Ringleader ASGD: The First Asynchronous SGD with Optimal Time Complexity under Data Heterogeneity

Artavazd Maranjyan

Mathematics and Applications Colloquium
KAUST, 28 October 2025

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

AI = Optimization

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ f(x) := \mathbb{E}_{\xi \sim \mathcal{D}} \left[f(x; \xi) \right] \right\}$$

Model parameters we want to find

The distribution of the training dataset

$$f(x; \xi) := \text{Loss}\left(\text{Model}\left(x; \xi^{\text{input}}\right), \xi^{\text{label}}\right)$$

Distributed Learning

$\mathcal{D}_1$

$f_1(x) := \mathbb{E}_{\xi_1 \sim \mathcal{D}_1}\left[f_1(x;\xi_1)\right]$

$\mathcal{D}_2$

$f_2(x) := \mathbb{E}_{\xi_2 \sim \mathcal{D}_2}\left[f_2(x;\xi_2)\right]$

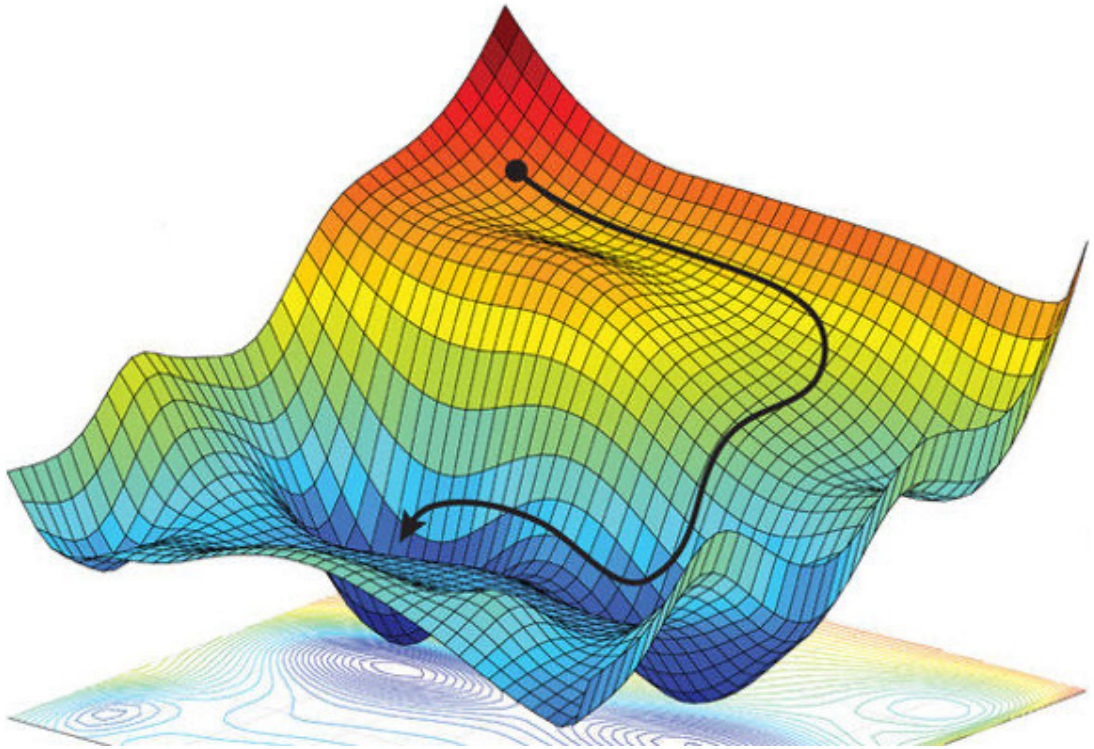
$\mathcal{D}_3$

$f_3(x) := \mathbb{E}_{\xi_3 \sim \mathcal{D}_3}\left[f_3(x;\xi_3)\right]$

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

Server

A common method in ML is Stochastic Gradient Descent (SGD)



Stepsize / Learning rate

$$x^{k+1} = x^k - \gamma g(x^k)$$

Unbiased gradient estimator, i.e.

$$\mathbb{E}\left[g(x^k)\right] = f(x^k)$$

$$g(x^k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k; \xi_i^k)$$

How to parallelize SGD in heterogeneous systems?

$\nabla f_1(x;\xi_1)$

Compute time = τ_1

τ_1

$\nabla f_2(x;\xi_2)$

Compute time = τ_2

τ_2

$\nabla f_3(x;\xi_3)$

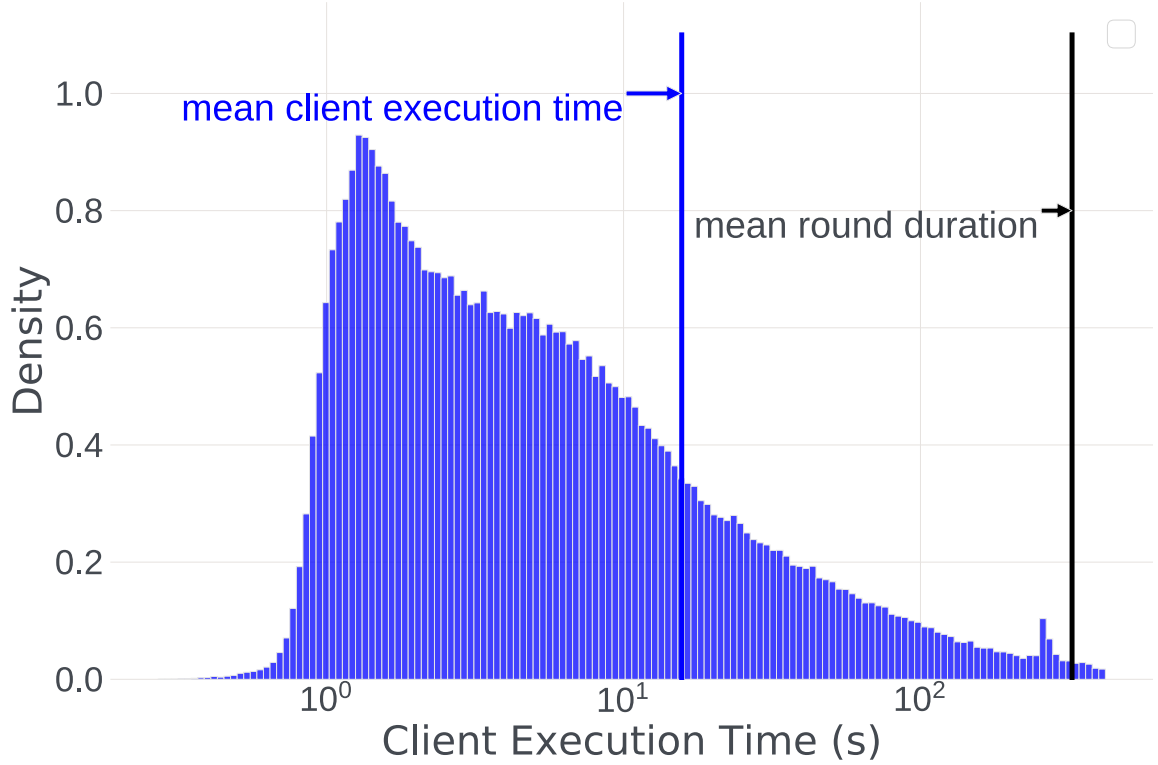
Compute time = τ_3

τ_3

Server

Client Compute Heterogeneity is Real

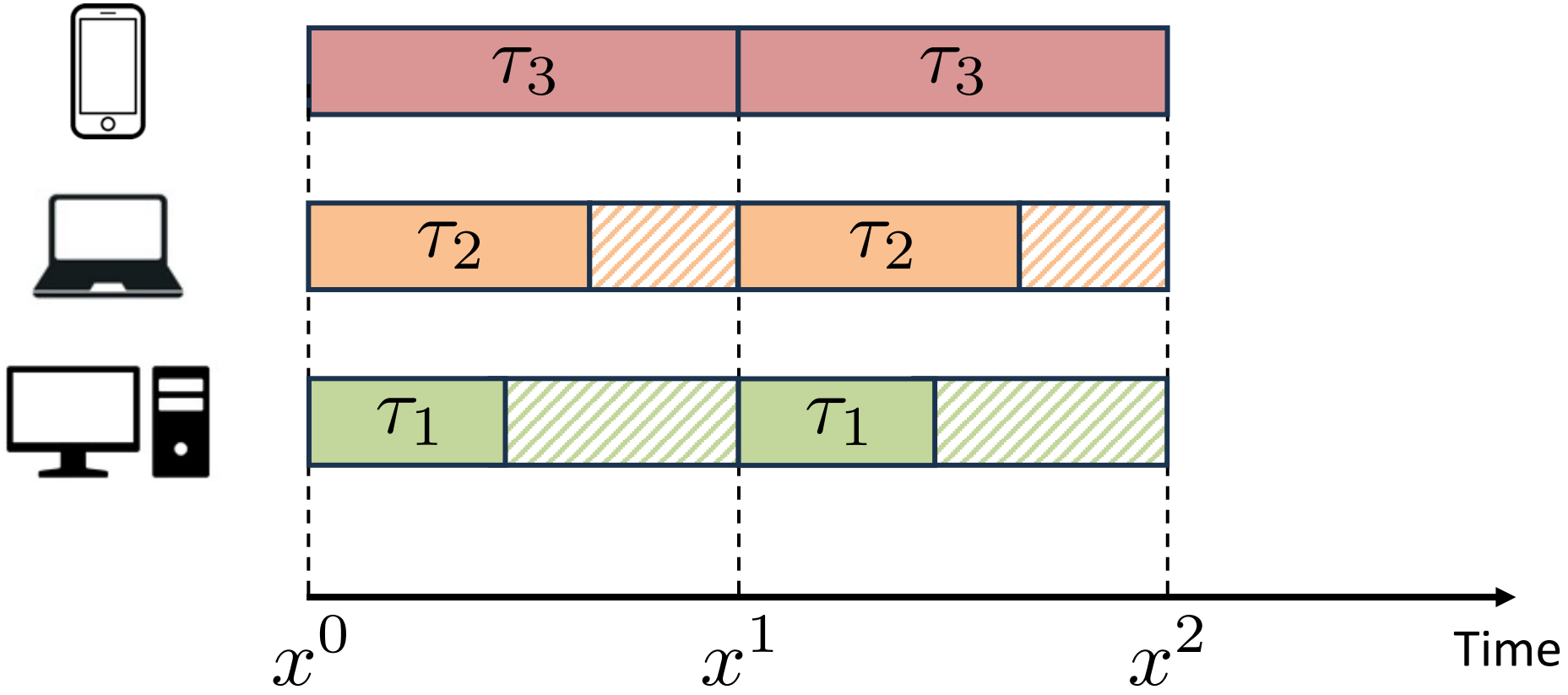
An LSTM-based language model
on nearly 100 million Android phones.



Papaya: Practical, Private, and Scalable Federated Learning
Huba et al. (2022)

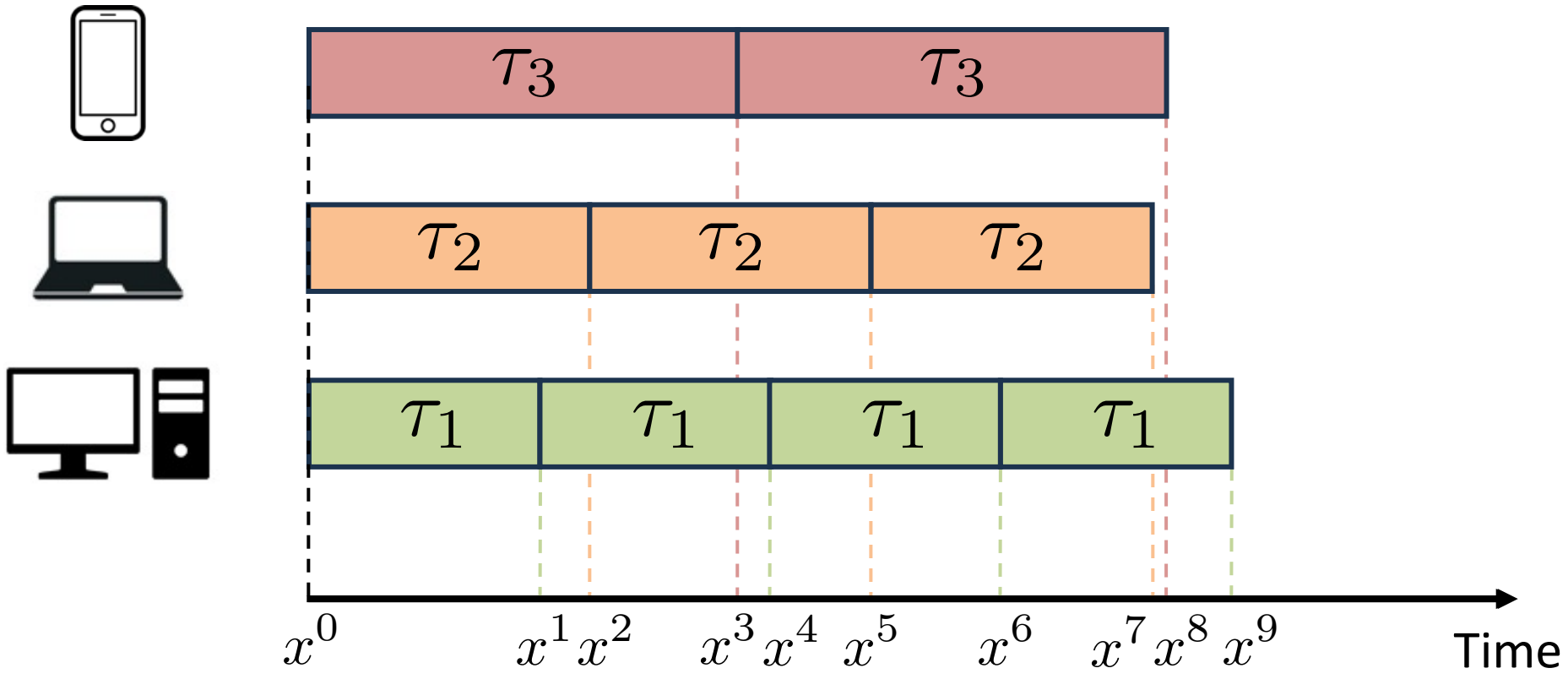
Minibatch SGD:
Each worker does one job only

$n = 3$



$$x^{k+1} = x^k - \gamma \frac{1}{n} \sum_{i=1}^{n} \nabla f_i\left(x^k; \xi_i^k\right)$$

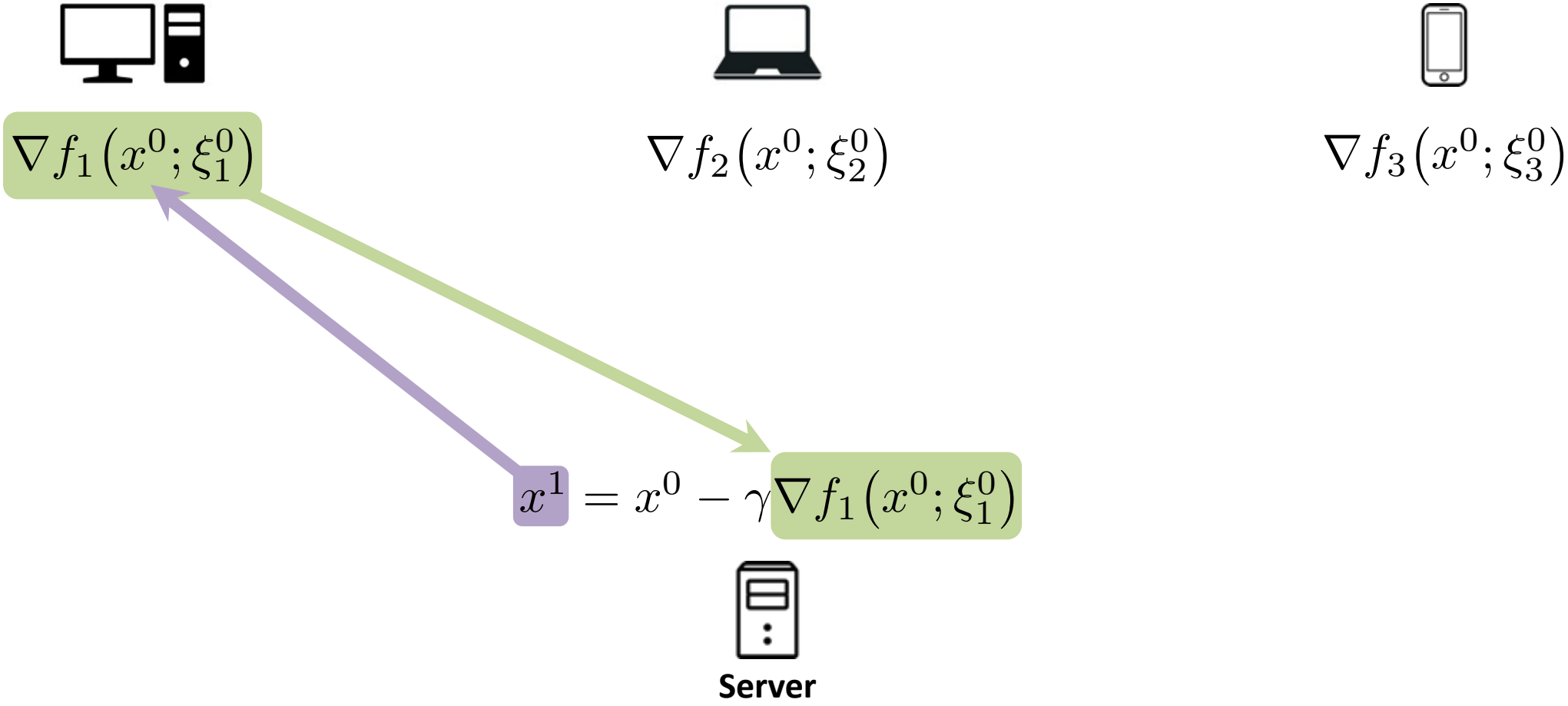
Asynchronous SGD
Remove the synchronization



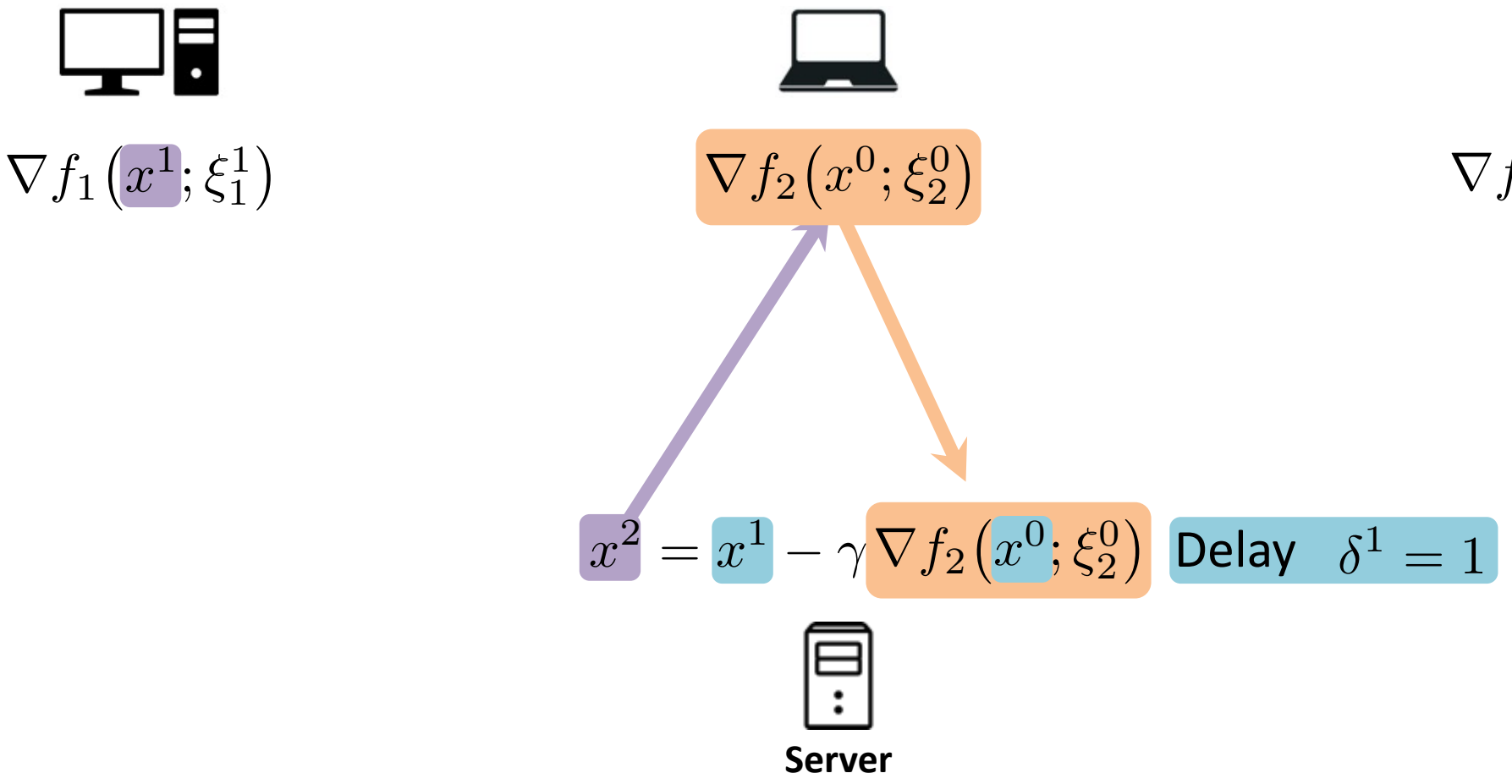
The first Asynchronous SGD

B Recht, C Re, S Wright, F Niu (2011)
HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent

Updates of Asynchronous SGD has delayed stochastic gradients



Updates of Asynchronous SGD has delayed stochastic gradients



Updates of Asynchronous SGD has delayed stochastic gradients

$\nabla f_1\left(x^1;\xi_1^1\right)$

$\nabla f_2\left(x^2;\xi_2^2\right)$

$\nabla f_3\left(x^0;\xi_3^0\right)$

$x^3 = x^2 - \gamma \nabla f_3\left(x^0;\xi_3^0\right)$ Delay $\delta^2 = 2$

Server

Updates of Asynchronous SGD has delayed stochastic gradients

Delay of worker i^k at iteration k

$$x^{k+1} = x^k - \gamma \nabla f_{i^k}\left(x^{k-\delta_{i^k}^k}; \xi^{k-\delta_{i^k}^k}\right)$$

Uses a gradient from a single worker

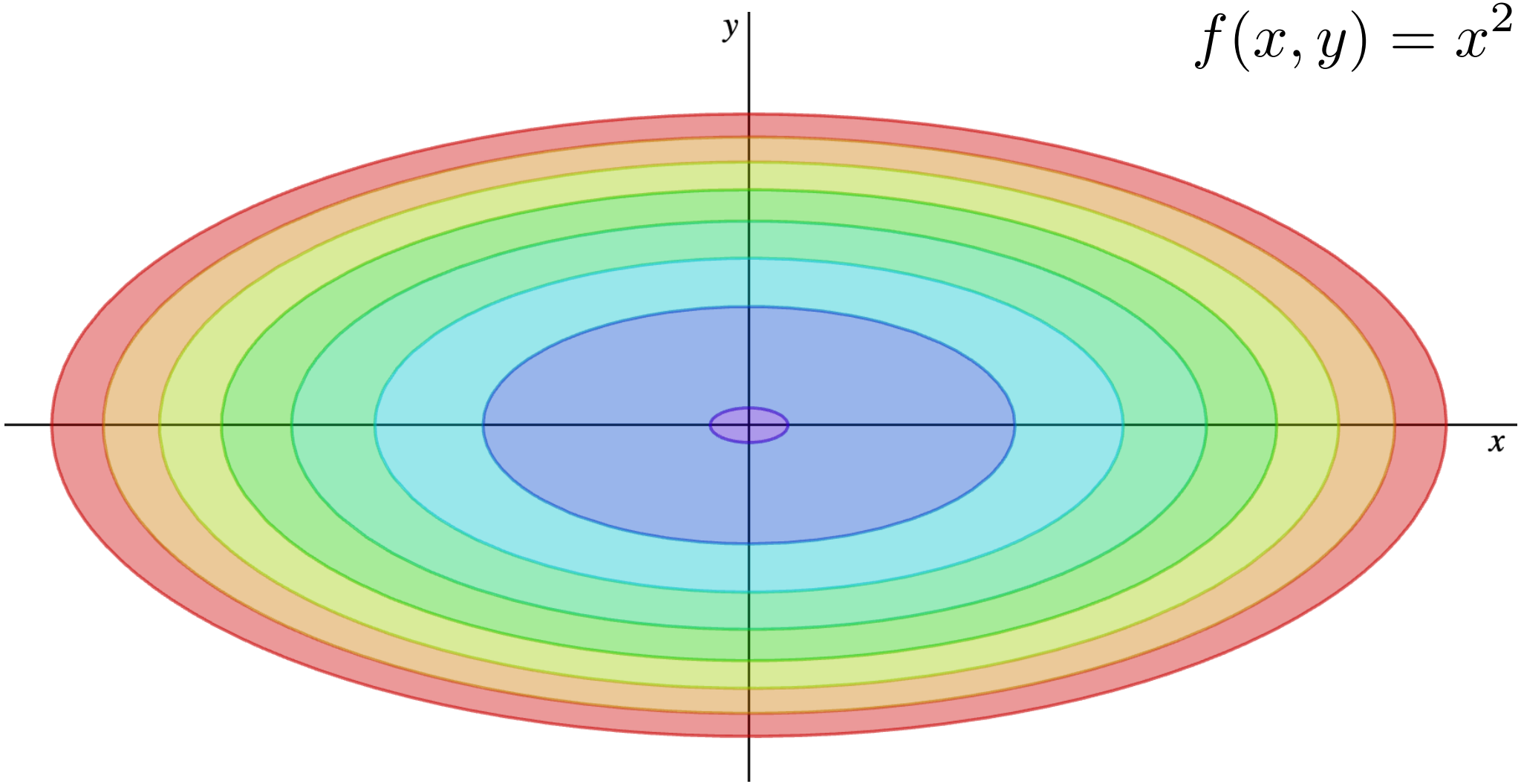
Similarity assumption

$$\|\nabla f_i(x) - \nabla f(x)\| \leq \zeta$$

Server

Asynchronous SGD can get wild: delays can degrade performance

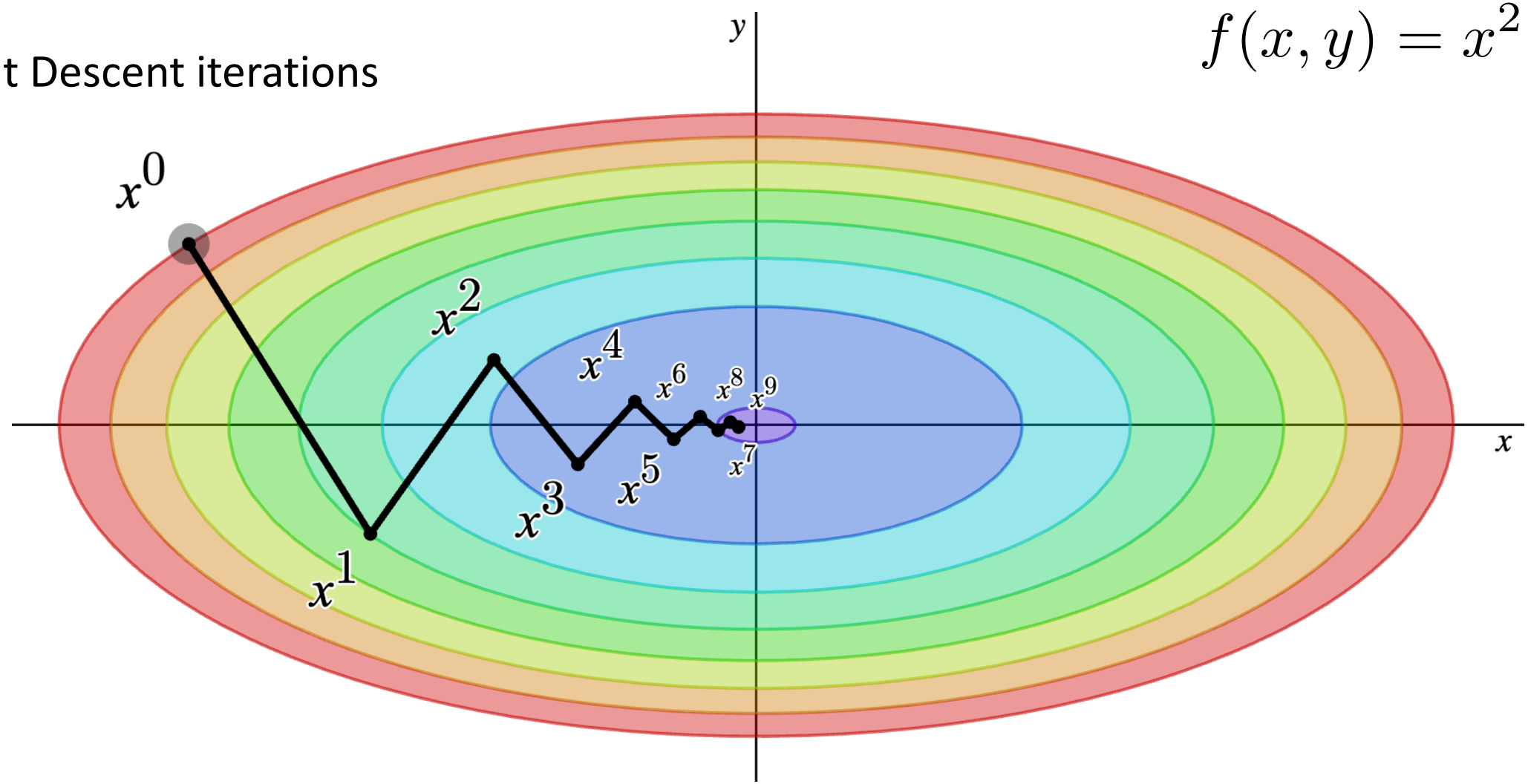
$$f(x, y) = x^2 + 5y^2$$



Asynchronous SGD can get wild: delays can degrade performance

Gradient Descent iterations

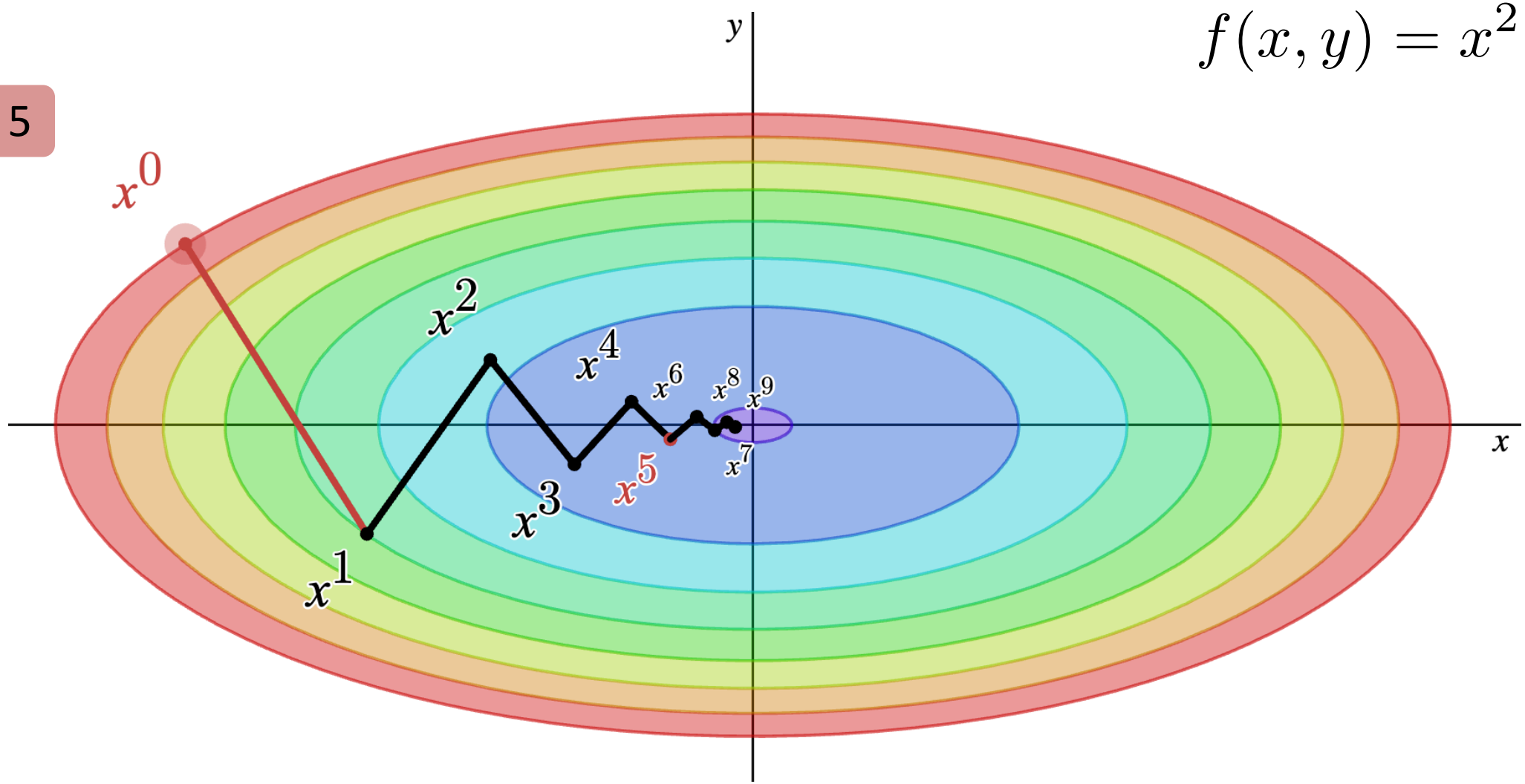
$$f(x, y) = x^2 + 5y^2$$



Asynchronous SGD can get wild: delays can degrade performance

$$f(x, y) = x^2 + 5y^2$$

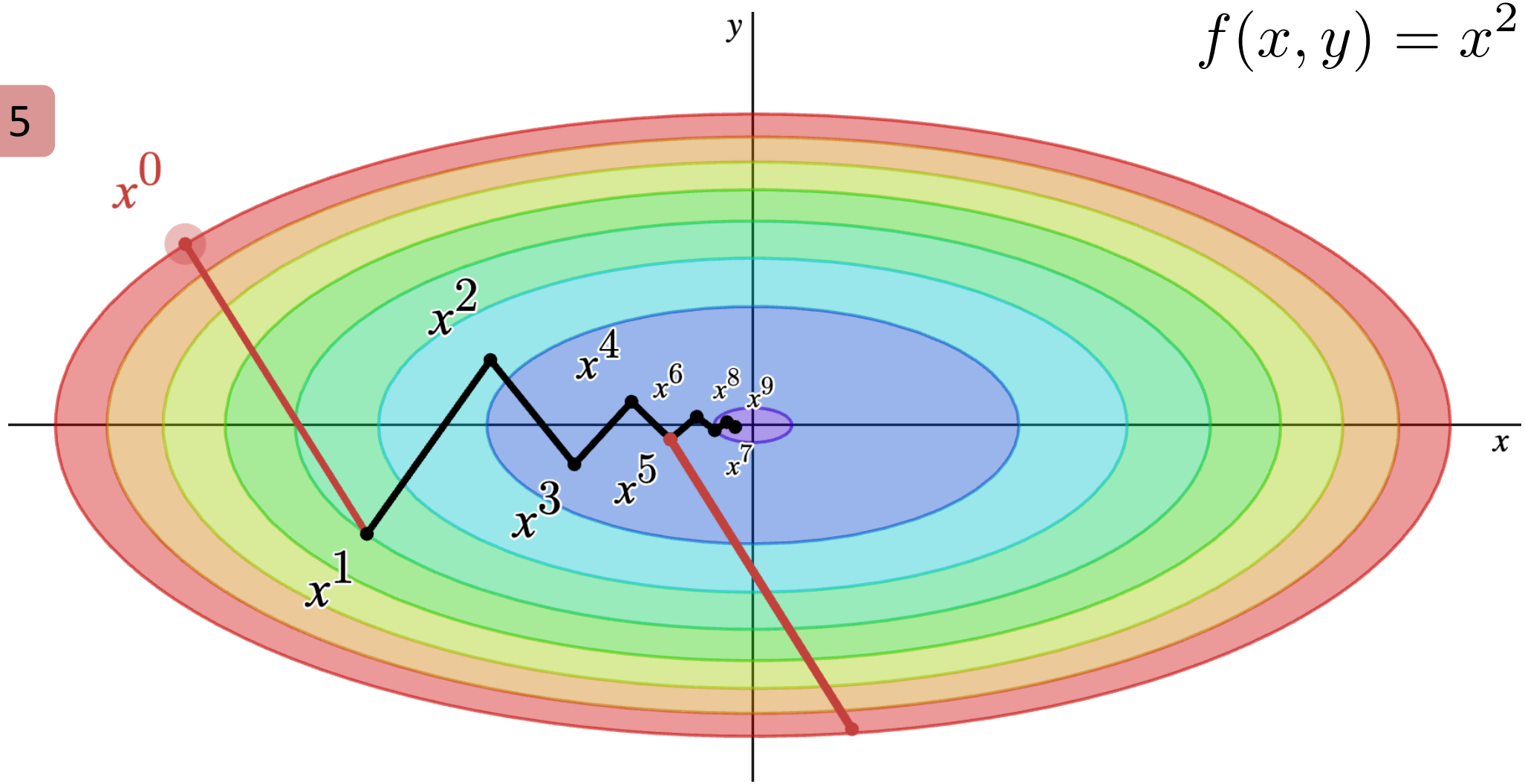
Delay = 5



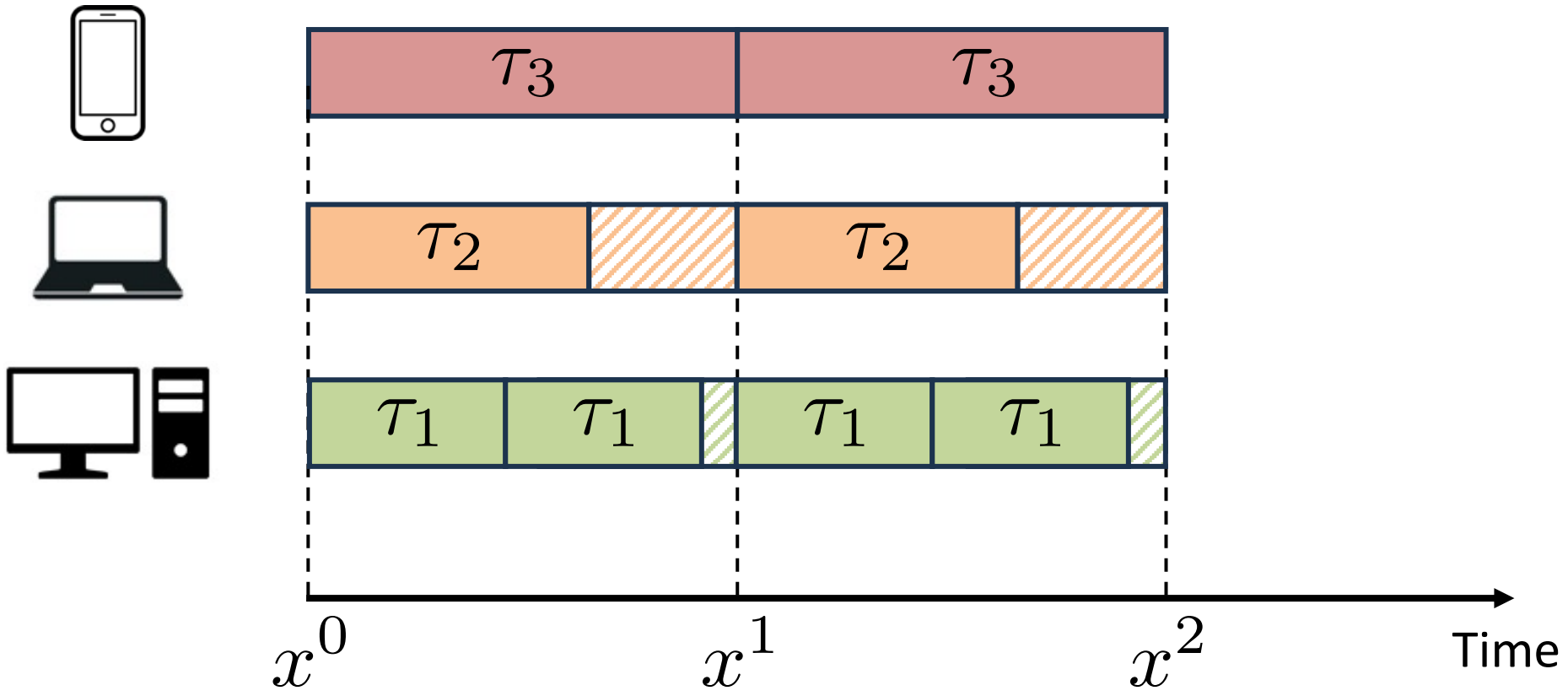
Asynchronous SGD can get wild: delays can degrade performance

$$f(x, y) = x^2 + 5y^2$$

Delay = 5



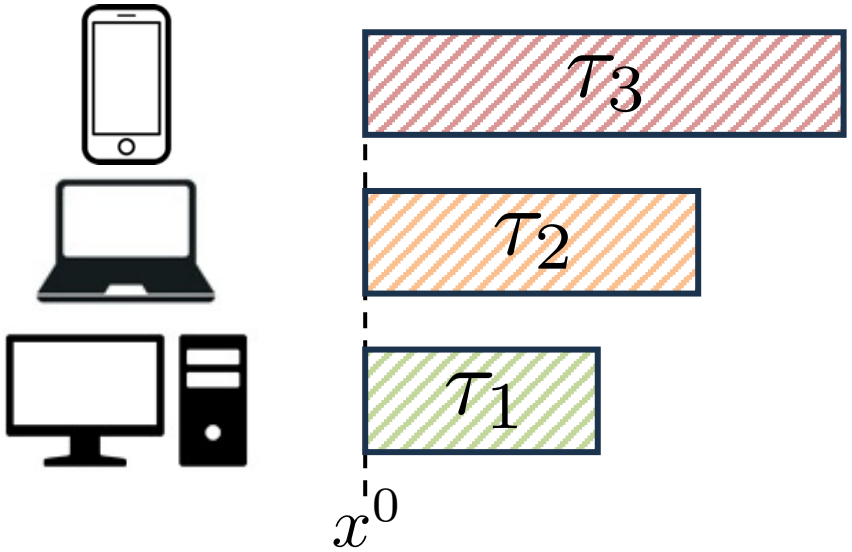
Back to Synchrony: Malenia SGD



$$x^{k+1} = x^k - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{1}{b_i^k} \sum_{j=1}^{b_i^k} \nabla f_i\left(x^k; \xi_i^{k,j}\right)$$

Tyurin and Richtárik (2024)

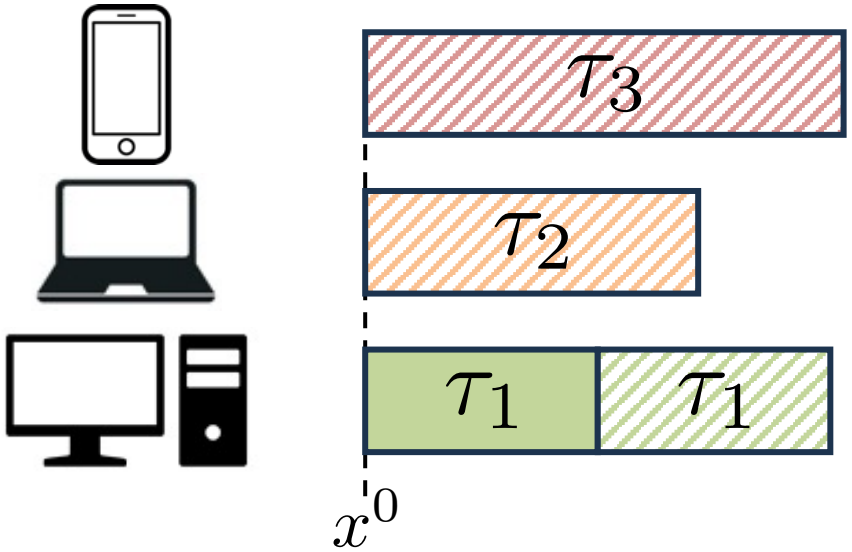
Malenia SGD with Gradient Table



$$b \qquad 0 \quad 0 \quad 0$$

$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix}$$

Malenia SGD with Gradient Table

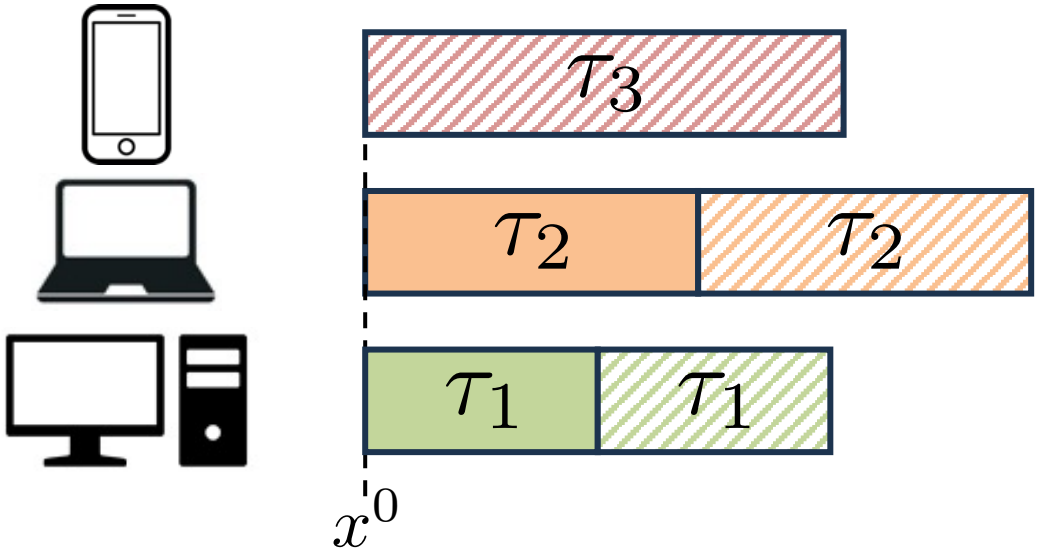


$$b \qquad 1 \quad 0 \quad 0$$

$$\begin{bmatrix} & & \\ G_1 & & \\ & & \end{bmatrix}$$

$$G_1 = \nabla f_1\left(x^0; \xi_1^{0,1}\right)$$

Malenia SGD with Gradient Table

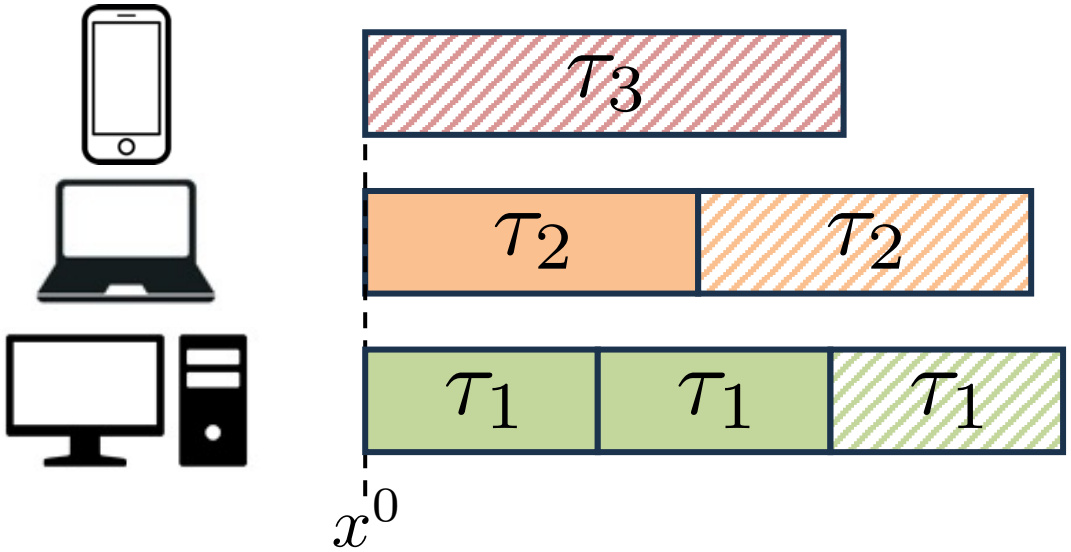


$$b \qquad 1 \quad 1 \quad 0$$

$$\begin{bmatrix} & & \\ G_1 & G_2 & \\ & & \end{bmatrix}$$

$$G_2 = \nabla f_2\left(x^0; \xi_2^{0,1}\right)$$

Malenia SGD with Gradient Table

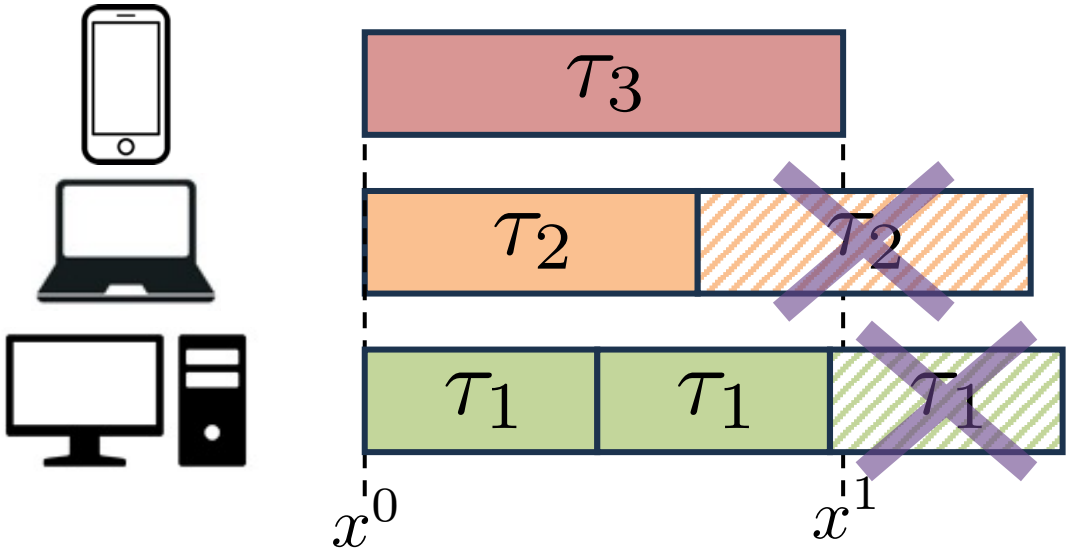


$$b \qquad 2 \qquad 1 \qquad 0$$

$$\begin{bmatrix} & | & | & \\ & G_1 & G_2 & \\ & | & | & \end{bmatrix}$$

$$G_1 = \sum_{j=1}^{2} \nabla f_1\left(x^0; \xi_1^{0,j}\right)$$

Malenia SGD with Gradient Table



Discard!

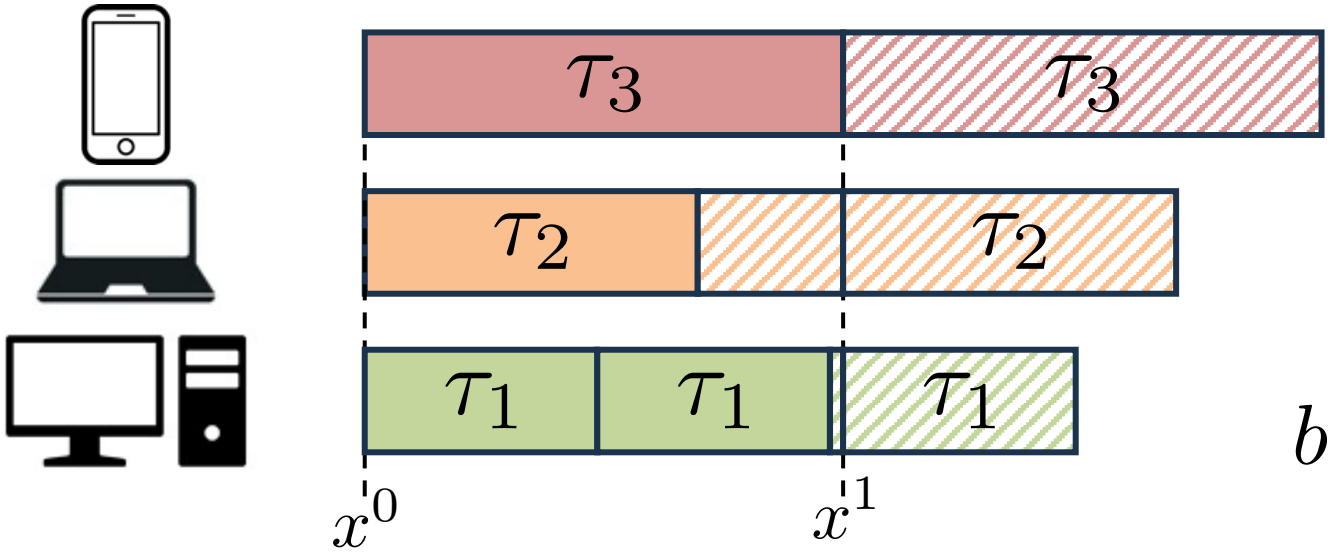
$$x^1 = x^0 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

$$b \quad 2 \quad 1 \quad 1$$

$$\begin{bmatrix} & & \\ G_1 & G_2 & G_3 \\ & & \end{bmatrix}$$

$$G_3 = \nabla f_3\left(x^0; \xi_3^{0,1}\right)$$

Malenia SGD with Gradient Table

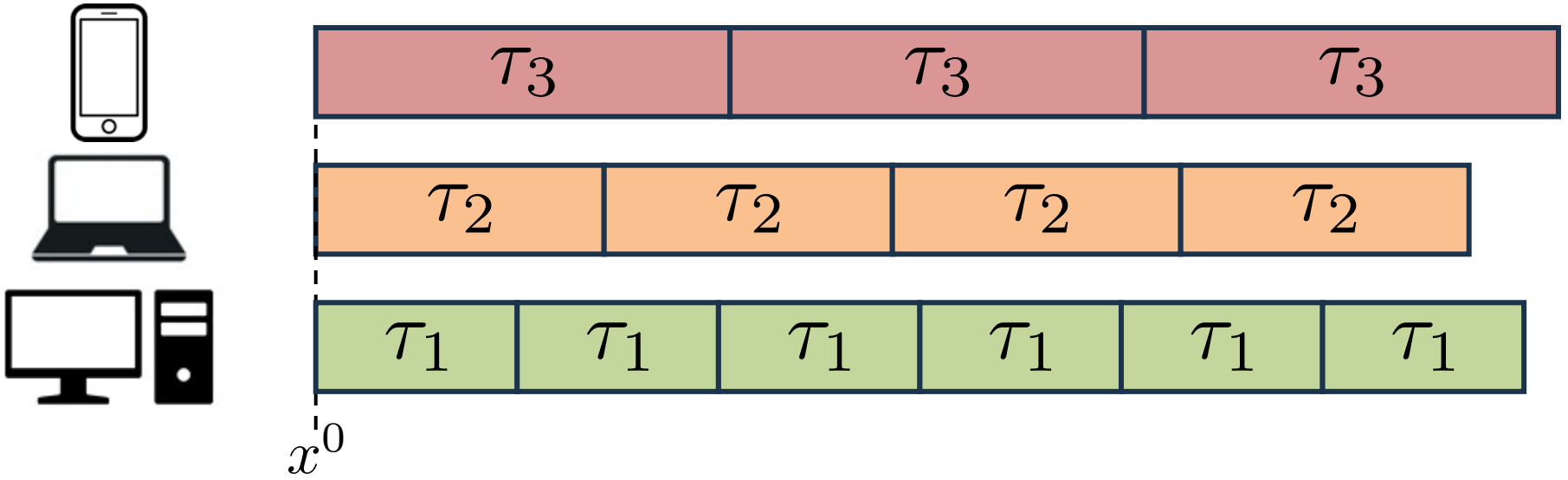


Synchronize and Repeat

$$b \qquad 0 \qquad 0 \qquad 0$$

$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix}$$

Can we achieve optimal time complexity with asynchronous methods?



Yes — Ringleader ASGD can:
Make It Less Wild

Yes — Ringleader ASGD can: Make It Less Wild

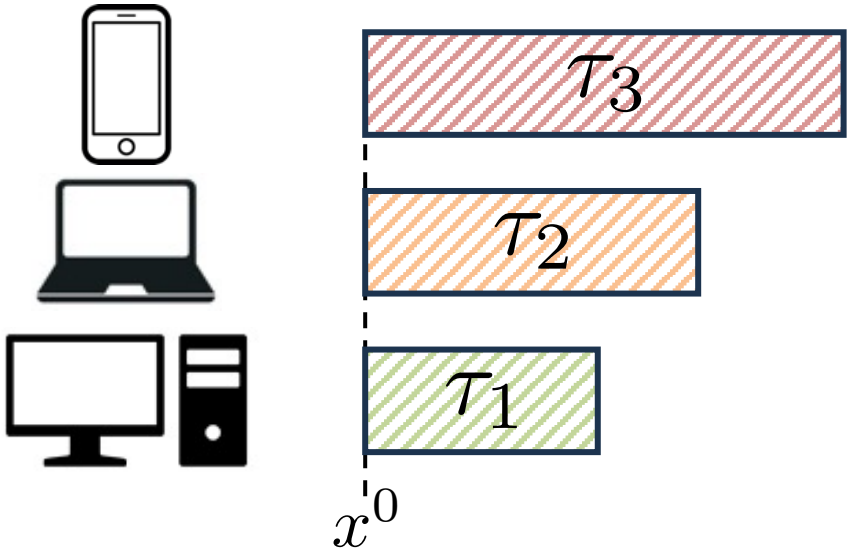
Phase 1 (collect)

Wait to receive at least one gradient from each worker

Phase 2 (update)

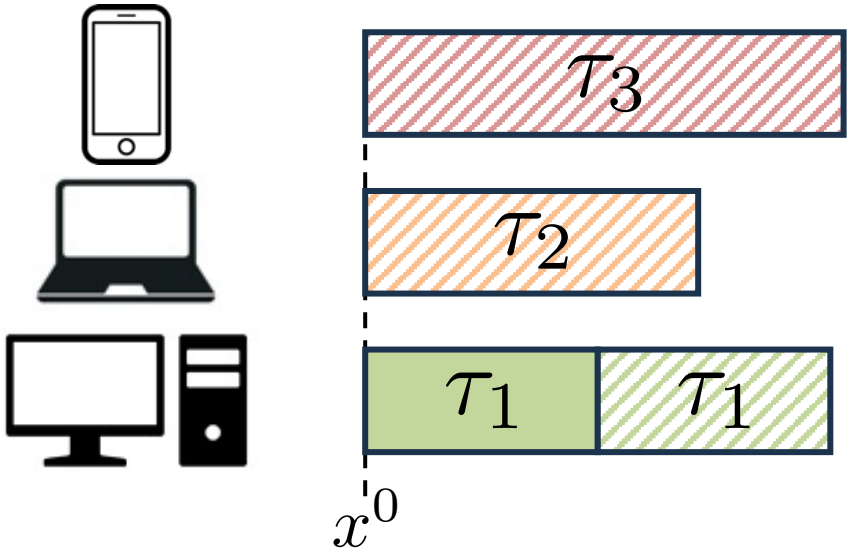
Perform a single update per worker

Ringleader ASGD: Phase 1 (collect)



$$\begin{bmatrix} b & 0 & 0 & 0 \\ & & & \end{bmatrix}$$

Ringleader ASGD: Phase 1 (collect)

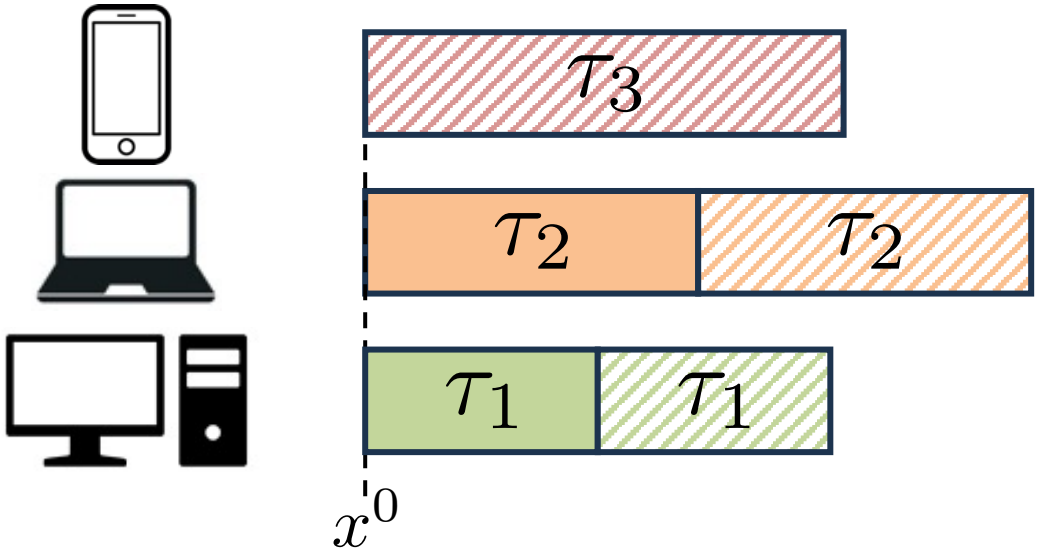


$$b \qquad 1 \quad 0 \quad 0$$

$$\begin{bmatrix} & & \\ & G_1 & \\ & & \end{bmatrix}$$

$$G_1 = \nabla f_1\left(x^0; \xi_1^{0,1}\right)$$

Ringleader ASGD: Phase 1 (collect)

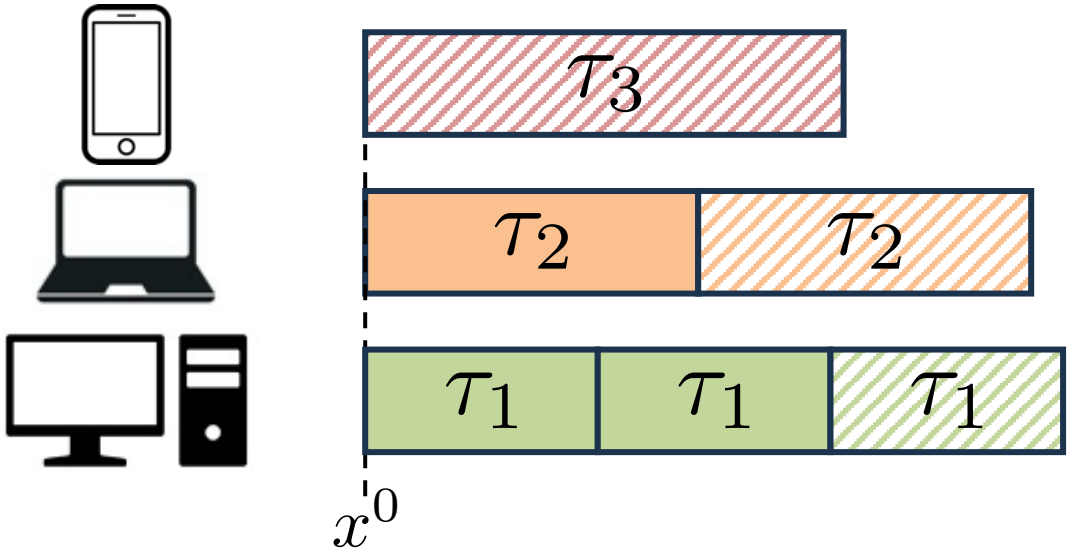


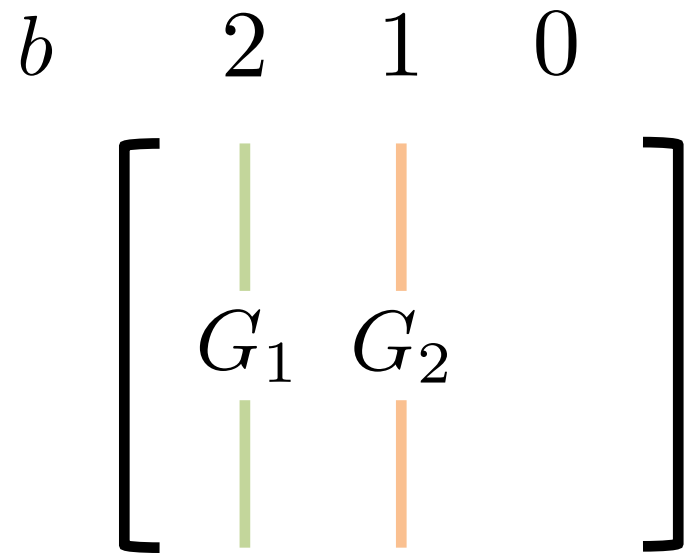
$$b \qquad 1 \quad 1 \quad 0$$

$$\begin{bmatrix} & & \\ G_1 & G_2 & \\ & & \end{bmatrix}$$

$$G_2 = \nabla f_2\left(x^0; \xi_2^{0,1}\right)$$

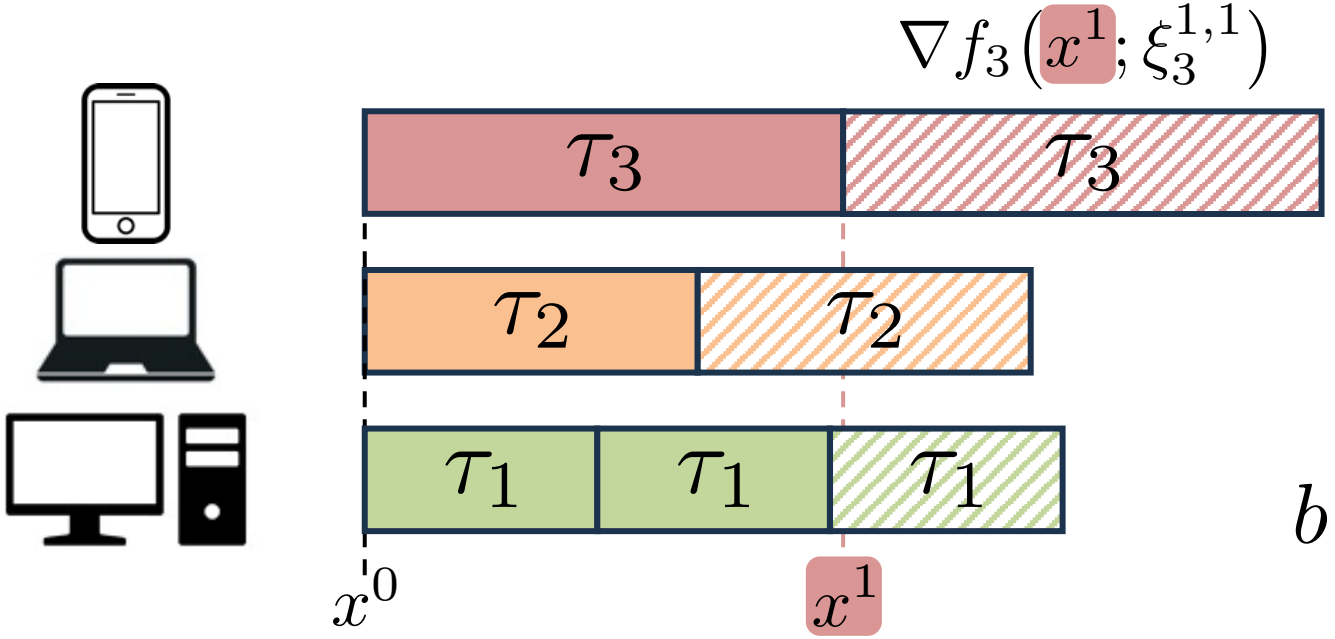
Ringleader ASGD: Phase 1 (collect)





$$G_1 = \sum_{j=1}^{2} \nabla f_1\left(x^0; \xi_1^{0,j}\right)$$

Ringleader ASGD: Phase 2 (update)



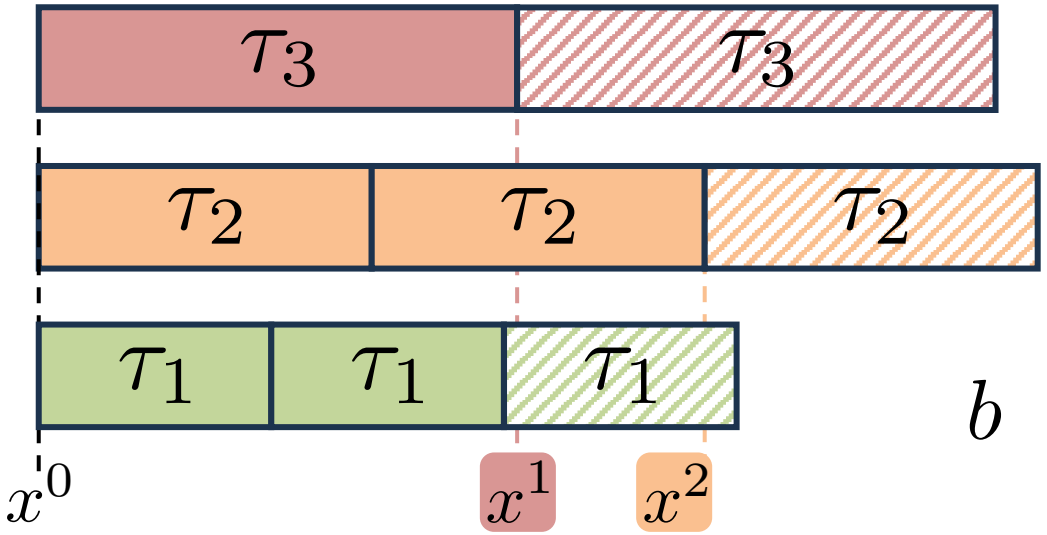
$$x^1 = x^0 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

$$b \quad 2 \quad 1 \quad 1$$

$$\begin{bmatrix} & & \\ G_1 & G_2 & G_3 \\ & & \end{bmatrix}$$

$$G_3 = \nabla f_3\left(x^0; \xi_3^{0,1}\right)$$

Ringleader ASGD: Phase 2 (update)



$$\nabla f_2\left(x^2; \xi_2^{2,1}\right)$$

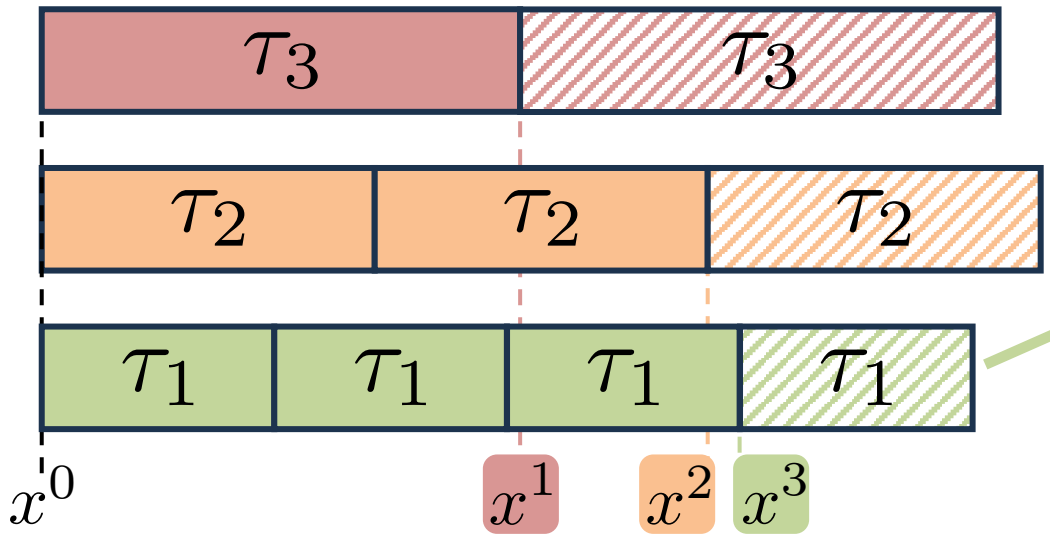
$$b \quad 2 \quad 2 \quad 1$$

$$\begin{bmatrix} & & \\ G_1 & G_2 & G_3 \\ & & \end{bmatrix}$$

$$x^2 = x^1 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

$$G_2 = \sum_{j=1}^{2} \nabla f_2\left(x^0; \xi_2^{0,j}\right)$$

Ringleader ASGD: Phase 2 (update)



$$\nabla f_1\left(x^3; \xi_1^{3,1}\right)$$

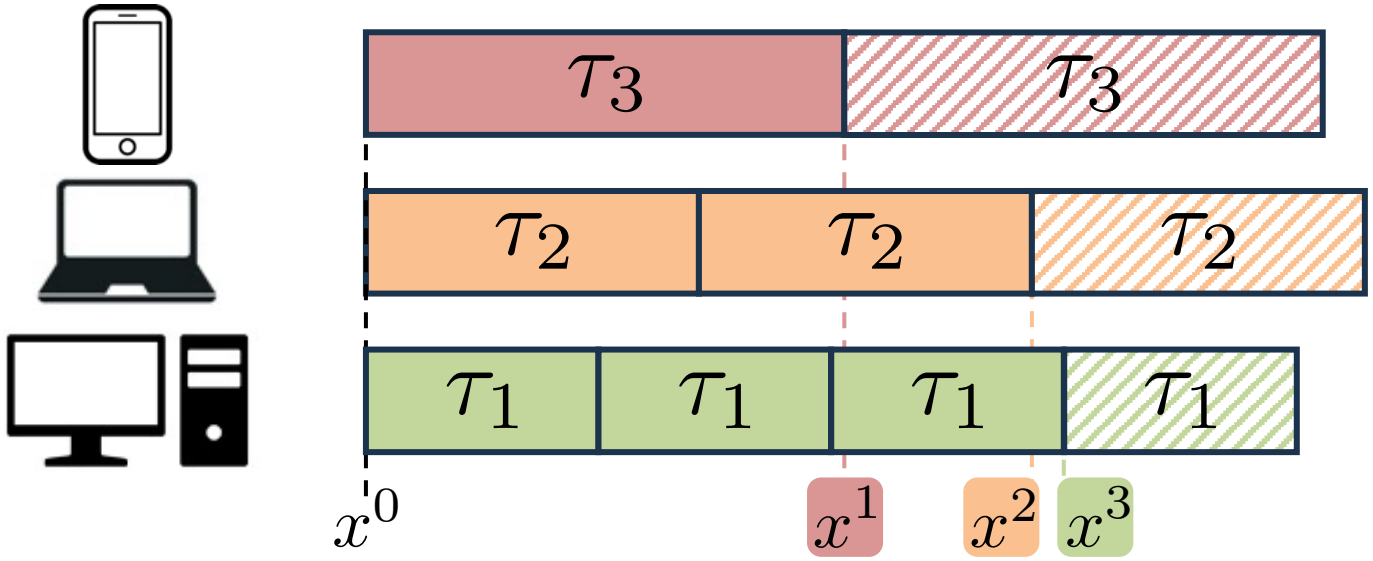
$$3 \quad 2 \quad 1 \quad b$$

$$\begin{bmatrix} & & & \\ G_1 & G_2 & G_3 & \\ & & & \end{bmatrix}$$

$$x^3 = x^2 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

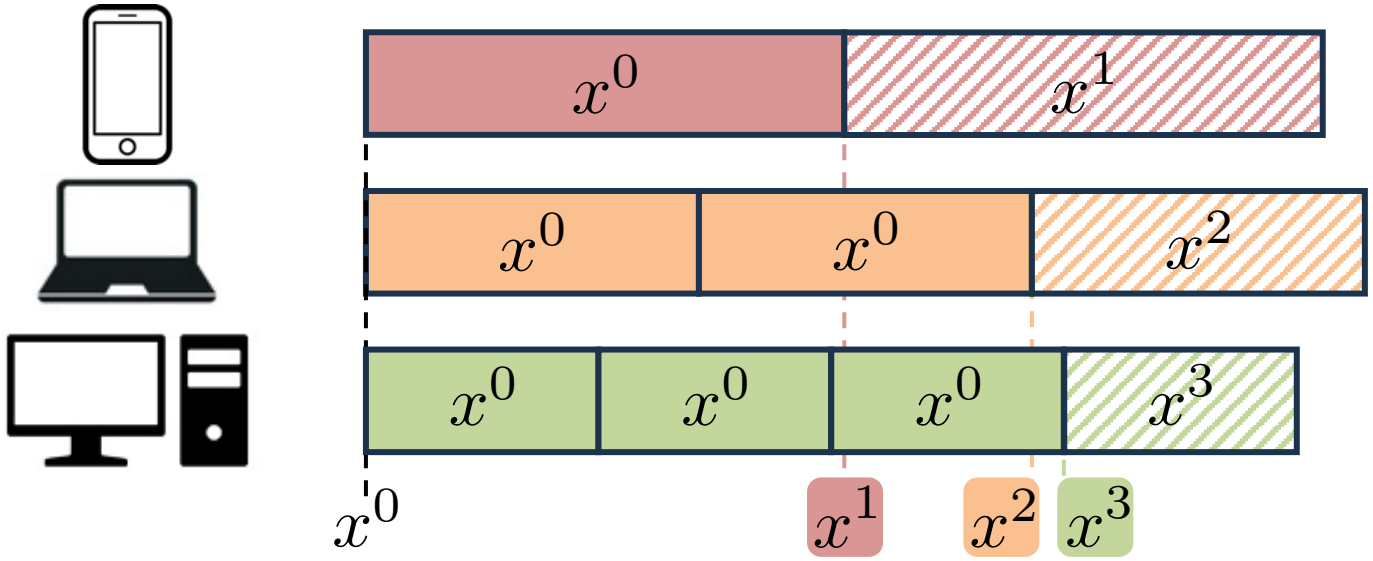
$$G_1 = \sum_{j=1}^{3} \nabla f_1\left(x^0; \xi_1^{0,j}\right)$$

Ringleader ASGD: Phase 1 (collect)



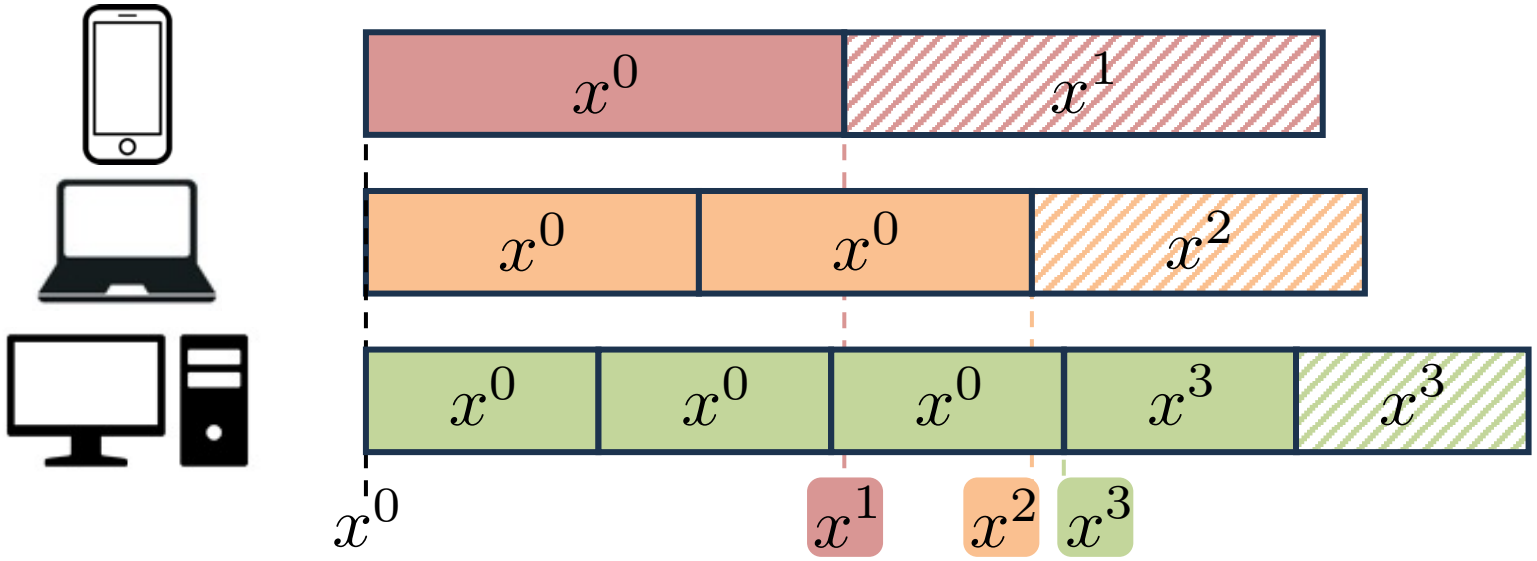
$$\begin{bmatrix} 0 & 0 & 0 & b \\ & & & \end{bmatrix}$$

Ringleader ASGD: Phase 1 (collect)



$$\begin{bmatrix} 0 & 0 & 0 & b \end{bmatrix}$$

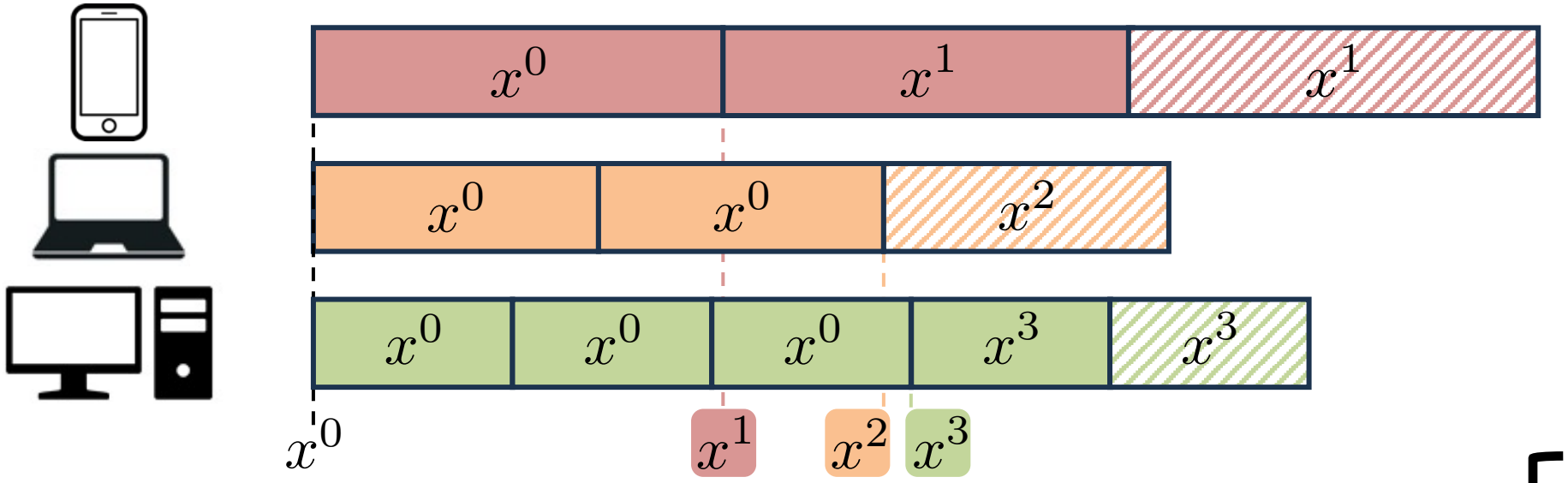
Ringleader ASGD: Phase 1 (collect)

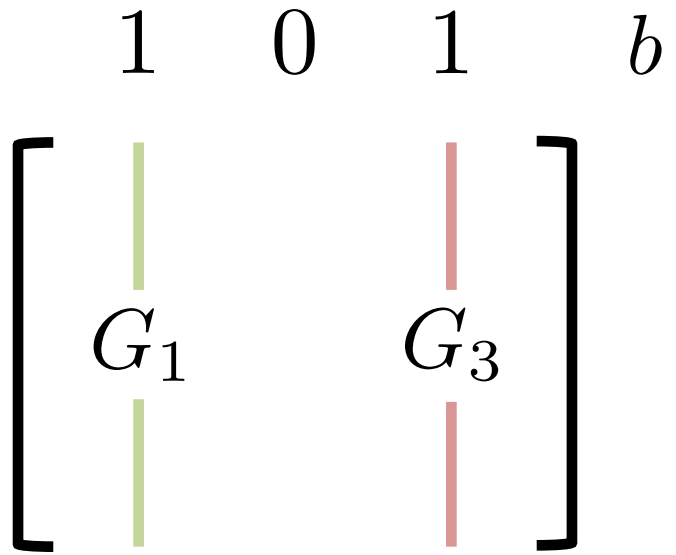


$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & b \\ & & & \\ & G_1 & & \\ & & & \end{array}\right]$$

$$G_1 = \nabla f_1\left(x^3; \xi_1^{3,1}\right)$$

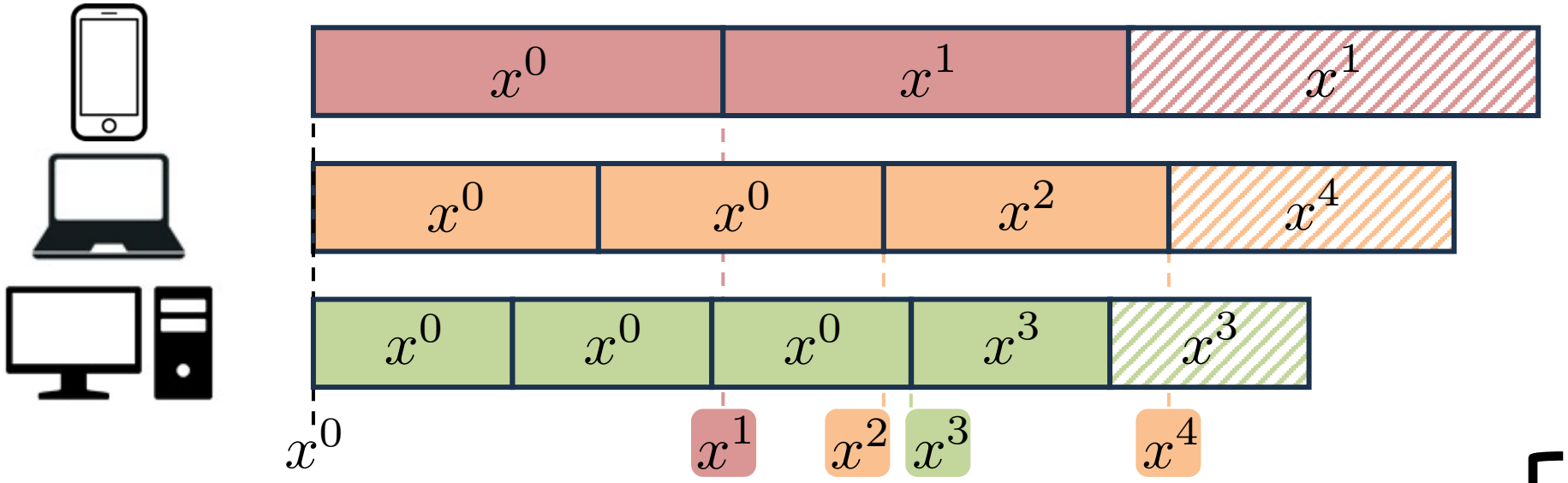
Ringleader ASGD: Phase 1 (collect)





$$G_3 = \nabla f_3\left(x^1; \xi_3^{1,1}\right)$$

Ringleader ASGD: Phase 2 (update)



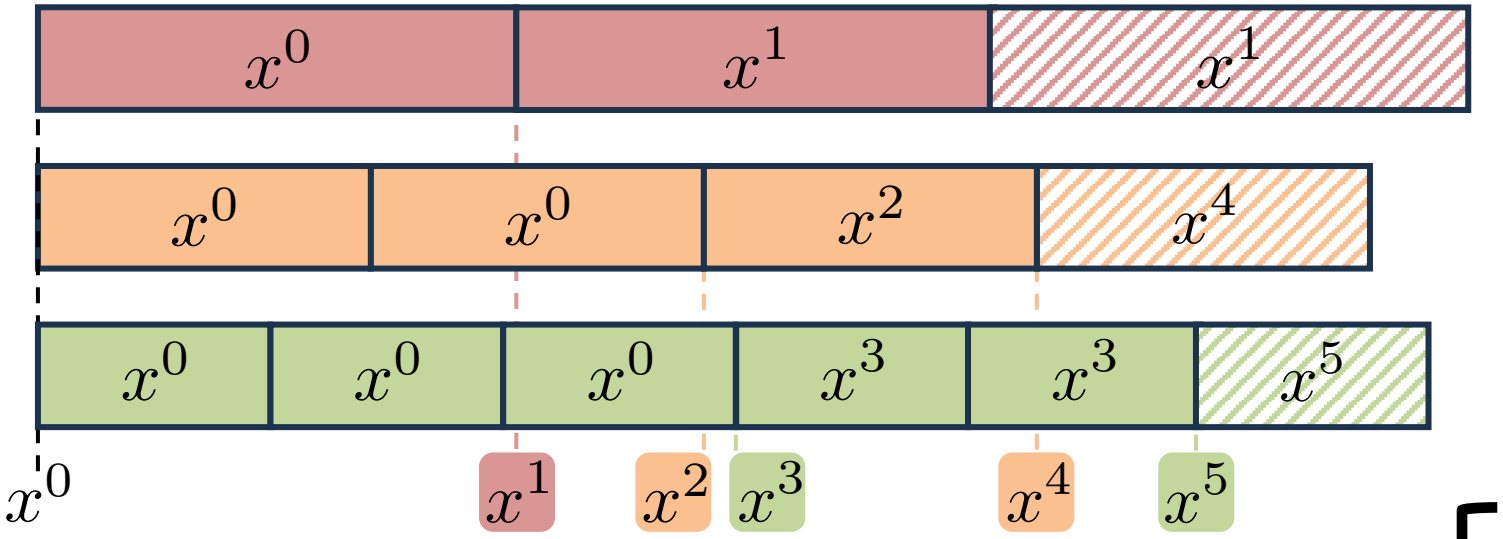
$$x^4 = x^3 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

$$\begin{matrix} 1 & 1 & 1 & b \end{matrix}$$

$$\begin{bmatrix} & & & \\ G_1 & G_2 & G_3 & \\ & & & \end{bmatrix}$$

$$G_2 = \nabla f_2\left(x^2; \xi_2^{2,1}\right)$$

Ringleader ASGD: Phase 2 (update)



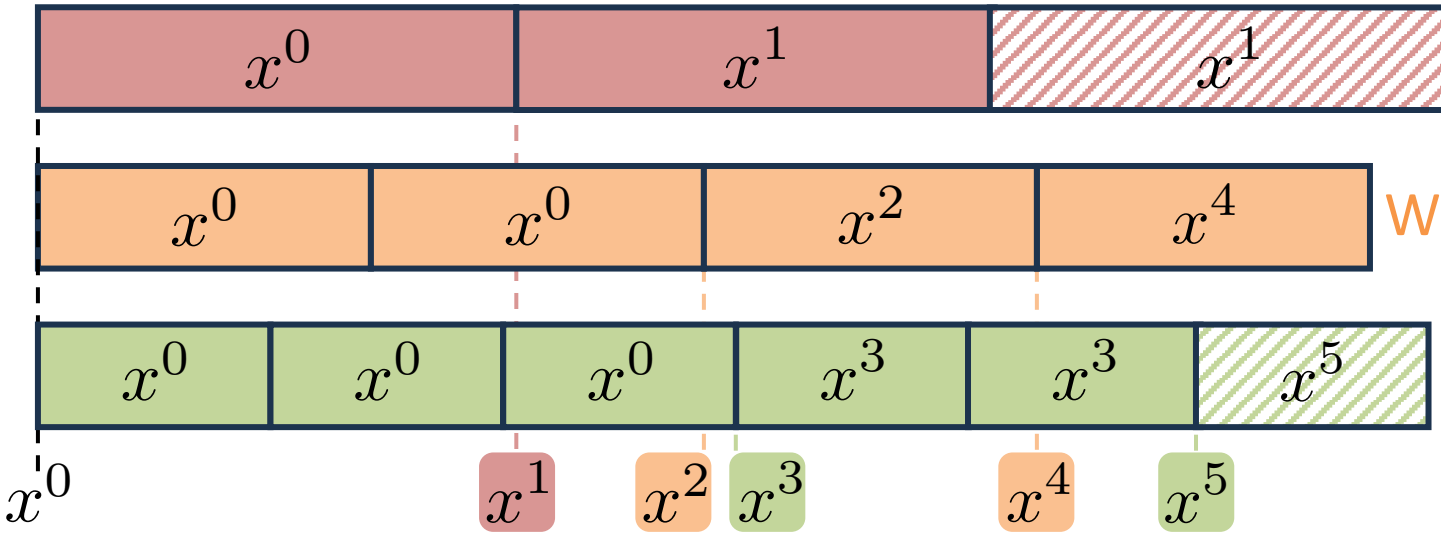
$$x^5 = x^4 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i}$$

$$\begin{matrix} 2 & 1 & 1 & b \end{matrix}$$

$$\begin{bmatrix} G_1 & G_2 & G_3 \end{bmatrix}$$

$$G_1 = \sum_{j=1}^{2} \nabla f_1\left(x^3; \xi_1^{3,j}\right)$$

Ringleader ASGD: Phase 2 (update)



What to do with this gradient?

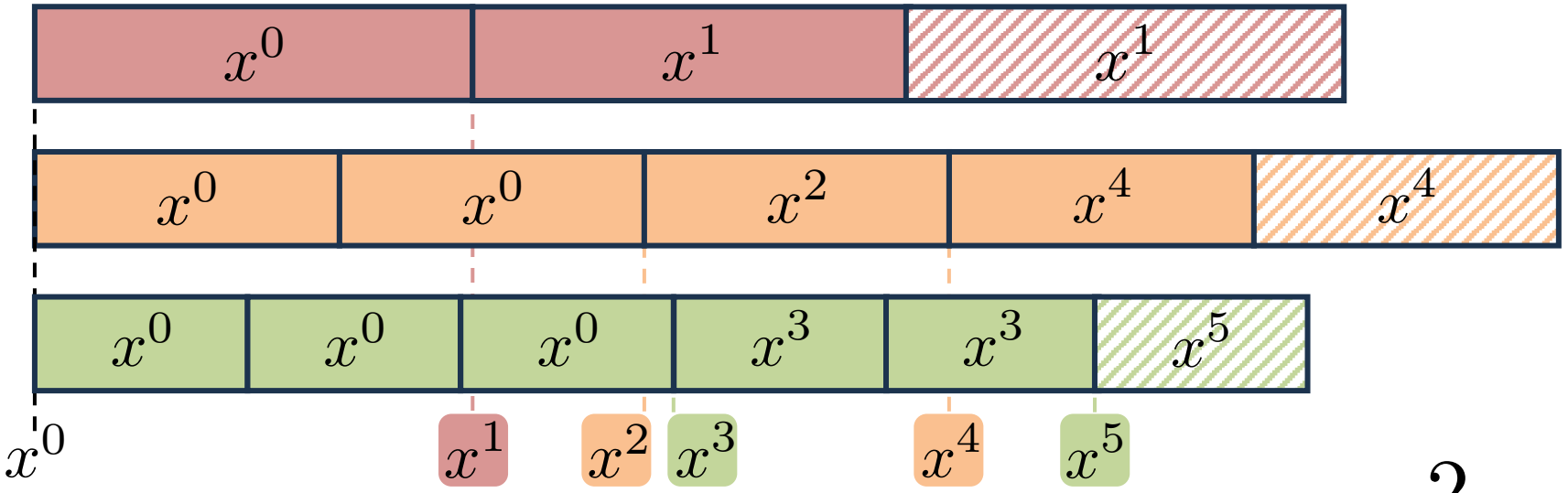
$$\begin{bmatrix} & G_2^+ & \end{bmatrix}$$

Store it for later use!

$$G_2^+ = \nabla f_2\left(x^4; \xi_2^{4,1}\right)$$

$$\begin{array}{cccc} 2 & 1 & 1 & b \\ \begin{bmatrix} \end{bmatrix} & \begin{bmatrix} \end{bmatrix} & \begin{bmatrix} \end{bmatrix} & \\ G_1 & G_2 & G_3 & \end{array}$$

Ringleader ASGD: Phase 2 (update)



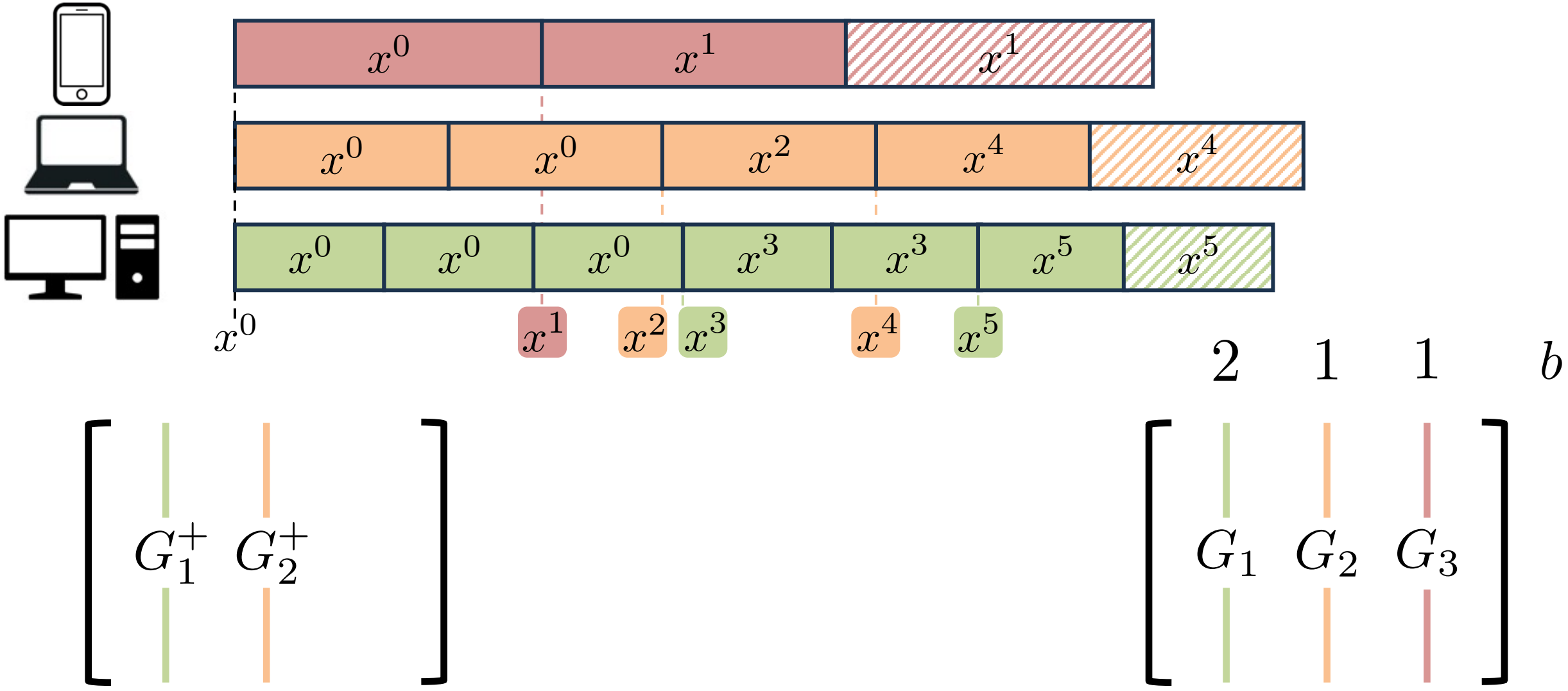
$$\begin{matrix} 2 & 1 & 1 & b \end{matrix}$$

$$\begin{bmatrix} & G_2^+ & \end{bmatrix} \qquad \begin{bmatrix} G_1 & G_2 & G_3 \end{bmatrix}$$

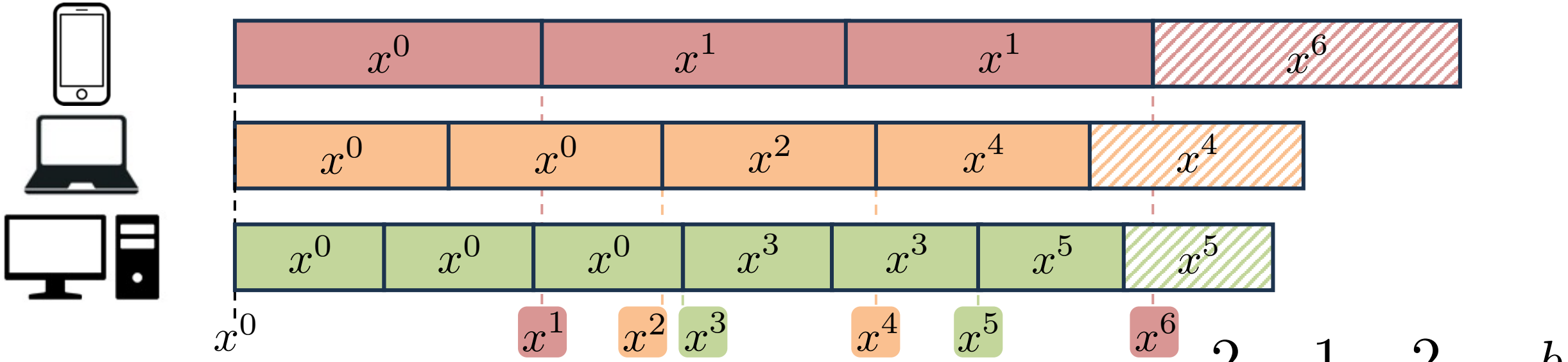
Store it for later use!

$$G_2^+ = \nabla f_2\left(x^4; \xi_2^{4,1}\right)$$

Ringleader ASGD: Phase 2 (update)



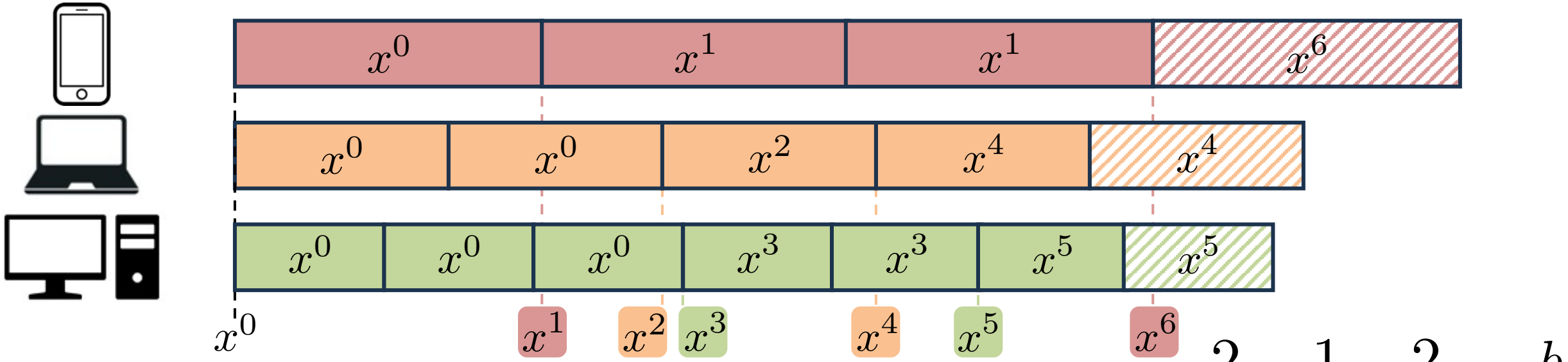
Ringleader ASGD: Phase 2 (update)



$$\begin{matrix} 2 & 1 & 2 & b \end{matrix}$$

$$\begin{bmatrix} | & | \\ G_1^+ & G_2^+ \\ | & | \end{bmatrix} \qquad x^6 = x^5 - \gamma \frac{1}{n} \sum_{i=1}^{n} \frac{G_i}{b_i} \qquad \begin{bmatrix} | & | & | \\ G_1 & G_2 & G_3 \\ | & | & | \end{bmatrix}$$

Ringleader ASGD: Prepare for the Next Round

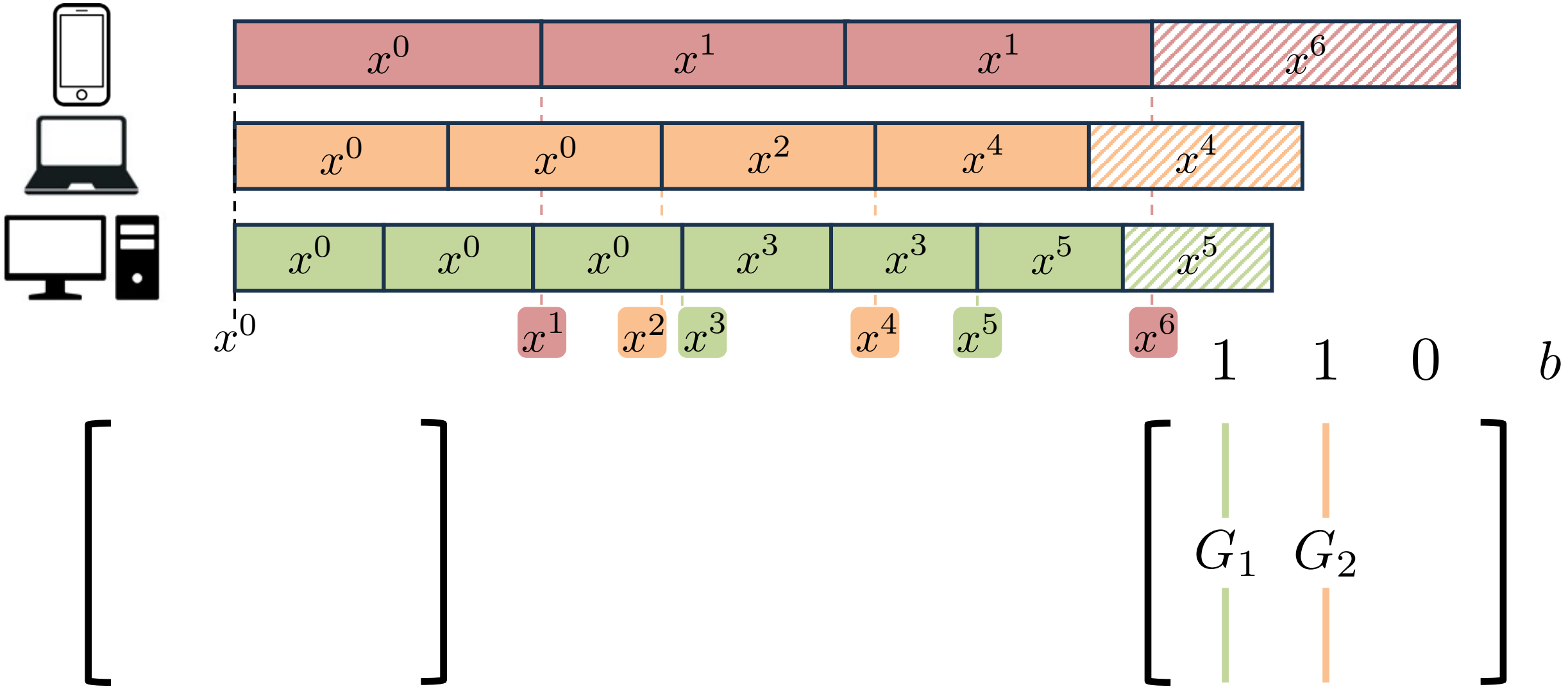


$$\begin{bmatrix} G_1^+ & G_2^+ \end{bmatrix}$$

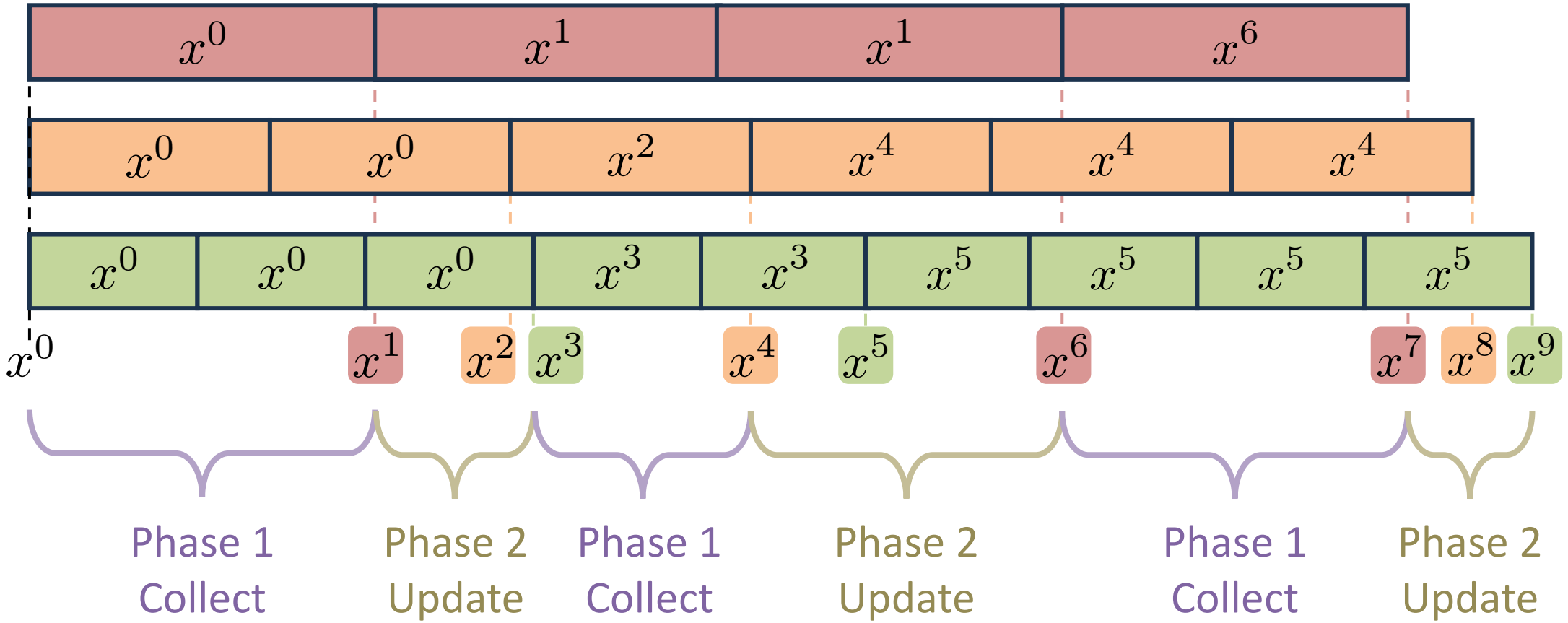
Move the stored ones
back to the original table

$$\begin{bmatrix} G_1 & G_2 & G_3 \end{bmatrix}$$

Ringleader ASGD: Start with Phase 1

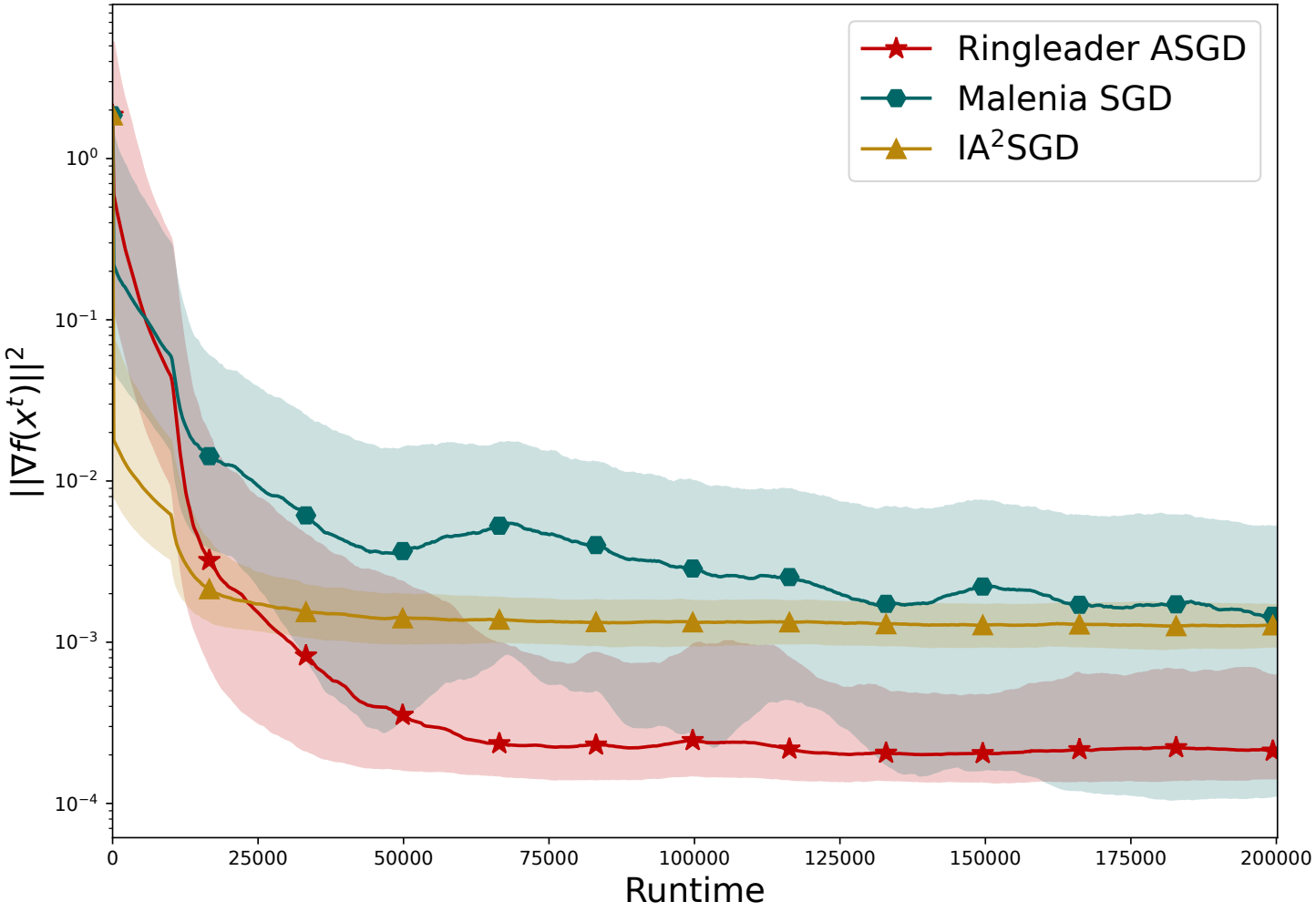


Ringleader ASGD



$$\delta_i^k \leq 2n - 2$$

Ringleader ASGD outperforms existing baselines



Two-layer MLP

MNIST

$$n = 100$$

$$\tau_i = i + |\eta_i|$$

$$\eta_i \sim \mathcal{N}(0, i)$$

KHALAS

More on my PhD Defense
December 4
(lmk if you want to attend)