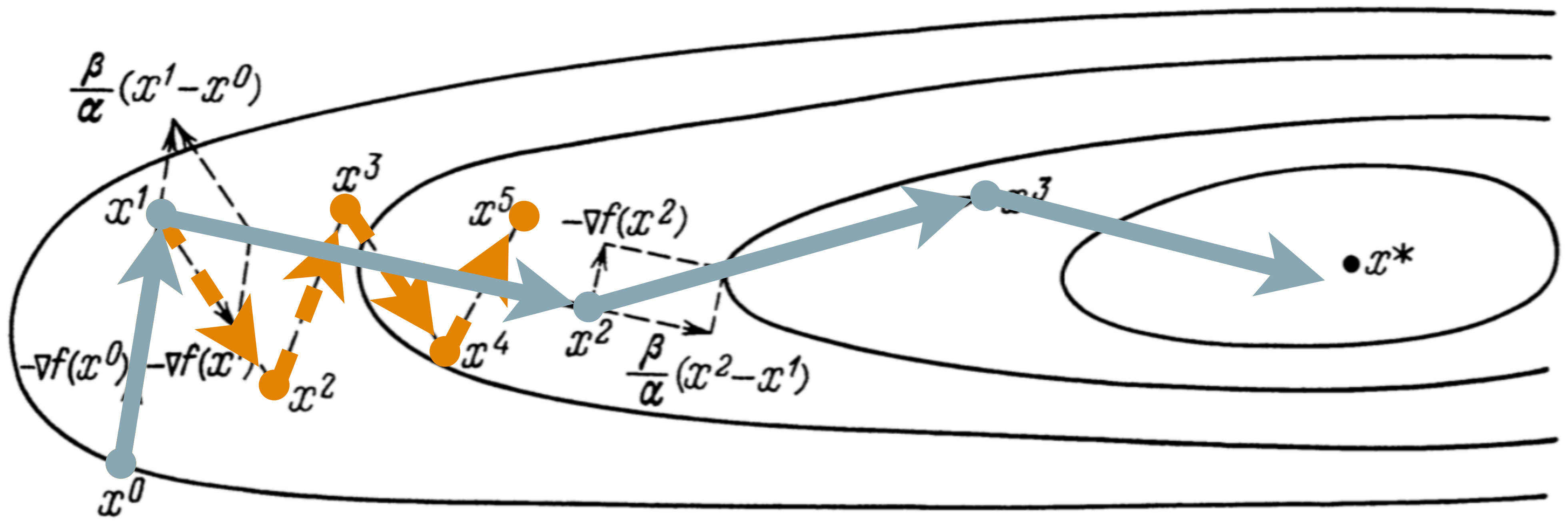


Empirical risk minimization and SGD/SGDM

$$x_{t+1} = x_t - \eta \nabla f_{i_t}(x_t) + \beta(x_t - x_{t-1})$$

Momentum parameter

BUT:



1. SGDM/accelerated SGD may diverge for step sizes that SGD converges

[Liu and Belkin (2019). "Accelerating SGD with momentum for over-parameterized learning"]
[Kidambi, Rahul et al. (2018). "On the insufficiency of existing momentum schemes for Stochastic Optimization"]

2. Accelerated SGD may diverge even for quadratic objectives with usual choices of η and β

[Assran and Rabbat (2020). "On the Convergence of Nesterov's Accelerated Gradient Method in Stochastic Settings"]

Is there other method that is *numerically stable* AND *fast*?

Stochastic proximal point algorithm

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^p} \left\{ f_{i_t}(x) + \frac{1}{2\eta} \|x - x_t\|_2^2 \right\} \qquad f(x) = \cancel{\frac{1}{n}\sum_{i=1}^{n}} f_i(x)$$

$$x_{t+1} = x_t - \eta \nabla f_{i_t}\left(x_{t+1}\right)$$

- SPPA enjoys the same convergence rate as SGD
- SPPA is much more numerically stable

$$\text{SGD} : \mathbb{E}\|x_t - x^\star\|_2^2 \leq 2\exp(4L^2\eta_1^2\log(t))\|x_0 - x^\star\|_2^2 \cdots$$

$$\text{SPPA} : \mathbb{E}\|x_t - x^\star\|_2^2 \leq \exp(-\log(1 + 2\eta_1\mu)\log(t))\|x_0 - x^\star\|_2^2 \cdots$$

[Ryu and Boyd (2017). "Stochastic Proximal Iteration: A Non-Asymptotic Improvement Upon Stochastic Gradient Descent"]
[Toulis et al. (2021) "The proximal Robbins–Monro method"]

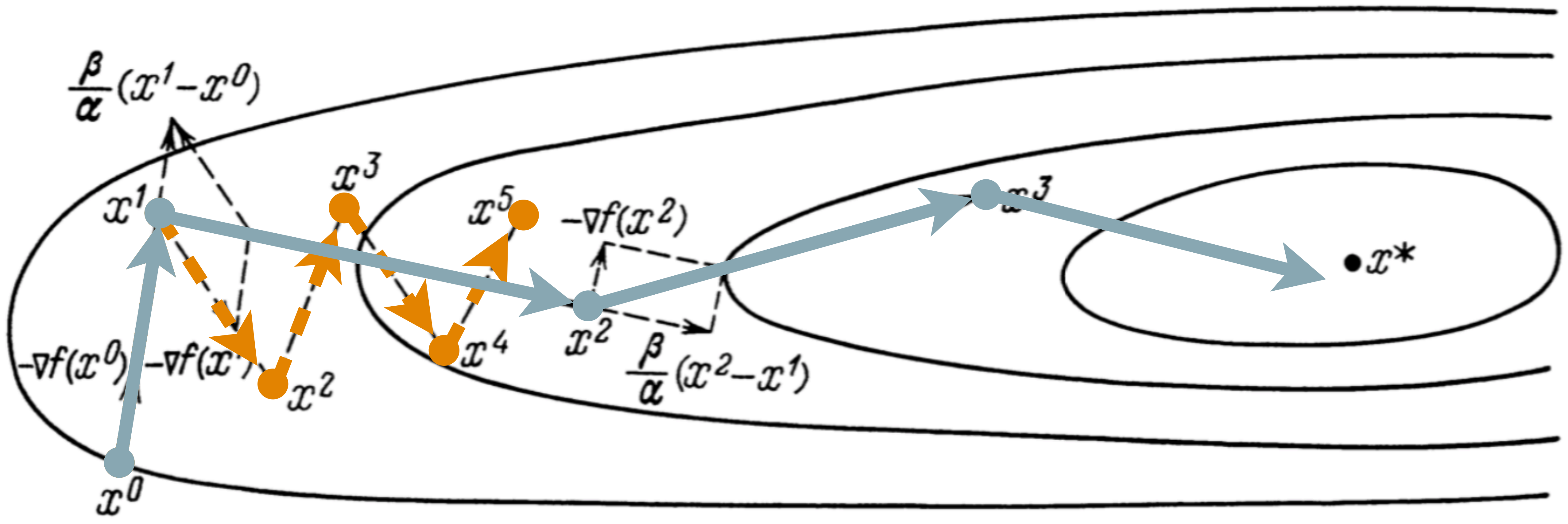
Can we accelerate SPPA while preserving numerical stability?

SPPAM : Stochastic Proximal Point Algorithm with Momentum

$$x_{t+1} = x_t - \eta\left(\nabla f(x_{t+1}) + \varepsilon_{t+1}\right) + \beta\left(x_t - x_{t-1}\right)$$

- Disregarding the stochastic error for simplicity, above can be written as the solution to:

$$\arg\min_{x\in\mathbb{R}^p}\left\{f(x)+\frac{1}{2\eta}\|x-x_t\|_2^2-\frac{\beta}{\eta}\langle x_t-x_{t-1},x\rangle\right\}$$



SPPAM: theoretical analysis

Assumption 1: The objective function f is a μ-strongly convex function. That is, for some fixed $\mu > 0$ and for all x and y,

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|_2^2$$

Assumption 2: For SPPAM, there exists fixed $\sigma^2 > 0$ such that, given the natural filtration $\mathcal{F}_{t-1}$,

$$\mathbb{E}\left[\varepsilon_t \mid \mathcal{F}_{t-1}\right] = 0 \quad \text{and} \quad \mathbb{E}\left[\|\varepsilon_t\|^2 \mid \mathcal{F}_{t-1}\right] \leq \sigma^2 \quad \text{for all } t.$$

Theorem 1: Suppose Assumptions 1 and 2 hold. SPPAM satisfies the following iteration invariant bound:

$$\mathbb{E}[\|x_{t+1} - x^\star\|_2^2] \leq \frac{4}{(1+\eta\mu)^2}\mathbb{E}[\|x_t - x^\star\|_2^2] + \frac{4\beta^2}{(1+\eta\mu)^2\big(4-(1+\beta)^2\big)}\mathbb{E}[\|x_{t-1} - x^\star\|_2^2] + \eta^2\sigma^2$$

$$\begin{bmatrix} \mathbb{E}\|x_{t+1} - x^\star\|_2^2 \\ \mathbb{E}\|x_t - x^\star\|_2^2 \end{bmatrix} \leq A \cdot \begin{bmatrix} \mathbb{E}\|x_t - x^\star\|_2^2 \\ \mathbb{E}\|x_{t-1} - x^\star\|_2^2 \end{bmatrix} + \begin{bmatrix} \eta^2\sigma^2 \\ 0 \end{bmatrix} \qquad A = \begin{bmatrix} \frac{4}{(1+\eta\mu)^2} & \frac{4\beta^2}{(1+\eta\mu)^2\big(4-(1+\beta)^2\big)} \\ 1 & 0 \end{bmatrix}$$

Lemma: The maximum eigenvalue of A, which determines the convergence rate of SPPAM, is:

$$\frac{2}{(1+\eta\mu)^2} + \sqrt{\frac{4}{(1+\eta\mu)^4} + \frac{4\beta^2}{(1+\eta\mu)^2(4-(1+\beta)^2)}} \approx O\left(\frac{1}{\eta^2}\right)$$

Corollary 1: For μ-strongly convex f, SPPAM enjoys smaller contraction factor than SPPA if:

$$\frac{4\beta^2}{4-(1+\beta)^2} < \frac{\eta^2\mu^2 - 6\eta\mu - 3}{(1+\eta\mu)^2}$$

VS. Contraction factor of SPPA: $\frac{1}{1+2\eta\mu} \approx O\left(\frac{1}{\eta}\right)$

Acceleration

SPPAM: theoretical analysis

Stability (w.r.t. hyperparameters)

$$\begin{bmatrix}\mathbb{E}\|x_{t+1}-x^\star\|_2^2\\ \mathbb{E}\|x_t-x^\star\|_2^2\end{bmatrix} \leq A\cdot\begin{bmatrix}\mathbb{E}\|x_t-x^\star\|_2^2\\ \mathbb{E}\|x_{t-1}-x^\star\|_2^2\end{bmatrix}+\begin{bmatrix}\eta^2\sigma^2\\ 0\end{bmatrix} \longrightarrow \begin{bmatrix}\mathbb{E}\|x_T-x^\star\|_2^2\\ \mathbb{E}\|x_{T-1}-x^\star\|_2^2\end{bmatrix} \leq \boxed{A^T}\cdot\begin{bmatrix}\|x_0-x^\star\|_2^2\\ \|x_{-1}-x^\star\|_2^2\end{bmatrix}+\boxed{\left(\sum_{i=1}^{T-1}A^i\right)}\begin{bmatrix}1\\ 0\end{bmatrix}\eta^2\sigma^2.$$

Theorem 2: Suppose Assumptions 1 (f is μ-strongly convex) and 2 hold. Further, suppose that SPPAM is initialized with $x_0 = x_{-1}$. Then, after T iterations, SPPAM satisfies:

$$\mathbb{E}\|x_T-x^\star\|_2^2 \leq \boxed{\frac{2\sigma_1^T}{\sigma_1-\sigma_2}}\left(\left(\|x_0-x^\star\|_2^2+\frac{\eta^2\sigma^2}{1-\theta}\right)\cdot(1+\theta)\right)+\frac{\eta^2\sigma^2}{1-\theta},$$

where $\theta = \frac{4}{(1+\eta\mu)^2}+\frac{4\beta^2}{(1+\eta\mu)^2(4-(1+\beta)^2)}$. Further, $\sigma_{1,2}$ are the eigenvalues of A, and

$$\boxed{\frac{2\sigma_1^T}{\sigma_1-\sigma_2} = \tau^{-1}\cdot\left(\frac{2}{(1+\eta\mu)^2}+\tau\right)^T}$$

with $\tau = \sqrt{\frac{4}{(1+\eta\mu)^4}+\frac{4\beta^2}{(1+\eta\mu)^2(4-(1+\beta)^2)}}$.

Corollary 2: Suppose the following condition hold:

$$\boxed{\tau = \sqrt{\frac{4}{(1+\eta\mu)^4}+\frac{4\beta^2}{(1+\eta\mu)^2(4-(1+\beta)^2)}} < \frac{1}{2}.}$$

Then, under Assumptions 1 and 2, the initial conditions of SPPAM exponentially discount. That is,

🧐 Easy to satisfy?

$$\frac{2\sigma_1^T}{\sigma_1-\sigma_2} = \tau^{-1}\cdot\left(\frac{2}{(1+\eta\mu)^2}+\tau\right)^T = C^T, \quad \text{where} \quad C\in(0,1).$$

Unfair comparison [Assran and Rabbat (2020)]

Accelerated SGD (strongly convex *quadratic*):

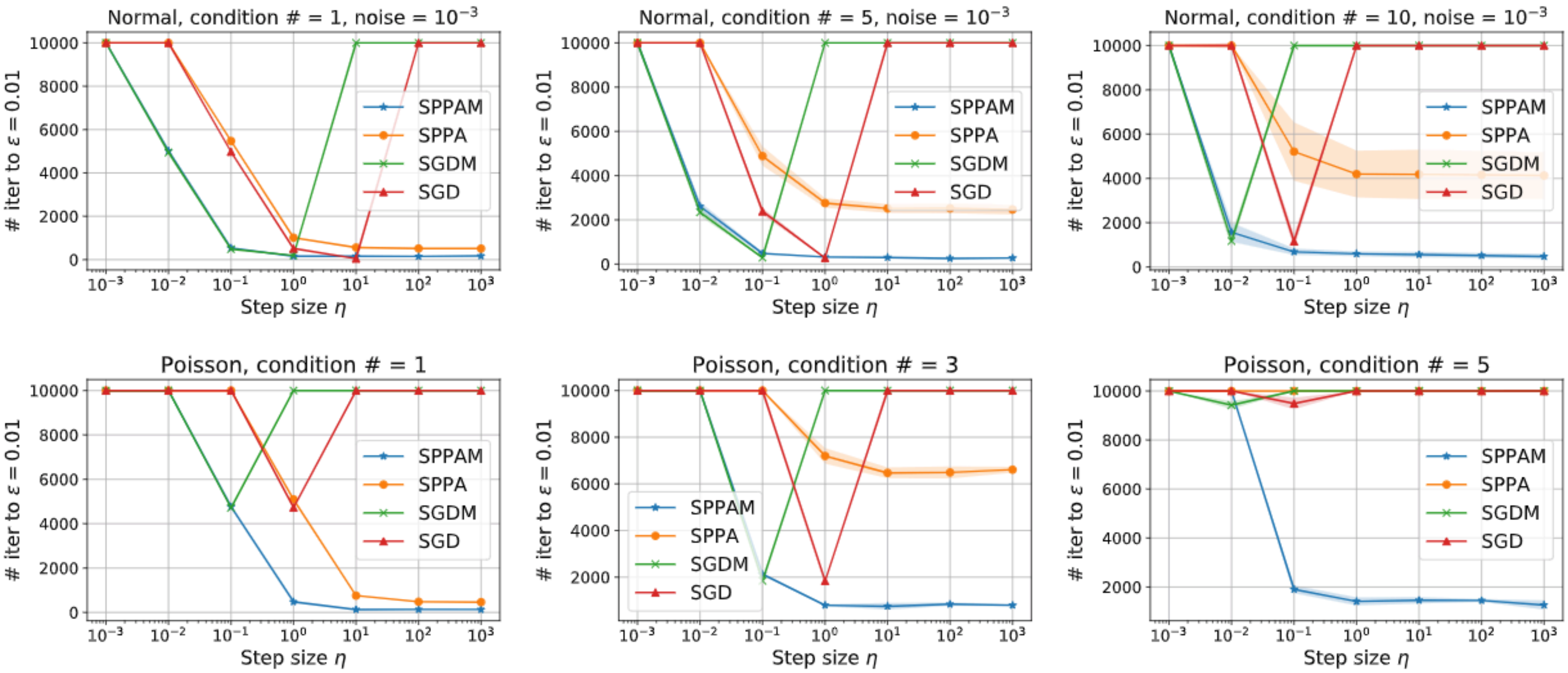
$0.0028 \approx \frac{1}{361} \leq \eta\lambda \leq \frac{24}{19} \approx 1.26$ for $\lambda \in \{\mu, L\}$

SPPAM (strongly convex):

$\eta\mu > 4.81$ with $\beta = 0.9$

Experiments

1. SPPAM converges faster when SPPA converges (acceleration)
2. SPPAM converges at the same rate as SGDM when the latter converges but for much wider range (stability)



"Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum"

J. L. Kim, P. Toulis, A. Kyrillidis., L4DC 2022.

1. Acceleration — SPPAM converges faster than SPPA

2. Stability — SPPAM converges for wider range of hyperparameters than SGD/SGDM

3. Above holds both in theory and practice

But what if we cannot implement $x_{t+1} = x_t - \eta \nabla f(x_{t+1})$?

Road Map: Algorithmic, Structural, and Pragmatic Acceleration

Algorithmic Pragmatic

Stochastic Proximal Point Method with Momentum

Keyword: implicit method, acceleration, stability

[**J. L. Kim,** P. Toulis, A. Kyrillidis. "Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum," **L4DC 2022**]

"Plug-and-play"

Adaptive step size

Structural Pragmatic

Adaptive Federated Learning with Auto-Tuned Clients

Keyword: adaptive step-size, federated learning

[**J. L. Kim**, M. T. Toghani, C. A. Uribe, A. Kyrillidis. "Adaptive Federated Learning with Auto-Tuned Clients", **ICLR 2024**]

“Adaptive Federated Learning with Auto-tuned Clients”

J. L. Kim (Rice CS)
M. T. Toghani (Rice ECE)
C. A. Uribe (Rice ECE)
A. Kyrillidis (Rice CS)

To appear in “International Conference on Learning Representations” (ICLR), 2024.

What is Federated Learning?

- Distributed ML framework where a **global model** is trained via multiple collaborative steps by participating **clients**, without sharing data (E.g., next word prediction in smartphone).

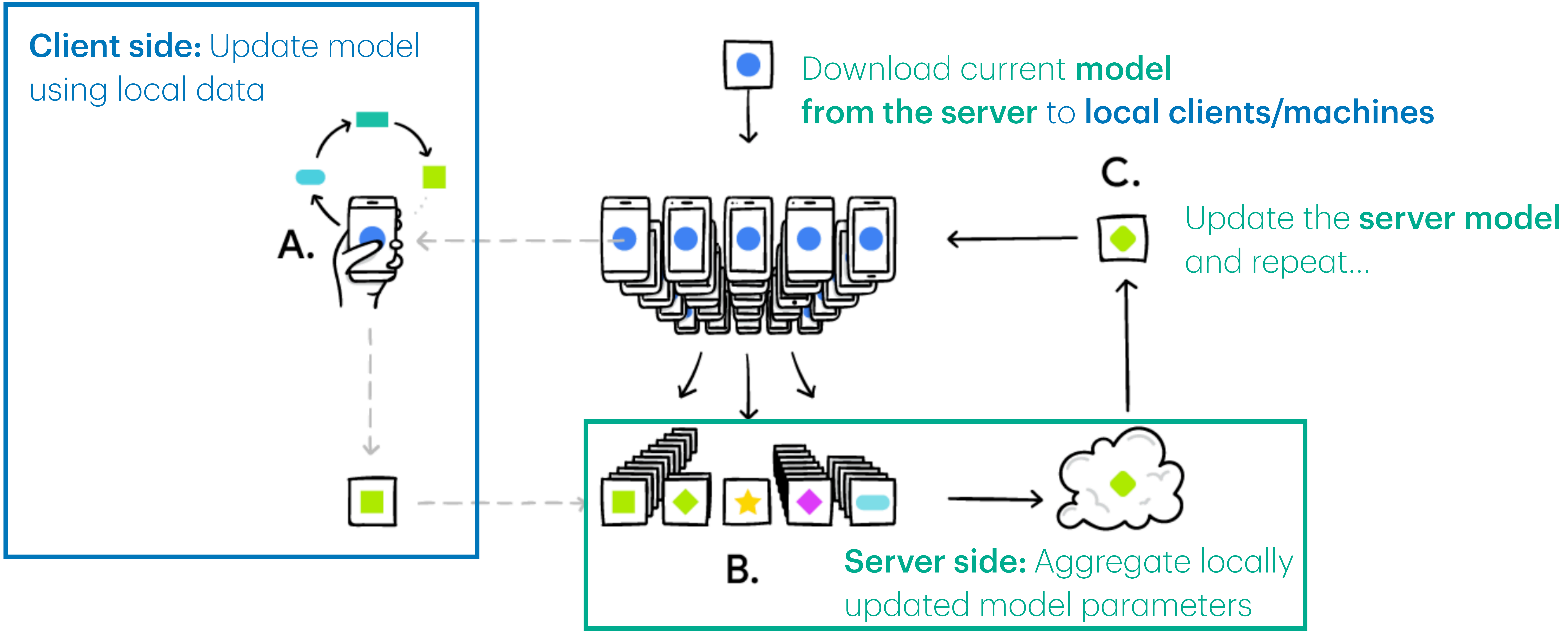


Image source: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

What is Federated Learning?

Number of clients

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

Shared model parameter

Individual loss function:

$$f_i(x) := \mathbb{E}_{z \sim \mathcal{D}_i}[F_i(x, z)]$$

Why Federated Learning?

$f_i(\cdot)$ *differs for each client* i

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x) \leftarrow \mathbb{E}_{z \sim \mathcal{D}_i}[F_i(x, z)]$$

Flexiblity

- Number of clients m
- Client participation rate
- Computing power

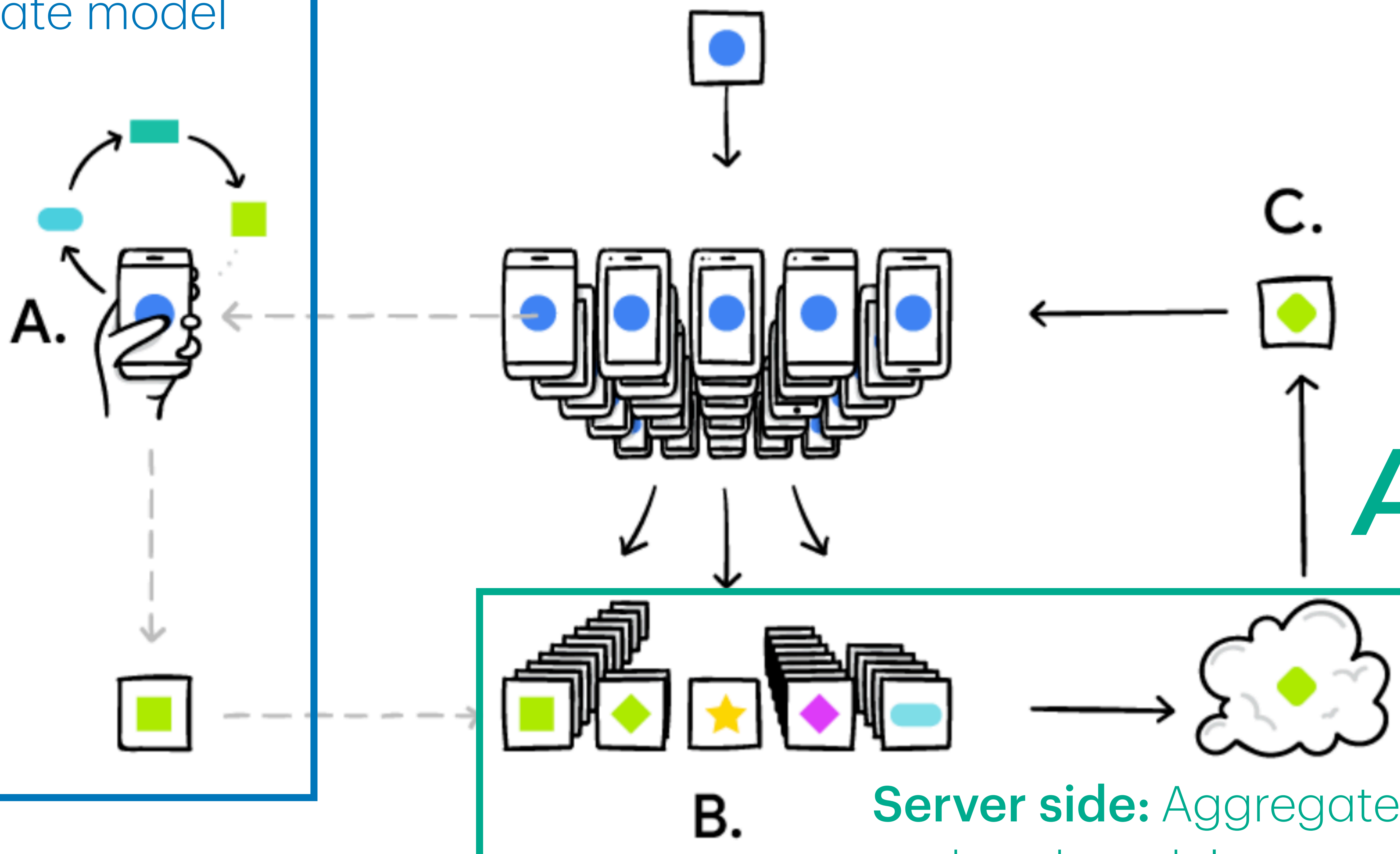
Privacy

- Local data never shared
- $\mathcal{D}_i$ differs for each client i
- Also number of samples $z \sim \mathcal{D}_i$

[McMahan et al., 2017 "Communication-Efficient Learning of Deep Networks from Decentralized Data."]

Client side: Update model using local data

SGD



Averaging

Server side: Aggregate locally updated model parameters

Image source: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Main challenges in Federated Learning

Algorithm 2 FedAvg

input: $x_0 \in \mathbb{R}^d$, $\eta > 0$, and $p \in (0, 1)$.
for each round $t = 0, 1, \ldots, T-1$ **do**
  sample a subset $\mathcal{S}_t$ of clients with size $|\mathcal{S}_t| = p \cdot m$
  for each machine in parallel for $i \in \mathcal{S}_t$ **do**
    Set $x^i_{t,0} = x_t$
    for local step $k \in [K]$ **do**
      Compute an estimate $g^i_{t,k-1}$ of $\nabla f_i(x^i_{t,k-1})$
      $x^i_{t,k} = x^i_{t,k-1} - \eta g^i_{t,k-1}$
    end for
  end for
  $x_{t+1} = \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} x^i_{t,K}$
end for
return x_T

Client side: How do we make sure each client meaningfully "learns" using local data?

E.g., How do we tune η?

Does it make sense to use "same" η for all clients?

If not, how do we tune individual step size η_i?

Server side: How do we smartly aggregate the local information coming from each participating client?

Federated optimization framework

Algorithm 2 FedAvg

1: **input**: $x_0 \in \mathbb{R}^d$, $\eta > 0$, and $p \in (0,1)$.
2: **for** each round $t = 0, 1, \ldots, T-1$ **do**
3: sample a subset $\mathcal{S}_t$ of clients with size $|\mathcal{S}_t| = p \cdot m$
4: **for** each machine in parallel for $i \in \mathcal{S}_t$ **do**
5: Set $x^i_{t,0} = x_t$
6: **for** local step $k \in [K]$ **do**
7: Compute an estimate $g^i_{t,k-1}$ of $\nabla f_i(x^i_{t,k-1})$
8: $x^i_{t,k} = x^i_{t,k-1} - \eta g^i_{t,k-1}$
9: **end for**
10: **end for**
11: $x_{t+1} = \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} x^i_{t,K}$
12: **end for**
13: **return** x_T

$$\boxed{x_{t+1} = \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} x^i_{t,K}} = x_t - \underbrace{\frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} (x_t - x^i_{t,K})}_{\text{“Pseudo-gradient”}}$$

["Adaptive Federated Optimization" Reddi et al. (2021)]

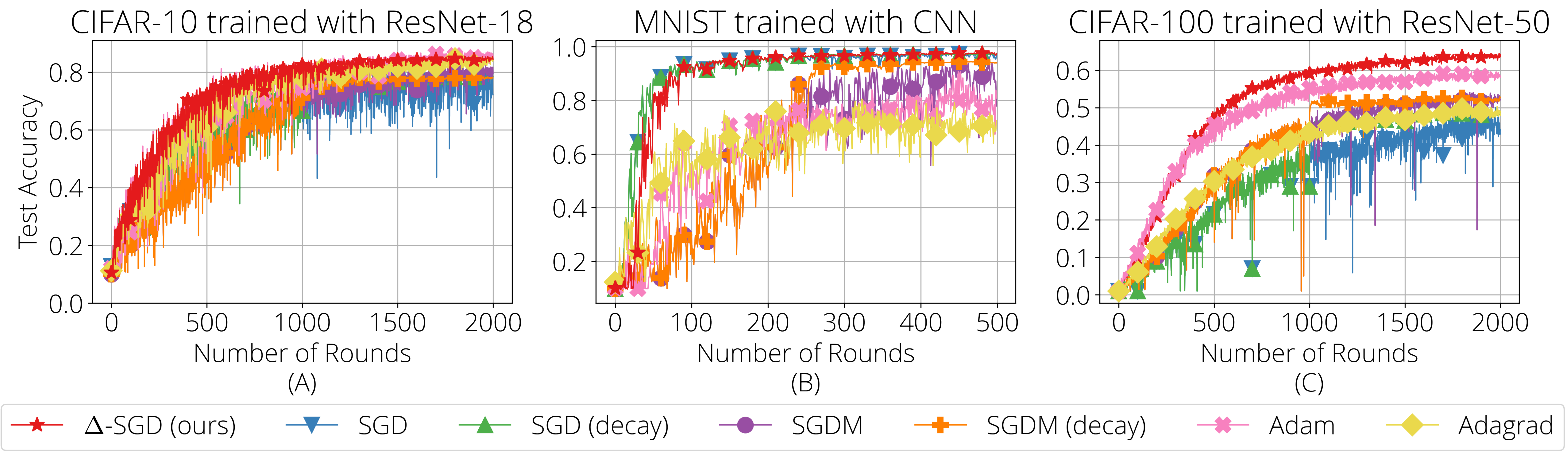
Typically 3-4 orders of magnitude more expensive than local computation [G. Lan, et al., 2020]

What about client optimizer?

- Communication is expensive
- Many more local updates compared to server update

- FedAdam
- FedAdagrad
- FedYogi

Client optimization is more challenging...?



- **FedAvg:** grid-search of typically 11-13 client SGD step sizes
- **FedAdam*:** grid-search of 6 different client step sizes (best usually different for each task)
- Above assume the same step size is used for all clients (can we do better?)

Our proposed step size for Δ-SGD

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

$$\eta_t^i = \min\left\{\boxed{\frac{\|x_t^i - x_{t-1}^i\|}{2\|\nabla f_i(x_t^i) - \nabla f_i(x_{t-1}^i)\|}}, \boxed{\sqrt{1+\theta_{t-1}^i}\,\eta_{t-1}^i}\right\}, \quad \theta_{t-1}^i = \eta_{t-1}^i/\eta_{t-2}^i$$

Don't increase too much!

Client i local updates: $x_{t+1}^i = x_t^i - \eta_t^i \nabla f_{i_t}(x_t^i)$

Adapts to local smoothness: $\|\nabla f_i(x_t^i) - \nabla f_i(x_{t-1}^i)\| \leq \frac{1}{2\eta_t^i}\|x_t^i - x_{t-1}^i\| \approx L_t^i \|x_t^i - x_{t-1}^i\|$

- Each client uses its **own** step size
- Step size only requires known quantities (i.e., no tuning required)
- Individual step size is **adaptive** to the **local smoothness** of f_i

Extension to FL setting: Δ-SGD

Algorithm 1 DELTA(Δ)-SGD: **D**istribut**E**d **L**ocali**T**y **A**daptive SGD

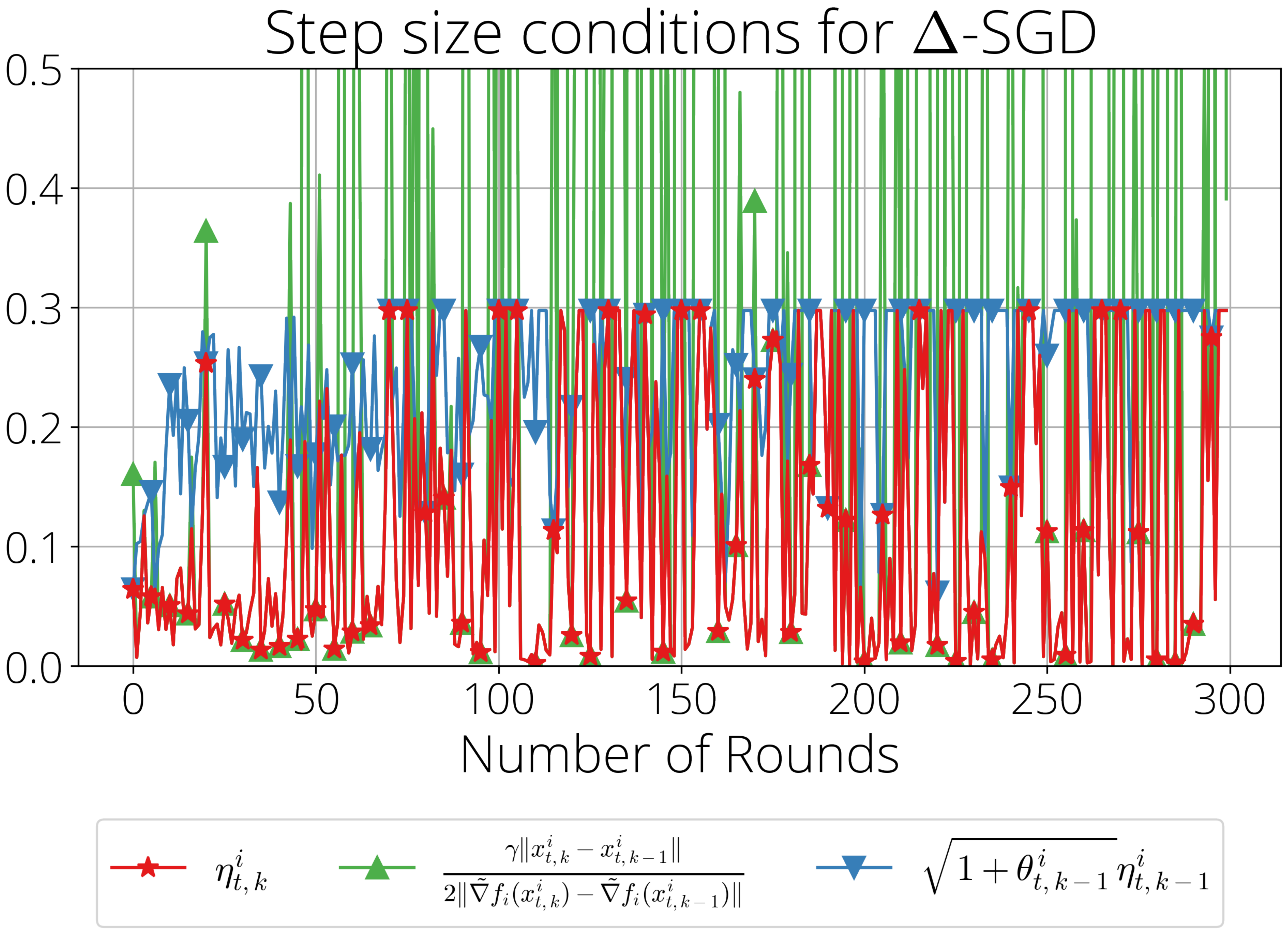
```
input: x_0 ∈ R^d, η_0, θ_0, γ > 0, and p ∈ (0,1).
for each round t = 0, 1, ..., T−1 do
    sample a subset S_t of clients with size |S_t| = p · m
    for each machine in parallel for i ∈ S_t do
        set x^i_{t,0} = x_t
        set η^i_{t,0} = η_0 and θ^i_{t,0} = θ_0
        for local step k ∈ [K] do
            x^i_{t,k} = x^i_{t,k-1} − η^i_{t,k-1} ∇̃f_i(x^i_{t,k-1})
            η^i_{t,k} = min{ γ‖x^i_{t,k} − x^i_{t,k-1}‖ / (2‖∇̃f_i(x^i_{t,k}) − ∇̃f_i(x^i_{t,k-1})‖), sqrt(1 + θ^i_{t,k-1}) η^i_{t,k-1} }
            θ^i_{t,k} = η^i_{t,k} / η^i_{t,k-1}
        end for
    end for
    x_{t+1} = (1/|S_t|) Σ_{i∈S_t} x^i_{t,K}
end for
return x_T
```

$$\eta^i_{t,k} = \min\left\{ \frac{\gamma\|x^i_{t,k} - x^i_{t,k-1}\|}{2\|\tilde{\nabla} f_i(x^i_{t,k}) - \tilde{\nabla} f_i(x^i_{t,k-1})\|}, \sqrt{1+\theta^i_{t,k-1}}\,\eta^i_{t,k-1} \right\}$$

$$x_{t+1} = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t} x^i_{t,K}$$

We can apply server-side adaptive method too
from ["Adaptive Federated Optimization" Reddi et al. (2021)]

Δ-SGD step size in practice



Assumption 1: There exist nonnegative constants $\sigma, \rho,$ and G such that for all $i \in [M]$ and $x \in \mathbb{R}^d$,

(1a)	$\mathbb{E}\|\nabla F_i(x,z) - \nabla f_i(x)\|^2 \le \sigma^2,$	(bounded variance)
(1b)	$\|\nabla f_i(x)\| \le G,$	(bounded gradient)
(1c)	$\|\nabla f_i(x) - \nabla f(x)\|^2 \le \rho\|\nabla f(x)\|^2.$	(strong growth of dissimilarity)

Theorem 1: Let Assumption 1 hold, with $\rho = \mathcal{O}(1)$. Further, suppose that $\gamma = \mathcal{O}\left(\frac{1}{K\sqrt{T}}\right)$, and $\eta_0 = \mathcal{O}(\gamma)$. Then, the following property holds for Algorithm Δ-SGD, for T sufficiently large:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(x_t)\right\|^2 \le \mathcal{O}\left(\frac{\Psi_1}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{\boxed{\tilde{L}^2}\Psi_2}{T}\right) + \mathcal{O}\left(\frac{\tilde{L}^3\Psi_2}{\sqrt{T^3}}\right),$$

where $\Psi_1 = \max\left\{\frac{\sigma^2}{b}, f(x_0) - f(x^\star)\right\}$ and $\Psi_2 = \left(\frac{\sigma^2}{b} + G^2\right)$ are global constants, with $b = |\mathcal{B}|$ being the batch size; $\tilde{L}$ is a constant at most the maximum of local smoothness, i.e., $\max_{i,t} \tilde{L}_{i,t}$, where $\tilde{L}_{i,t}$ the local smoothness of f_i at round t.

Experimental setup

- Datasets: MNIST, FMINIST, CIFAR-10, and CIFAR-100
- Architecture: shallow CNN, ResNet-18, and ResNet-50
- Level of heterogeneity: latent Dirichilet allocation (LDA), $\alpha \in \{1, 0.1, 0.01\}$

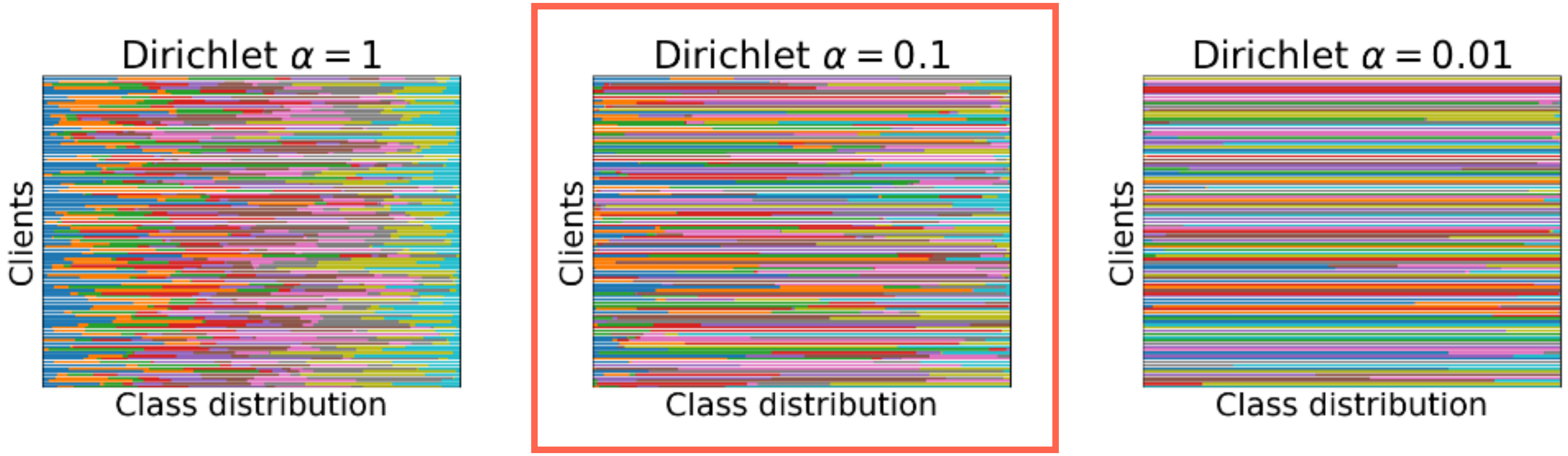


Figure 5: Illustration of the degree of heterogeneity induced by using different concentration parameter α for Dirichlet distribution, for CIFAR-10 dataset (10 colors) and 100 clients (100 rows on y-axis).

Results

Green: < 0.5% from best performance

(x.x): performance difference from the best highlighted when > 2%

Non-iidness	*Optimizer*	*Dataset / Model*				
Dir($\alpha \cdot \mathbf{p}$)		MNIST CNN	FMNIST CNN	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	CIFAR-100 ResNet-50
$\alpha = 1$	SGD	**98.3** ↓(0.2)	86.5 ↓(0.8)	87.7 ↓(2.1)	57.7 ↓(4.2)	53.0 ↓(12.8)
	SGD (↓)	97.8 ↓(0.7)	86.3 ↓(1.0)	87.8 ↓(2.0)	**61.9** ↓(0.0)	60.9 ↓(4.9)
	SGDM	**98.5** ↓(0.0)	85.2 ↓(2.1)	88.7 ↓(1.1)	58.8 ↓(3.1)	60.5 ↓(5.3)
	SGDM (↓)	**98.4** ↓(0.1)	**87.2** ↓(0.1)	**89.3** ↓(0.5)	**61.4** ↓(0.5)	63.3 ↓(2.5)
	Adam	94.7 ↓(3.8)	71.8 ↓(15.5)	**89.4** ↓(0.4)	55.6 ↓(6.3)	61.4 ↓(4.4)
	Adagrad	64.3 ↓(34.2)	45.5 ↓(41.8)	86.6 ↓(3.2)	53.5 ↓(8.4)	51.9 ↓(13.9)
	SPS	10.1 ↓(88.4)	85.9 ↓(1.4)	82.7 ↓(7.1)	1.0 ↓(60.9)	50.0 ↓(15.8)
	Δ-SGD	**98.4** ↓(0.1)	**87.3** ↓(0.0)	**89.8** ↓(0.0)	**61.5** ↓(0.4)	**65.8** ↓(0.0)
$\alpha = 0.1$	SGD	**98.1** ↓(0.0)	83.6 ↓(2.8)	72.1 ↓(12.9)	54.4 ↓(6.7)	44.2 ↓(19.9)
	SGD (↓)	**98.0** ↓(0.1)	84.7 ↓(1.7)	78.4 ↓(6.6)	59.3 ↓(1.8)	48.7 ↓(15.4)
	SGDM	**97.6** ↓(0.5)	83.6 ↓(2.8)	79.6 ↓(5.4)	58.8 ↓(2.3)	52.3 ↓(11.8)
	SGDM (↓)	**98.0** ↓(0.1)	**86.1** ↓(0.3)	77.9 ↓(7.1)	60.4 ↓(0.7)	52.8 ↓(11.3)
	Adam	96.4 ↓(1.7)	80.4 ↓(6.0)	**85.0** ↓(0.0)	55.4 ↓(5.7)	58.2 ↓(5.9)
	Adagrad	89.9 ↓(8.2)	46.3 ↓(40.1)	84.1 ↓(0.9)	49.6 ↓(11.5)	48.0 ↓(16.1)
	SPS	96.0 ↓(2.1)	85.0 ↓(1.4)	70.3 ↓(14.7)	42.2 ↓(18.9)	42.2 ↓(21.9)
	Δ-SGD	**98.1** ↓(0.0)	**86.4** ↓(0.0)	**84.5** ↓(0.5)	**61.1** ↓(0.0)	**64.1** ↓(0.0)
$\alpha = 0.01$	SGD	96.8 ↓(0.7)	79.0 ↓(1.2)	22.6 ↓(11.3)	30.5 ↓(1.3)	24.3 ↓(7.1)
	SGD (↓)	**97.2** ↓(0.3)	79.3 ↓(0.9)	**33.9** ↓(0.0)	30.3 ↓(1.5)	24.6 ↓(6.8)
	SGDM	77.9 ↓(19.6)	75.7 ↓(4.5)	28.4 ↓(5.5)	24.8 ↓(7.0)	22.0 ↓(9.4)
	SGDM (↓)	94.0 ↓(3.5)	79.5 ↓(0.7)	29.0 ↓(4.9)	20.9 ↓(10.9)	14.7 ↓(16.7)
	Adam	80.8 ↓(16.7)	60.6 ↓(19.6)	22.1 ↓(11.8)	18.2 ↓(13.6)	22.6 ↓(8.8)
	Adagrad	72.4 ↓(25.1)	45.9 ↓(34.3)	12.5 ↓(21.4)	25.8 ↓(6.0)	22.2 ↓(9.2)
	SPS	69.7 ↓(27.8)	44.0 ↓(36.2)	21.5 ↓(12.4)	22.0 ↓(9.8)	17.4 ↓(14.0)
	Δ-SGD	**97.5** ↓(0.0)	**80.2** ↓(0.0)	31.6 ↓(2.3)	**31.8** ↓(0.0)	**31.4** ↓(0.0)

Step size grid search done in this setting

TOP-1 in 73%, TOP-2 in 100% of the experiments without additional tuning

“Adaptive Federated Learning with Auto-tuend Clients”

J. L. Kim, M. T. Toghani, C. A. Uribe, A. Kyrillidis. ICLR 2024.

1. Algorithmic:
   Adaptive SGD scheme that utilize the local smoothness

2. Setup / Modeling:
   Federated/Collaborative protocol that allows local updates

3. Practical Importance:
   Extensive experimental results achieving better or similar performance across different FL scenarios without tuning

Road Map: Algorithmic, Structural, and Pragmatic Acceleration

Algorithmic | Structural

Accelerated Factored Gradient Descent

Keyword: non-convex, matrix factorization

[**J. L. Kim**, G. Kollias, A. Kalev, K. X. Wei, A. Kyrillidis. "Fast Quantum State Reconstruction via Accelerated Non-Convex Programming" **Photonics 2023** / Quantum Information Processing (QIP) 2023 (poster)]

"Momentum"

Algorithmic | Pragmatic

Stochastic Proximal Point Method with Momentum

Keyword: implicit method, acceleration, stability

[**J. L. Kim**, P. Toulis, A. Kyrillidis. "Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum," **L4DC 2022**]

Application:

Quantum State Tomography

"Plug-and-play"

Adaptive step size

Structural | Pragmatic

Local Stochastic Factored Gradient Descent

Keyword: distributed optimization, local updates

[**J. L. Kim**, M. T. Toghani, C. A. Uribe, A. Kyrillidis. "Local Stochastic Factored Gradient Descent for Distributed Quantum State Tomography" **Control Systems Letters (L-CSS), IEEE 2022** / Quantum Information Processing (QIP) 2023 (poster)]

"Distributed Optimization"

Structural | Pragmatic

Adaptive Federated Learning with Auto-Tuned Clients

Keyword: adaptive step-size, federated learning

[**J. L. Kim**, M. T. Toghani, C. A. Uribe, A. Kyrillidis. "Adaptive Federated Learning with Auto-Tuned Clients", **ICLR 2024**]

“Fast Quantum State Reconstruction via Accelerated Non-convex Programming”

J. L. Kim (Rice CS), G. Kollias (IBM), A. Kalev (USC), K. X. Wei (IBM), A. Kyrillidis (Rice CS)

Published in Photonics, 2023.

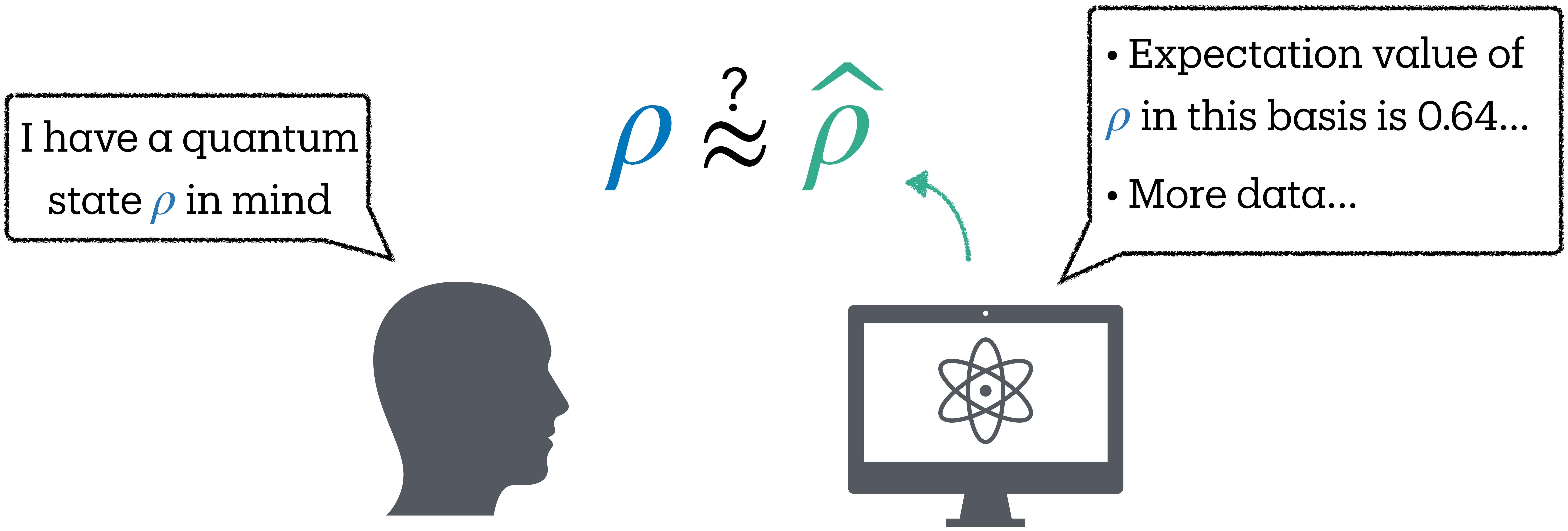
“Local Stochastic Factored Gradient Descent for Distributed Quantum State Tomography”

J. L. Kim (Rice CS), M. T. Toghani (Rice ECE), C. A. Uribe (Rice ECE), A. Kyrillidis (Rice CS)

Published in Control System Letters, 2022.

Quantum State Tomogrpahy (QST)

- Electrical engineers use multimeters and oscilloscopes to verify that circuit works as expected.
- We need similar verification tools in quantum computing. QST is one such tool.
- QST is the task to reconstruct the density matrix of a given quantum state from measurement data.



Quantum State

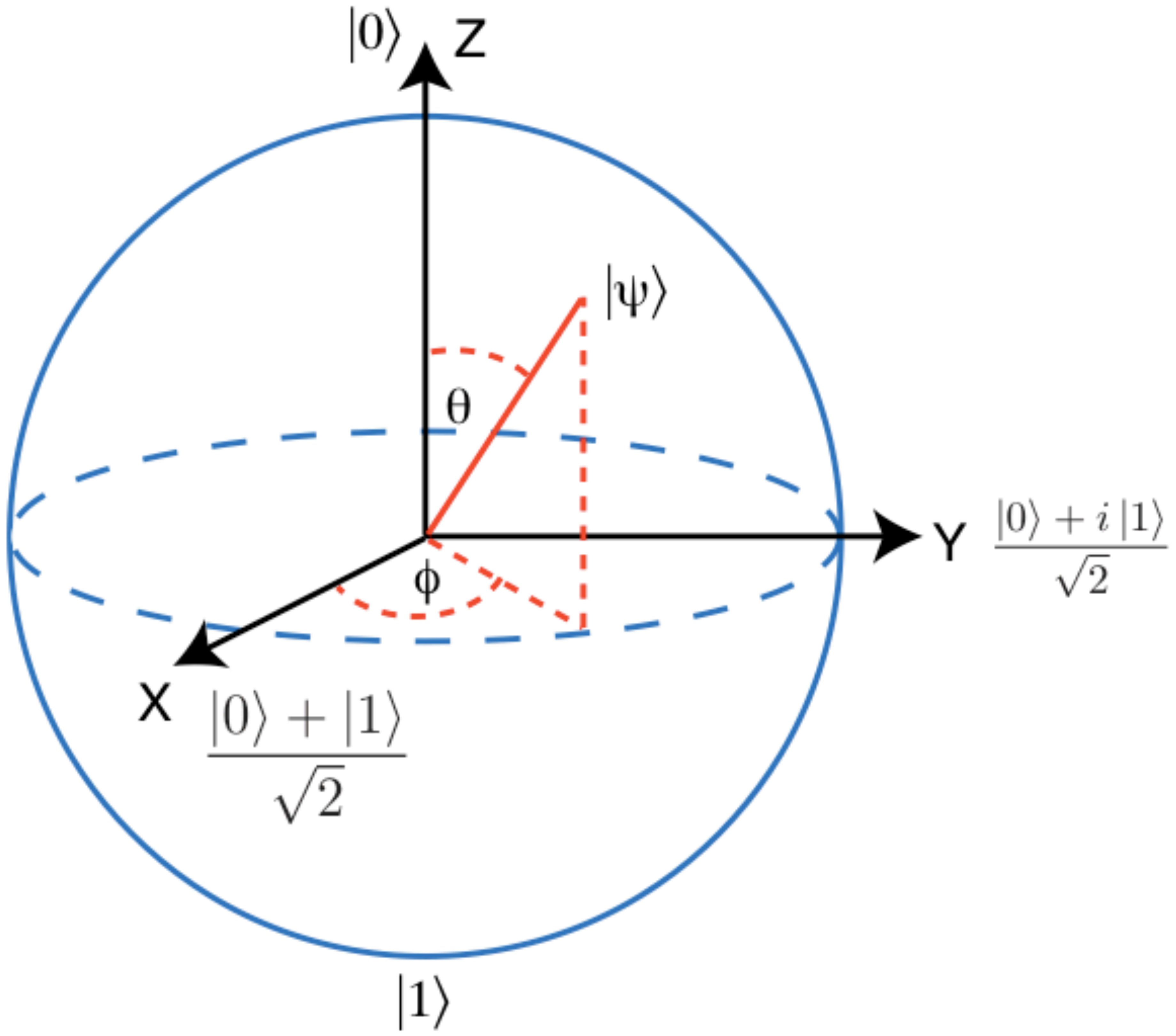
- We represent quantum bits (qubits) $|0\rangle$ and $|1\rangle$ as vectors:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- A state $|\psi\rangle$ can be written as a superposition of $|0\rangle$ and $|1\rangle$, e.g., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ Outcome $|1\rangle$ w.p. $|\beta|^2$
- 2-qubit state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$:

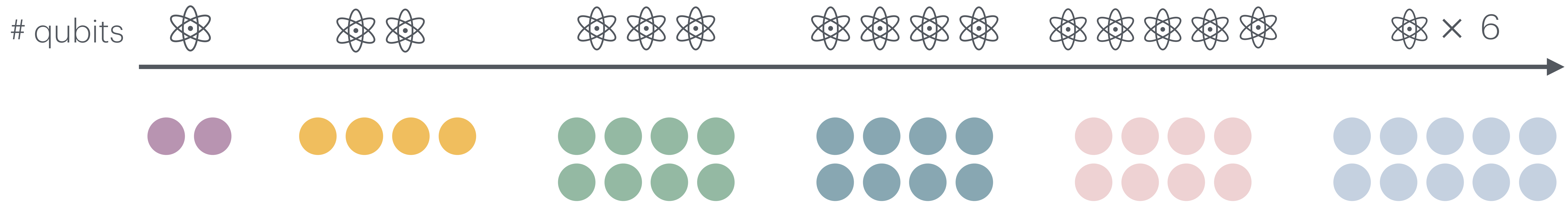
$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \delta \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

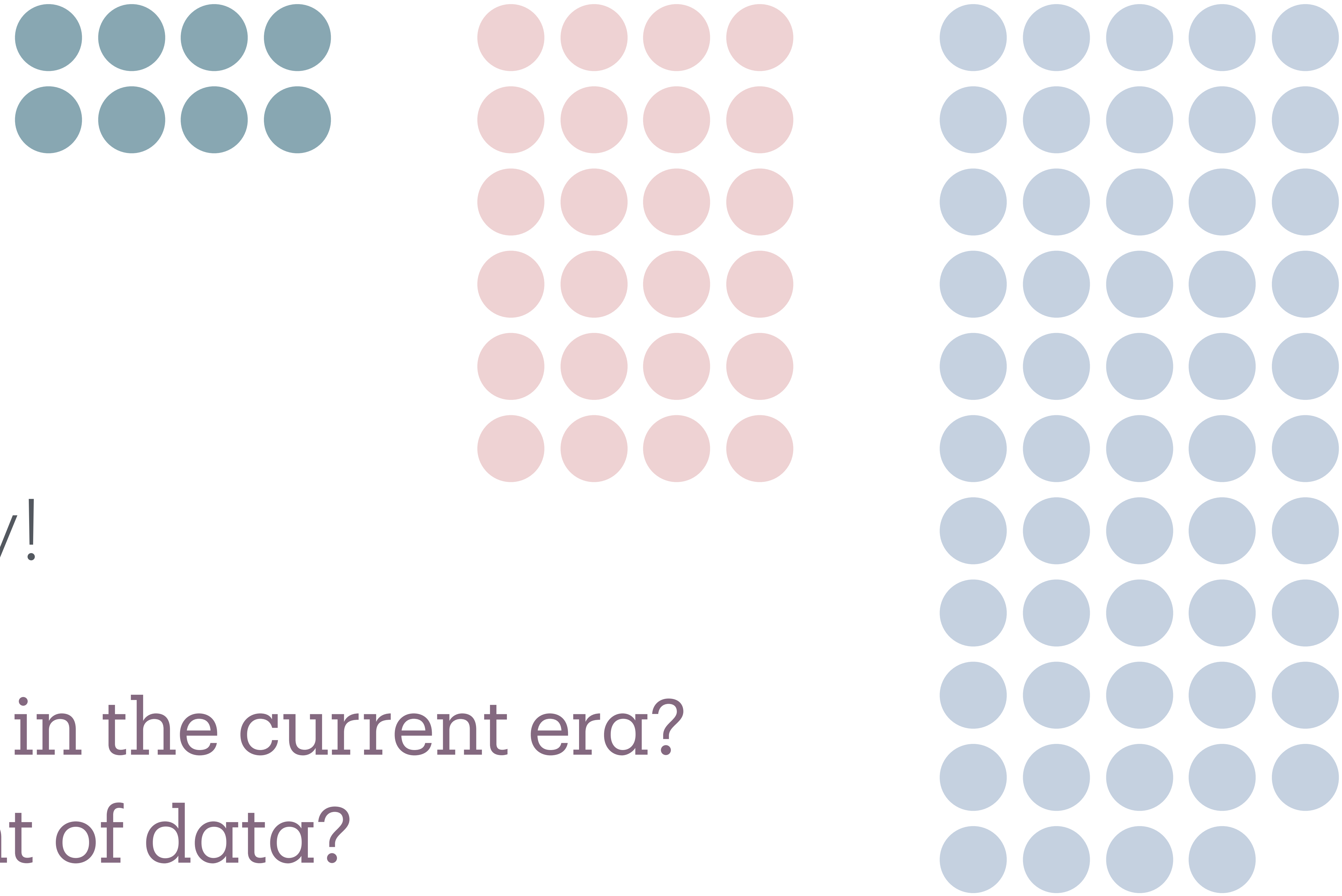
- A pure state of n qubits can be represented by column vectors in $\mathbb{C}^d$ space with $d = 2^n$



[Figure source: https://qiskit.org/textbook/ch-states/introduction.html]

What about Quantum Computing?





$\times 100 \Rightarrow$

$$1.26765 \times 10^{30}$$

complex numbers required to describe the system classically!

🧐 How to process meaningfully in the current era?

🤔 How to even store this amount of data?

Quantum State and Density matrix

- $|\psi\rangle$ is a column vector, called "ket"
- $\langle\psi|$ is a row vector, called "bra", with complex conjugates
- Inner product: $\langle\phi|\psi\rangle$ is a number
- Outer product: $|\phi\rangle\langle\psi|$ is a matrix
  - A pure state $|\psi\rangle$ can be written as $\rho = |\psi\rangle\langle\psi|$
  - A mixed state can be written as $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$

E.g. $|\psi\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix} \rightarrow \langle\psi| = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \end{bmatrix}$

Density matrix ρ

- PSD: $\rho \succeq 0$
- Unit trace: $\mathrm{Tr}(\rho) = 1$

QST objective

$\left(\mathscr{A}(\rho)\right)_i = \mathrm{Tr}(\rho A_i)$ where $A_i \in \mathbb{C}^{d\times d},\ i = 1,\ldots,m$

$$\underset{\rho\in\mathbb{C}^{d\times d}}{\text{minimize}} \quad F(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \rho \succeq 0,\ \mathrm{Tr}(\rho) = 1$$

measured data

How does it scale?

- **Optimization:** the space of $\rho \in \mathbb{C}^{d\times d}$ grows exponentially ($d = 2^n$, where n is the number of qubits)
- **Amount of data:** from $\mathscr{A}(\rho) = y$, if we have access to $y_1, \ldots, y_m$ and $A_1, \ldots, A_m$ that form an orthonormal basis for $\mathbb{C}^{d\times d}$ (i.e. $m = d^2$), we can reconstruct ρ with linear inversion

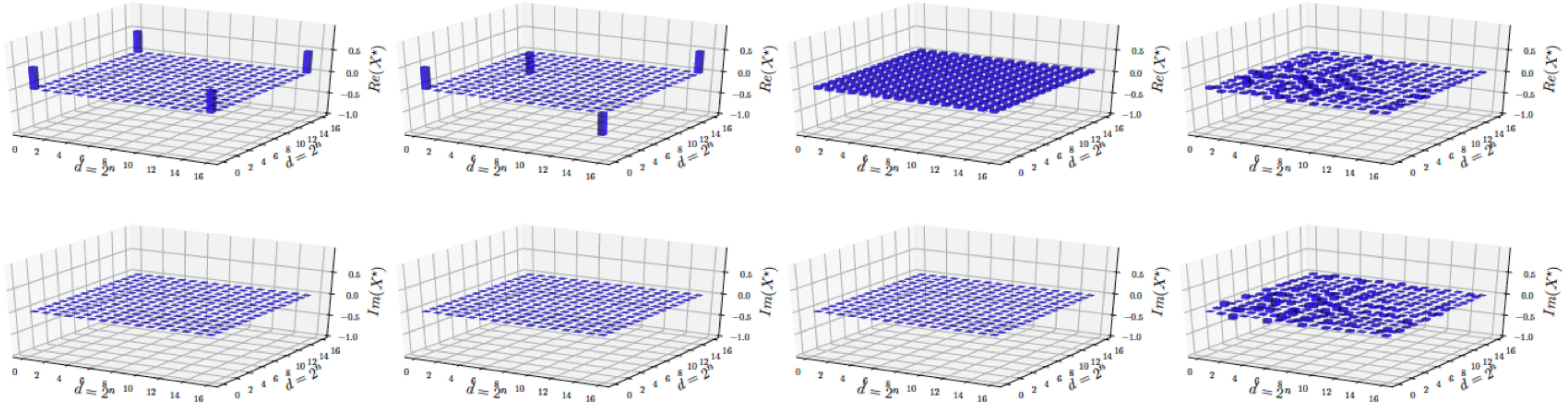
For $n = 16$ qubits, $\rho \in \mathbb{C}^{d\times d}$ where $d = 65{,}536$

We need $m = O\left(2^{32}\right) \approx 4{,}294{,}967{,}296$ measurements

Structured density matrices

- Optimization: $\rho \in \mathbb{C}^{d \times d}$ where $d = 2^n$
- Amount of data: $O(d^2)$ without any prior

GHZ | GHZminus | Hadamard | Random



Compressive sensing + QST

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- Amount of data: $O(d^2)$ without any prior

[Kalev et al., 2015]:

Tr$(\rho) = 1$ constraint can be removed without affecting the final estimate

$$\begin{aligned} &\underset{\rho\in\mathbb{C}^{d\times d}}{\text{minimize}} && F(\rho) := \tfrac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2 \\ &\text{subject to} && \boxed{\text{rank}(\rho) \le r,}\ \rho \succeq 0,\ \cancel{\text{Tr}(\rho) = 1} \end{aligned}$$

Restricted Isometry Property (RIP) for rank-r matrices

[B. Recht et al., 2010]

A linear operator $\mathscr{A} : \mathbb{C}^{d\times d} \to \mathbb{R}^m$ satisfies the RIP on rank-r matrices, with parameter $\delta_{2r} \in (0,1)$, if the following holds for any rank-r matrix $X \in \mathbb{C}^{d\times d}$, with high probability:

$$(1 - \delta_{2r}) \cdot \|X_1 - X_2\|_F^2 \le \|\mathscr{A}(X_1 - X_2)\|_2^2 \le (1 + \delta_{2r}) \cdot \|X_1 - X_2\|_F^2.$$

[D. Gross et al., 2010]: can reconstruct rank-r density matrix $\rho \in \mathbb{C}^{d\times d}$ using $O\left(r \cdot d \cdot \text{poly}(\log d)\right)$ measurements

[Y.K. Liu, 2010]: $P_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ satisfies RIP for rank-r matrices

Factored objective for QST

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- Amount of data: $O(d^2)$ without any prior

$$\underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} \quad F(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \cancel{\rho \succeq 0, \text{rank}(\rho) \leq r} \leftarrow \rho = UU^\dagger$$

Convex constraint ($\rho \succeq 0$)

Non-convex constraint ($\text{rank}(\rho) \leq r$)

$$\underset{U \in \mathbb{C}^{d\times r}}{\text{minimize}} \quad F(UU^\dagger) := \frac{1}{2}\|\mathscr{A}(UU^\dagger) - y\|_2^2$$

Smaller space ($\mathbb{C}^{d\times r}$) than original space ($\mathbb{C}^{d\times d}$)

Constraints automatically satisfied

[J. L. Kim, et al., Photonics 2023]
- Accelerated linear convergence
- Extensive experimental results using real quantum data (IBM)

Factored Gradient Descent

[Kyrillidis et al., 2019]

$$U_{t+1} = U_t - \eta \nabla F(U_t U_t^\dagger) \cdot U_t$$
$$= U_t - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \{\text{Tr}(A_i U_i U_i^\dagger - y_i\} A_i \right) \cdot U_t$$

Momentum-inspired Factored Gradient Descent

$$U_{t+1} = Z_t - \eta \mathscr{A}^\dagger \left(\mathscr{A}(Z_t Z_t^\dagger) - y \right) \cdot Z_t$$
$$Z_{t+1} = U_{t+1} + \mu \left(U_{t+1} - U_t \right)$$

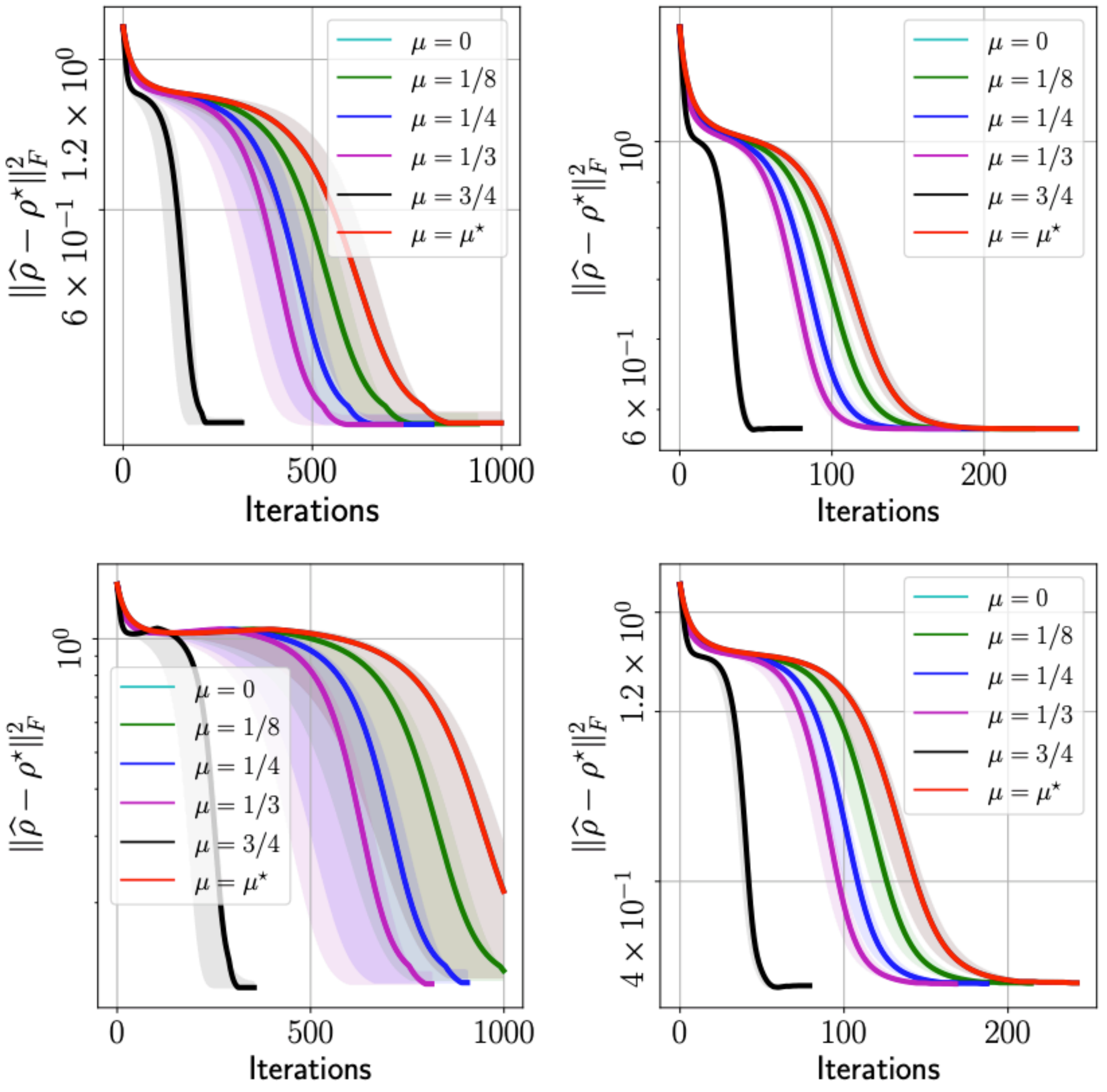
MiFGD performance on real quantum data (`IBM QPU`)

GHZminus (6)

GHZminus (8)

Hadamard (6)

Hadamard (8)



$[m = 20\% \cdot d^2]$

Compressive sensing + QST

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- Amount of data: $O(d^2)$ without any prior

[Kalev et al., 2015]:

Tr$(\rho) = 1$ constraint can be removed without affecting the final estimate

$$\underset{\rho\in\mathbb{C}^{d\times d}}{\text{minimize}} \quad F(\rho) := \frac{1}{2}\|\mathcal{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \boxed{\text{rank}(\rho) \leq r,} \; \rho \succeq 0, \; \cancel{\text{Tr}(\rho) = 1}$$

Restricted Isometry Property (RIP) for rank-r matrices

[B. Recht et al., 2010]

A linear operator $\mathcal{A} : \mathbb{C}^{d\times d} \to \mathbb{R}^m$ satisfies the RIP on rank-r matrices, with parameter $\delta_{2r} \in (0,1)$, if the following holds for any rank-r matrix $X \in \mathbb{C}^{d\times d}$, with high probability:

$$(1 - \delta_{2r}) \cdot \|X_1 - X_2\|_F^2 \leq \|\mathcal{A}(X_1 - X_2)\|_2^2 \leq (1 + \delta_{2r}) \cdot \|X_1 - X_2\|_F^2.$$

[D. Gross et al., 2010]: can reconstruct rank-r density matrix $\rho \in \mathbb{C}^{d\times d}$ using $O\big(r \cdot d \cdot \text{poly}(\log d)\big)$ measurements $\approx 9.65 \times 10^{14}$ with $r = 100,\ n = 30$

[Y.K. Liu, 2010]: $P_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ satisfies RIP for rank-r matrices

Factored objective for QST

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- Amount of data: $O(d^2)$ without any prior

$$\underset{\rho\in\mathbb{C}^{d\times d}}{\text{minimize}} \quad F(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \cancel{\rho \succeq 0, \ \text{rank}(\rho) \leq r} \longleftarrow \rho = UU^\dagger$$

Convex constraint ($\rho \succeq 0$)

Non-convex constraint ($\text{rank}(\rho) \leq r$)

$$\underset{U\in\mathbb{C}^{d\times r}}{\text{minimize}} \quad F(UU^\dagger) := \frac{1}{2}\|\mathscr{A}(UU^\dagger) - y\|_2^2$$

Smaller space ($\mathbb{C}^{d\times r}$) than original space ($\mathbb{C}^{d\times d}$)

Constraints automatically satisfied

[J. L. Kim, et al., Photonics 2023]
- Accelerated linear convergence
- Extensive experimental results using real quantum data (IBM)

Factored Gradient Descent

[Kyrillidis et al., 2019]

$$U_{t+1} = U_t - \eta \nabla F(U_t U_t^\dagger) \cdot U_t$$

$$= U_t - \eta \left(\frac{1}{n} \sum_{i=1}^{n} \{\text{Tr}(A_i U_i U_i^\dagger - y_i\} A_i \right) \cdot U_t$$

still large! $\approx 9.65 \times 10^{14}$

Momentum-inspired Factored Gradient Descent

$$U_{t+1} = Z_t - \eta \mathscr{A}^\dagger \left(\mathscr{A}(Z_t Z_t^\dagger) - y \right) \cdot Z_t$$

$$Z_{t+1} = U_{t+1} + \mu \left(U_{t+1} - U_t \right)$$

Distributed objective

- We consider the setting where the measurements $y \in \mathbb{R}^m$ and the sensing matrices $\mathscr{A} : \mathbb{C}^{d\times d} \to \mathbb{R}^m$ from a central quantum computer are locally stored across M different classical machines.
- These classical machines perform some local operations based on their local data, and communicate back and forth with the central quantum server.

Centralized

$$\min_{U\in\mathbb{C}^{d\times r}} \; G(U) := F(UU^\dagger) = \frac{1}{2}\|\mathscr{A}(UU^\dagger) - y\|_2^2$$

$$\downarrow$$

Distributed

$$\min_{U\in\mathbb{C}^{d\times r}} \left\{ g(U) = \frac{1}{M}\sum_{i=1}^{M} g_i(U) \right\},$$

where $\quad g_i(U) := \mathbb{E}_{j\sim\mathscr{D}_i}\|\mathscr{A}_i^j(UU^\dagger) - y_i^j\|_2^2$

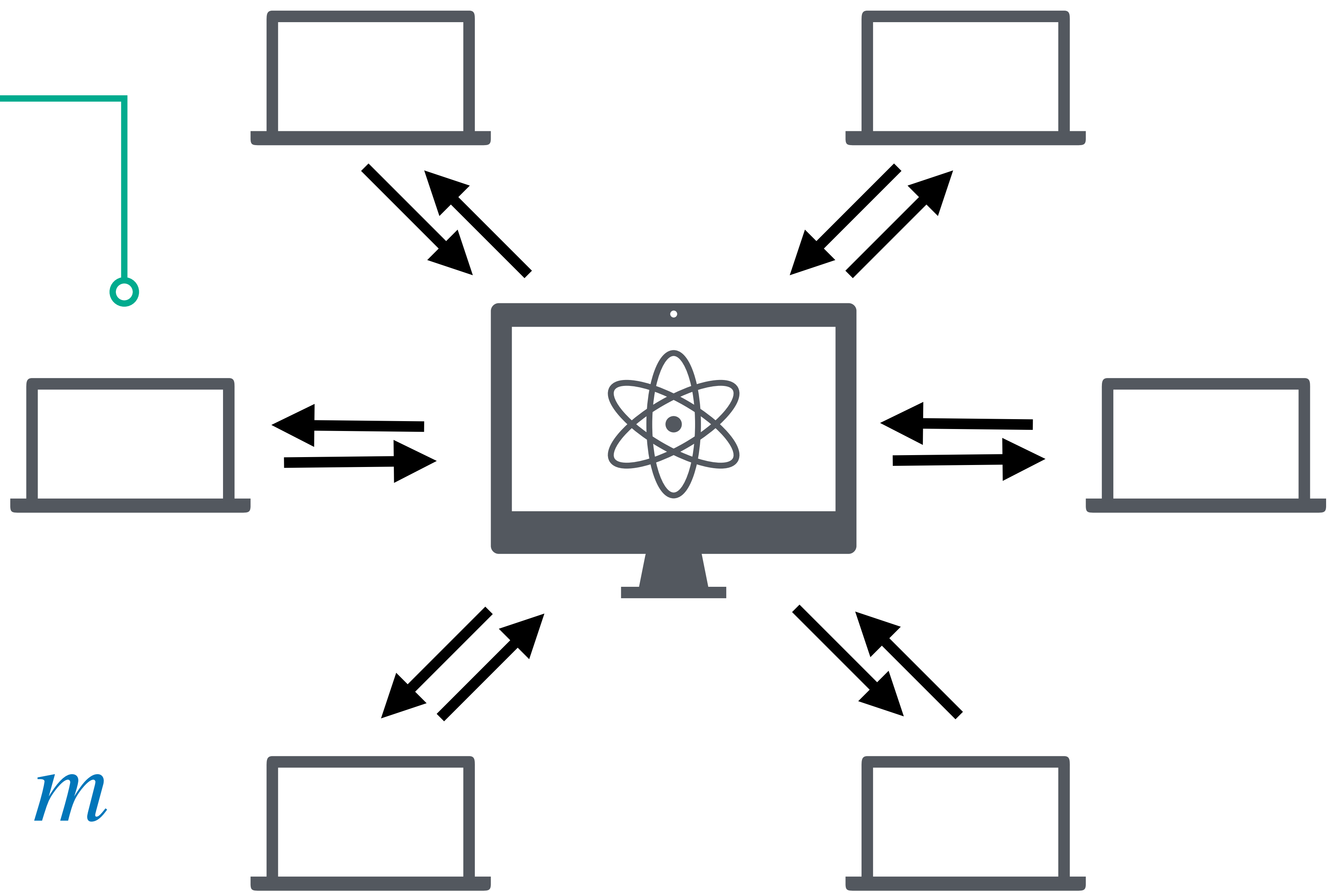
$$\min_{U \in \mathbb{C}^{d \times r}} \left\{ g(U) = \frac{1}{M} \sum_{i=1}^{M} g_i(U) \right\},$$

where $\quad g_i(U) := \mathbb{E}_{j \sim \mathcal{D}_i} \| \mathcal{A}_i^j(UU^\dagger) - y_i^j \|_2^2$

i-th machine

$\mathcal{A}_i : \mathbb{C}^{d \times d} \rightarrow \mathbb{R}^{m_i}$

$y_i \in \mathbb{R}^{m_i}$



$\bigcup_{i=1}^{M} \mathcal{A}_i = \mathcal{A} \qquad \sum_{i=1}^{M} m_i = m$

BUT:

- Intra-node communication is much more expensive than— typically about 3 orders of magnitude— local computation [G. Lan et al., 2020]

Local Stochastic Factored Gradient Descent

Algorithm 1 Local SFGD

Set number of iterations $T > 0$, synchronization time steps $t_1, t_2, \ldots$, and initialize $U_0^i = U_0$ as below:

$$U_0^i = \texttt{SVD}\Big(- \sum_{i=1}^{M} \frac{m_i}{m} \nabla f_i(0) \Big) \quad \forall i \in [M], \tag{7}$$

where SVD denotes the singular value decomposition.

for each round $t = 0, \ldots T$ **do**

  for in parallel for $i \in [M]$ **do**

    Sample j_t uniformly at random from $[m_i]$.

    if $t = t_p$ for some $p \in \mathbb{N}$ **then**

      $U_{t+1}^i = \frac{1}{M} \sum_{i=1}^{M} \left(U_t^i - \eta_t \nabla g_i^{j_t}(U_t^i) \right)$

    else

      $U_{t+1}^i = U_t^i - \eta_t \nabla g_i^{j_t}(U_t^i)$

    end if

  end for

end for

return $\hat{U}_{T+1} := \frac{1}{M} \sum_{i=1}^{M} U_{T+1}^i$.

Lemma 1: Let Assumption 1 hold. Assume that $D^2(U_0^i, U^\star) \leq \frac{\sigma_r(X^\star)}{100 \cdot \kappa \cdot \sigma_1(X^\star)}$, where $\sigma_k(X^\star)$ is the k-th singular value of $X^\star$, and $\kappa = L/\mu$. Then:
$\left\langle U_t^i - U^\star R^\star, \nabla g_i(U_t^i) \right\rangle \geq \frac{2\eta_t}{3} \|\nabla g_i(U_t^i)\|_F^2 + \frac{3\mu}{20} \sigma_r(X^\star) \cdot D^2(U_t^i, U^\star)$.

- Initialization scheme in (7) in Algorithm 1 is modified from [S. Bhojanapalli et al., 2016] to distributed version, and satisfies the initialization condition in Lemma 1.

Lemma 2: Let Assumptions 1 and (2c) hold. Then, the output of Algorithm 1 with $\max_p |t_{p+1} - t_p| \leq h$ satisfies:

$$\frac{1}{M} \sum_{i=1}^{M} \mathbb{E}\left[\|\hat{U}_t - U_t^i\|_F^2 \right] \leq \eta_{t_q}^2 (h-1)^2 G^2,$$

where t_q is the synchronization step immediately before t.

Local linear convergence

Theorem 1: Let Assumptions 1, 2, and the initialization condition of Lemma 1 hold. Moreover, let $\eta_t = \eta < \frac{1}{\alpha}$ for $t \in [0 : T]$ and $\max_p | t_p - t_{p+1} | \leq h$. Then, the output of Algorithm 1 has the following property:

$$\mathbb{E}\left[D^2(\hat{U}_{T+1}, U^\star)\right] \leq \left(1 - \eta\alpha\right)^{T+1} D^2(\hat{U}_0, U^\star) + \eta\left(\frac{4(h-1)^2 G^2}{\alpha} + \frac{\sigma^2}{M\alpha}\right),$$

where $X^\star$ is the optimum of f over the set of PSD matrices such that rank$(X^\star) = r$, $U^\star$ is such that $X^\star = U^\star U^{\star\top}$, and $\alpha = \frac{3\mu}{10}\sigma_r(X^\star)$ is a global constant.

- Notice the last variance term $\frac{\sigma^2}{M\alpha}$, which disappears in the noiseless case, is reduced by the number of machines M.
- Single-batch is assumed in the proof; by using batch size $b > 1$, this term can be further divided by b.
- By plugging in $h = 1$ (i.e., synchronization happens on every iteration), the first variance term disappears, exhibiting similar local linear convergence to SFGD.
- Can achieve exact (local) convergence with decaying step size at the expense of sublinear rate

“Fast Quantum State Reconstruction via Accelerated Non-convex Programming” **J. L. Kim,** G. Kollias, A. Kalev, K. X. Wei, A. Kyrillidis. Photonics 2023

“Local Stochastic Factored Gradient Descent for Distributed Quantum State Tomography”

J. L. Kim, M. T. Toghani, C. A. Uribe, A. Kyrillidis. L-CSS 2022

1. Algorithmic:
   Accelerated linear convergence of MiFGD that utilize (non-convex) low-rank matrix factorization + acceleration

2. Setup / Modeling:
   Distributed QST with Stochastic FGD that allows local updates with rigorous theory

3. Practical Importance:
   Extensive experimental results using real quantum data / Open source software compatible with Qiskit

Road Map: Algorithmic, Structural, and Pragmatic Acceleration

Algorithmic Structural

Accelerated Factored Gradient Descent

Keyword: non-convex, matrix factorization

[**J. L. Kim**, G. Kollias, A. Kalev, K. X. Wei, A. Kyrillidis. "Fast Quantum State Reconstruction via Accelerated Non-Convex Programming" **Photonics 2023** / Quantum Information Processing (QIP) 2023 (poster)]

"Momentum"

Algorithmic Pragmatic

Stochastic Proximal Point Method with Momentum

Keyword: implicit method, acceleration, stability

[**J. L. Kim,** P. Toulis, A. Kyrillidis. "Convergence and Stability of the Stochastic Proximal Point Algorithm With Momentum," **L4DC 2022**]

Application:

Quantum State Tomography

Momentum Extragradient: A Polynomial-Based Analysis

[**J. L. Kim**, G. Gidel, A. Kyrillidis, F. Pedregosa. "When is MEG Optimal?," **TMLR 2024**]

Algorithmic Pragmatic

Solving QLSP via PPM

[**J. L. Kim**, N. H. Chia, A. Kyrillidis. **WIP**]

"Plug-and-play"

Adaptive step size

Structural Pragmatic

Local Stochastic Factored Gradient Descent

Keyword: distributed optimization, local updates

[**J. L. Kim**, M. T. Toghani, C. A. Uribe, A. Kyrillidis. "Local Stochastic Factored Gradient Descent for Distributed Quantum State Tomography" **Control Systems Letters (L-CSS), IEEE 2022** / Quantum Information Processing (QIP) 2023 (poster)]

"Distributed Optimization"

Structural Pragmatic

Adaptive Federated Learning with Auto-Tuned Clients

Keyword: adaptive step-size, federated learning

[**J. L. Kim**, M. T. Toghani, C. A. Uribe, A. Kyrillidis. "Adaptive Federated Learning with Auto-Tuned Clients", **ICLR 2024**]

Beyond Non-convex, Distributed, and Stable Optimization

Analysis

Lyapunov Functions
Spectral Analysis
Polynomial

Quantum Linear System with PPM

Generative Models

Acceleration with momentum

Quantum Subroutines

Optimization

Centralized GD

Algorithm

Factored Gradient Descent
Proximal Point Method
Extragradient
Adaptive Methods

Model Structure

Distributed Optimization
Minimax/Game Optimization
Federated Learning
Quantum Optimization

Local SFGD

Momentum Extragradient

Thank you!

Back up slides

SPPAM

Proximal point algorithm

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^p} \left\{ f(x) + \frac{1}{2\eta} \|x - x_t\|_2^2 \right\}$$

- PPA changes the conditioning of the problem
- Equivalent to implicit gradient descent (IGD):

$$x_{t+1} = x_t - \eta \nabla f\left(x_{t+1}\right)$$

- PPA enjoys remarkable convergence behavior. For convex f:

$$f(x_T) - f(x^\star) \leq O\left(\frac{1}{\sum_{t=1}^T \eta_t}\right)$$

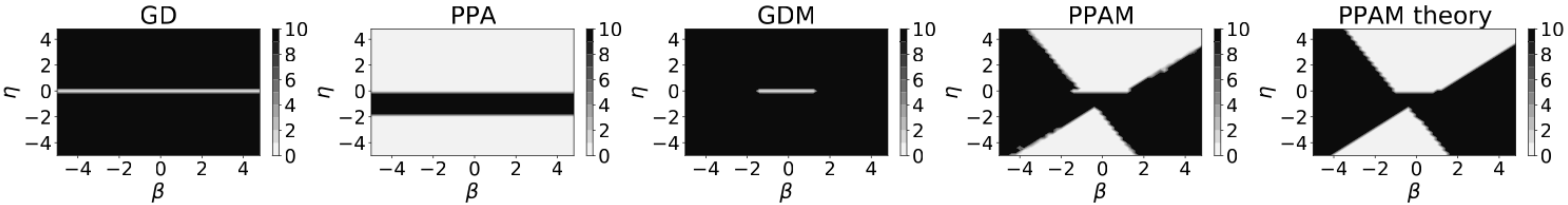
"Arbitrarily fast convergence"

Guller (1991). "On the Convergence of the Proximal Point Algorithm for Convex Minimization"

What about stochastic settings?

The quadratic model case

Conditions on η and β for different algorithms to solve: $f(x) = \frac{1}{2}x^\top A x - b^\top x$



Proposition 1 (GD *(Goh 2017)*). *To minimize (10) with gradient descent, the step size η needs to satisfy $0 < \eta < \frac{2}{\lambda_i}$, $\forall i$, where λ_i is the i-th eigenvalue of A.*

Proposition 2 (PPA/IGD). *To minimize (10) with PPA, the step size η needs to satisfy $\left|\frac{1}{1+\eta\lambda_i}\right| < 1$.*

Proposition 3 (GDM *(Goh 2017)*). *To minimize (10) with gradient descent with momentum, the step size η needs to satisfy $0 < \eta\lambda_i < 2 + 2\beta$, for $\forall i$ and $0 \leq \beta \leq 1$.*

Proposition 4 (PPAM). *Let $\delta_i = \left(\frac{\beta+1}{1+\eta\lambda_i}\right)^2 - \frac{4\beta}{1+\eta\lambda_i}$. To minimize (10) with PPAM, the step size η and momentum β need to satisfy:*

- $\eta > \frac{\beta-1}{\lambda_i}$, *if* $\delta_i \leq 0$;
- $\frac{\beta+1}{1+\eta\lambda_i} + \sqrt{\delta_i} < 2$, *if* $\delta_i > 0$ *and* $\frac{\beta+1}{1+\eta\lambda_i} \geq 0$;
- $\frac{\beta+1}{1+\eta\lambda_i} - \sqrt{\delta_i} > -2$, *otherwise.*

Theory continued...

Unfair comparison

Assran and Rabbat (2020). "On the Convergence of Nesterov's Accelerated Gradient Method in Stochastic Settings"

$$\max\{\rho_\mu(\eta,\beta),\ \rho_L(\eta,\beta)\} < 1,$$

where $\rho_\lambda(\eta,\beta)$ for $\lambda \in \{\mu, L\}$ is defined as:

$$\rho_\lambda(\eta,\beta) = \begin{cases} \frac{|(1+\beta)(1-\eta\lambda)|}{2} + \frac{\sqrt{\Delta_\lambda}}{2} & \text{if } \Delta_\lambda \geq 0, \\ \sqrt{\beta(1-\eta\lambda)} & \text{otherwise,} \end{cases}$$

with $\Delta_\lambda = (1+\beta)^2(1-\eta\lambda)^2 - 4\beta(1-\eta\lambda)$.

strongly convex quadratic $f(\cdot)$

$$\begin{cases} \eta\lambda \geq 1, & \text{Converges if } -\psi_{\beta,\eta,\lambda} + \sqrt{\Delta_\lambda} < 2, \\ \frac{(1-\beta)^2}{(1+\beta)^2} \leq \eta\lambda < 1, & \text{Always converges,} \\ \eta\lambda < \frac{(1-\beta)^2}{(1+\beta)^2}, & \text{Converges if } \psi_{\beta,\eta,\lambda} + \sqrt{\Delta_\lambda} < 2. \end{cases}$$

$\beta = 0.9$

Theorem 4. *Let the following condition hold:*

$$\tau = \sqrt{\frac{4}{(1+\eta\mu)^4} + \frac{4\beta^2}{(1+\eta\mu)^2(4-(1+\beta)^2)}} < \frac{1}{2}. \qquad (18)$$

Then, for $\boxed{\mu\text{-strongly convex } f(\cdot)}$*, the initial conditions of SPPAM exponentially discount: i.e., in* (16),

$$\frac{2\sigma_1^T}{\sigma_1 - \sigma_2} = \tau^{-1} \cdot \left(\frac{2}{(1+\eta\mu)^2} + \tau\right)^T = C^T,$$

where $C \in (0,1)$.

Accelerated SGD (strongly convex *quadratic*):

$0.0028 \approx \frac{1}{361} \leq \eta\lambda \leq \frac{24}{19} \approx 1.26$ for $\lambda \in \{\mu, L\}$

SPPAM (strongly convex):

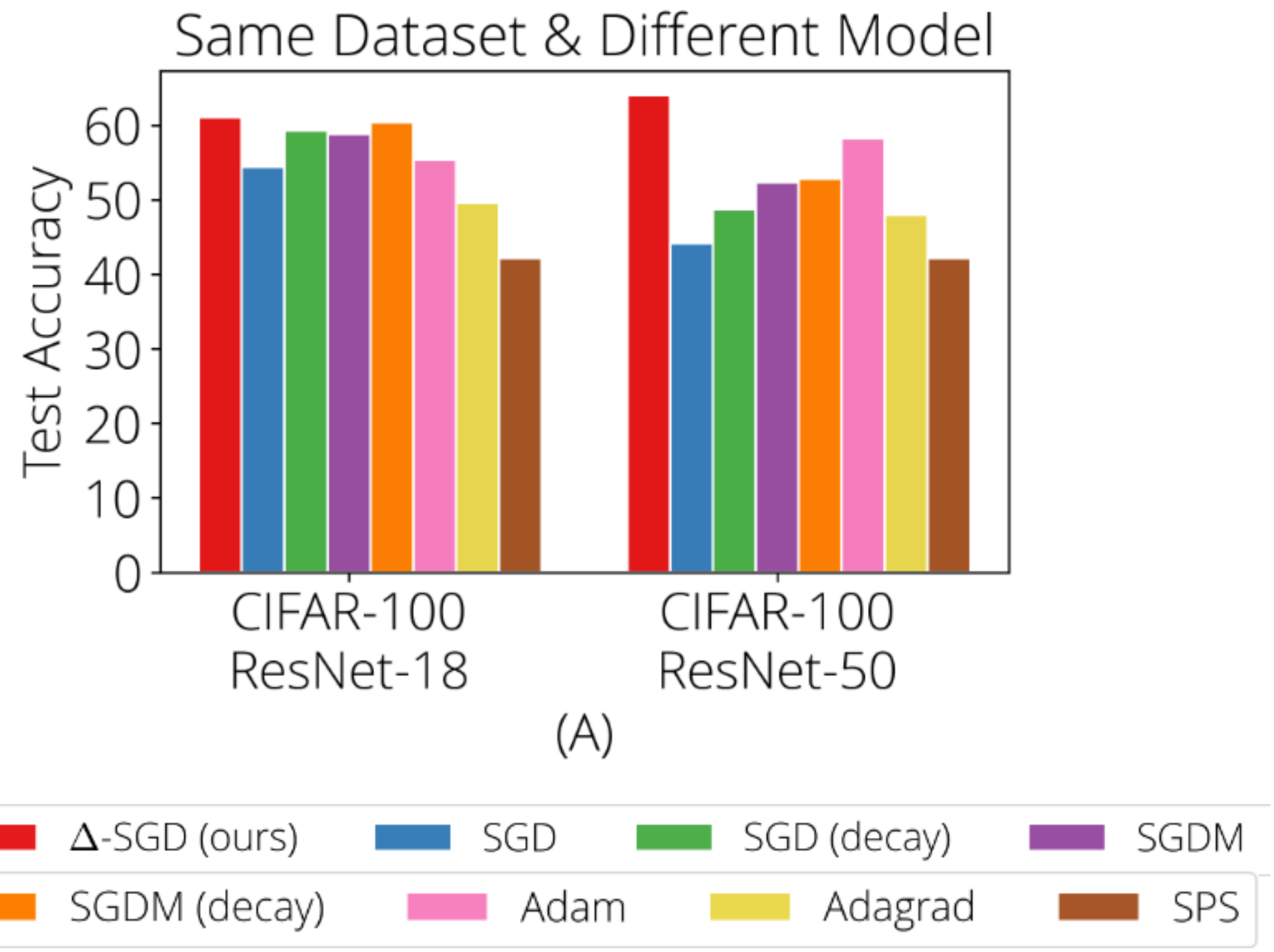
$\eta\mu > 4.81$ with $\beta = 0.9$

Adaptive FL

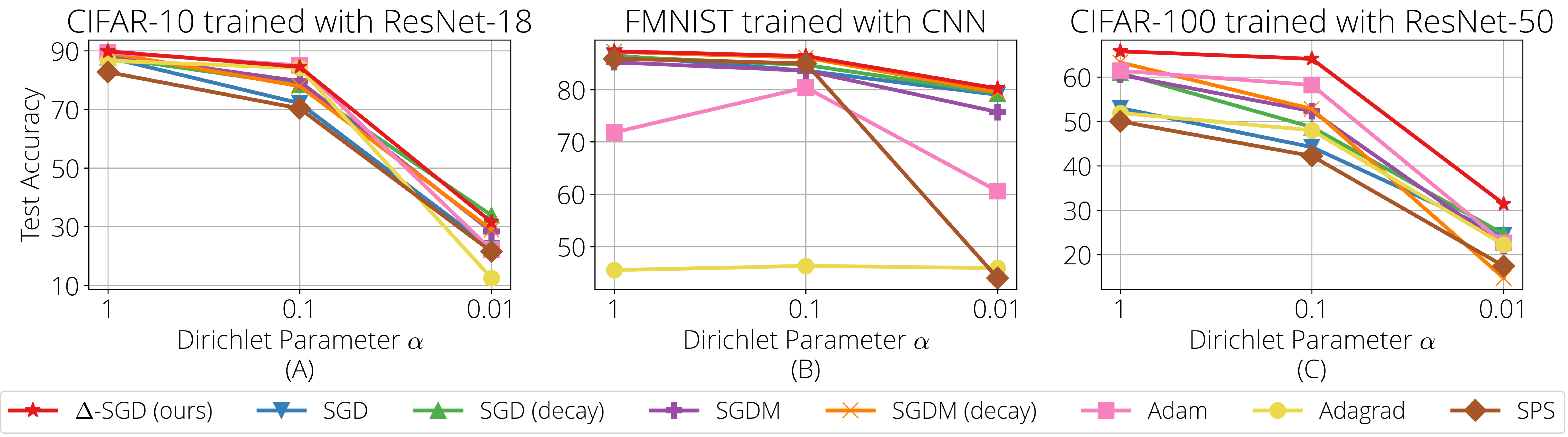
Same dataset & different model

Non-iidness	*Optimizer*	*Dataset / Model*				
Dir($\alpha \cdot \mathbf{p}$)		MNIST CNN	FMNIST CNN	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	CIFAR-100 ResNet-50
$\alpha = 1$	SGD	**98.3**$_{\downarrow(0.2)}$	86.5$_{\downarrow(0.8)}$	87.7$_{\downarrow(2.1)}$	57.7$_{\downarrow(4.2)}$	53.0$_{\downarrow(12.8)}$
	SGD (↓)	97.8$_{\downarrow(0.7)}$	86.3$_{\downarrow(1.0)}$	87.8$_{\downarrow(2.0)}$	**61.9**$_{\downarrow(0.0)}$	60.9$_{\downarrow(4.9)}$
	SGDM	**98.5**$_{\downarrow(0.0)}$	85.2$_{\downarrow(2.1)}$	88.7$_{\downarrow(1.1)}$	58.8$_{\downarrow(3.1)}$	60.5$_{\downarrow(5.3)}$
	SGDM (↓)	**98.4**$_{\downarrow(0.1)}$	**87.2**$_{\downarrow(0.1)}$	**89.3**$_{\downarrow(0.5)}$	**61.4**$_{\downarrow(0.5)}$	63.3$_{\downarrow(2.5)}$
	Adam	94.7$_{\downarrow(3.8)}$	71.8$_{\downarrow(15.5)}$	**89.4**$_{\downarrow(0.4)}$	55.6$_{\downarrow(6.3)}$	61.4$_{\downarrow(4.4)}$
	Adagrad	64.3$_{\downarrow(34.2)}$	45.5$_{\downarrow(41.8)}$	86.6$_{\downarrow(3.2)}$	53.5$_{\downarrow(8.4)}$	51.9$_{\downarrow(13.9)}$
	SPS	10.1$_{\downarrow(88.4)}$	85.9$_{\downarrow(1.4)}$	82.7$_{\downarrow(7.1)}$	1.0$_{\downarrow(60.9)}$	50.0$_{\downarrow(15.8)}$
	Δ-SGD	**98.4**$_{\downarrow(0.1)}$	**87.3**$_{\downarrow(0.0)}$	**89.8**$_{\downarrow(0.0)}$	**61.5**$_{\downarrow(0.4)}$	**65.8**$_{\downarrow(0.0)}$
$\alpha = 0.1$	SGD	**98.1**$_{\downarrow(0.0)}$	83.6$_{\downarrow(2.8)}$	72.1$_{\downarrow(12.9)}$	54.4$_{\downarrow(6.7)}$	44.2$_{\downarrow(19.9)}$
	SGD (↓)	**98.0**$_{\downarrow(0.1)}$	84.7$_{\downarrow(1.7)}$	78.4$_{\downarrow(6.6)}$	59.3$_{\downarrow(1.8)}$	48.7$_{\downarrow(15.4)}$
	SGDM	**97.6**$_{\downarrow(0.5)}$	83.6$_{\downarrow(2.8)}$	79.6$_{\downarrow(5.4)}$	58.8$_{\downarrow(2.3)}$	52.3$_{\downarrow(11.8)}$
	SGDM (↓)	**98.0**$_{\downarrow(0.1)}$	**86.1**$_{\downarrow(0.3)}$	77.9$_{\downarrow(7.1)}$	60.4$_{\downarrow(0.7)}$	52.8$_{\downarrow(11.3)}$
	Adam	96.4$_{\downarrow(1.7)}$	80.4$_{\downarrow(6.0)}$	**85.0**$_{\downarrow(0.0)}$	55.4$_{\downarrow(5.7)}$	58.2$_{\downarrow(5.9)}$
	Adagrad	89.9$_{\downarrow(8.2)}$	46.3$_{\downarrow(40.1)}$	84.1$_{\downarrow(0.9)}$	49.6$_{\downarrow(11.5)}$	48.0$_{\downarrow(16.1)}$
	SPS	96.0$_{\downarrow(2.1)}$	85.0$_{\downarrow(1.4)}$	70.3$_{\downarrow(14.7)}$	42.2$_{\downarrow(18.9)}$	42.2$_{\downarrow(21.9)}$
	Δ-SGD	**98.1**$_{\downarrow(0.0)}$	**86.4**$_{\downarrow(0.0)}$	**84.5**$_{\downarrow(0.5)}$	**61.1**$_{\downarrow(0.0)}$	**64.1**$_{\downarrow(0.0)}$
$\alpha = 0.01$	SGD	96.8$_{\downarrow(0.7)}$	79.0$_{\downarrow(1.2)}$	22.6$_{\downarrow(11.3)}$	30.5$_{\downarrow(1.3)}$	24.3$_{\downarrow(7.1)}$
	SGD (↓)	**97.2**$_{\downarrow(0.3)}$	79.3$_{\downarrow(0.9)}$	**33.9**$_{\downarrow(0.0)}$	30.3$_{\downarrow(1.5)}$	24.6$_{\downarrow(6.8)}$
	SGDM	77.9$_{\downarrow(19.6)}$	75.7$_{\downarrow(4.5)}$	28.4$_{\downarrow(5.5)}$	24.8$_{\downarrow(7.0)}$	22.0$_{\downarrow(9.4)}$
	SGDM (↓)	94.0$_{\downarrow(3.5)}$	79.5$_{\downarrow(0.7)}$	29.0$_{\downarrow(4.9)}$	20.9$_{\downarrow(10.9)}$	14.7$_{\downarrow(16.7)}$
	Adam	80.8$_{\downarrow(16.7)}$	60.6$_{\downarrow(19.6)}$	22.1$_{\downarrow(11.8)}$	18.2$_{\downarrow(13.6)}$	22.6$_{\downarrow(8.8)}$
	Adagrad	72.4$_{\downarrow(25.1)}$	45.9$_{\downarrow(34.3)}$	12.5$_{\downarrow(21.4)}$	25.8$_{\downarrow(6.0)}$	22.2$_{\downarrow(9.2)}$
	SPS	69.7$_{\downarrow(27.8)}$	44.0$_{\downarrow(36.2)}$	21.5$_{\downarrow(12.4)}$	22.0$_{\downarrow(9.8)}$	17.4$_{\downarrow(14.0)}$
	Δ-SGD	**97.5**$_{\downarrow(0.0)}$	**80.2**$_{\downarrow(0.0)}$	31.6$_{\downarrow(2.3)}$	**31.8**$_{\downarrow(0.0)}$	**31.4**$_{\downarrow(0.0)}$

$\alpha = 0.1$

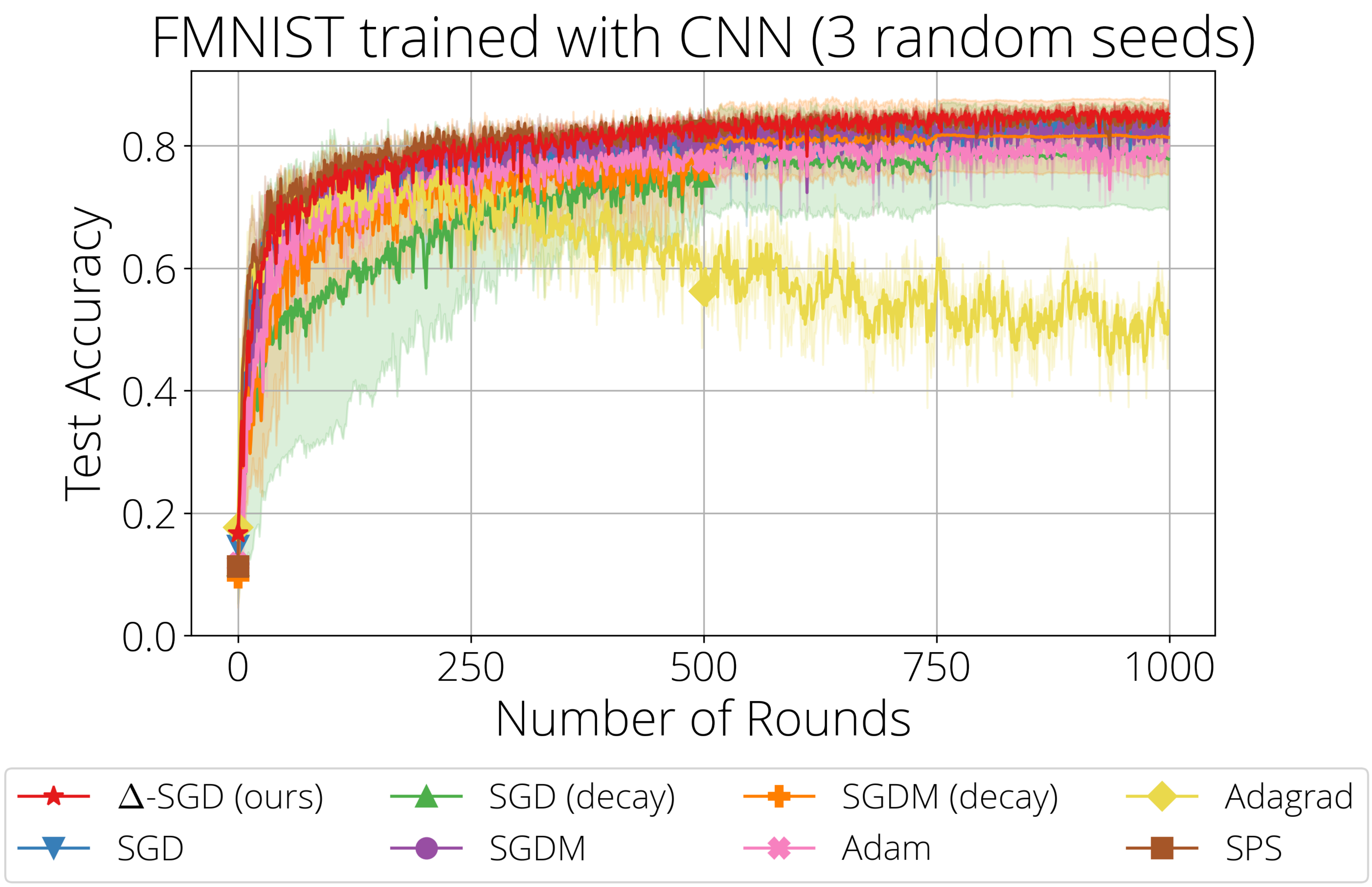


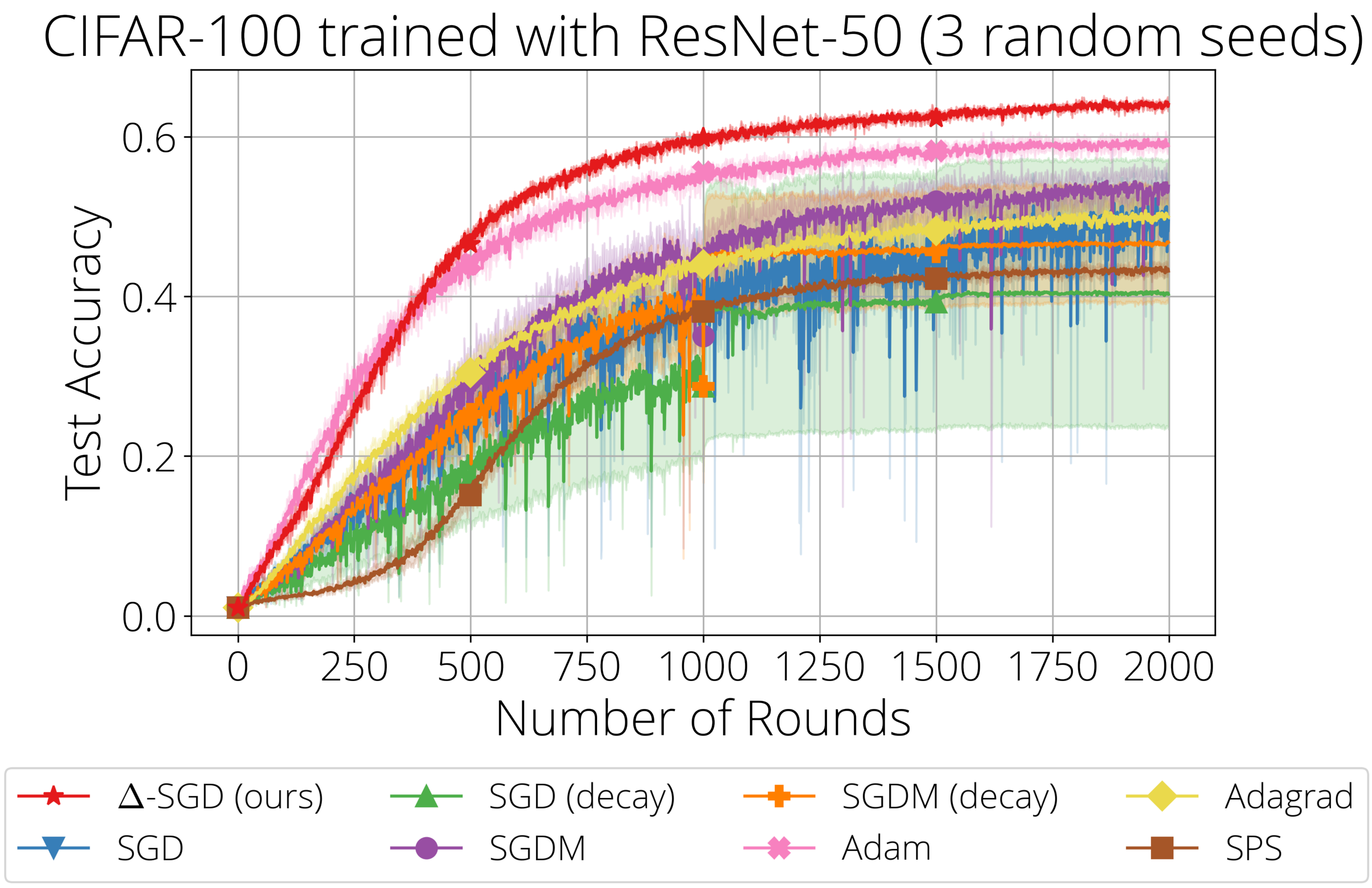
Effect of different level of non-iidness



Once you change the dataset / architecture / heterogeneity, you have to fine-tune your client optimizer again to ensure proper learning. Δ-SGD performs well and robustly in different FL settings.

Average across three random seeds





Different loss functions

FedProx	*Dataset / Model*	
$\alpha = 0.01$ *Optimizer*	CIFAR-10 ResNet-18	CIFAR-100 ResNet-50
SGD	$20.0_{\downarrow(13.8)}$	$25.2_{\downarrow(5.9)}$
SGD (↓)	$31.3_{\downarrow(2.5)}$	$20.2_{\downarrow(10.8)}$
SGDM	$29.3_{\downarrow(4.4)}$	$23.8_{\downarrow(7.2)}$
SGDM (↓)	$25.3_{\downarrow(8.5)}$	$15.0_{\downarrow(16.0)}$
Adam	$28.1_{\downarrow(5.7)}$	$22.6_{\downarrow(8.4)}$
Adagrad	$19.3_{\downarrow(14.5)}$	$4.1_{\downarrow(26.9)}$
SPS	$27.6_{\downarrow(6.2)}$	$16.5_{\downarrow(14.5)}$
Δ-SGD	$\mathbf{33.8}_{\downarrow(0.0)}$	$\mathbf{31.0}_{\downarrow(0.0)}$

FedProx

Non-iidness	*Optimizer*	*Dataset / Model*
$\text{Dir}(\alpha \cdot \mathbf{p})$		CIFAR-10 ResNet-18
$\alpha = 0.1$	SGD	$78.2_{\downarrow(4.9)}$
	SGD (↓)	$74.2_{\downarrow(8.9)}$
	SGDM	$76.4_{\downarrow(6.7)}$
	SGDM (↓)	$75.5_{\downarrow(14.1)}$
	Adam	$82.4_{\downarrow(0.6)}$
	Adagrad	$81.3_{\downarrow(1.8)}$
	SPS	$9.57_{\downarrow(73.5)}$
	Δ-SGD	$\mathbf{83.1}_{\downarrow(0.0)}$

MOON

Our proposed step size for Δ-SGD

$$\min_{x \in \mathbb{R}^d} \; f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

Where does this come from?

$$\eta_t^i = \min \left\{ \frac{\|x_t^i - x_{t-1}^i\|}{2\|\nabla f_i(x_t^i) - \nabla f_i(x_{t-1}^i)\|}, \sqrt{1 + \theta_{t-1}^i \eta_{t-1}^i} \right\}, \quad \theta_{t-1}^i = \eta_{t-1}^i / \eta_{t-2}^i$$

Client i local updates: $x_{t+1}^i = x_t^i - \eta_t^i \nabla f_i(x_t^i)$

- Each client uses its **own** step size
- Step size only requires known quantities (i.e., no tuning required)
- Individual step size is **adaptive** to the **local smoothness** of f_i

What is a good step size for gradient descent?

L-smooth functions: $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \quad \forall x, y$

Gradient descent: $x_{t+1} = x_t - \eta \nabla f\left(x_t\right)$

$$f(x_{t+1}) \leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2$$

$$= f(x_t) - \boxed{\eta\left(1 - \frac{\eta \cdot L}{2}\right)}\|\nabla f(x_t)\|^2$$

$\eta = \frac{1}{L}$ "Optimal"

$$f(x_{t+1}) - f(x^\star) \leq \frac{L\|x_0 - x^\star\|^2}{2(2t+1)}$$

Hard to estimate L in practice

Even trickier in distributed/FL scenario

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x) \leftarrow \mathbb{E}_{z \sim \mathcal{D}_i}[F_i(x, z)]$$

Assuming $f_i(\cdot)$ is L_i smooth (hard to estimate in practice),
step size of the form $\frac{1}{L_{max}}$ is often used for all i, where $L_{max} := \max_i L_i$

Suboptimal! f_i can be different for each i

E.g., $\min_{x \in \mathbb{R}^d} \frac{1}{2}\left(f_1(x) + f_2(x)\right)$ where f_1 is 1-smooth and f_2 is 10000-smooth

Local smoothness?

$$\|\nabla f(x) - \nabla f(y)\| \leq L \cdot \|x - y\| \quad \forall x, y \implies \frac{1}{L} \leq \frac{\|x - y\|}{\|\nabla f(x) - \nabla f(y)\|}$$

$$\eta_t = \min\left\{\frac{\|x_t - x_{t-1}\|}{2\|\nabla f(x_t) - \nabla f(x_{t-1})\|}, \boxed{\sqrt{1 + \theta_{t-1}}\eta_{t-1}}\right\}, \quad \theta_{t-1} = \eta_{t-1}/\eta_{t-2}$$

$$\|\nabla f(x_t) - \nabla f(x_{t-1})\| \leq L_t \cdot \|x_t - x_{t-1}\|, \quad \forall t = 1,2,\ldots$$

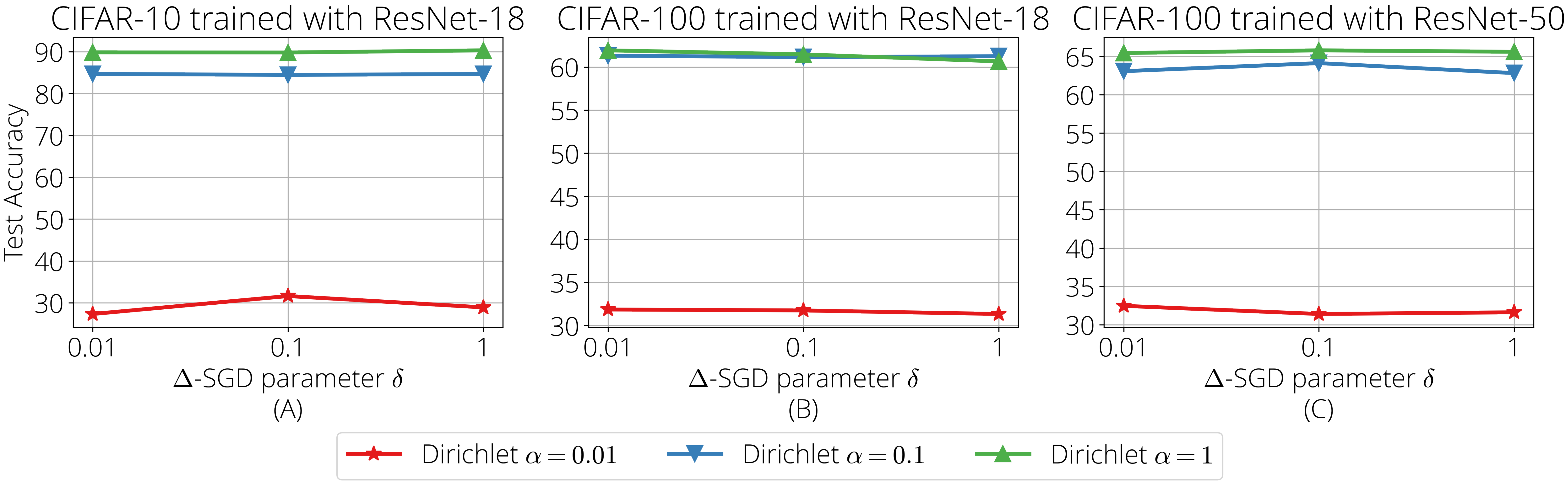
$$2\eta_{t+1}\theta_{t+1} \leq 2\eta_t(1 + \theta_t)$$

$$\|x_{t+1} - x^\star\|^2 + \frac{1}{2}\|x_{t+1} - x_t\|^2 + \boxed{2\eta_t(1 + \theta_t)}(f(x_t) - f(x^\star))$$
$$\leq \|x_t - x^\star\|^2 + \frac{1}{2}\|x_t - x_{t-1}\|^2 + 2\eta_t\theta_t(f(x_{t-1}) - f(x^\star))$$

[Malisky & Mishchenko (2020), "Adaptive Gradient Descent without Descent"]

"Almost" no tuning

$$\eta_{t,k}^i = \min\left\{ \frac{\gamma \|x_{t,k}^i - x_{t,k-1}^i\|}{2\|\tilde{\nabla} f_i(x_{t,k}^i) - \tilde{\nabla} f_i(x_{t,k-1}^i)\|}, \sqrt{1 + \delta\theta_{t,k-1}^i}\,\eta_{t,k-1}^i \right\}$$



Additional experiments with FedAdam

Additional experiments using FedAdam (Reddi et al., 2021)						
	Non-iidness	*Optimizer*	*Dataset / Model*			
	$\mathrm{Dir}(\alpha \cdot \mathbf{p})$		MNIST CNN	FMNIST CNN	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18
FedAdam	$\alpha = 0.1$	SGD	$97.3_{\downarrow(0.6)}$	$83.7_{\downarrow(2.1)}$	$52.0_{\downarrow(11.8)}$	$46.7_{\downarrow(2.5)}$
		SGD (↓)	$96.4_{\downarrow(1.4)}$	$80.9_{\downarrow(4.9)}$	$49.1_{\downarrow(14.7)}$	**49.2**$_{\downarrow(0.0)}$
		SGDM	**97.5**$_{\downarrow(0.4)}$	$84.6_{\downarrow(1.2)}$	$53.7_{\downarrow(10.1)}$	$13.3_{\downarrow(35.9)}$
		SGDM (↓)	$96.4_{\downarrow(1.5)}$	$81.8_{\downarrow(4.0)}$	$53.3_{\downarrow(10.5)}$	$16.8_{\downarrow(32.4)}$
		Adam	$96.4_{\downarrow(1.5)}$	$81.5_{\downarrow(4.3)}$	$27.8_{\downarrow(36.0)}$	$38.3_{\downarrow(10.9)}$
		Adagrad	$95.7_{\downarrow(2.2)}$	$82.1_{\downarrow(3.7)}$	$10.4_{\downarrow(53.4)}$	$1.0_{\downarrow(48.2)}$
		SPS	$96.6_{\downarrow(1.3)}$	$85.0_{\downarrow(0.8)}$	$21.6_{\downarrow(42.2)}$	$1.6_{\downarrow(47.6)}$
		Δ-SGD	**97.9**$_{\downarrow(0.0)}$	**85.8**$_{\downarrow(0.0)}$	**63.8**$_{\downarrow(0.0)}$	$41.9_{\downarrow(7.3)}$
FedAvg	$\alpha = 0.1$	SGD	**98.1**$_{\downarrow(0.0)}$	$83.6_{\downarrow(2.8)}$	$72.1_{\downarrow(12.9)}$	$54.4_{\downarrow(6.7)}$
		SGD (↓)	**98.0**$_{\downarrow(0.1)}$	$84.7_{\downarrow(1.7)}$	$78.4_{\downarrow(6.6)}$	$59.3_{\downarrow(1.8)}$
		SGDM	**97.6**$_{\downarrow(0.5)}$	$83.6_{\downarrow(2.8)}$	$79.6_{\downarrow(5.4)}$	$58.8_{\downarrow(2.3)}$
		SGDM (↓)	**98.0**$_{\downarrow(0.1)}$	**86.1**$_{\downarrow(0.3)}$	$77.9_{\downarrow(7.1)}$	$60.4_{\downarrow(0.7)}$
		Adam	$96.4_{\downarrow(1.7)}$	$80.4_{\downarrow(6.0)}$	**85.0**$_{\downarrow(0.0)}$	$55.4_{\downarrow(5.7)}$
		Adagrad	$89.9_{\downarrow(8.2)}$	$46.3_{\downarrow(40.1)}$	$84.1_{\downarrow(0.9)}$	$49.6_{\downarrow(11.5)}$
		SPS	$96.0_{\downarrow(2.1)}$	$85.0_{\downarrow(1.4)}$	$70.3_{\downarrow(14.7)}$	$42.2_{\downarrow(18.9)}$
		Δ-SGD	**98.1**$_{\downarrow(0.0)}$	**86.4**$_{\downarrow(0.0)}$	**84.5**$_{\downarrow(0.5)}$	**61.1**$_{\downarrow(0.0)}$

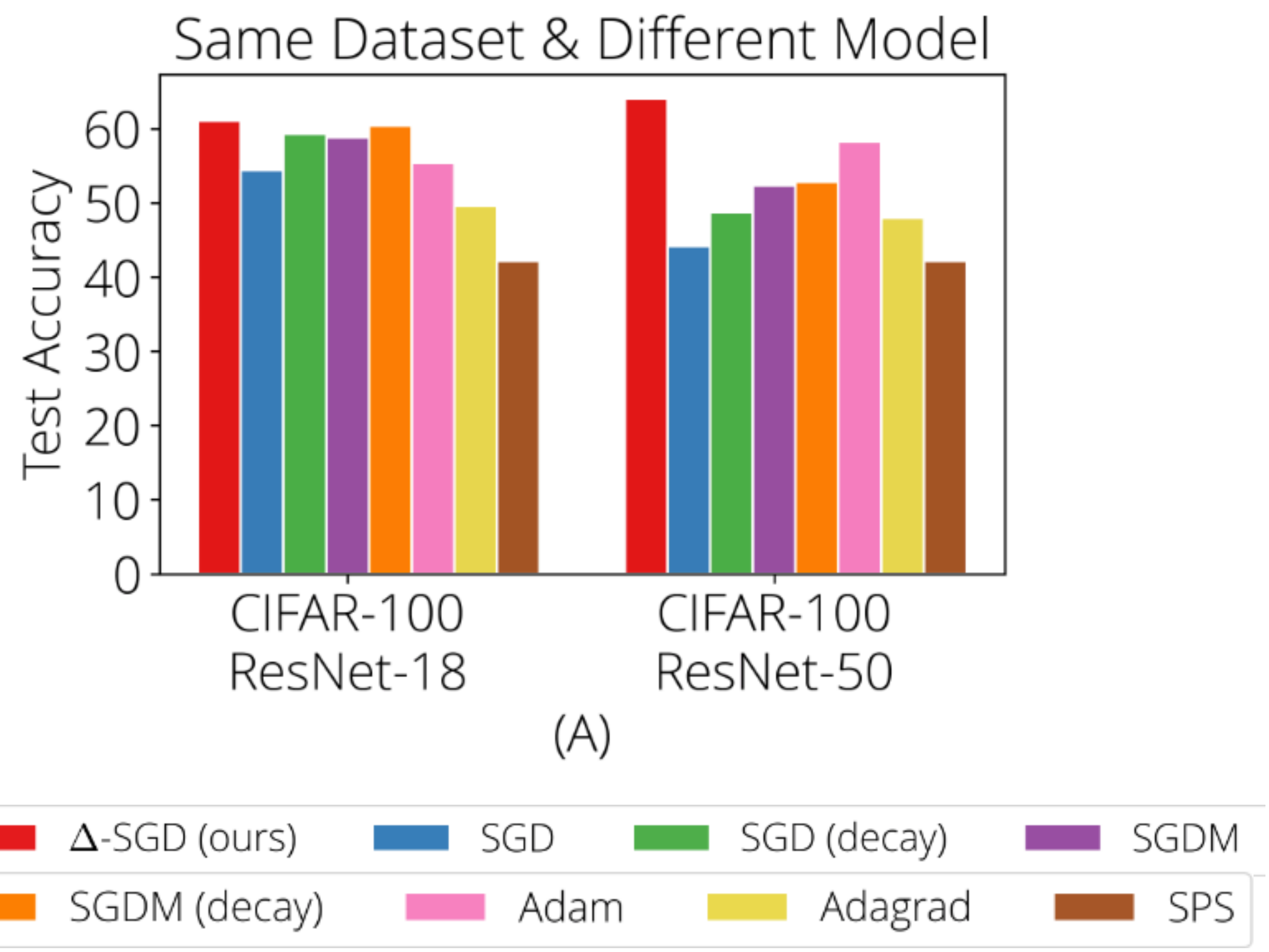
Changing the domain: text classification

Text classification	*Dataset / Model*	
$\alpha = 1$	Agnews	Dbpedia
Optimizer	DistillBERT	
SGD	**91.1**$_{\downarrow(0.5)}$	96.0$_{\downarrow(2.9)}$
SGD ($\downarrow$)	**91.6**$_{\downarrow(0.0)}$	**98.7**$_{\downarrow(0.2)}$
SGDM	25.0$_{\downarrow(66.6)}$	7.1$_{\downarrow(91.8)}$
SGDM ($\downarrow$)	25.0$_{\downarrow(66.6)}$	7.1$_{\downarrow(91.8)}$
Adam	25.0$_{\downarrow(66.6)}$	7.1$_{\downarrow(91.8)}$
Adagrad	25.0$_{\downarrow(66.6)}$	7.1$_{\downarrow(91.8)}$
SPS	**91.5**$_{\downarrow(0.1)}$	**98.9**$_{\downarrow(0.0)}$
Δ-SGD	90.7$_{\downarrow(0.9)}$	**98.6**$_{\downarrow(0.3)}$

Same dataset & different model

Non-iidness	*Optimizer*	*Dataset / Model*				
Dir($\alpha \cdot \mathbf{p}$)		MNIST CNN	FMNIST CNN	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	CIFAR-100 ResNet-50
$\alpha = 1$	SGD	$\mathbf{98.3}_{\downarrow(0.2)}$	$86.5_{\downarrow(0.8)}$	$87.7_{\downarrow(2.1)}$	$57.7_{\downarrow(4.2)}$	$53.0_{\downarrow(12.8)}$
	SGD (↓)	$97.8_{\downarrow(0.7)}$	$86.3_{\downarrow(1.0)}$	$87.8_{\downarrow(2.0)}$	$\mathbf{61.9}_{\downarrow(0.0)}$	$60.9_{\downarrow(4.9)}$
	SGDM	$\mathbf{98.5}_{\downarrow(0.0)}$	$85.2_{\downarrow(2.1)}$	$88.7_{\downarrow(1.1)}$	$58.8_{\downarrow(3.1)}$	$60.5_{\downarrow(5.3)}$
	SGDM (↓)	$\mathbf{98.4}_{\downarrow(0.1)}$	$\mathbf{87.2}_{\downarrow(0.1)}$	$\mathbf{89.3}_{\downarrow(0.5)}$	$\mathbf{61.4}_{\downarrow(0.5)}$	$63.3_{\downarrow(2.5)}$
	Adam	$94.7_{\downarrow(3.8)}$	$71.8_{\downarrow(15.5)}$	$\mathbf{89.4}_{\downarrow(0.4)}$	$55.6_{\downarrow(6.3)}$	$61.4_{\downarrow(4.4)}$
	Adagrad	$64.3_{\downarrow(34.2)}$	$45.5_{\downarrow(41.8)}$	$86.6_{\downarrow(3.2)}$	$53.5_{\downarrow(8.4)}$	$51.9_{\downarrow(13.9)}$
	SPS	$10.1_{\downarrow(88.4)}$	$85.9_{\downarrow(1.4)}$	$82.7_{\downarrow(7.1)}$	$1.0_{\downarrow(60.9)}$	$50.0_{\downarrow(15.8)}$
	Δ-SGD	$\mathbf{98.4}_{\downarrow(0.1)}$	$\mathbf{87.3}_{\downarrow(0.0)}$	$\mathbf{89.8}_{\downarrow(0.0)}$	$\mathbf{61.5}_{\downarrow(0.4)}$	$\mathbf{65.8}_{\downarrow(0.0)}$
$\alpha = 0.1$	SGD	$\mathbf{98.1}_{\downarrow(0.0)}$	$83.6_{\downarrow(2.8)}$	$72.1_{\downarrow(12.9)}$	$54.4_{\downarrow(6.7)}$	$44.2_{\downarrow(19.9)}$
	SGD (↓)	$\mathbf{98.0}_{\downarrow(0.1)}$	$84.7_{\downarrow(1.7)}$	$78.4_{\downarrow(6.6)}$	$59.3_{\downarrow(1.8)}$	$48.7_{\downarrow(15.4)}$
	SGDM	$\mathbf{97.6}_{\downarrow(0.5)}$	$83.6_{\downarrow(2.8)}$	$79.6_{\downarrow(5.4)}$	$58.8_{\downarrow(2.3)}$	$52.3_{\downarrow(11.8)}$
	SGDM (↓)	$\mathbf{98.0}_{\downarrow(0.1)}$	$\mathbf{86.1}_{\downarrow(0.3)}$	$77.9_{\downarrow(7.1)}$	$60.4_{\downarrow(0.7)}$	$52.8_{\downarrow(11.3)}$
	Adam	$96.4_{\downarrow(1.7)}$	$80.4_{\downarrow(6.0)}$	$\mathbf{85.0}_{\downarrow(0.0)}$	$55.4_{\downarrow(5.7)}$	$58.2_{\downarrow(5.9)}$
	Adagrad	$89.9_{\downarrow(8.2)}$	$46.3_{\downarrow(40.1)}$	$84.1_{\downarrow(0.9)}$	$49.6_{\downarrow(11.5)}$	$48.0_{\downarrow(16.1)}$
	SPS	$96.0_{\downarrow(2.1)}$	$85.0_{\downarrow(1.4)}$	$70.3_{\downarrow(14.7)}$	$42.2_{\downarrow(18.9)}$	$42.2_{\downarrow(21.9)}$
	Δ-SGD	$\mathbf{98.1}_{\downarrow(0.0)}$	$\mathbf{86.4}_{\downarrow(0.0)}$	$\mathbf{84.5}_{\downarrow(0.5)}$	$\mathbf{61.1}_{\downarrow(0.0)}$	$\mathbf{64.1}_{\downarrow(0.0)}$
$\alpha = 0.01$	SGD	$96.8_{\downarrow(0.7)}$	$79.0_{\downarrow(1.2)}$	$22.6_{\downarrow(11.3)}$	$30.5_{\downarrow(1.3)}$	$24.3_{\downarrow(7.1)}$
	SGD (↓)	$\mathbf{97.2}_{\downarrow(0.3)}$	$79.3_{\downarrow(0.9)}$	$\mathbf{33.9}_{\downarrow(0.0)}$	$30.3_{\downarrow(1.5)}$	$24.6_{\downarrow(6.8)}$
	SGDM	$77.9_{\downarrow(19.6)}$	$75.7_{\downarrow(4.5)}$	$28.4_{\downarrow(5.5)}$	$24.8_{\downarrow(7.0)}$	$22.0_{\downarrow(9.4)}$
	SGDM (↓)	$94.0_{\downarrow(3.5)}$	$79.5_{\downarrow(0.7)}$	$29.0_{\downarrow(4.9)}$	$20.9_{\downarrow(10.9)}$	$14.7_{\downarrow(16.7)}$
	Adam	$80.8_{\downarrow(16.7)}$	$60.6_{\downarrow(19.6)}$	$22.1_{\downarrow(11.8)}$	$18.2_{\downarrow(13.6)}$	$22.6_{\downarrow(8.8)}$
	Adagrad	$72.4_{\downarrow(25.1)}$	$45.9_{\downarrow(34.3)}$	$12.5_{\downarrow(21.4)}$	$25.8_{\downarrow(6.0)}$	$22.2_{\downarrow(9.2)}$
	SPS	$69.7_{\downarrow(27.8)}$	$44.0_{\downarrow(36.2)}$	$21.5_{\downarrow(12.4)}$	$22.0_{\downarrow(9.8)}$	$17.4_{\downarrow(14.0)}$
	Δ-SGD	$\mathbf{97.5}_{\downarrow(0.0)}$	$\mathbf{80.2}_{\downarrow(0.0)}$	$31.6_{\downarrow(2.3)}$	$\mathbf{31.8}_{\downarrow(0.0)}$	$\mathbf{31.4}_{\downarrow(0.0)}$

$\alpha = 0.1$

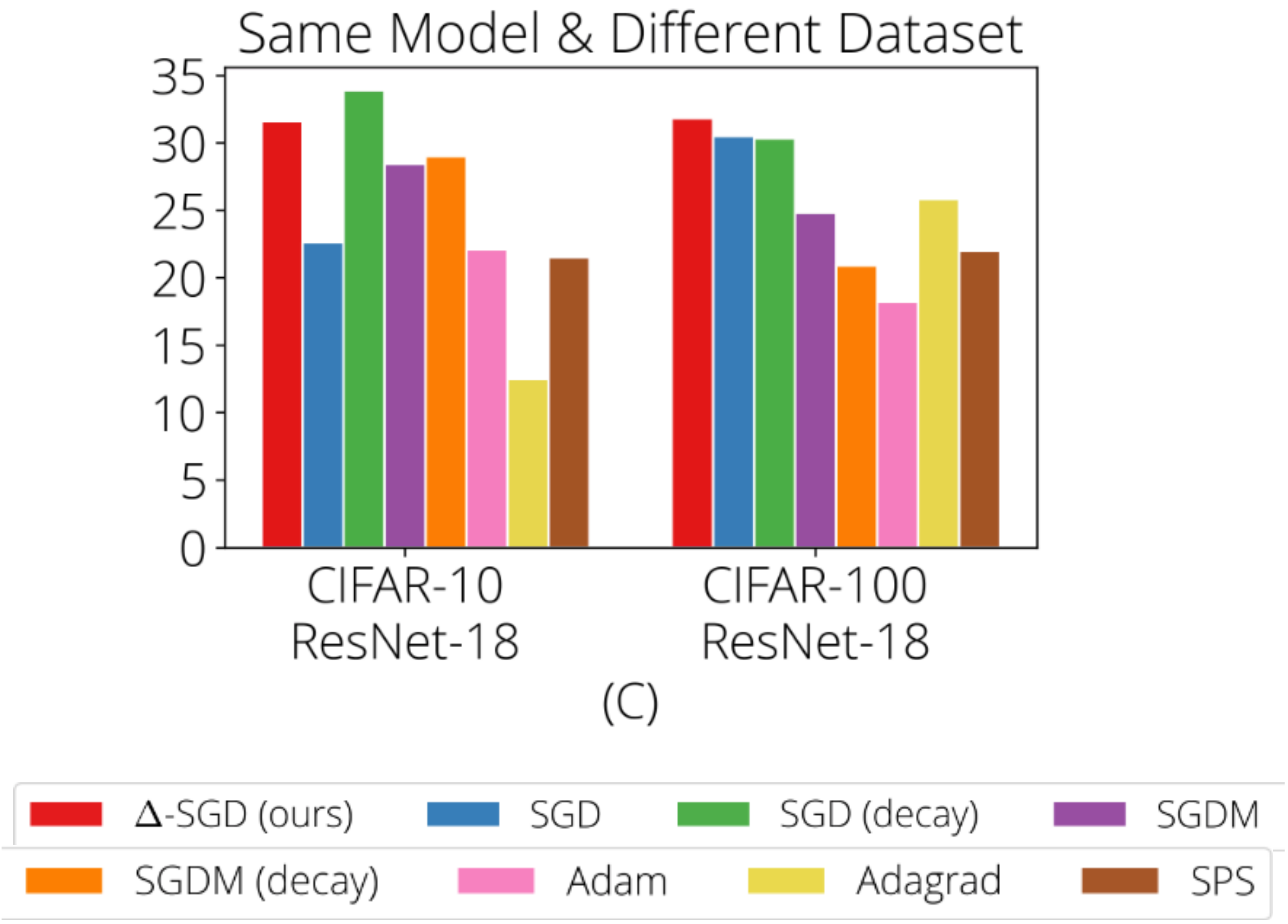


(A)

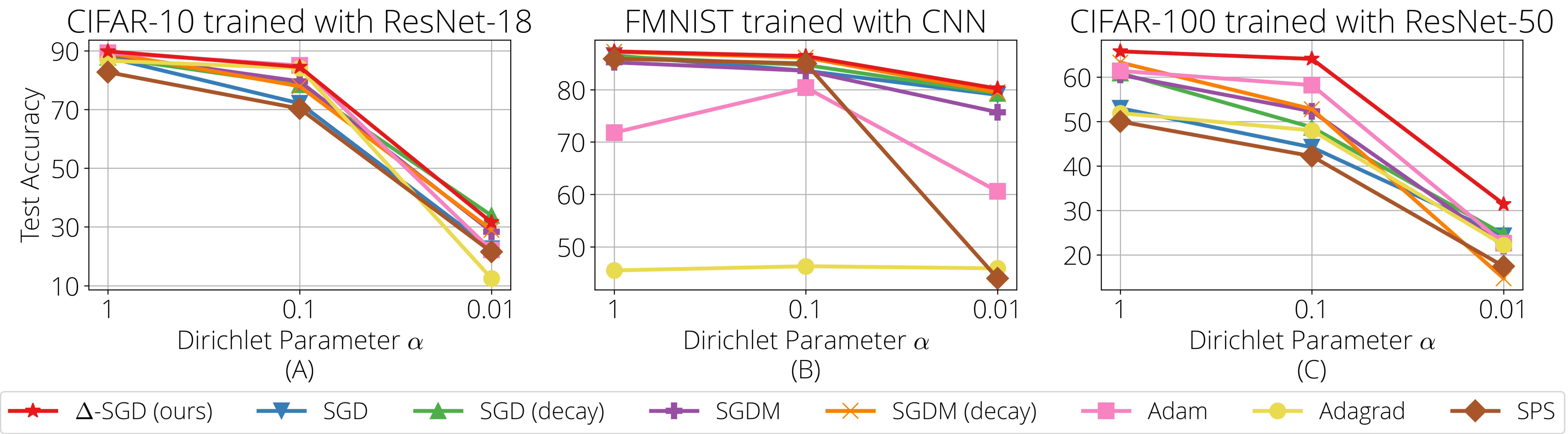
Same model & different dataset

Non-iidness	*Optimizer*	*Dataset / Model*				
Dir($\alpha \cdot \mathbf{p}$)		MNIST CNN	FMNIST CNN	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	CIFAR-100 ResNet-50
$\alpha = 1$	SGD	**98.3**$_{\downarrow(0.2)}$	86.5$_{\downarrow(0.8)}$	87.7$_{\downarrow(2.1)}$	57.7$_{\downarrow(4.2)}$	53.0$_{\downarrow(12.8)}$
	SGD (↓)	97.8$_{\downarrow(0.7)}$	86.3$_{\downarrow(1.0)}$	87.8$_{\downarrow(2.0)}$	**61.9**$_{\downarrow(0.0)}$	60.9$_{\downarrow(4.9)}$
	SGDM	**98.5**$_{\downarrow(0.0)}$	85.2$_{\downarrow(2.1)}$	88.7$_{\downarrow(1.1)}$	58.8$_{\downarrow(3.1)}$	60.5$_{\downarrow(5.3)}$
	SGDM (↓)	**98.4**$_{\downarrow(0.1)}$	**87.2**$_{\downarrow(0.1)}$	**89.3**$_{\downarrow(0.5)}$	**61.4**$_{\downarrow(0.5)}$	63.3$_{\downarrow(2.5)}$
	Adam	94.7$_{\downarrow(3.8)}$	71.8$_{\downarrow(15.5)}$	**89.4**$_{\downarrow(0.4)}$	55.6$_{\downarrow(6.3)}$	61.4$_{\downarrow(4.4)}$
	Adagrad	64.3$_{\downarrow(34.2)}$	45.5$_{\downarrow(41.8)}$	86.6$_{\downarrow(3.2)}$	53.5$_{\downarrow(8.4)}$	51.9$_{\downarrow(13.9)}$
	SPS	10.1$_{\downarrow(88.4)}$	85.9$_{\downarrow(1.4)}$	82.7$_{\downarrow(7.1)}$	1.0$_{\downarrow(60.9)}$	50.0$_{\downarrow(15.8)}$
	Δ-SGD	**98.4**$_{\downarrow(0.1)}$	**87.3**$_{\downarrow(0.0)}$	**89.8**$_{\downarrow(0.0)}$	**61.5**$_{\downarrow(0.4)}$	**65.8**$_{\downarrow(0.0)}$
$\alpha = 0.1$	SGD	**98.1**$_{\downarrow(0.0)}$	83.6$_{\downarrow(2.8)}$	72.1$_{\downarrow(12.9)}$	54.4$_{\downarrow(6.7)}$	44.2$_{\downarrow(19.9)}$
	SGD (↓)	**98.0**$_{\downarrow(0.1)}$	84.7$_{\downarrow(1.7)}$	78.4$_{\downarrow(6.6)}$	59.3$_{\downarrow(1.8)}$	48.7$_{\downarrow(15.4)}$
	SGDM	**97.6**$_{\downarrow(0.5)}$	83.6$_{\downarrow(2.8)}$	79.6$_{\downarrow(5.4)}$	58.8$_{\downarrow(2.3)}$	52.3$_{\downarrow(11.8)}$
	SGDM (↓)	**98.0**$_{\downarrow(0.1)}$	**86.1**$_{\downarrow(0.3)}$	77.9$_{\downarrow(7.1)}$	60.4$_{\downarrow(0.7)}$	52.8$_{\downarrow(11.3)}$
	Adam	96.4$_{\downarrow(1.7)}$	80.4$_{\downarrow(6.0)}$	**85.0**$_{\downarrow(0.0)}$	55.4$_{\downarrow(5.7)}$	58.2$_{\downarrow(5.9)}$
	Adagrad	89.9$_{\downarrow(8.2)}$	46.3$_{\downarrow(40.1)}$	84.1$_{\downarrow(0.9)}$	49.6$_{\downarrow(11.5)}$	48.0$_{\downarrow(16.1)}$
	SPS	96.0$_{\downarrow(2.1)}$	85.0$_{\downarrow(1.4)}$	70.3$_{\downarrow(14.7)}$	42.2$_{\downarrow(18.9)}$	42.2$_{\downarrow(21.9)}$
	Δ-SGD	**98.1**$_{\downarrow(0.0)}$	**86.4**$_{\downarrow(0.0)}$	**84.5**$_{\downarrow(0.5)}$	**61.1**$_{\downarrow(0.0)}$	**64.1**$_{\downarrow(0.0)}$
$\alpha = 0.01$	SGD	96.8$_{\downarrow(0.7)}$	79.0$_{\downarrow(1.2)}$	22.6$_{\downarrow(11.3)}$	30.5$_{\downarrow(1.3)}$	24.3$_{\downarrow(7.1)}$
	SGD (↓)	**97.2**$_{\downarrow(0.3)}$	79.3$_{\downarrow(0.9)}$	**33.9**$_{\downarrow(0.0)}$	30.3$_{\downarrow(1.5)}$	24.6$_{\downarrow(6.8)}$
	SGDM	77.9$_{\downarrow(19.6)}$	75.7$_{\downarrow(4.5)}$	28.4$_{\downarrow(5.5)}$	24.8$_{\downarrow(7.0)}$	22.0$_{\downarrow(9.4)}$
	SGDM (↓)	94.0$_{\downarrow(3.5)}$	79.5$_{\downarrow(0.7)}$	29.0$_{\downarrow(4.9)}$	20.9$_{\downarrow(10.9)}$	14.7$_{\downarrow(16.7)}$
	Adam	80.8$_{\downarrow(16.7)}$	60.6$_{\downarrow(19.6)}$	22.1$_{\downarrow(11.8)}$	18.2$_{\downarrow(13.6)}$	22.6$_{\downarrow(8.8)}$
	Adagrad	72.4$_{\downarrow(25.1)}$	45.9$_{\downarrow(34.3)}$	12.5$_{\downarrow(21.4)}$	25.8$_{\downarrow(6.0)}$	22.2$_{\downarrow(9.2)}$
	SPS	69.7$_{\downarrow(27.8)}$	44.0$_{\downarrow(36.2)}$	21.5$_{\downarrow(12.4)}$	22.0$_{\downarrow(9.8)}$	17.4$_{\downarrow(14.0)}$
	Δ-SGD	**97.5**$_{\downarrow(0.0)}$	**80.2**$_{\downarrow(0.0)}$	31.6$_{\downarrow(2.3)}$	**31.8**$_{\downarrow(0.0)}$	**31.4**$_{\downarrow(0.0)}$

$\alpha = 0.01$



Effect of different level of non-iidness



Once you change the dataset / architecture / heterogeneity, you have to fine-tune your client optimizer again to ensure proper learning. ***Δ****-SGD performs well and robustly in different FL settings.*

Well-known adaptive step sizes have limitations

$$x_{t+1} = x_t - \eta_t \nabla f\left(x_t\right)$$

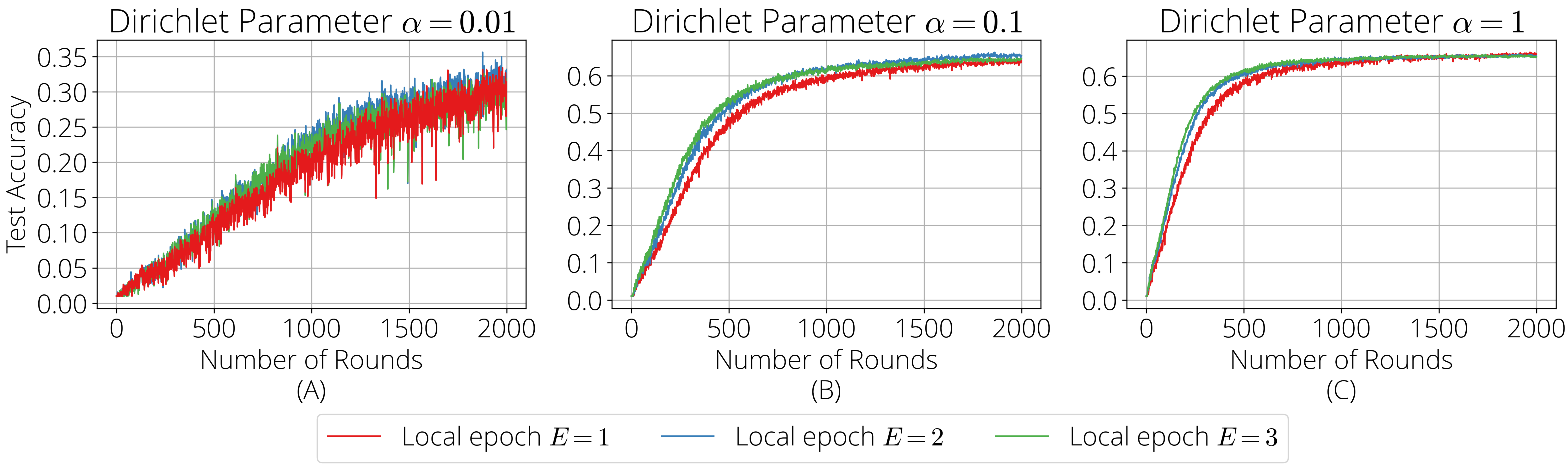
Line search: $\eta_t = \arg\min_{\eta} f\left(x_t - \eta \nabla f(x_t)\right)$ Additional computation

Polyak step size: $\eta_t = \dfrac{f(x_t) - f(x^\star)}{\|\nabla f(x_t)\|^2}$ Knowledge of $f(x^\star)$

(Norm) Adagrad: $\eta_t = \dfrac{\|x_0 - x^\star\|}{\sqrt{\sum_{i=0}^{t} \|\nabla f(x_t)\|^2}}$ Knowledge of $\|x_0 - x^\star\|$

Little effect on different number of local epochs

CIFAR-100 + ResNet-50



Different number of local data per client

Non-iidness	*Optimizer*	*Dataset / Model*	
$\mathrm{Dir}(\alpha \cdot \mathbf{p})$		CIFAR-10 ResNet-18	CIFAR-100 ResNet-50
$\alpha = 1$	SGD	**80.7**$_{\downarrow(0.0)}$	53.5$_{\downarrow(4.0)}$
	SGD (↓)	78.8$_{\downarrow(1.9)}$	53.6$_{\downarrow(3.9)}$
	SGDM	75.0$_{\downarrow(5.7)}$	53.9$_{\downarrow(3.6)}$
	SGDM (↓)	66.6$_{\downarrow(14.1)}$	53.1$_{\downarrow(4.4)}$
	Adam	79.9$_{\downarrow(0.8)}$	51.1$_{\downarrow(6.4)}$
	Adagrad	79.3$_{\downarrow(1.4)}$	44.5$_{\downarrow(13.0)}$
	SPS	64.4$_{\downarrow(16.3)}$	37.2$_{\downarrow(20.3)}$
	Δ-SGD	**80.4**$_{\downarrow(0.3)}$	**57.5**$_{\downarrow(0.0)}$

QST

Quantum State Tomography, Single Qubit Case

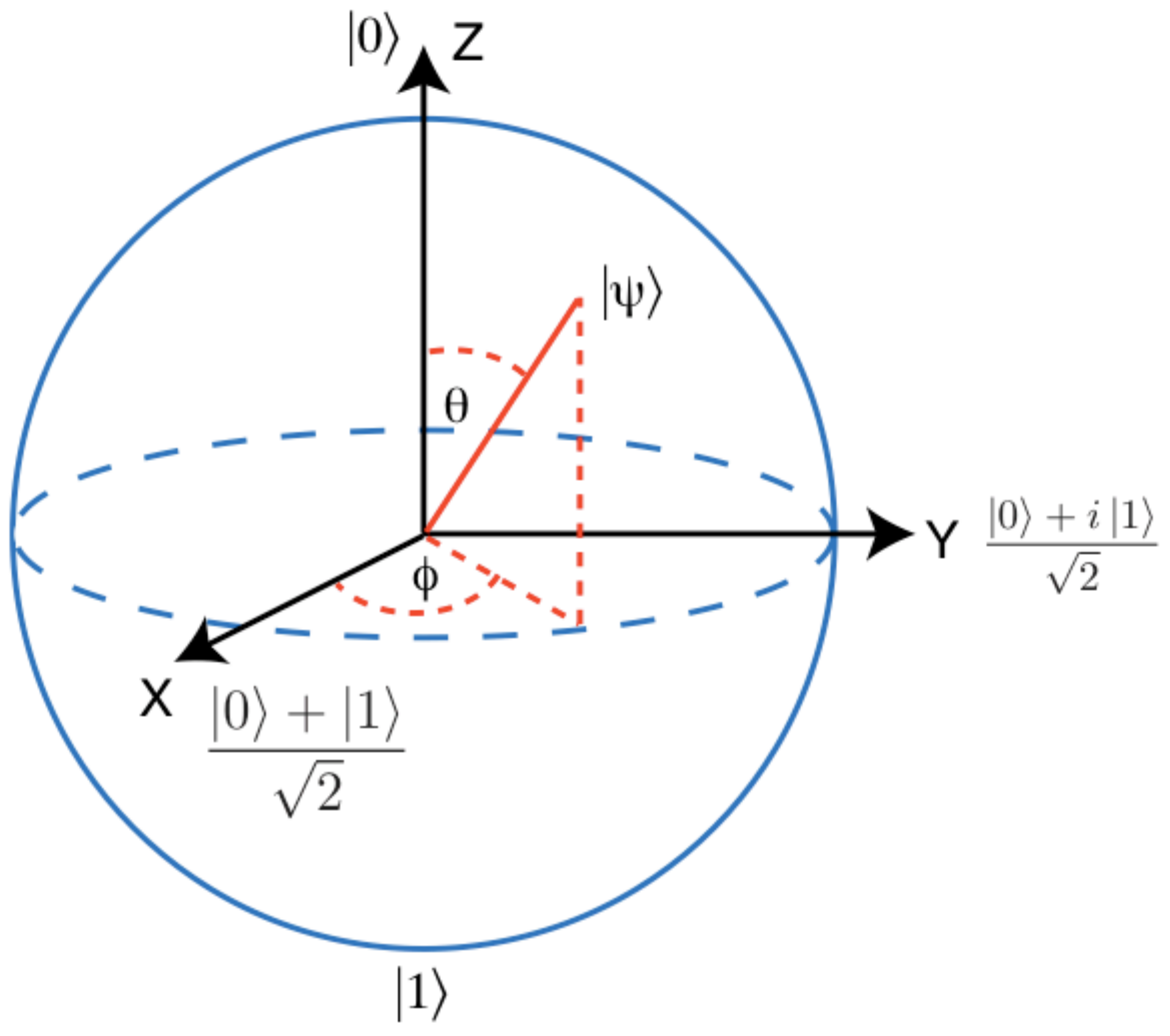
- Any single qubit state can be written as

"Pauli matrices"

$$\rho = \frac{1}{2}\left(I + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\right) \qquad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

where $r_\alpha = \mathrm{Tr}(\rho\sigma_\alpha)$, for $\alpha = x, y, z$

Expectation value of σ_α w.r.t ρ



- How do we "measure" $r_\alpha = \mathrm{Tr}(\rho\sigma_\alpha)$?
  - Prepare M number of copies of the state ρ

E.g. $M = 1000$

Measurement using σ_z, suppose we find the qubit in state $|0_z\rangle$ 400 times, and in state $|1_z\rangle$ 600 times.

  - Measure the projection of ρ onto eigenvectors of σ_α resulting in $\alpha_1, \alpha_2, \ldots, \alpha_M$
  - Approximation of $\mathrm{Tr}(\rho\sigma_\alpha)$ is given by $\frac{1}{M}\sum_{i=1}^{M}\alpha_i$

We can estimate

$\mathrm{Tr}\left(\rho\,|0_z\rangle\langle 0_z|\right) \approx \frac{400}{1000} := y_0^z$ and

$\mathrm{Tr}\left(\rho\,|1_z\rangle\langle 1_z|\right) \approx \frac{600}{1000} := y_1^z$

Quantum State Tomography, Single Qubit Case

- Once we have y_i^α for $i = \{0,1\}$ and $\alpha = \{x, y, z\}$, we can solve:

$$\underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} \quad f(\rho) := \sum_{\alpha=x,y,z} \sum_{i=0,1} \left(\text{Tr}(\rho A_i^\alpha) - y_i^\alpha\right)^2$$

$$\text{subject to} \quad \rho \succeq 0,\ \text{Tr}(\rho) = 1$$

$A_i^\alpha = |i_\alpha\rangle\langle i_\alpha|$ "rank-1 sensing matrix (outer product)"

- More generally we can solve:

$\left(\mathscr{A}(\rho)\right)_i = \text{Tr}(\rho A_i)$ where $A_i \in \mathbb{C}^{d\times d}$, $i = 1,\ldots,m$

$$\underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} \quad f(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \rho \succeq 0,\ \text{Tr}(\rho) = 1$$

measured data $y \in \mathbb{R}^m$

- How does it scale?
    - **Optimization**: the space of $\rho \in \mathbb{C}^{d\times d}$ grows exponentially (recall: $d = 2^n$)
    - **Amount of data**: from $\mathscr{A}(\rho) = y$, if we have access to $y_1, \ldots, y_m$ and $A_1, \ldots, A_m$ that form an orthonormal basis for $\mathbb{C}^{d\times d}$ (i.e. $m = d^2$), we can reconstruct ρ with linear inversion

For $n = 16$ qubits, $\rho \in \mathbb{C}^{d\times d}$ where $d = 65{,}536$

And we need $m = \mathcal{O}(2^{32}) \approx 4{,}294{,}96$ measurements

Convergence theory

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- ~~Amount of data: $O(d^2)$ without any~~

Theorem 3 (Accelerated convergence rate). *Assume that $\mathcal{A}$ satisfies the RIP with constant $\delta_{2r} \leq 1/10$. Let U_0 and U_{-1} be such that* $\boxed{\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F,\ \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F \leq \frac{\sqrt{\sigma_r(\rho^\star)}}{10^3\sqrt{\kappa\tau(\rho^\star)}}}$, *where* $\kappa := \frac{1+\delta_{2r}}{1-\delta_{2r}}$, $\tau(\rho) := \frac{\sigma_1(\rho)}{\sigma_r(\rho)}$ *for rank-r ρ, and $\sigma_i(\rho)$ is the ith singular value of ρ. Set step size η such that*

$$\boxed{\left[1 - \left(\frac{\sqrt{1+\delta_{2r}}-\sqrt{1-\delta_{2r}}}{(\sqrt{2}+1)\sqrt{1+\delta_{2r}}}\right)^4\right] \cdot \frac{10}{4\sigma_r(\rho^\star)(1-\delta_{2r})} \leq \eta \leq \frac{10}{4\sigma_r(\rho^\star)(1-\delta_{2r})},}$$

and the momentum parameter $\boxed{\mu = \frac{\varepsilon}{2\cdot 10^3 r\tau(\rho^\star)\sqrt{\kappa}},}$ *for user-defined $\varepsilon \in (0,1]$. For $y = \mathcal{A}(\rho^\star)$ where* $\texttt{rank}(\rho^\star) = r$, `MiFGD` *returns a solution such that*

$$\min_{R\in\mathcal{O}} \|U_{J+1} - U^\star R\|_F \leq \left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1} \left(\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F^2 + \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F^2\right)^{1/2}$$

$$\xi \cdot |\mu| \cdot \sigma_1(\rho^\star)^{1/2} \cdot r \cdot \left(1 - \left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1}\right)\left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{-1}$$

$$\lesssim \boxed{\left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1} \left(\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F^2 + \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F^2\right)^{1/2} + O(\mu),}$$

$(1 - 0.25)^6 \approx 0.1779$ VS. $\left(1 - \sqrt{0.25}\right)^6 \approx 0.0156$

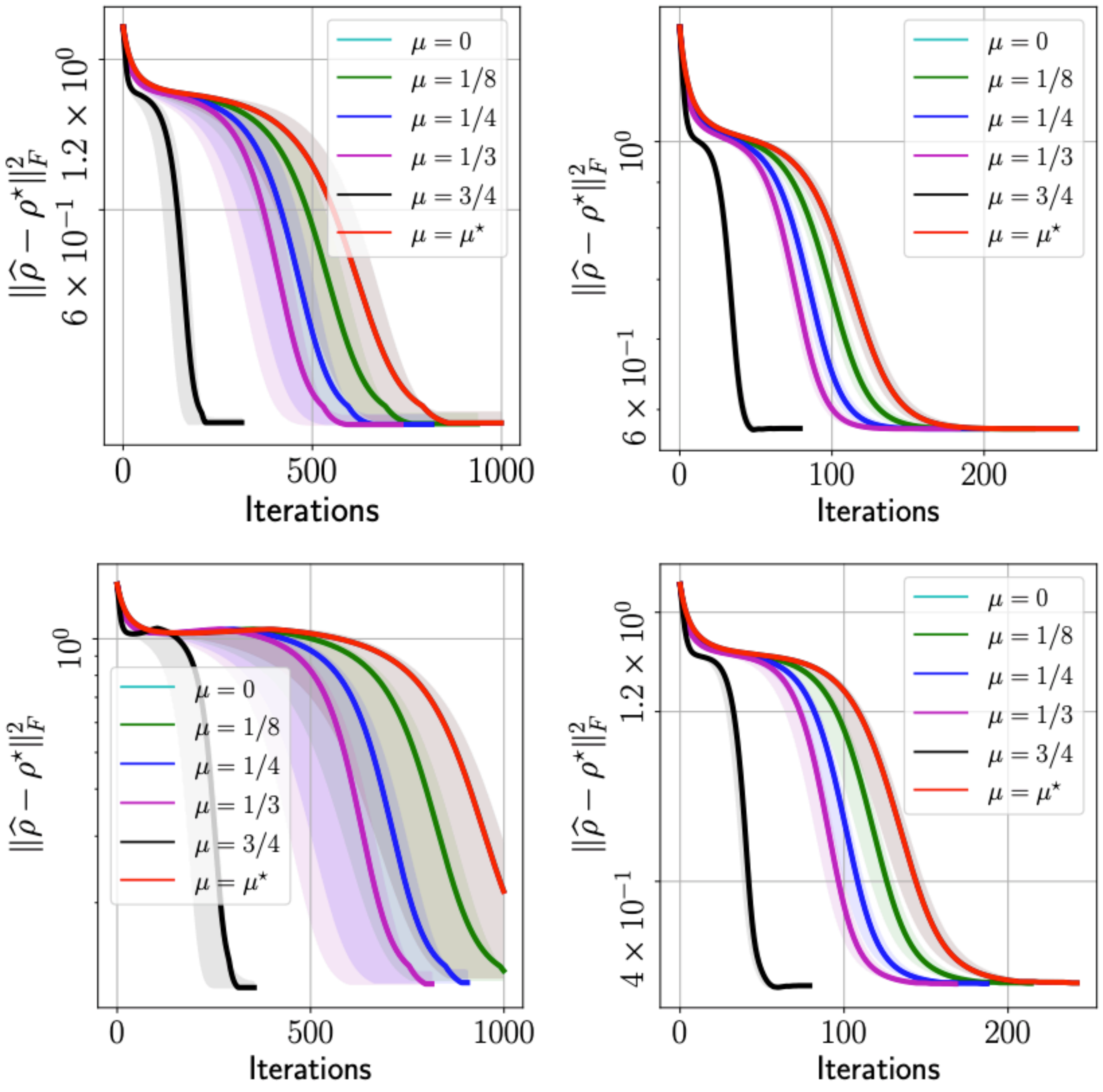
$\left(1 - \frac{1-\delta_{2r}}{1+\delta_{2r}}\right)^{J+1}$ VS.

where $\xi = \sqrt{1 - \frac{4\eta\sigma_r(\rho^\star)(1-\delta_{2r})}{10}}$. *That is, the algorithm has an accelerated linear convergence rate in iterate distances up to a constant proportional to the momentum parameter μ.*

MiFGD performance on real quantum data (IBM QPU)

GHZminus (6)

GHZminus (8)



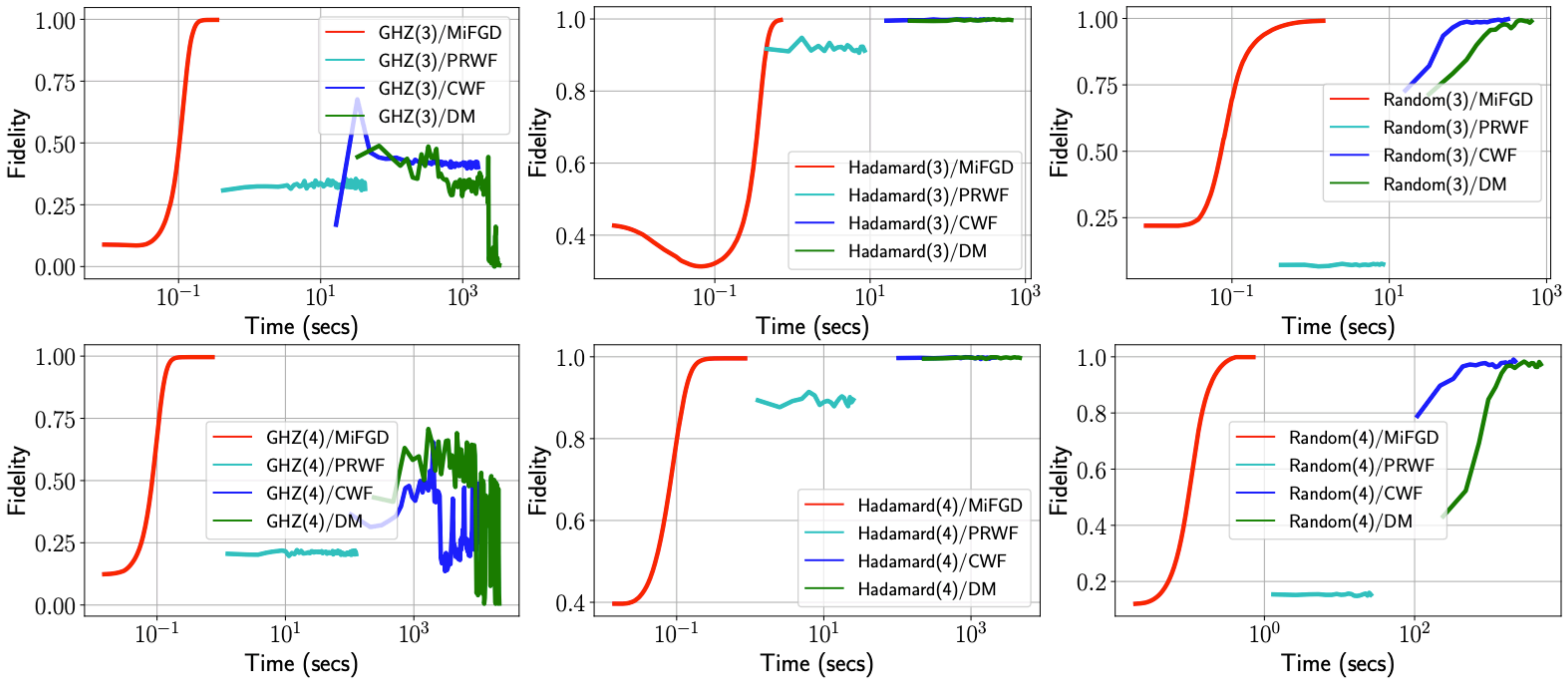
Hadamard (6)

Hadamard (8)

$[m = 20\% \cdot d^2]$

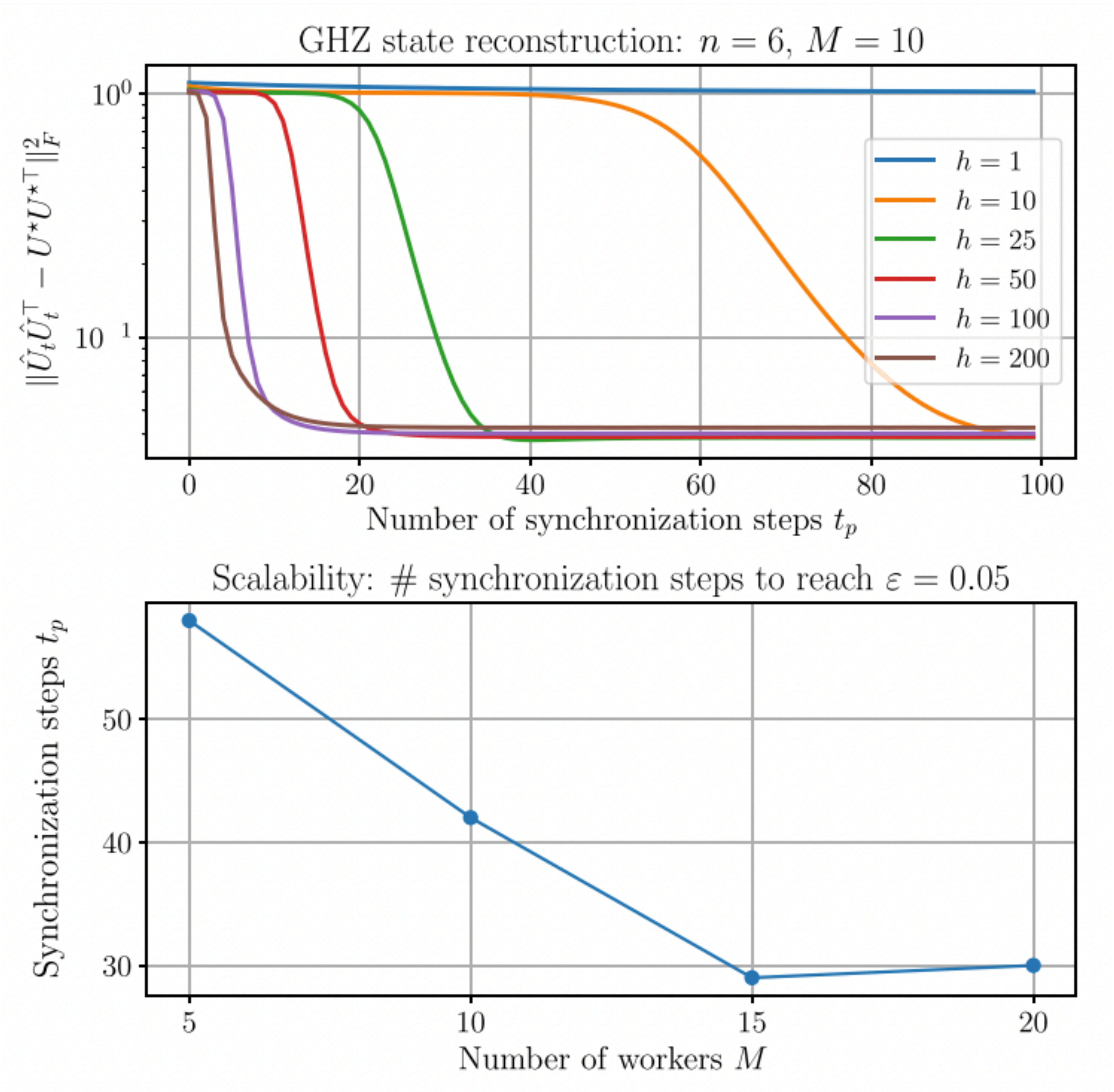
Comparison with SOTA: `Qucumber` NN methods

[Torlai et al., 2018]



$[m = 50\% \cdot d^2]$

Numerical Simulations



- Increasing number of local iterations lead to faster convergence in terms of the synchronization steps.
- Speed up gets marginal: there is not much difference between $h = 100$ and $h = 200$, indicating there is an "optimal" number of local iterations.
- Higher h leads to slightly worse final accuracy—consistent with Theorem 1

- Number of synchronization steps to reach $\varepsilon \leq 0.05$ while fixing $h = 20$. Each machine gets 200 measurements.
- Significant speed up from $M = 5$ to $M = 15$.

Function class and assumptions

Assumption 1: The function f_i is μ-restricted strongly convex and L-restricted smooth. That is, for all $X, Y \succeq 0$ and for all $i \in [M]$, it holds that:

$$f_i(Y) \geq f_i(X) + \langle \nabla f_i(X), Y - X \rangle + \frac{\mu}{2}\|X - Y\|_F^2 \quad \text{and} \qquad (1a)$$

$$\|\nabla f_i(X) - \nabla f_i(Y)\|_F \leq L\|X - Y\|_F \qquad (1b)$$

Assumption 2: The stochastic gradient ∇g_i^j is unbiased, has a bounded variance, and is bounded in expectation, for all $i \in [M]$. That is,

$$\mathbb{E}_j\left[\nabla g_i^j(U)\right] = \nabla g_i(U), \qquad (2a)$$

$$\mathbb{E}_j\left[\|\nabla g_i^j(U) - \nabla g_i(U)\|_F^2\right] \leq \sigma^2, \quad \text{and} \qquad (2b)$$

$$\mathbb{E}_j\left[\|\nabla g_i^j(U)\|_F^2\right] \leq G^2, \qquad (2c)$$

where j follows a uniform distribution.

Exact local sub-linear convergence

Theorem 2: Let Assumptions 1, 2, and the initialization condition of Lemma 1 hold. Moreover, let $\eta_t = \frac{2}{\alpha(t+2)}$ for $t \in [0:T]$ and $\max_p |t_p - t_{p+1}| \leq h$. Then, the output of Algorithm 1 has the following property:

$$\mathbb{E}\left[D^2(\hat{U}_{T+1}, U^\star)\right] \leq \frac{4C}{\alpha(T+3)},$$

where $X^\star$ is the optimum of f over the set of PSD matrices such that rank$(X^\star) = r$, $U^\star$ is such that $X^\star = U^\star U^{\star\top}$, and $\alpha = \frac{3\mu}{10}\sigma_r(X^\star)$ and $C = (h-1)^2(h+2)^2G^2 + \frac{\sigma^2}{M}$ are global constants.

- We can show the exact local convergence by using appropriately diminishing step sizes
- But the convergence rate reduces to a sub-linear rate.