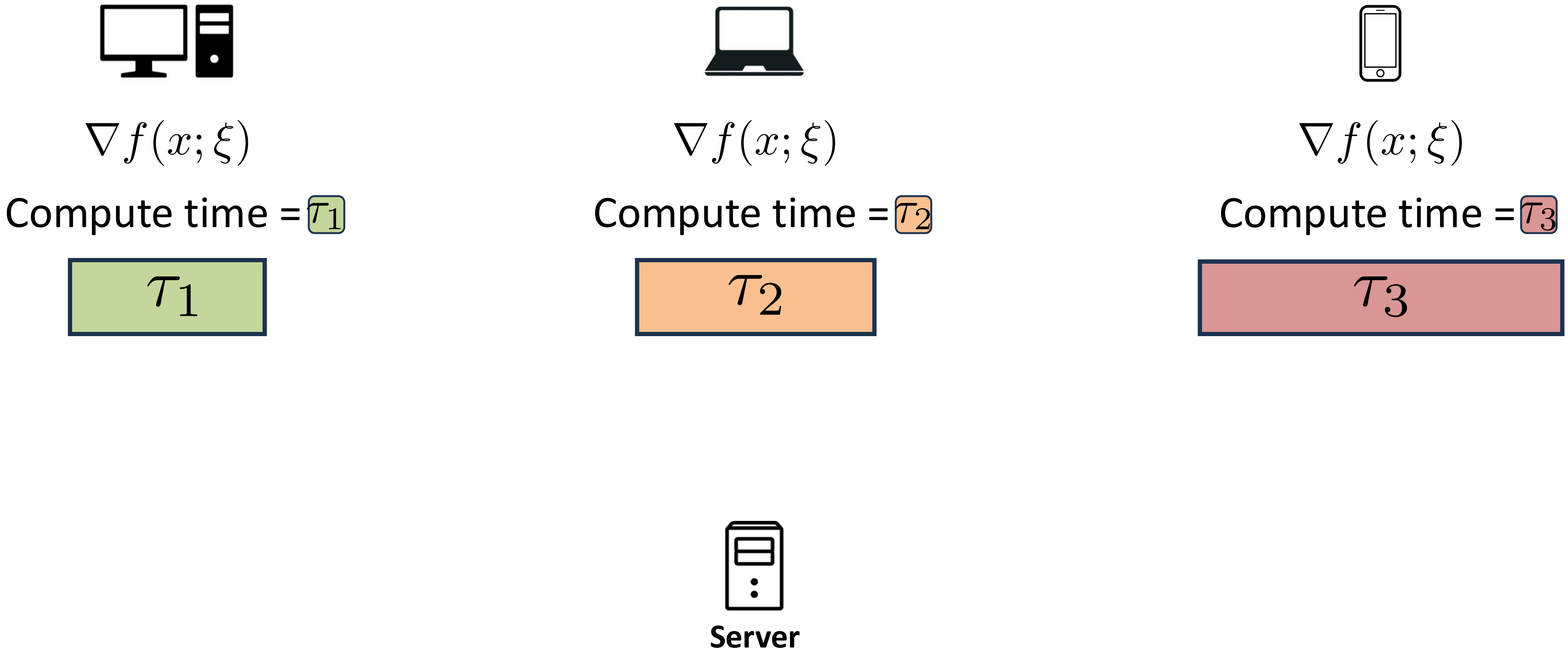
Ringmaster ASGD: The First Asynchronous SGD with Optimal Time Complexity

Artavazd Maranjyan

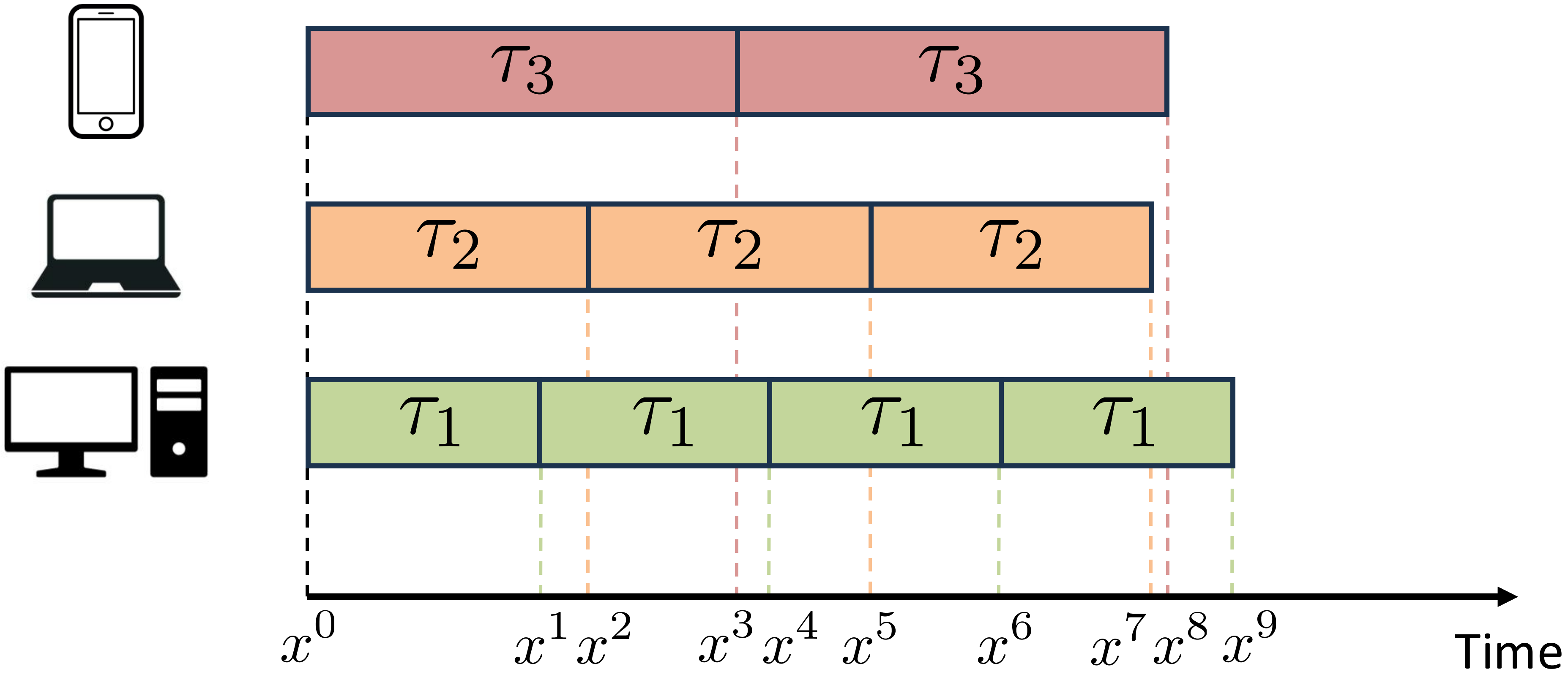
Flower AI Summit, London
March 26-27 2025

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

How to parallelize SGD in heterogeneous systems?

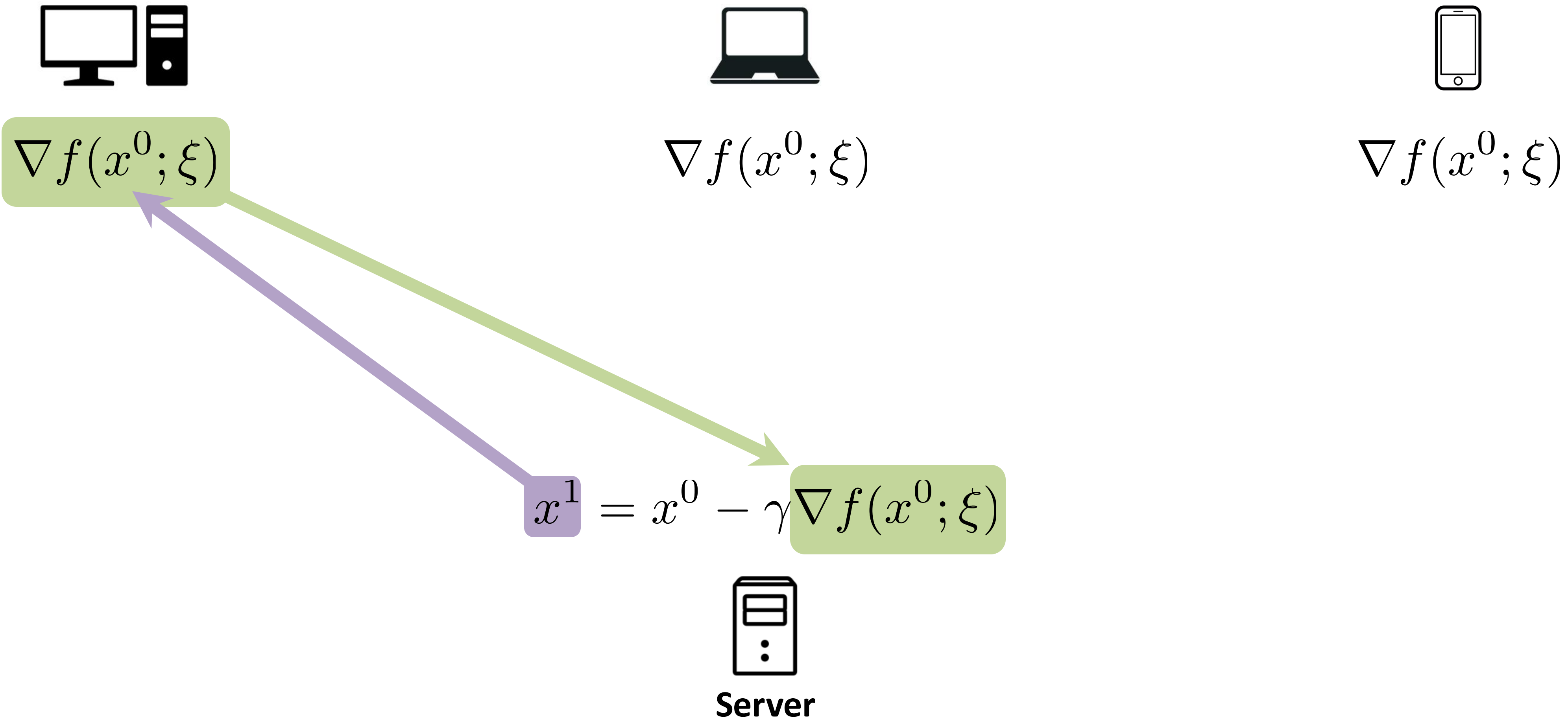


Asynchronous SGD
Remove the synchronization

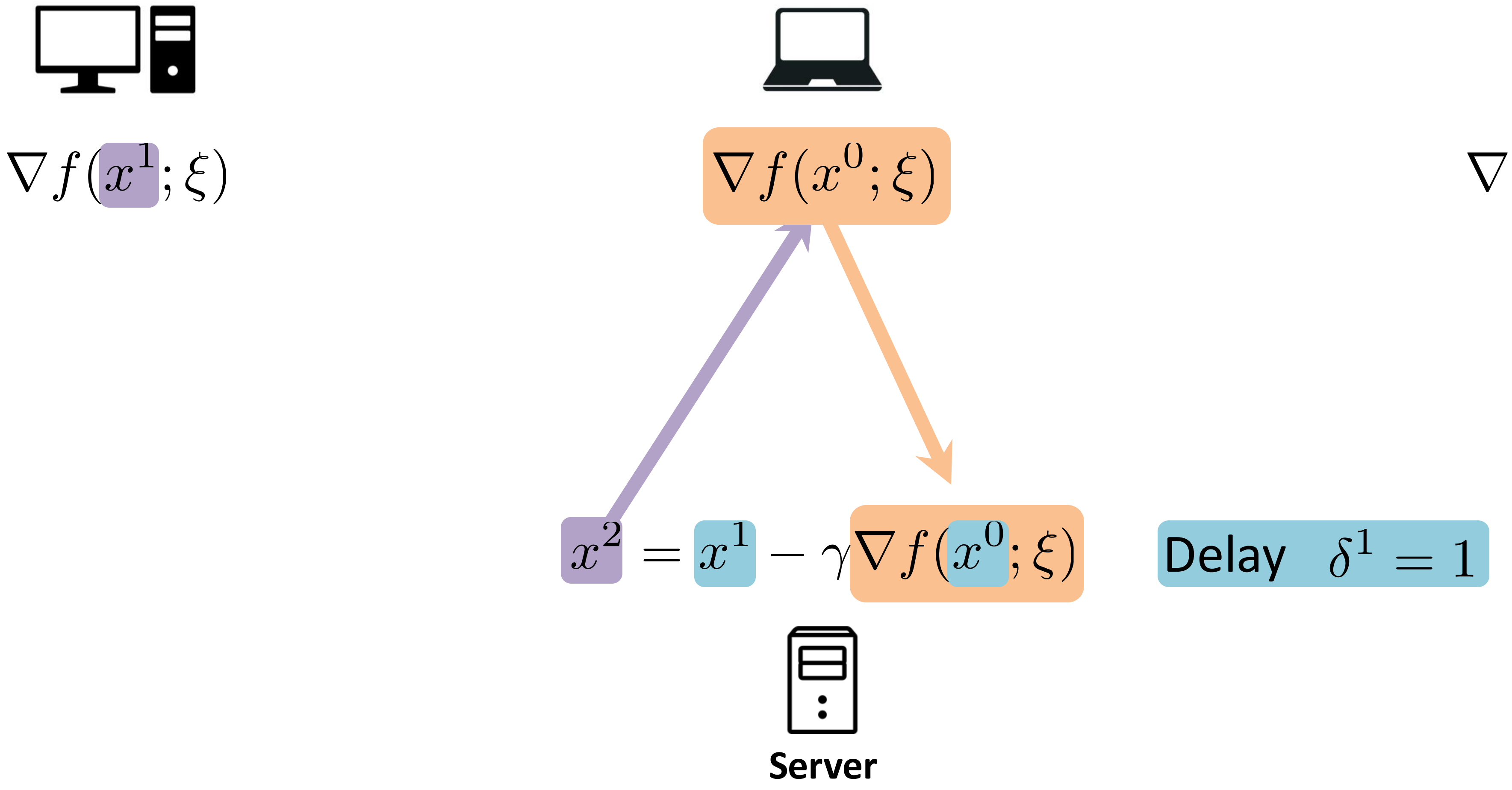


$$x^{k+1} = x^k - \gamma g(x^k)$$

Updates of Asynchronous SGD has delayed stochastic gradients



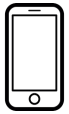
Updates of Asynchronous SGD has delayed stochastic gradients



Updates of Asynchronous SGD has delayed stochastic gradients

$\nabla f(x^1;\xi)$

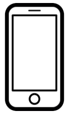
$\nabla f(x^2;\xi)$



$\nabla f(x^0;\xi)$

Server

Updates of Asynchronous SGD has delayed stochastic gradients



$$x^{k+1} = x^k - \gamma \nabla f(x^{k-\delta^k}; \xi)$$

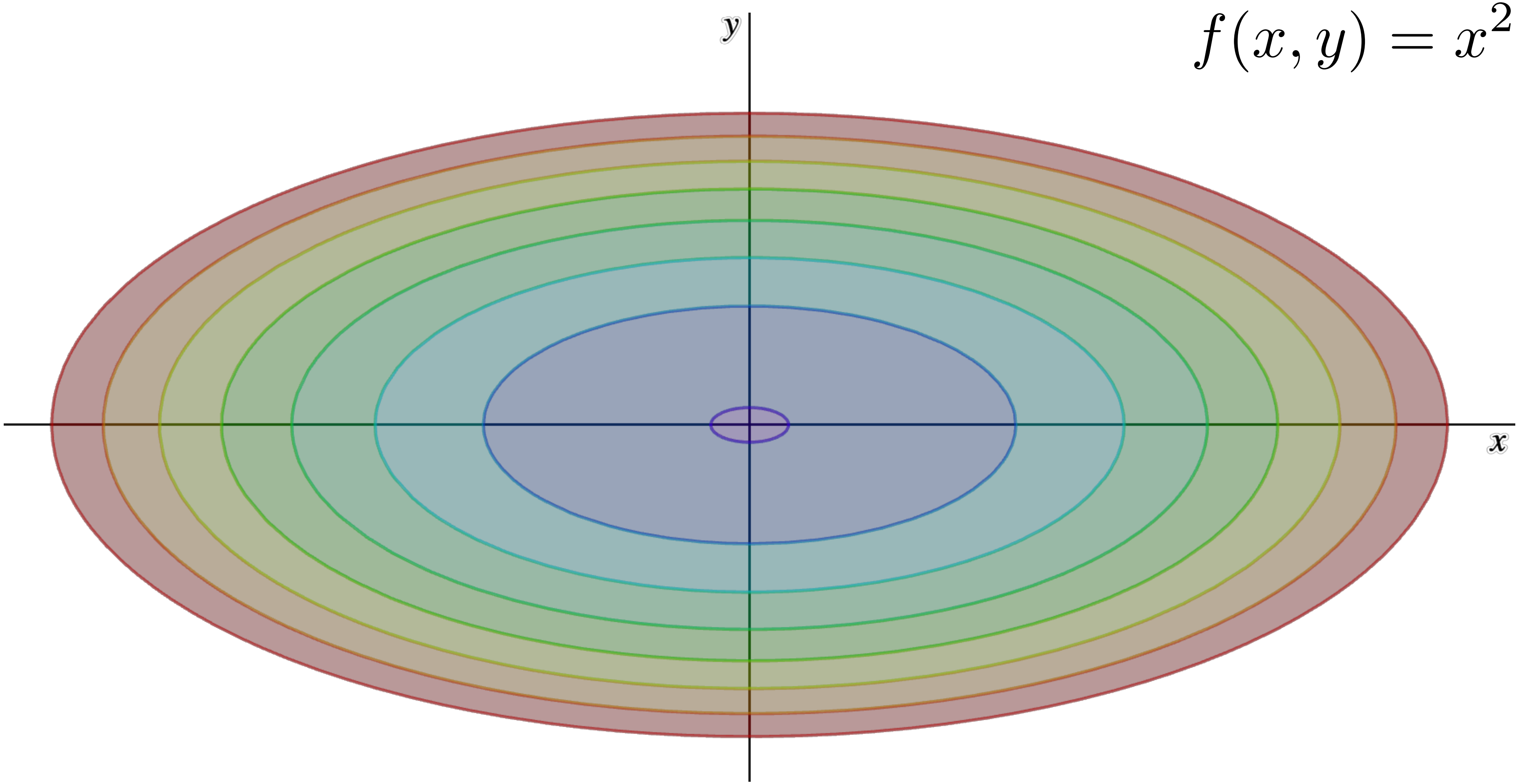
Delay δ^k

Server

Feng Niu, Benjamin Recht, Christopher Re, Stephen J. Wright, (2011). HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent.

Asynchronous SGD can get wild: delays can degrade performance

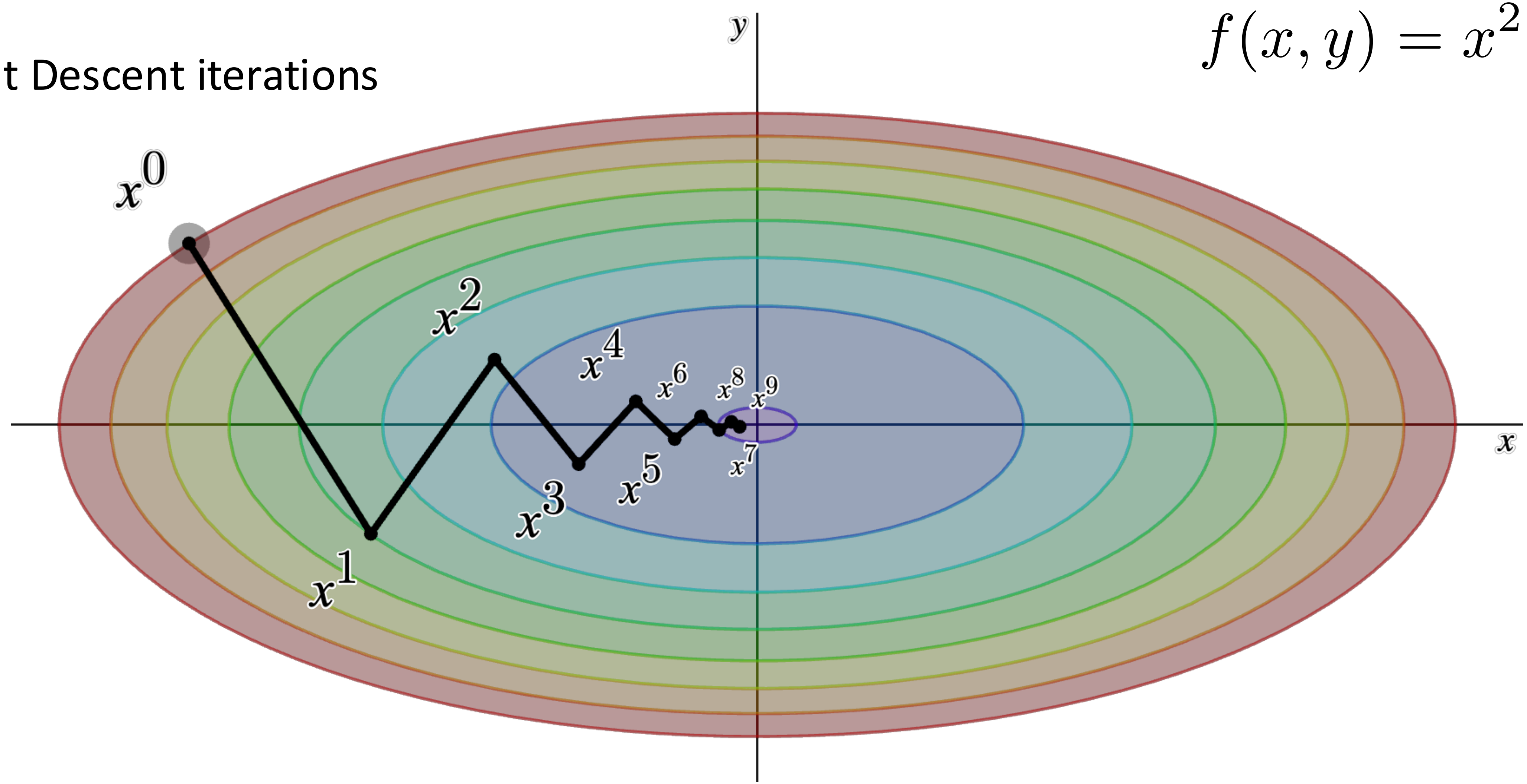
$$f(x, y) = x^2 + 5y^2$$



Asynchronous SGD can get wild: delays can degrade performance

Gradient Descent iterations

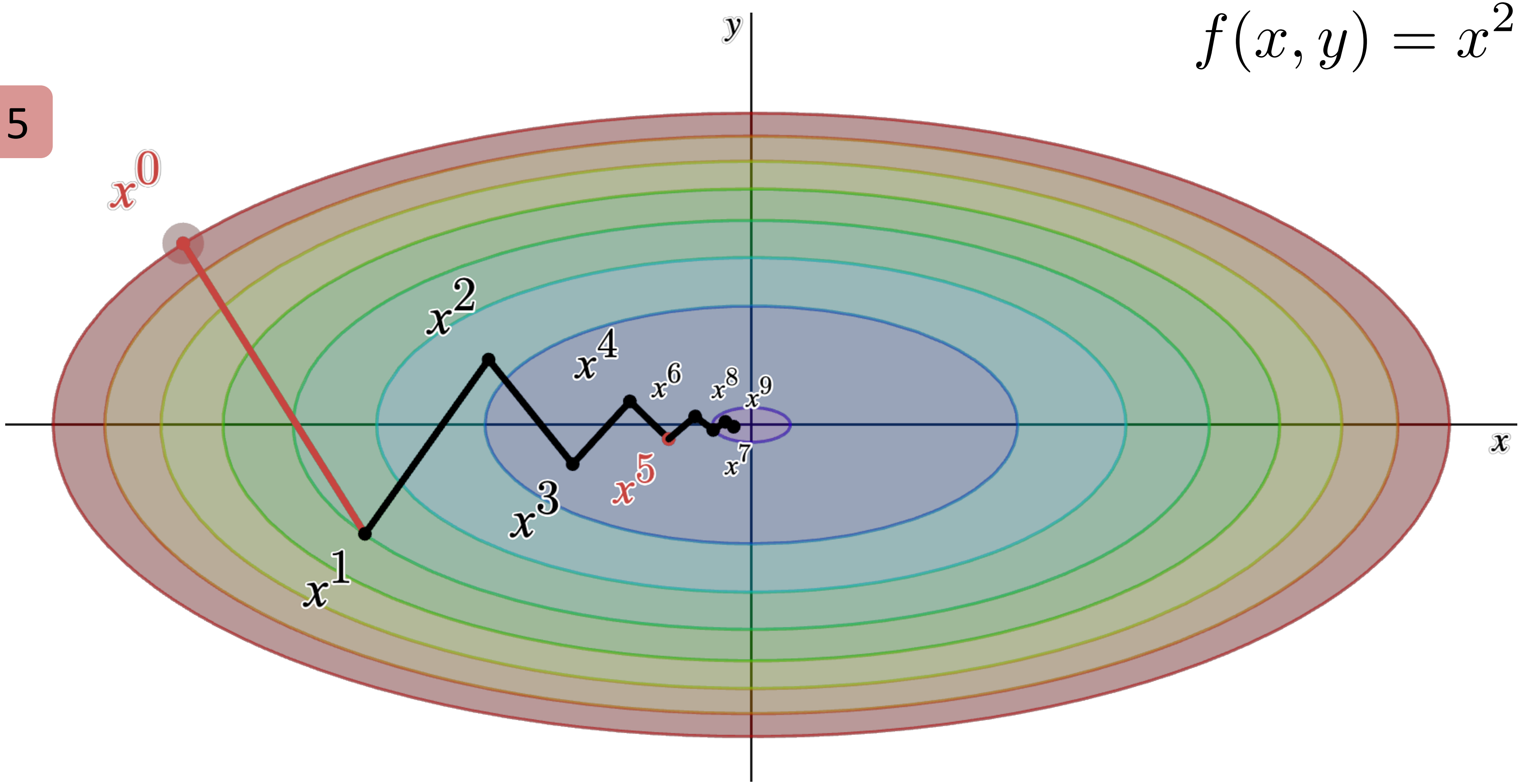
$$f(x, y) = x^2 + 5y^2$$



Asynchronous SGD can get wild: delays can degrade performance

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

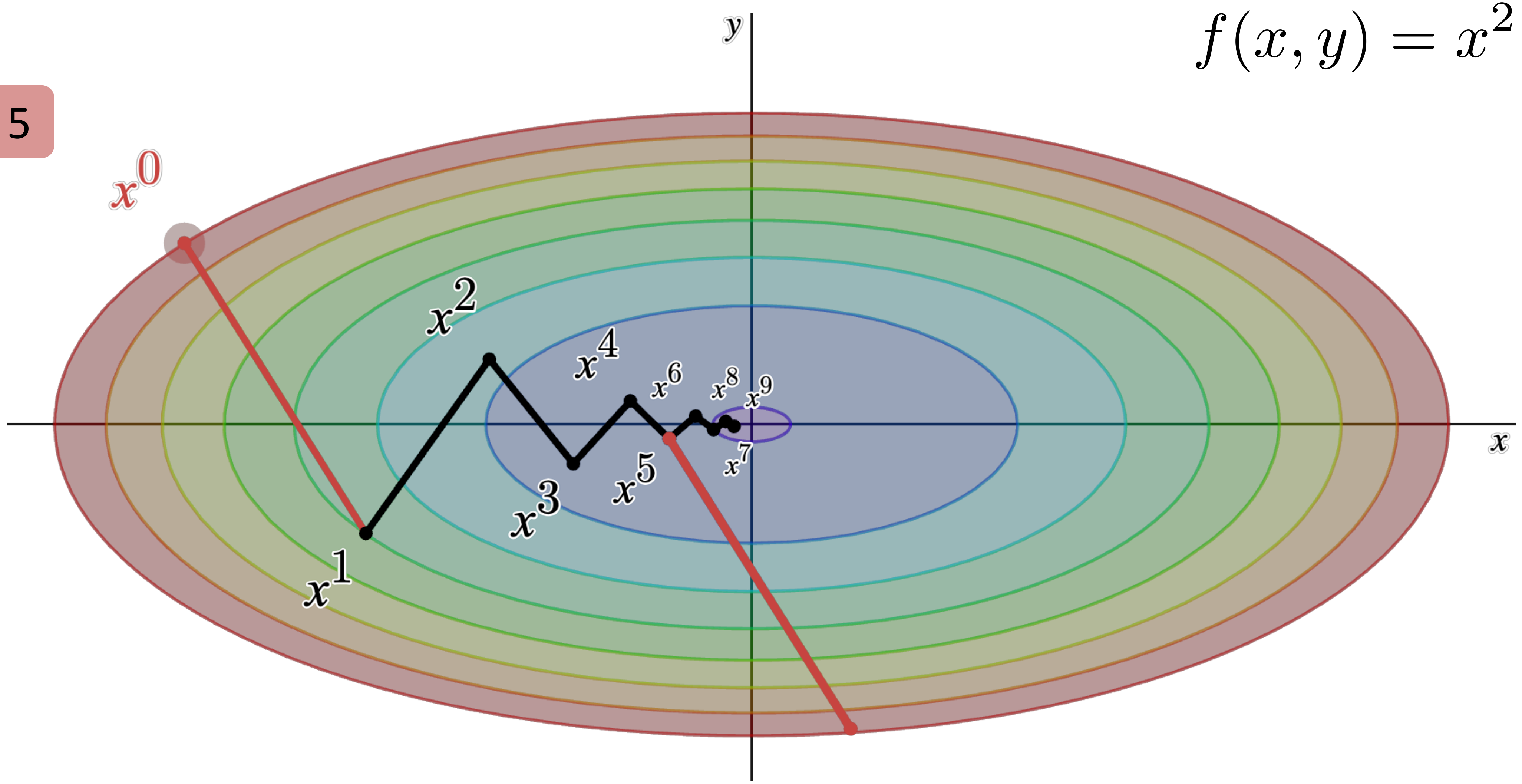
Delay = 5



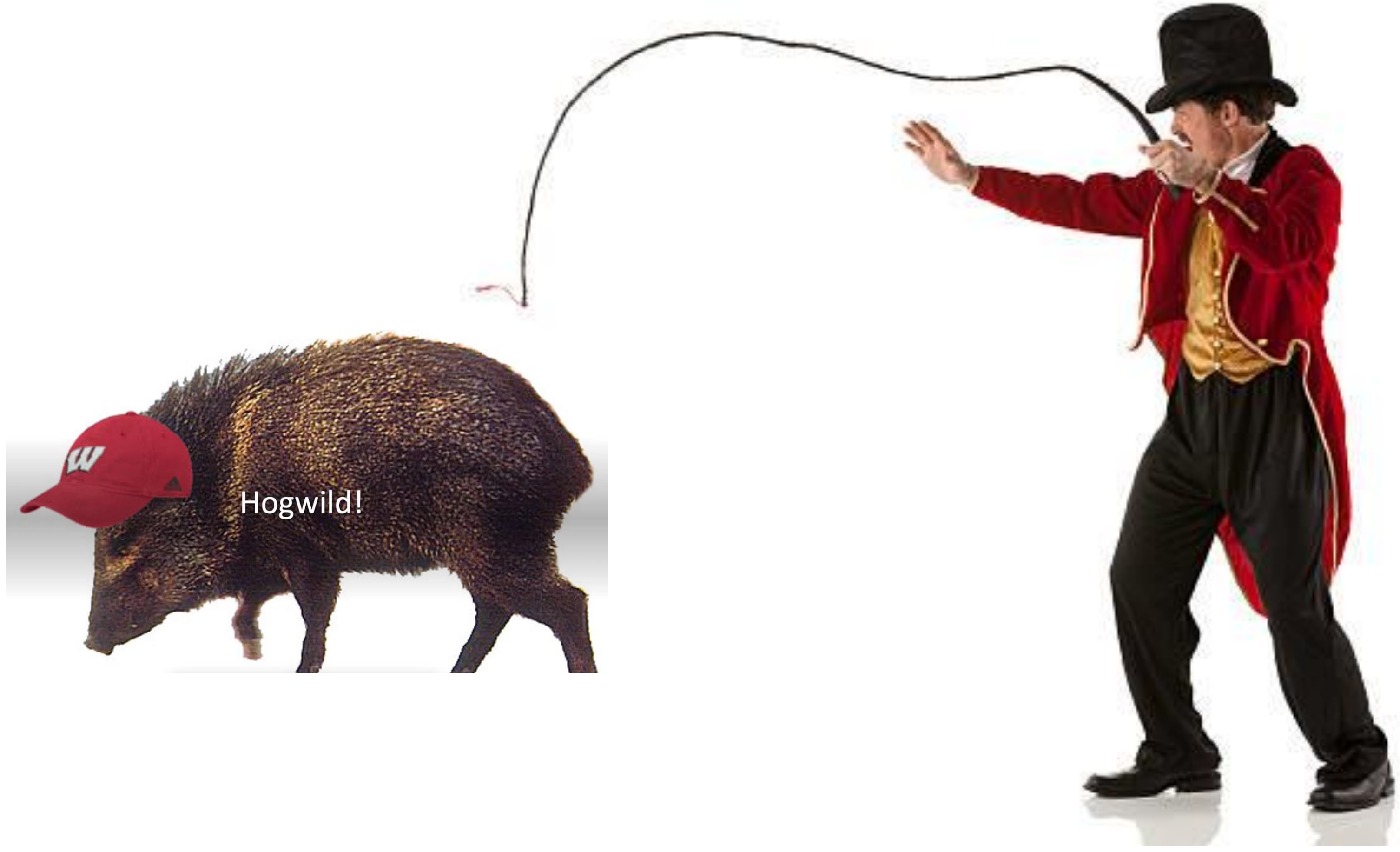
Asynchronous SGD can get wild: delays can degrade performance

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

Delay = 5



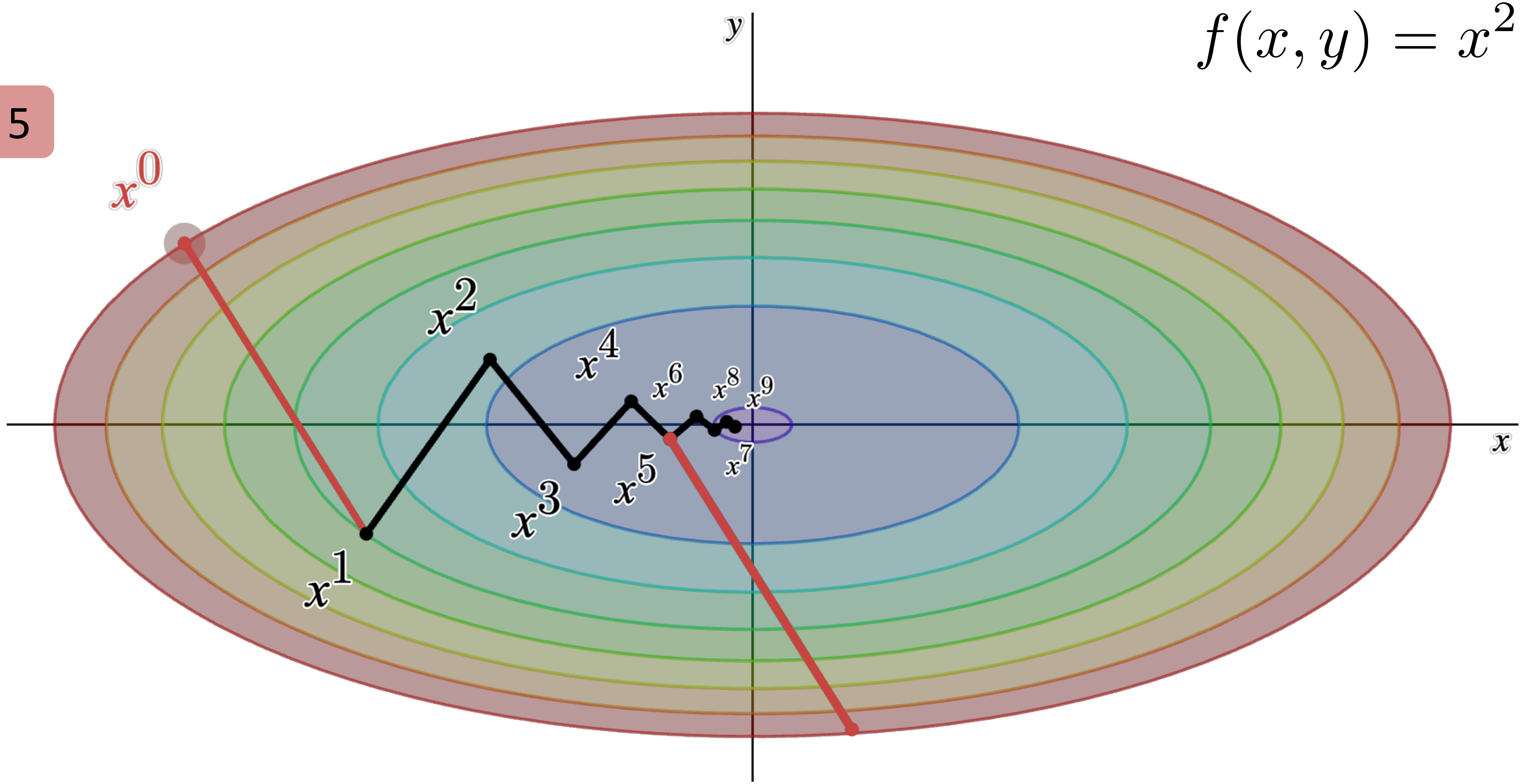
Asynchronous SGD is too wild: Ringmaster ASGD ***tames*** it



The smaller the delay, the better the gradient

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

Delay = 5

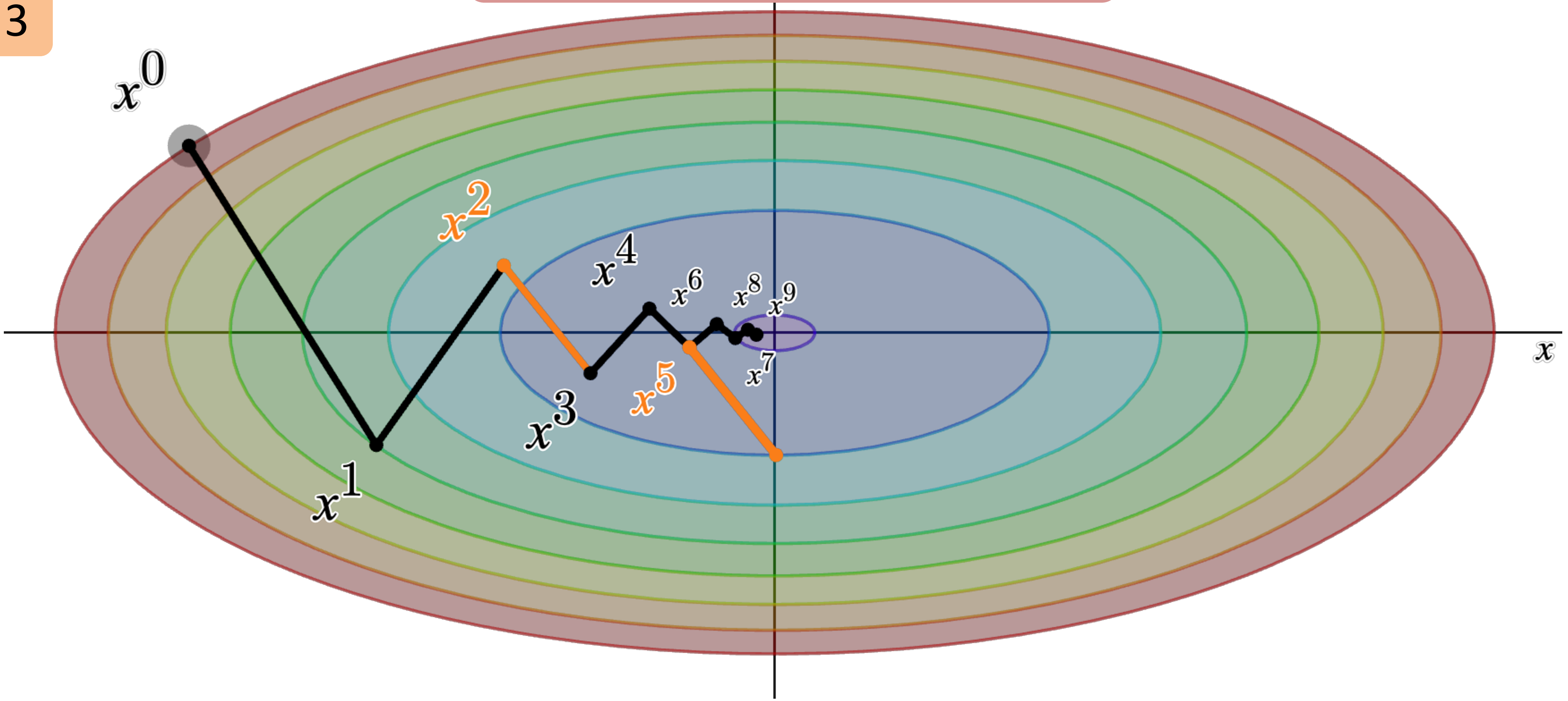


The smaller the delay, the better the gradient

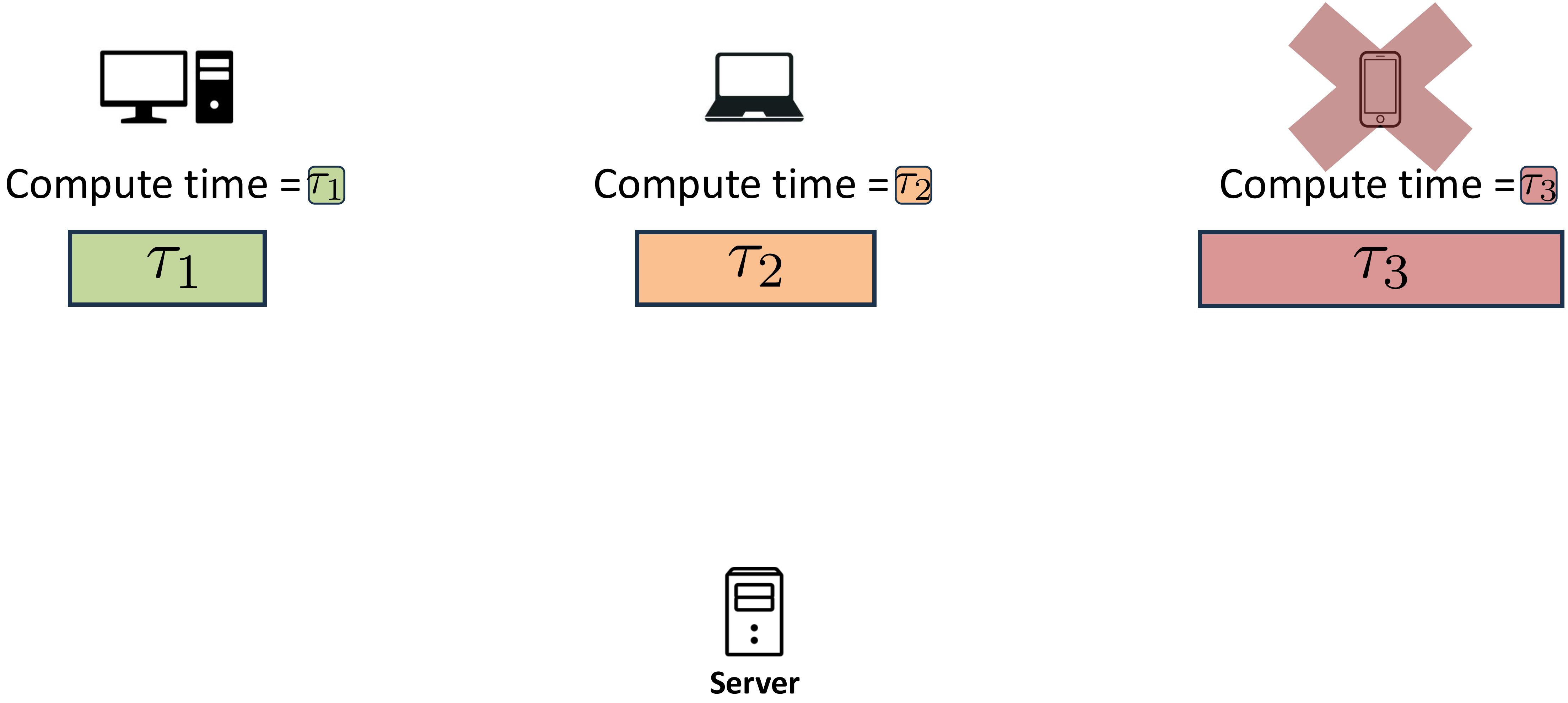
How can we reduce the delay?

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

Delay = 3



Naive approach:
Remove slow workers



Naive approach: Remove slow workers

$$\mathbb{E}\left[\|\nabla f(x;\xi) - \nabla f(x)\|^2\right] \leq \sigma^2$$

Use only the first $m_\star = \arg\min_{m \in [n]} \left\{ \left(\frac{1}{m} \sum_{i=1}^{m} \frac{1}{\tau_i} \right)^{-1} \left(1 + \frac{\sigma^2}{m\varepsilon} \right) \right\}$ fastest workers

$$\mathbb{E}\left[\|\nabla f(x)\|^2\right] \leq \varepsilon$$

Problem: τ_i-s may be unknown and dynamic

Ringmaster ASGD:
Have a threshold on delays

$$\nabla f\left(x^{k-\delta^k}; \xi_i^{k-\delta^k}\right)$$

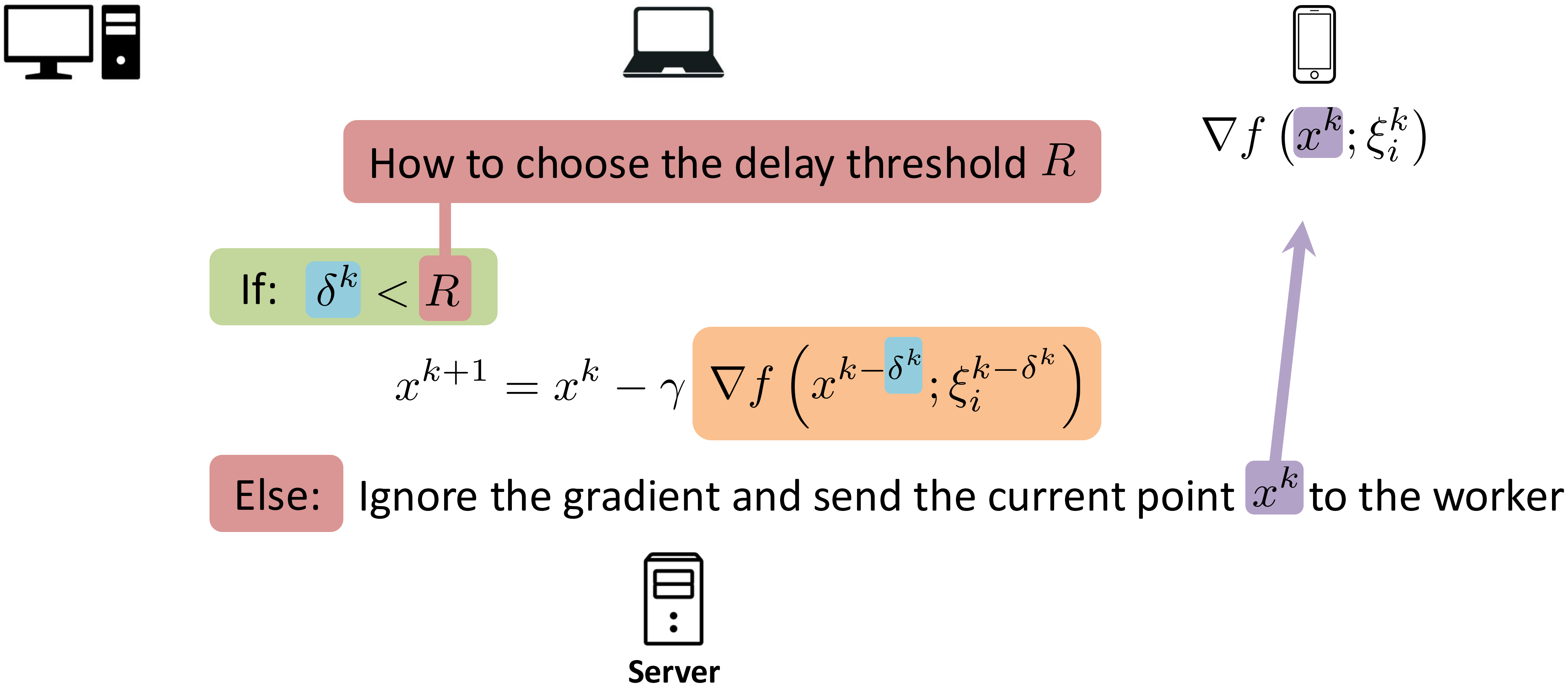
If: $\delta^k < R$

$$x^{k+1} = x^k - \gamma \nabla f\left(x^{k-\delta^k}; \xi_i^{k-\delta^k}\right)$$

Else: Ignore the gradient and send the current point x^k to the worker

Server

Ringmaster ASGD:
Have a threshold on delays



Certain threshold choices in Ringmaster ASGD recover previous methods

$$R = \max\left\{1, \left\lceil \frac{\sigma^2}{\varepsilon} \right\rceil\right\}$$

$R = 1$
Hero SGD

Sweet spot

$R = \infty$
HOGWILD!

Theoretical results validate our intuition

$$\mathcal{O}\left(\frac{R}{\varepsilon} + \frac{\sigma^2}{\varepsilon^2}\right)$$

Number of iterations

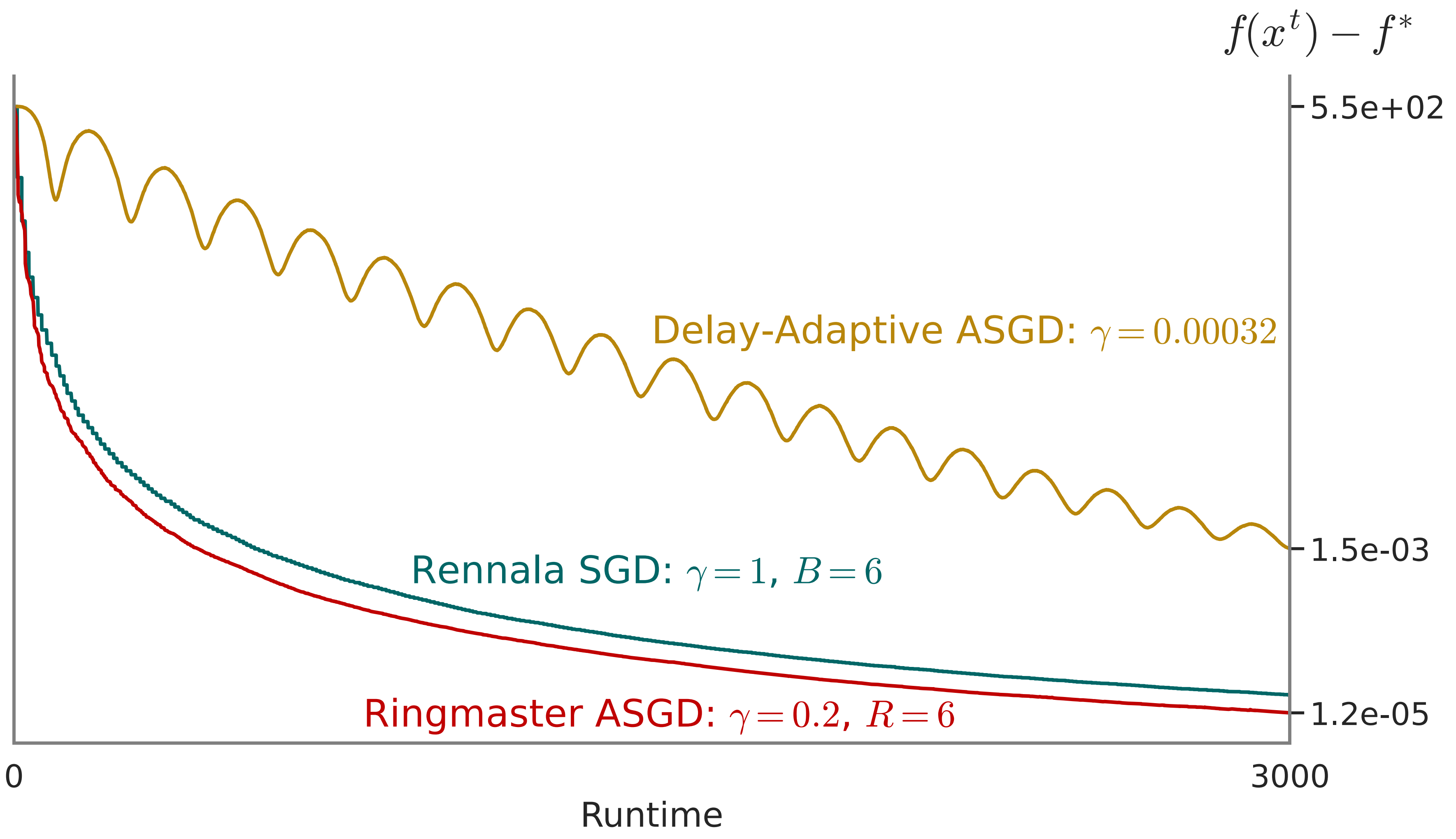
$$\mathcal{O}\left(\min_{m\in[n]}\left[\left(\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\right)^{-1}\left(\frac{1}{\varepsilon} + \frac{\sigma^2}{m\varepsilon^2}\right)\right]\right)$$

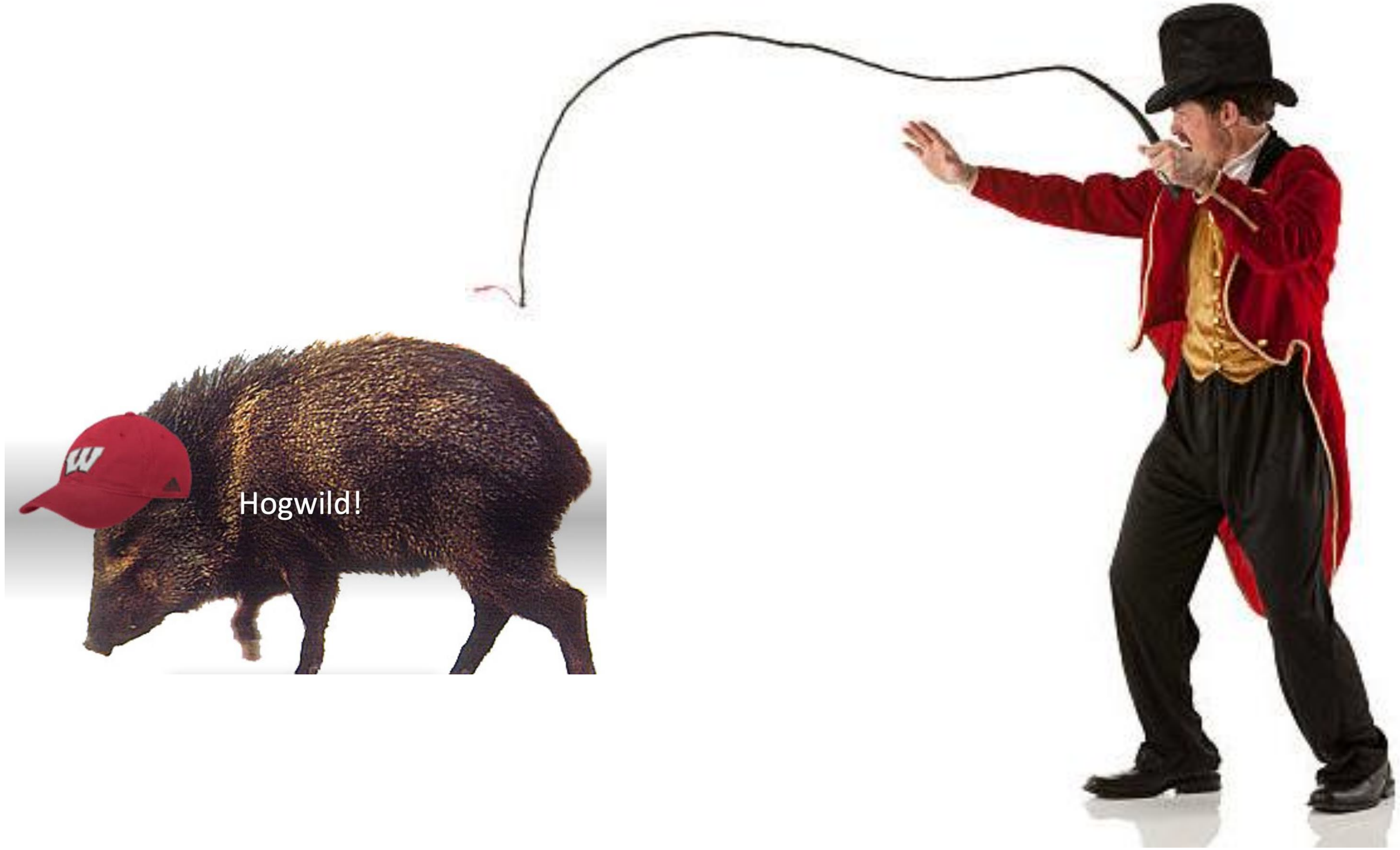
Time complexity

non-decreasing

decreasing

Ringmaster ASGD outperforms existing baselines




Hogwild!