Linearly Converging Error Compensated SGD

Eduard Gorbunov
MIPT, Yandex and Sirius

Dmitry Kovalev
KAUST

Dmitry Makarenko
MIPT

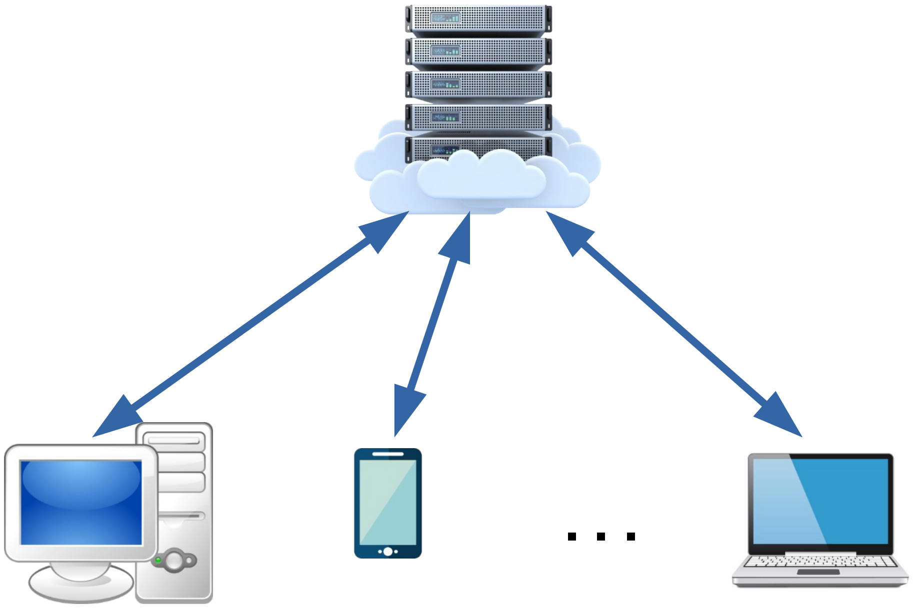
Peter Richtárik
KAUST

Dmitry Kovalev
PhD student
KAUST

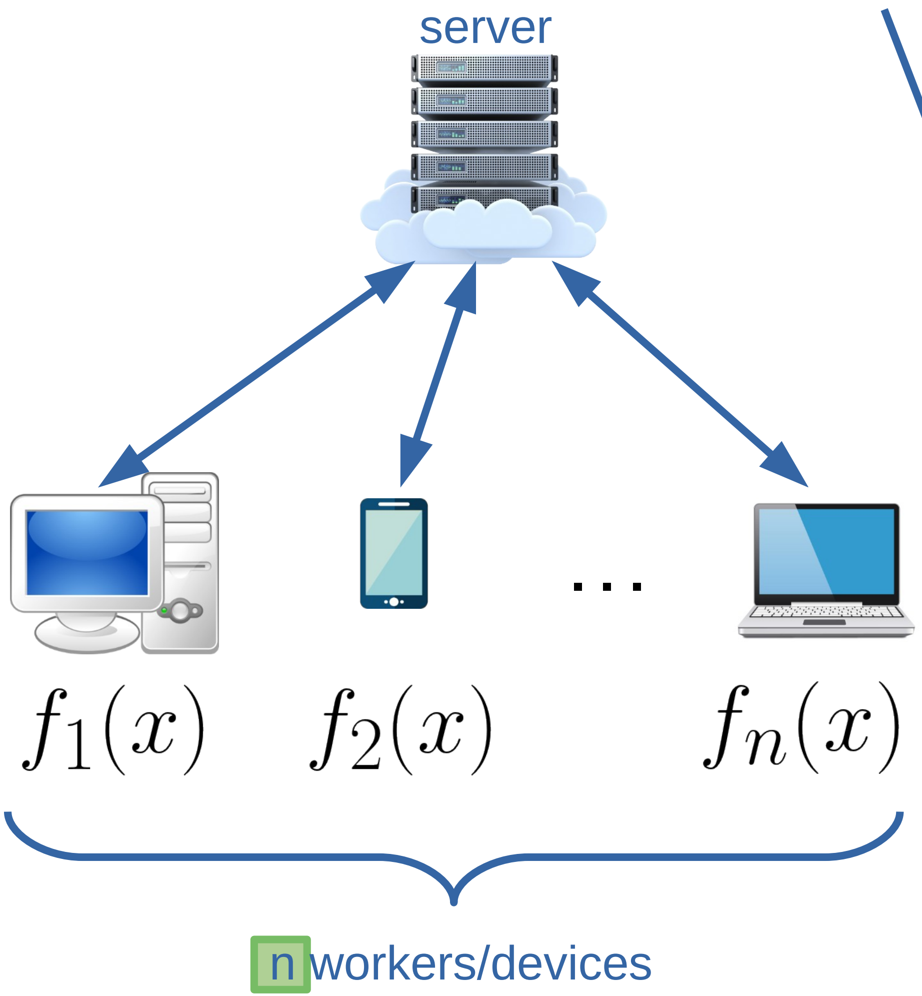
Dmitry Makarenko
PhD student
MIPT

Peter Richtárik
Professor of Computer Science
KAUST

1. The Problem



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



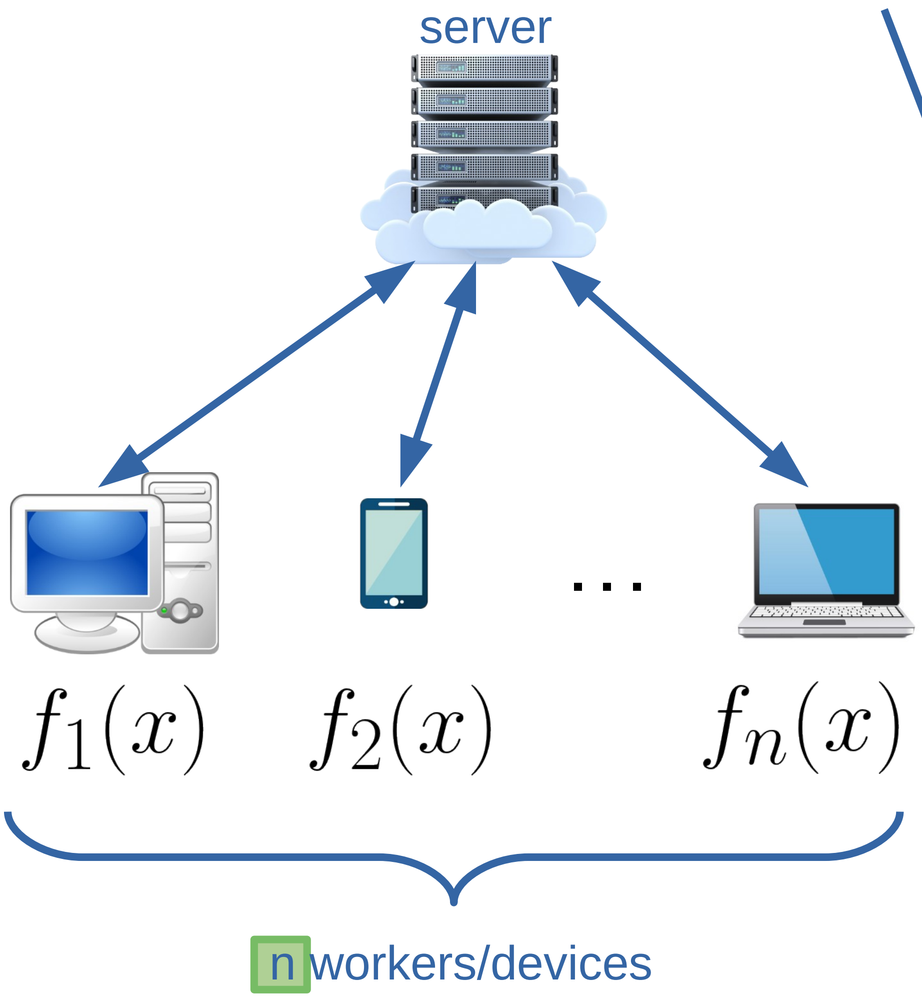
parameters of the model

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

\# of parameters

\# of workers/devices

loss on the data accessible by worker *i*



parameters of the model

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

# of parameters

# of workers/devices

loss on the data accessible by worker *i*

- $f_i(x) = \mathbf{E}_{\xi_i \sim \mathcal{D}_i}\left[f_{\xi_i}(x)\right]$
- $f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$

Assumptions

$$\boxed{\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|}$$

$$\boxed{f_i(x) - f_i(y) \geq \langle \nabla f_i(y), x - y \rangle}$$

- $f_1, f_2, \ldots, f_n$ – L-smooth and convex

Assumptions

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$$

$$f_i(x) - f_i(y) \geq \langle \nabla f_i(y), x - y \rangle$$

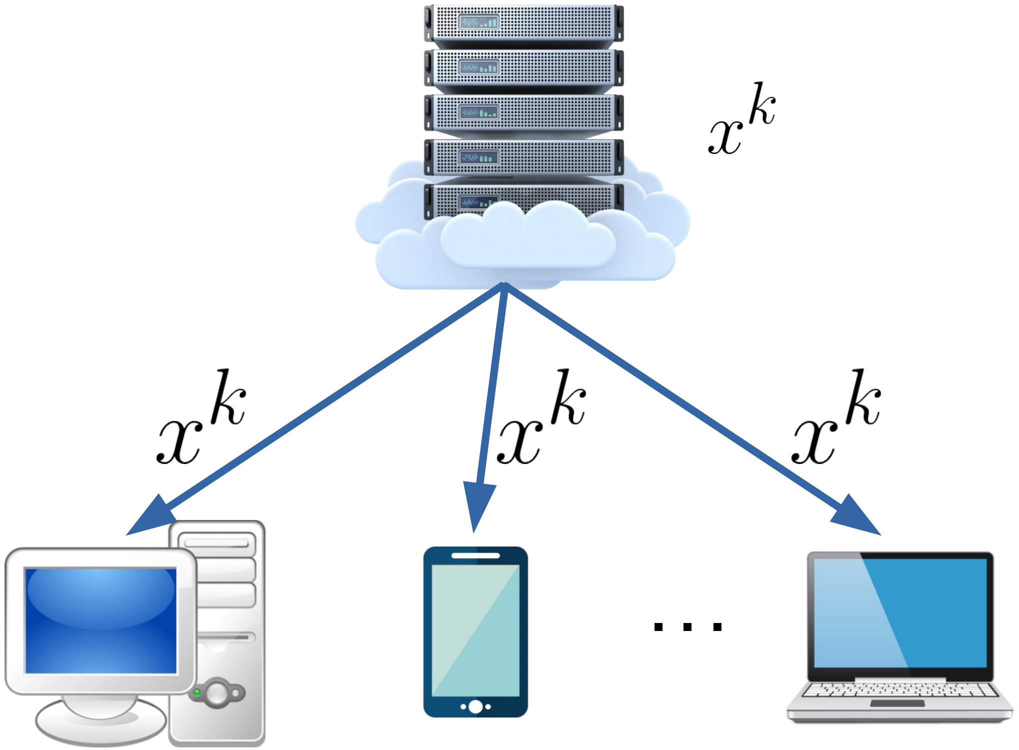
- $f_1, f_2, \ldots, f_n$ – L-smooth and convex
- f – strongly quasi-convex

$$f\left(x^*\right) \geq f(x) + \langle \nabla f(x), x^* - x \rangle + \frac{\mu}{2}\|x^* - x\|^2$$

the solution of the problem

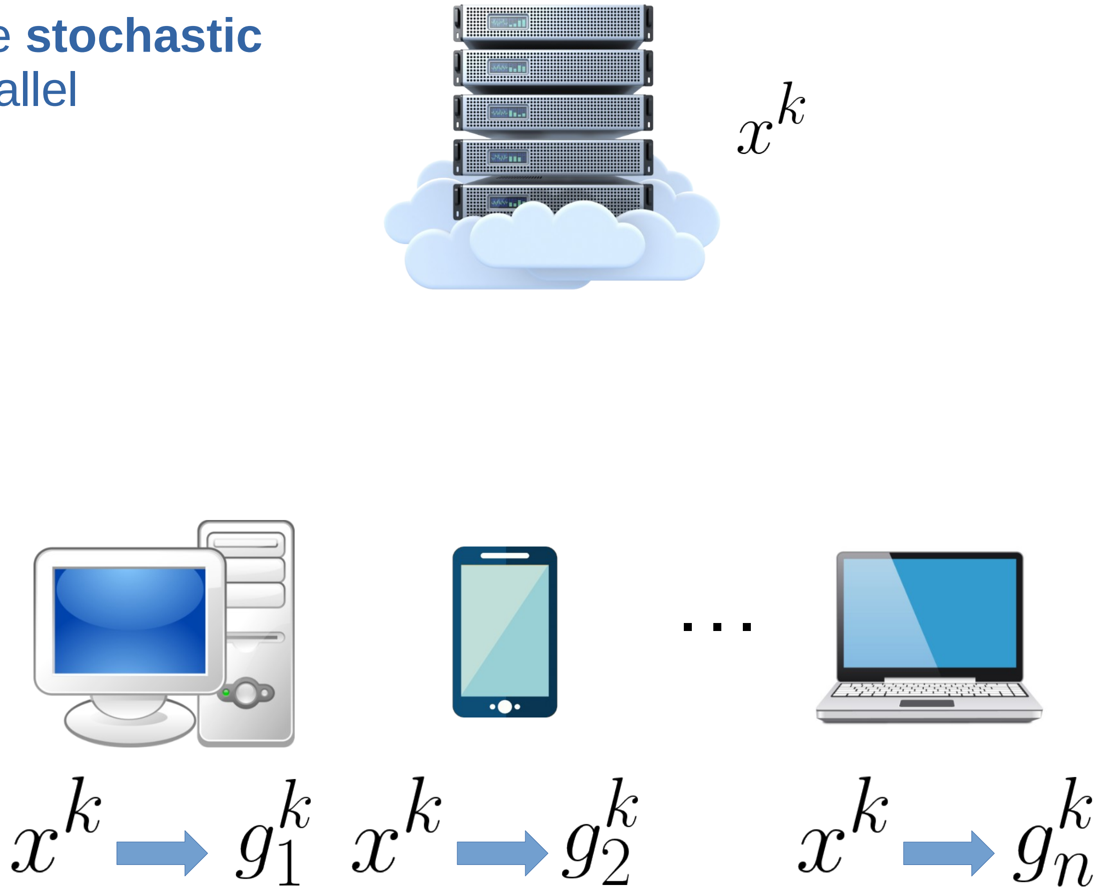
2. Parallel SGD

1 Server broadcasts the parameters

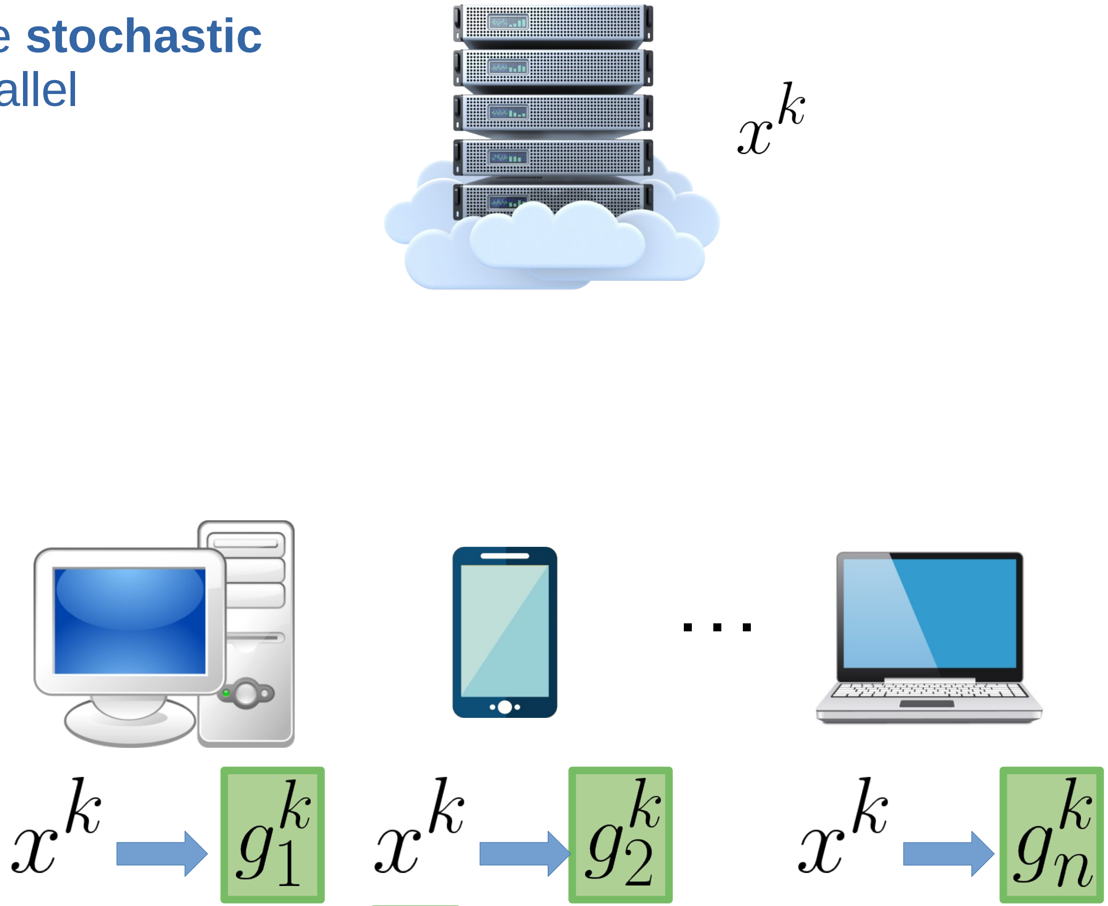


1 Server broadcasts the parameters

2 Devices compute **stochastic gradients** in parallel



1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel

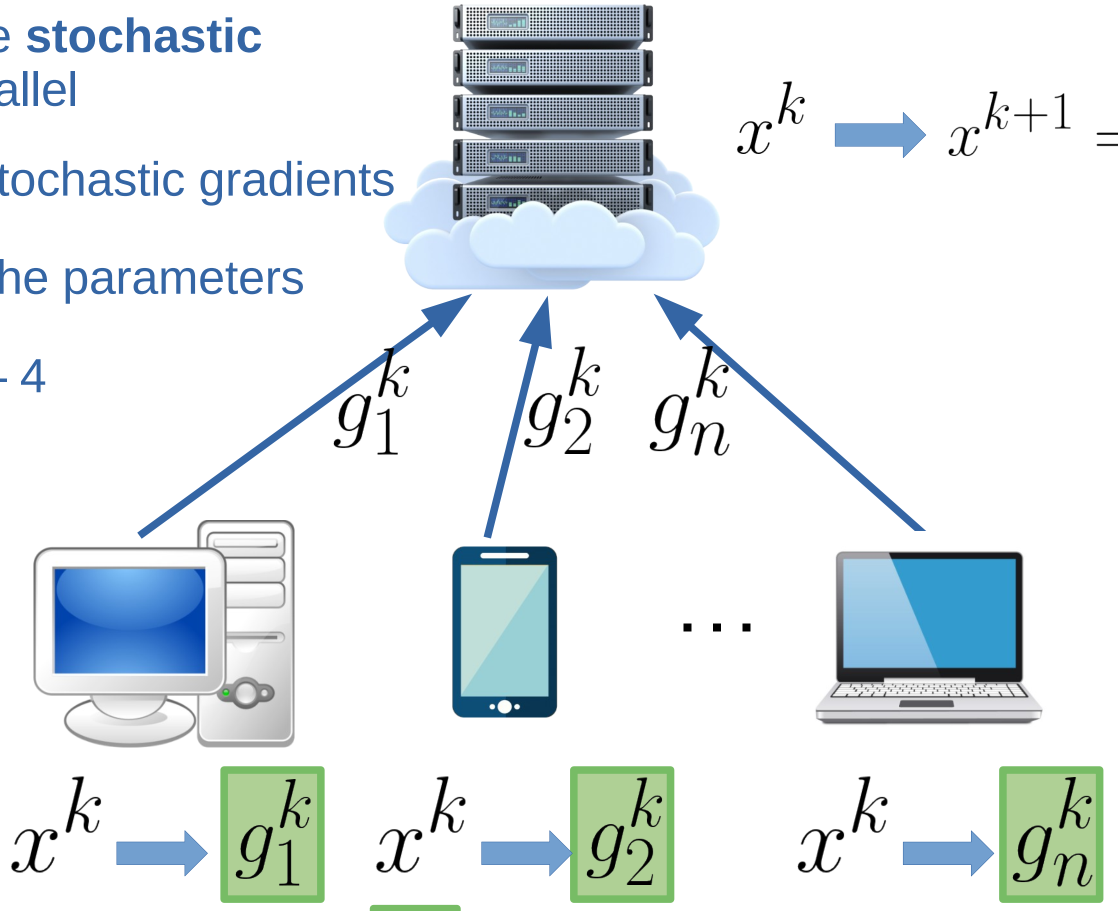


$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters
5. Repeat steps 1 – 4

stepsize

$$x^k \Rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$

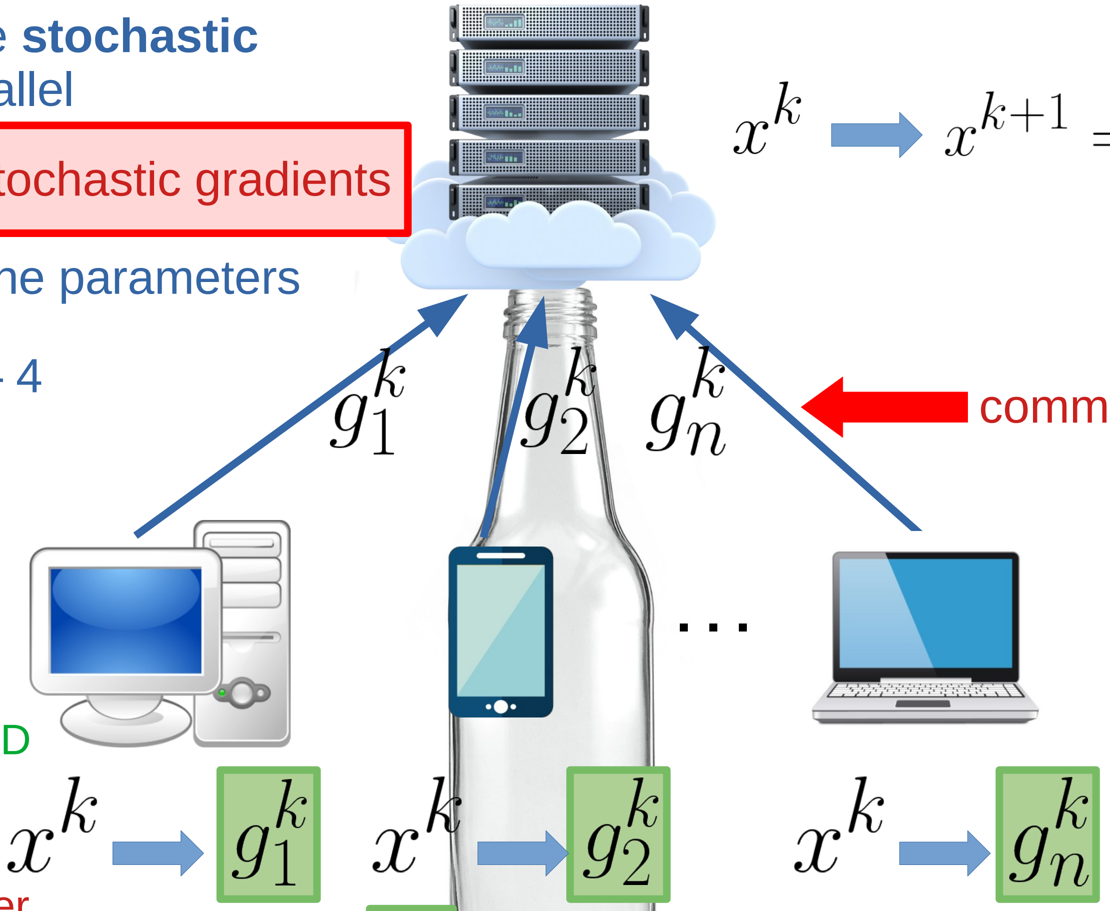


$$\mathbb{E}\boxed{g_i^k} = \nabla f_i(x^k)$$

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Server gathers stochastic gradients
4. Server updates the parameters
5. Repeat steps 1 – 4

stepsize

$$x^k \rightarrow x^{k+1} = x^k - \gamma \cdot \underbrace{\frac{1}{n}\sum_{i=1}^{n} g_i^k}_{g^k}$$



$$\mathbb{E} g_i^k = \nabla f_i(x^k)$$

Good news:

- Very simple algorithm
- Can be much faster than non-parallel SGD

Issues:

- Overload of the server

3. Communication Bottleneck

How to Handle Communication Bottleneck?

- Change the topology of the network → Decentralized optimization
- Do more work on each worker and communicate less → Local-SGD/Federated Averaging

How to Handle Communication Bottleneck?

- Change the topology of the network → Decentralized optimization
- Do more work on each worker and communicate less → Local-SGD/Federated Averaging
- Send less information to reduce the communication cost

Example:

Send $g_i^k = \begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix}$ → Send $\mathcal{C}(g_i^k) = \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

How to Handle Communication Bottleneck?

- Change the topology of the network → Decentralized optimization
- Do more work on each worker and communicate less → Local-SGD/Federated Averaging
- $\boxed{\text{Send less information to reduce the communication cost}}$

We focus on this approach

Example:

$$\text{Send} \quad \cancel{g_i^k = \begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix}} \quad \Rightarrow \quad \text{Send} \quad \mathcal{C}(g_i^k) = \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Compression operator

How to Handle Communication Bottleneck?

- Change the topology of the network → Decentralized optimization
- Do more work on each worker and communicate less → Local-SGD/Federated Averaging
- Send less information to reduce the communication cost

We focus on this approach

Example:

~~Send $g_i^k = \begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix}$~~ → Send $\mathcal{C}(g_i^k) = \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Compression operator

What are the options for choosing this?

Compression Operators

Unbiased compressors
(quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

Biased compressors

$$x \to \mathcal{C}(x)$$

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega\|x\|^2$$

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2$$

Compression Operators

Unbiased compressors (quantizations)

$$x \to \mathcal{Q}(x) \qquad \mathbb{E}[\mathcal{Q}(x)] = x$$

$$\mathbb{E}\|\mathcal{Q}(x) - x\|^2 \leq \omega\|x\|^2$$

Example: RandK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \frac{5}{\boxed{2}} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -7 \\ 0 \end{pmatrix}$$

for unbiasedness

Pick K = 2 components uniformly at random

Biased compressors

$$x \to \mathcal{C}(x)$$

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2$$

Example: TopK (for K = 2)

$$\begin{pmatrix} 1 \\ -15 \\ 0.2 \\ -7 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ -15 \\ 0 \\ 0 \\ 10 \end{pmatrix}$$

Pick K = 2 components with largest absolute value

Methods with Unbiased Compressors

- QSGD

Alistarh, Dan, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. **"QSGD: Communication-efficient SGD via gradient quantization and encoding."** *In Advances in Neural Information Processing Systems*, pp. 1709-1720. 2017.

- TernGrad

Wen, Wei, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. **"Terngrad: Ternary gradients to reduce communication in distributed deep learning."** *In Advances in neural information processing systems*, pp. 1509-1519. 2017.

- DQGD

Khirirat, Sarit, Hamid Reza Feyzmahdavian, and Mikael Johansson. **"Distributed learning with compressed gradients."** arXiv preprint arXiv:1806.06573 (2018).

Sublinear convergence rates even in the case when workers quantize full gradients

- DIANA

Mishchenko, Konstantin, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. **"Distributed learning with compressed gradient differences."** arXiv preprint arXiv:1901.09269 (2019).

Horváth, Samuel, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. **"Stochastic distributed learning with gradient quantization and variance reduction."** arXiv preprint arXiv:1904.05115 (2019).

Converges **linearly** when workers quantize full gradients

Parallel SGD with Biased Compressor Can Diverge at Exponential Rate

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

$$n = d = 3$$

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2$$

$$a = (-3, 2, 2)^\top \qquad b = (2, -3, 2)^\top \qquad c = (2, 2, -3)^\top$$

$$x^0 = (t, t, t)^\top$$

In this case Parallel SGD with Top1 compression operator satisfies

$$x^k = \left(1 + \frac{11\gamma}{6}\right)^k x^0$$

Parallel SGD with Biased Compressor Can Diverge at Exponential Rate

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

$$n = d = 3$$

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2 \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2$$

$$a = (-3, 2, 2)^\top \qquad b = (2, -3, 2)^\top \qquad c = (2, 2, -3)^\top$$

$$x^0 = (t, t, t)^\top$$

In this case Parallel SGD with Top1 compression operator satisfies

$$x^k = \left(1 + \frac{11\gamma}{6}\right)^k x^0$$

One can fix this using one special trick called ***error-compensation***

4. Error-Compensated SGD

Papers on EC-SGD

Seide, Frank, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. **"1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns."** *In Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

Stich, Sebastian U., Jean-Baptiste Cordonnier, and Martin Jaggi. **"Sparsified SGD with memory."** *In Advances in Neural Information Processing Systems*, pp. 4447-4458. 2018.

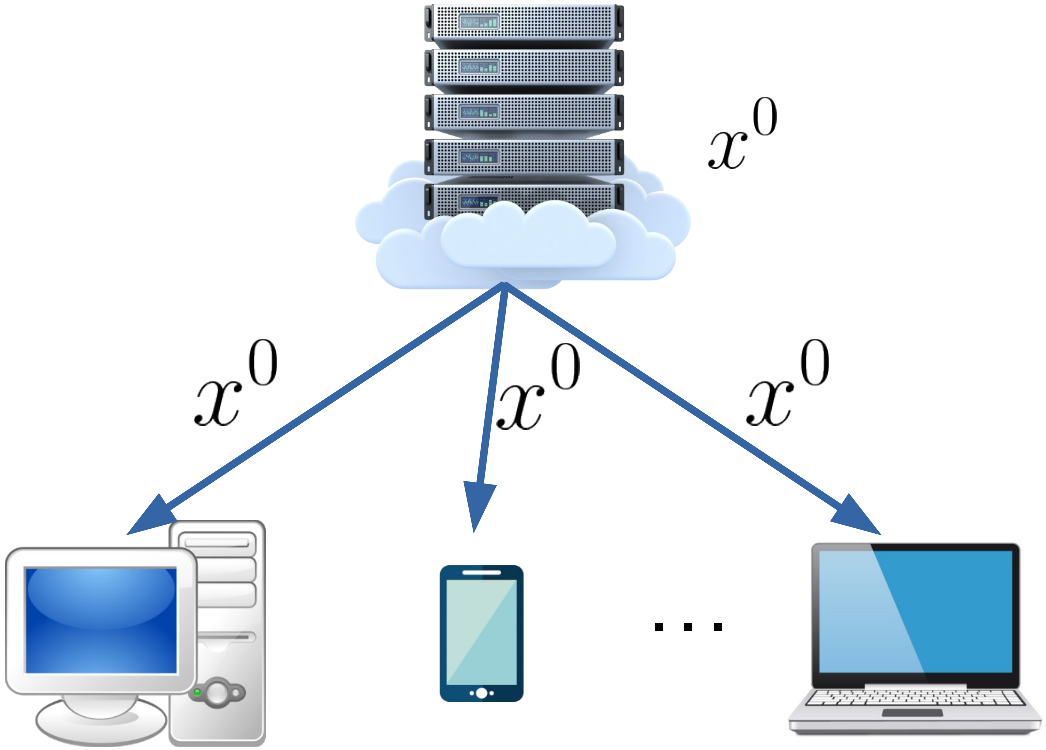
Karimireddy, Sai Praneeth, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. **"Error Feedback Fixes SignSGD and other Gradient Compression Schemes."** *In International Conference on Machine Learning*, pp. 3252-3261. 2019.

Stich, Sebastian U., and Sai Praneeth Karimireddy. **"The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication."** arXiv preprint arXiv:1909.05350 (2019).

Beznosikov, Aleksandr, Samuel Horváth, Peter Richtárik, and Mher Safaryan. **"On Biased Compression for Distributed Learning."** arXiv preprint arXiv:2002.12410 (2020).

1 Server broadcasts the parameters

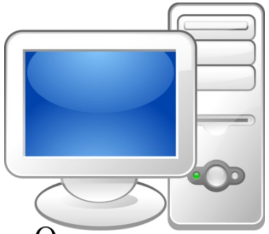
Step 1



Step 1

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel

x^0



$x^0 \rightarrow g_1^0$

$x^0 \rightarrow g_2^0$

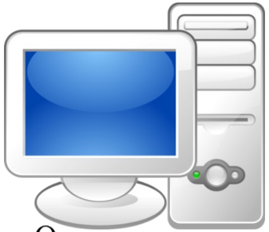
$\cdots$

$x^0 \rightarrow g_n^0$

Step 1

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression

x^0

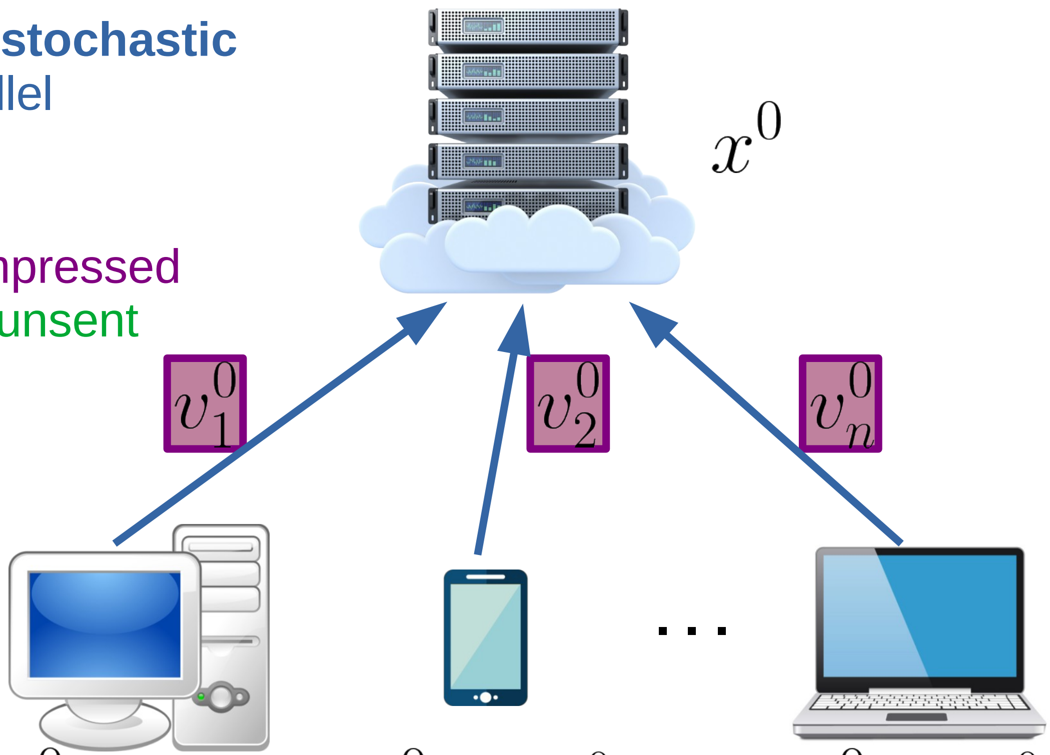


$x^0 \rightarrow g_1^0$ $\quad$ $x^0 \rightarrow g_2^0$ $\quad \dots \quad$ $x^0 \rightarrow g_n^0$

$$\boxed{v_1^0 = \mathcal{C}(\gamma g_1^0)} \quad \boxed{v_2^0 = \mathcal{C}(\gamma g_2^0)} \quad \boxed{v_n^0 = \mathcal{C}(\gamma g_n^0)}$$

Step 1

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and save unsent information



$x^0 \rightarrow g_1^0$ $\quad$ $x^0 \rightarrow g_2^0$ $\quad$ $x^0 \rightarrow g_n^0$

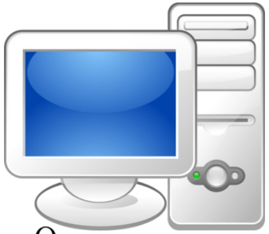
$v_1^0 = \mathcal{C}(\gamma g_1^0)$ $\quad$ $v_2^0 = \mathcal{C}(\gamma g_2^0)$ $\quad$ $v_n^0 = \mathcal{C}(\gamma g_n^0)$

$\boxed{e_1^1 = \gamma g_1^0 - v_1^0}$ $\quad$ $\boxed{e_2^1 = \gamma g_2^0 - v_2^0}$ $\quad$ $\boxed{e_n^1 = \gamma g_n^0 - v_n^0}$

Step 1

1. Server broadcasts the parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and save unsent information
5. Server gathers the information and updates the parameters

$$x^0 \rightarrow x^1 = x^0 - \frac{1}{n}\sum_{i=1}^{n} v_i^0$$



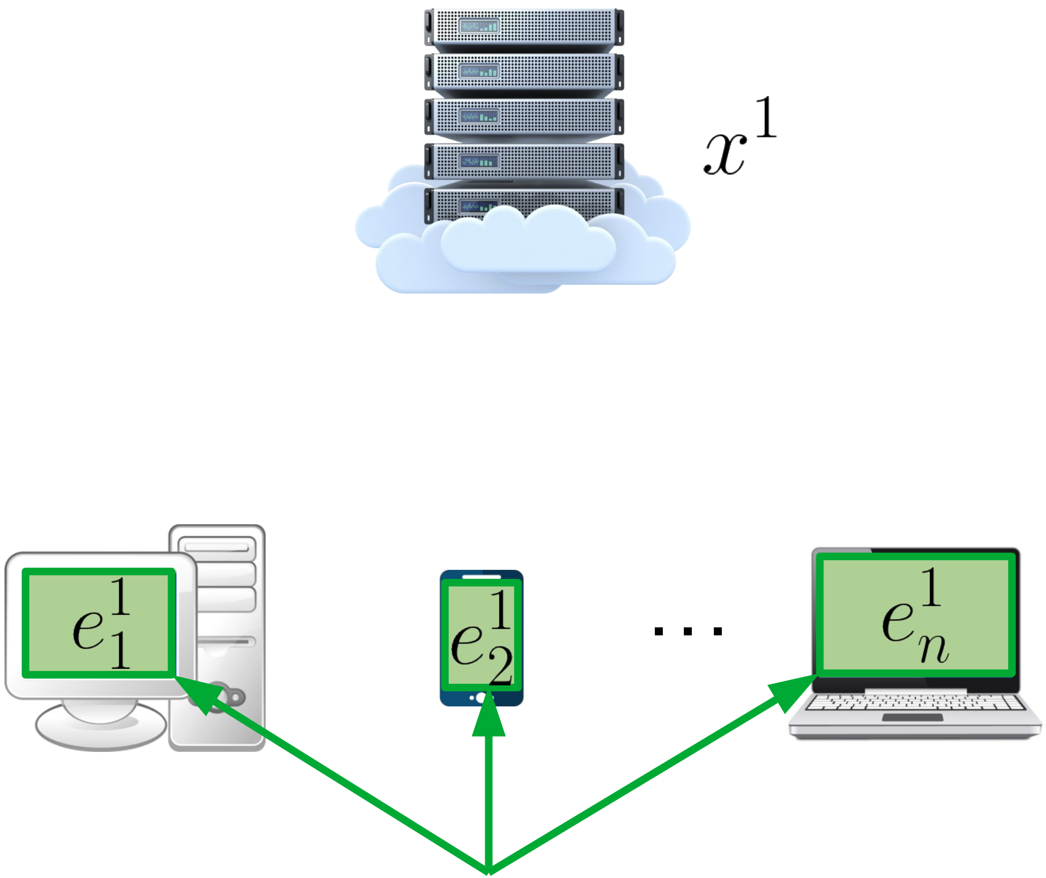
$\dots$

$$x^0 \rightarrow g_1^0 \qquad x^0 \rightarrow g_2^0 \qquad x^0 \rightarrow g_n^0$$

$$v_1^0 = \mathcal{C}(\gamma g_1^0) \quad v_2^0 = \mathcal{C}(\gamma g_2^0) \quad v_n^0 = \mathcal{C}(\gamma g_n^0)$$

$$e_1^1 = \gamma g_1^0 - v_1^0 \quad e_2^1 = \gamma g_2^0 - v_2^0 \quad e_n^1 = \gamma g_n^0 - v_n^0$$

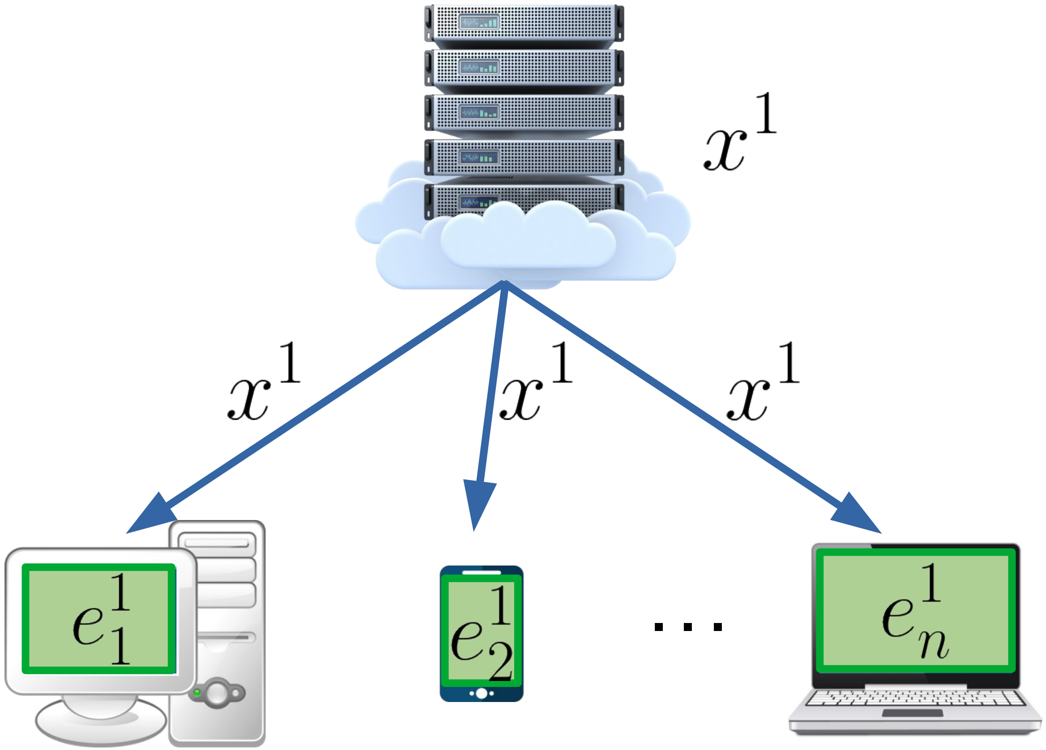
Step 1



devices keep these vectors for the next iterations
to *partially* send them later

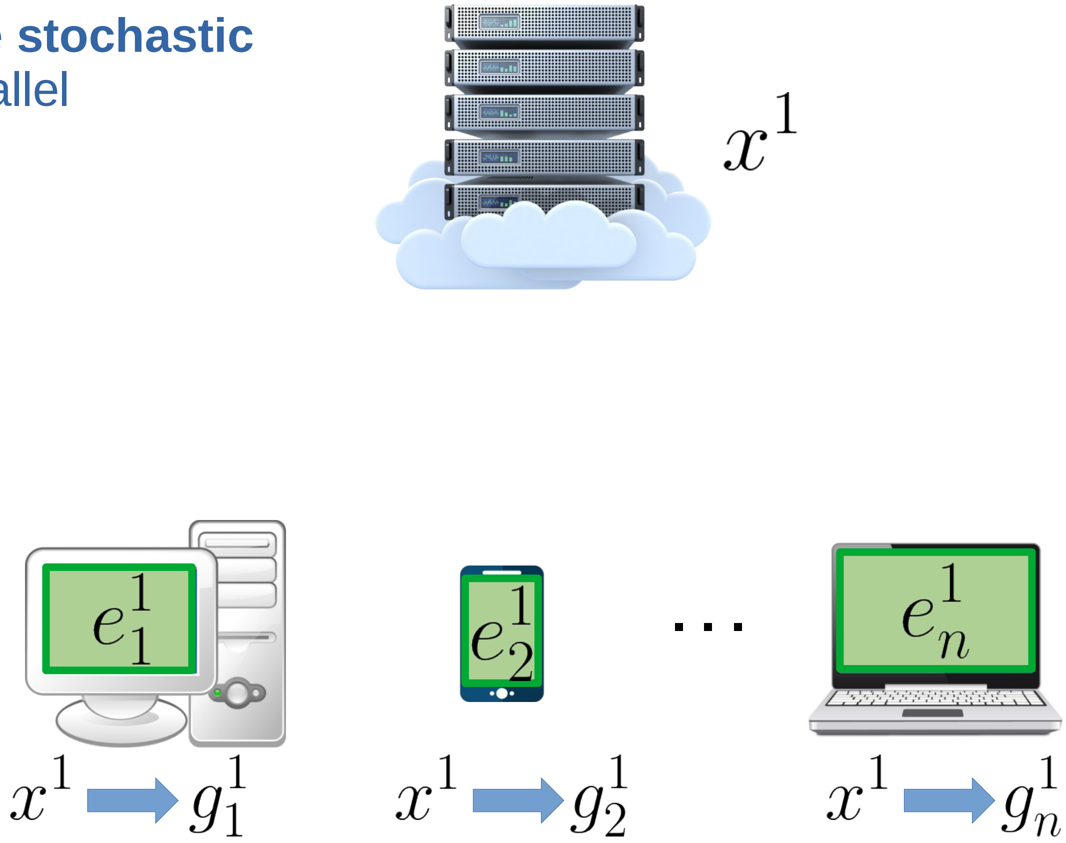
Step 2

1 Server broadcasts new parameters



Step 2

1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel



Step 2

1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression

x^1

$\cdots$

$x^1 \rightarrow g_1^1$ $\quad$ $x^1 \rightarrow g_2^1$ $\quad$ $x^1 \rightarrow g_n^1$

$\boxed{v_1^1 = \mathcal{C}(e_1^1 + \gamma g_1^1)}$ $\boxed{v_2^1 = \mathcal{C}(e_2^1 + \gamma g_2^1)}$ $\boxed{v_n^1 = \mathcal{C}(e_n^1 + \gamma g_n^1)}$

Step 2

1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression

x^1

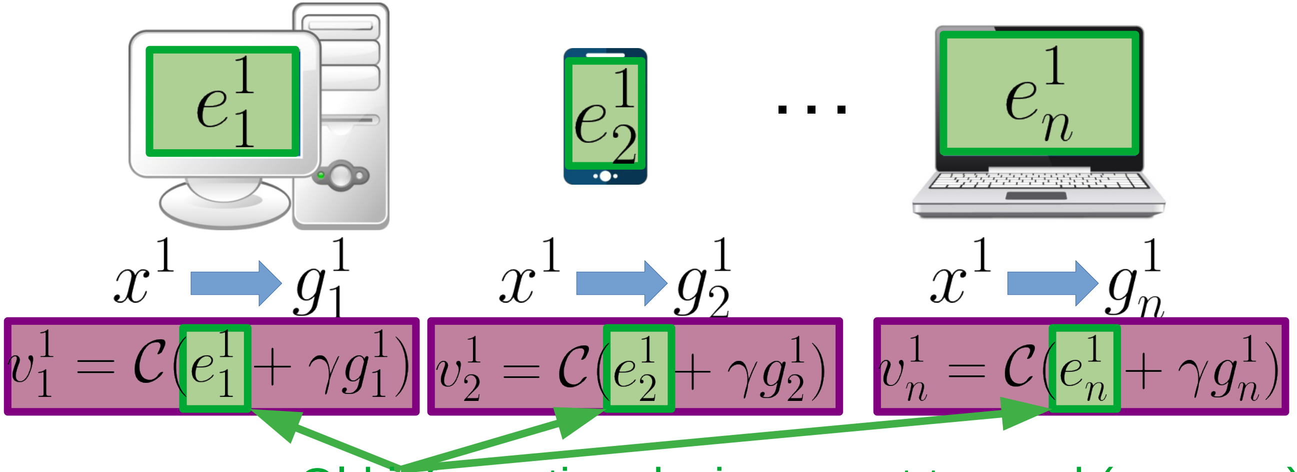
$x^1 \rightarrow g_1^1$ $\quad$ $x^1 \rightarrow g_2^1$ $\quad$ $x^1 \rightarrow g_n^1$

$\boxed{v_1^1 = \mathcal{C}(e_1^1 + \gamma g_1^1)}$ $\boxed{v_2^1 = \mathcal{C}(e_2^1 + \gamma g_2^1)}$ $\boxed{v_n^1 = \mathcal{C}(e_n^1 + \gamma g_n^1)}$

New information devices want to send

Step 2

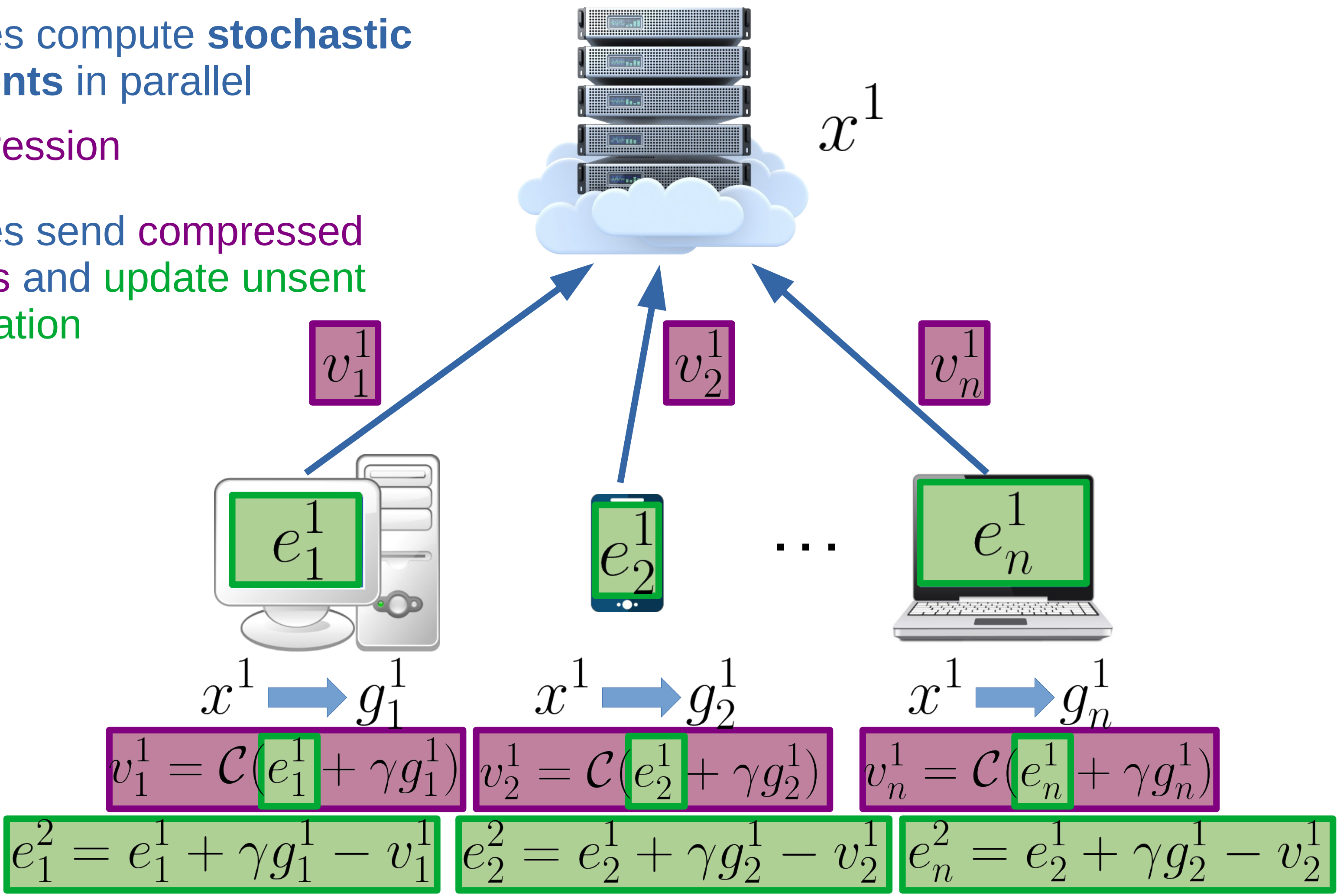
1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression



Old information devices want to send (memory)

Step 2

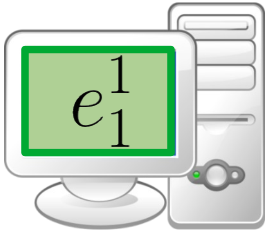
1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information



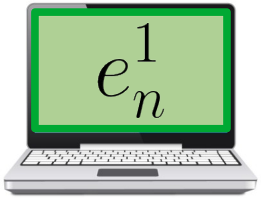
Step 2

1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information
5. Server gathers the information and updates the parameters

$$x^1 \rightarrow x^2 = x^1 - \frac{1}{n}\sum_{i=1}^{n} v_i^1$$



$\cdots$



$x^1 \rightarrow g_1^1$ $\quad$ $x^1 \rightarrow g_2^1$ $\quad$ $x^1 \rightarrow g_n^1$

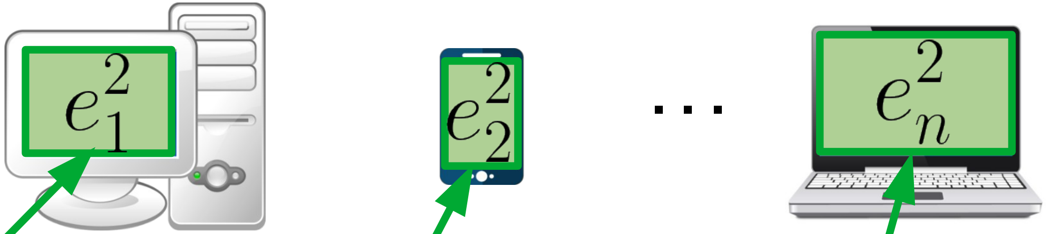
$v_1^1 = \mathcal{C}(e_1^1 + \gamma g_1^1)$ $\quad$ $v_2^1 = \mathcal{C}(e_2^1 + \gamma g_2^1)$ $\quad$ $v_n^1 = \mathcal{C}(e_n^1 + \gamma g_n^1)$

$e_1^2 = e_1^1 + \gamma g_1^1 - v_1^1$ $\quad$ $e_2^2 = e_2^1 + \gamma g_2^1 - v_2^1$ $\quad$ $e_n^2 = e_2^1 + \gamma g_2^1 - v_2^1$

Step 2

1. Server broadcasts new parameters
2. Devices compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information
5. Server gathers the information and updates the parameters
6. Devices update their memory

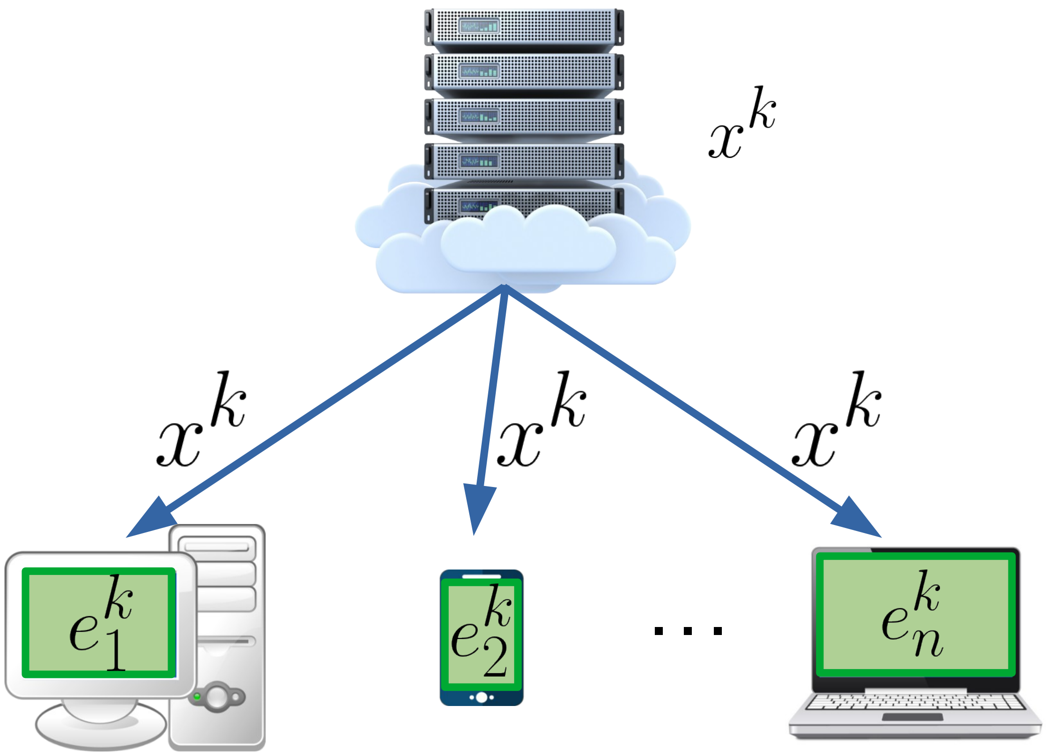
$$x^1 \rightarrow x^2 = x^1 - \frac{1}{n}\sum_{i=1}^{n} v_i^1$$



$$\boxed{e_1^2 = e_1^1 + \gamma g_1^1 - v_1^1} \quad \boxed{e_2^2 = e_2^1 + \gamma g_2^1 - v_2^1} \quad \boxed{e_n^2 = e_2^1 + \gamma g_2^1 - v_2^1}$$

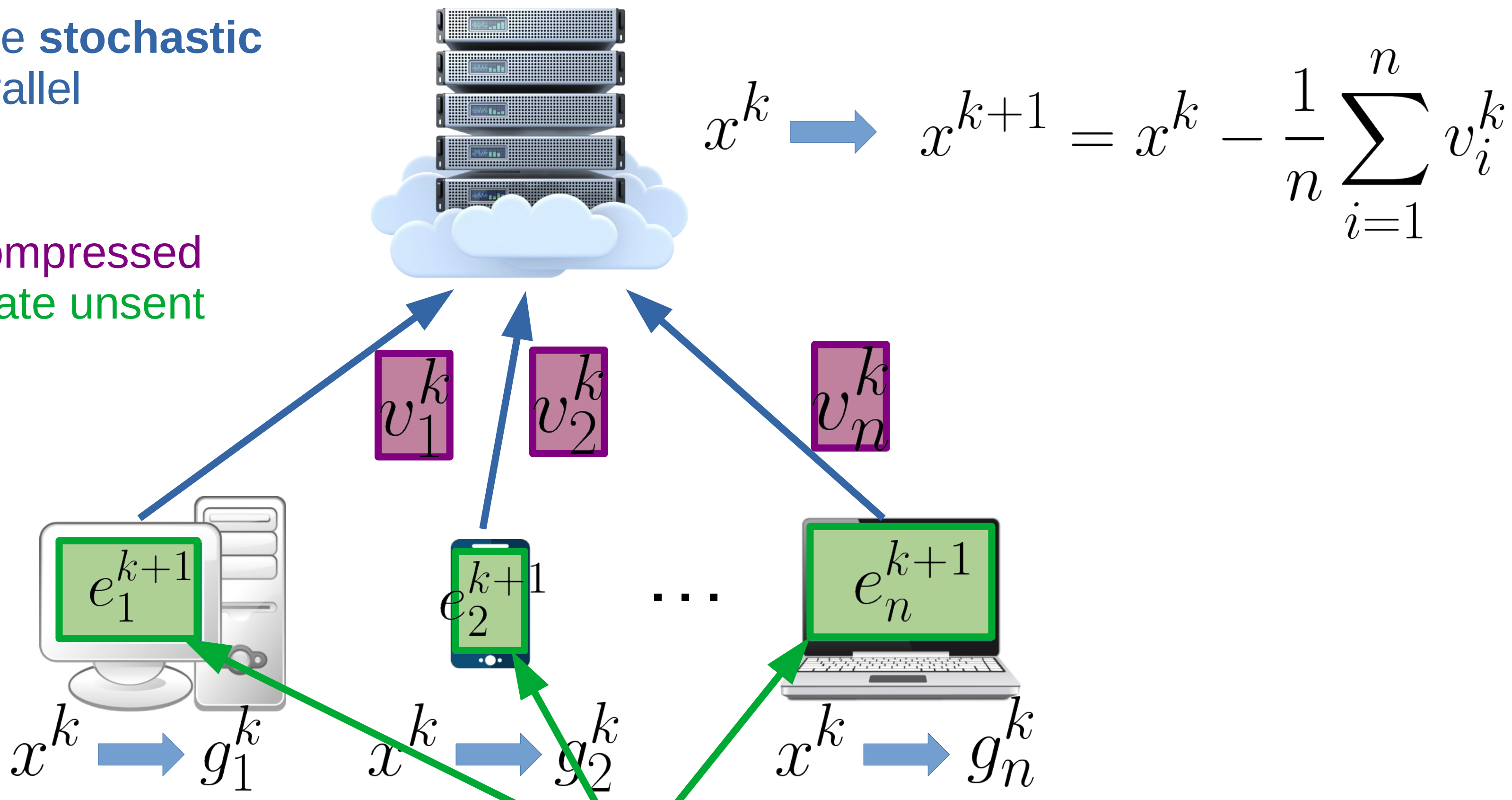
Step k+1

1 Server broadcasts new parameters



Step k+1

1. Server broadcasts new parameters
2. Workers compute **stochastic gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information
5. Server gathers the information and updates the parameters
6. Repeat steps 1 – 5



$$v_i^k = \mathcal{C}\left(e_i^k + \gamma g_i^k\right)$$

$$e_i^{k+1} = e_i^k + \gamma g_i^k - v_i^k$$

Error-Compensated SGD

✔ Converges even with biased compression operators

EC-SGD finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Hides logarithmical factors

$$\widetilde{\mathcal{O}}\left(\frac{L}{\boxed{\delta}\mu} + \frac{\boxed{\sigma^2}}{n\mu\varepsilon} + \frac{\sqrt{L(\boxed{\sigma^2} + \boxed{\zeta_*^2}/\boxed{\delta})}}{\mu\sqrt{\boxed{\delta}\varepsilon}}\right)$$

iterations

$$\boxed{\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2}$$

$$\boxed{\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x^k\right)\right\|^2 \mid x^k\right] \leq \sigma^2}$$

$$\boxed{\zeta_*^2 = \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x^*)\|^2}$$

Error-Compensated SGD

✓ Converges even with biased compression operators

✗ Fails to converge with **linear rate** even when workers compute full gradients

EC-SGD finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Hides logarithmical factors

$$\widetilde{\mathcal{O}}\left(\frac{L}{\delta\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L(\sigma^2 + \boxed{\zeta_*^2}/\delta)}}{\mu\sqrt{\delta\varepsilon}}\right)$$ iterations

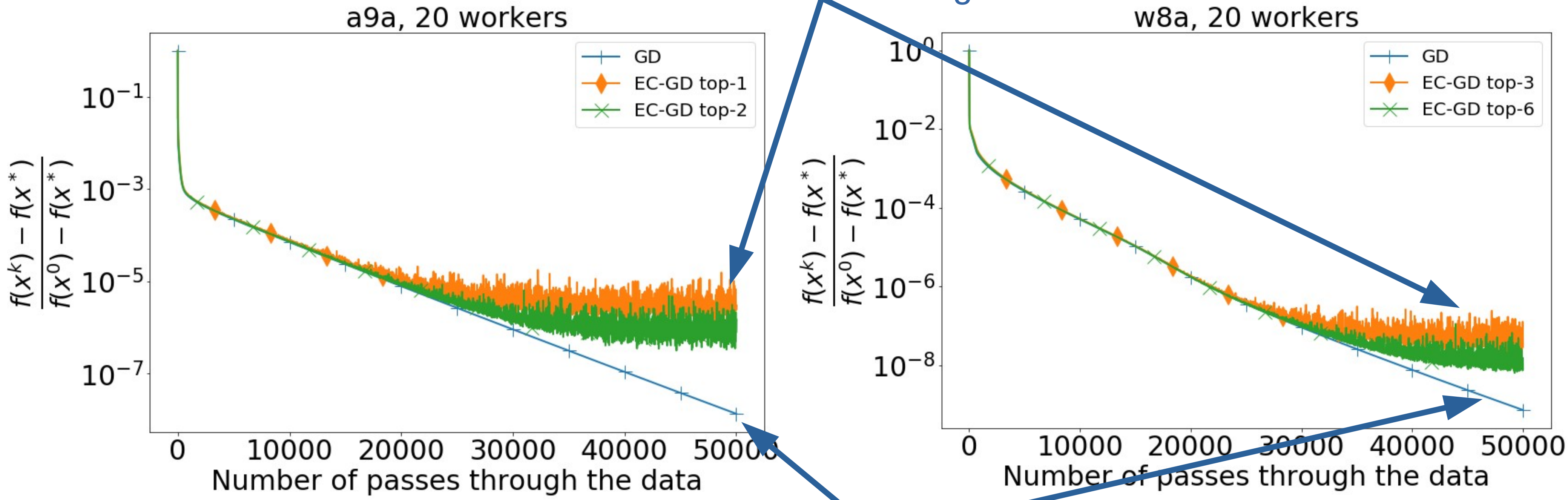
$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2 \qquad \mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x^k\right)\right\|^2 \mid x^k\right] \leq \sigma^2$$

$$\boxed{\zeta_*^2 = \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x^*)\|^2}$$

EC-GD and Logistic Regression

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \cdot (Ax)_i\right)\right) + \frac{\mu}{2}\|x\|^2 \right\}$$

sub-linear convergence



linear convergence

Error-Compensated SGD

✓ Converges even with biased compression operators

✗ Fails to converge with **linear rate** even when workers compute full gradients

Questions:

1. *Is it possible to design **linearly converging** SGD with error compensation when workers compute full gradients, i.e., linearly converging **EC-GD**?*

2. *Is it possible to design **linearly converging SGD** with error compensation when **the local loss functions have a finite-sum form**?*

The answer is *Yes* for both questions

5.1. New method: EC-GDstar

Error-Compensated GD

EC-GD finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Hides logarithmical factors

$$\widetilde{\mathcal{O}}\left(\frac{L}{\delta\mu} + \frac{\sqrt{L\boxed{\zeta_*^2}}}{\mu\delta\sqrt{\varepsilon}}\right)$$ iterations

$$\boxed{\zeta_*^2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x^*)\|^2}$$

Error-Compensated GD

EC-GD finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Hides logarithmical factors

$$\widetilde{\mathcal{O}}\left(\frac{L}{\delta\mu} + \frac{\sqrt{L\boxed{\zeta_*^2}}}{\mu\delta\sqrt{\varepsilon}}\right)$$

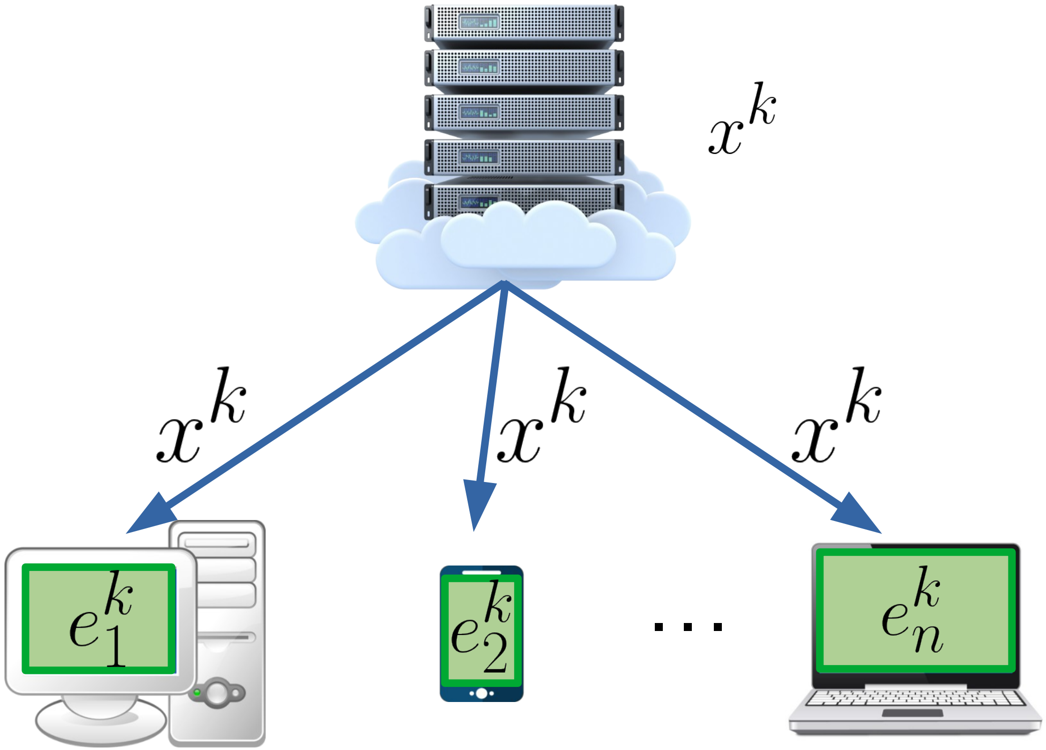
iterations

$$\boxed{\zeta_*^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*)\|^2}$$

What if devices know these vectors from the beginning?

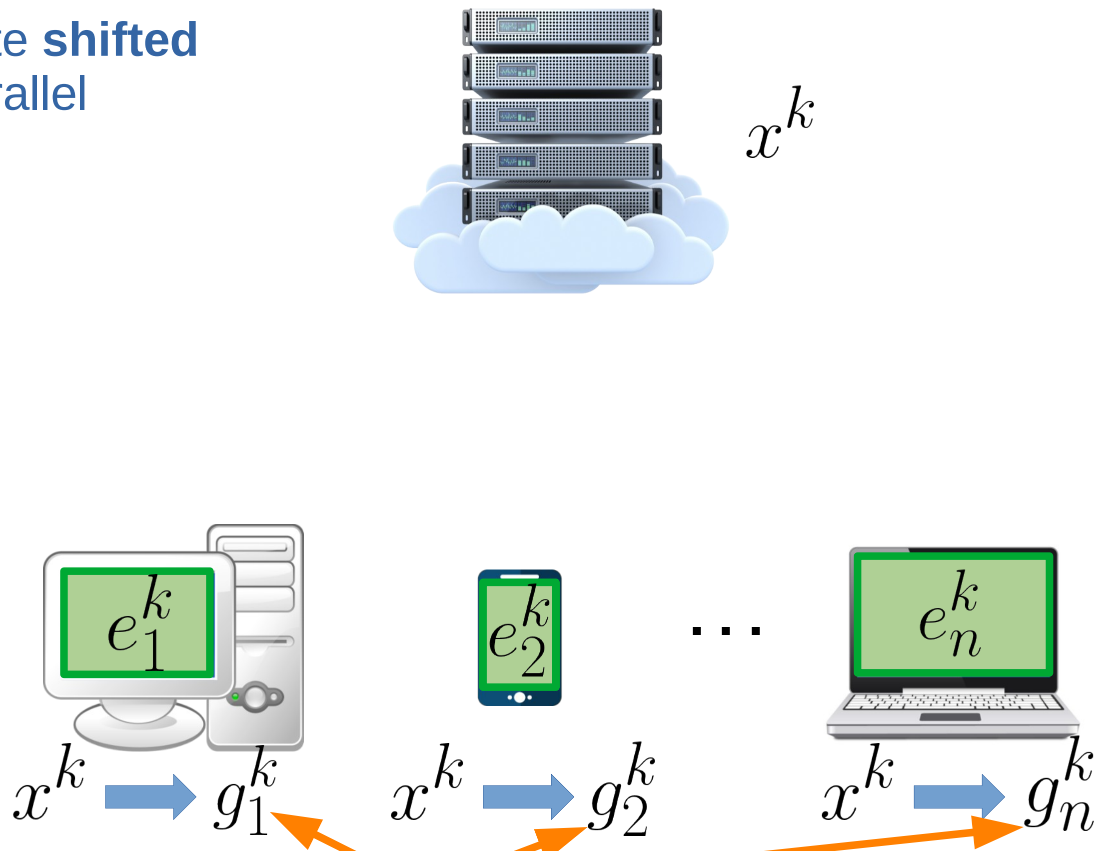
EC-GDstar

1 Server broadcasts new parameters



EC-GDstar

1. Server broadcasts new parameters
2. Workers compute **shifted gradients** in parallel

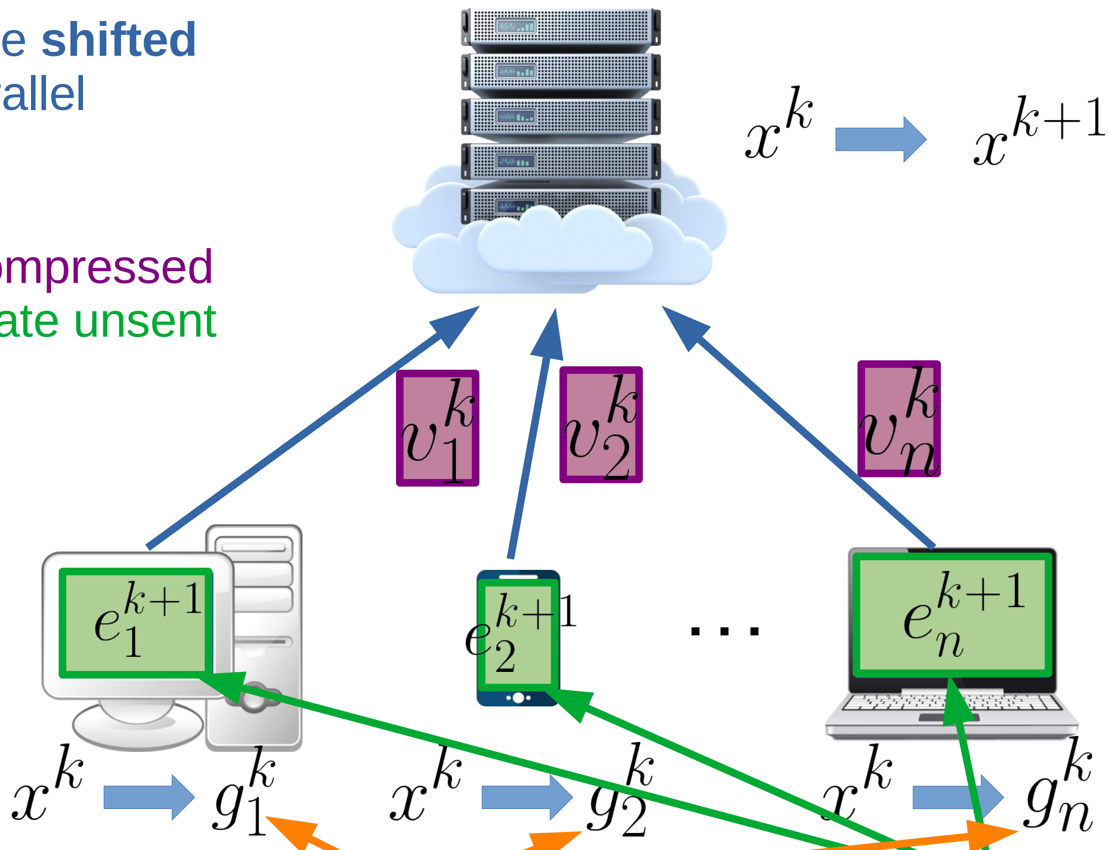


$$\boxed{g_i^k = \nabla f_i(x^k) - \nabla f_i(x^*)}$$

EC-GDstar

1. Server broadcasts new parameters
2. Workers compute **shifted gradients** in parallel
3. Compression
4. Devices send compressed vectors and update unsent information
5. Server gathers the information and updates the parameters
6. Repeat steps 1 – 5

$$x^k \rightarrow x^{k+1} = x^k - \frac{1}{n}\sum_{i=1}^{n} v_i^k$$



$$v_i^k = \mathcal{C}\left(e_i^k + \gamma g_i^k\right)$$

$$g_i^k = \nabla f_i(x^k) - \nabla f_i(x^*)$$

$$e_i^{k+1} = e_i^k + \gamma g_i^k - v_i^k$$

EC-GDstar: Rate of Convergence

EC-GDstar finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

$$\mathcal{O}\left(\frac{L}{\delta\mu} \ln \frac{1}{\varepsilon}\right)$$ iterations

✔ Linear rate

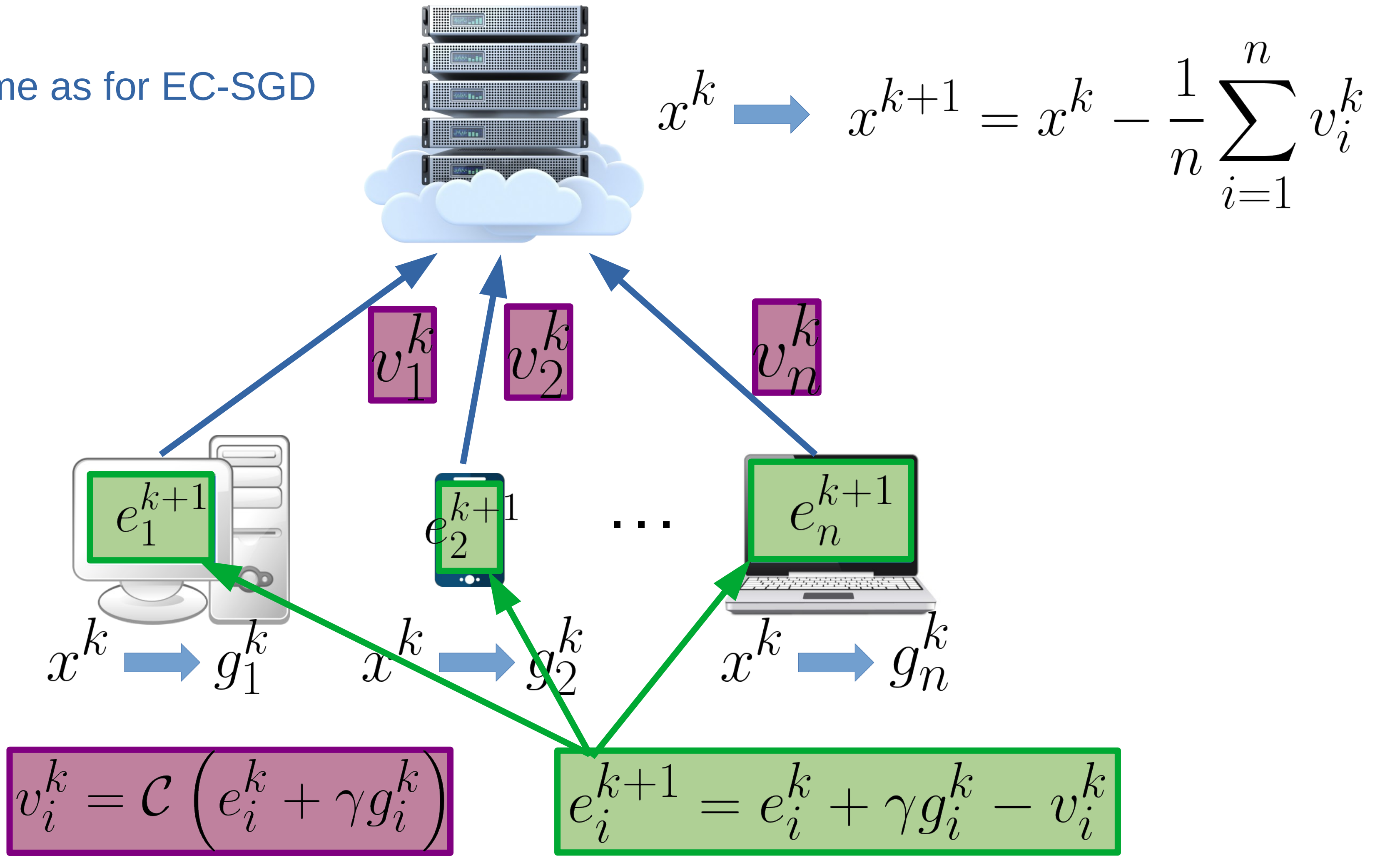
✖ The method is impractical: it uses the gradients at the solution

Can we develop a practical analog?

5.2. New method: EC-SGD-DIANA

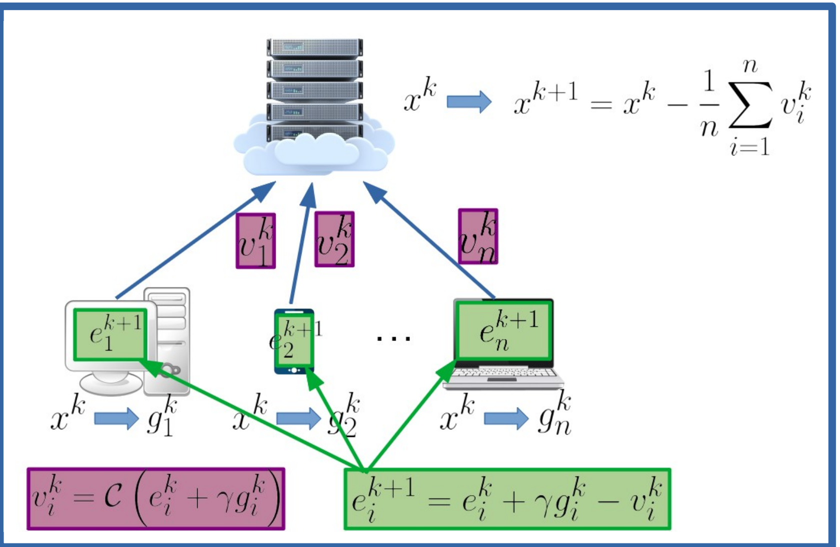
EC-SGD-DIANA

The same scheme as for EC-SGD



EC-SGD-DIANA

$$g_i^k = \hat{g}_i^k - h_i^k + h^k$$



EC-SGD-DIANA

The key insight why do we need $\{h_i^k\}_{i=1}^n$:

it reduces the variance coming from compressions via learning the gradients at the solution!

$$g_i^k = \boxed{\hat{g}_i^k} - h_i^k + h^k \longleftarrow h^k = \frac{1}{n}\sum_{i=1}^n h_i^k$$

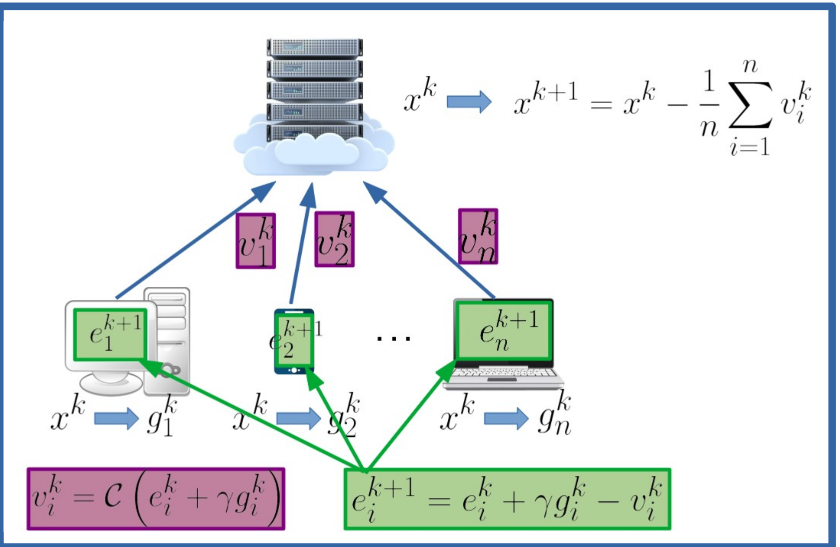
Server broadcasts this vector to the workers

$$\mathbb{E}\left[\boxed{\hat{g}_i^k} \mid x^k\right] = \nabla f_i(x^k)$$

stepsize

$$h_i^{k+1} = h_i^k + \alpha \boxed{\mathcal{Q}\left(\hat{g}_i^k - h_i^k\right)}$$

Workers send these vectors to the server



Works for both cases:

- $f_i(x) = \mathbf{E}_{\xi_i \sim \mathcal{D}_i}\left[f_{\xi_i}(x)\right]$ $\quad \boxed{\hat{g}_i^k} = \nabla f_{\xi_i}(x^k)$
- $f_i(x) = \frac{1}{m}\sum_{j=1}^m f_{ij}(x)$ $\quad \boxed{\hat{g}_i^k} = \nabla f_{il}(x^k)$

$l \sim [m]$ uniformly at random

EC-SGD-DIANA: Rate of Convergence

EC-SGD-DIANA finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Option I: $$\widetilde{\mathcal{O}}\left(\omega + \frac{L}{\delta\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$$

iterations

Hides logarithmical factors

Option II: $$\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{L}{\delta\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$$

EC-SGD-DIANA: Rate of Convergence

EC-S**GD**-DIANA finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

Option I: $$\widetilde{\mathcal{O}}\left(\omega + \frac{L}{\delta\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$$

Hides logarithmical factors

Option II: $$\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{L}{\delta\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$$

iterations

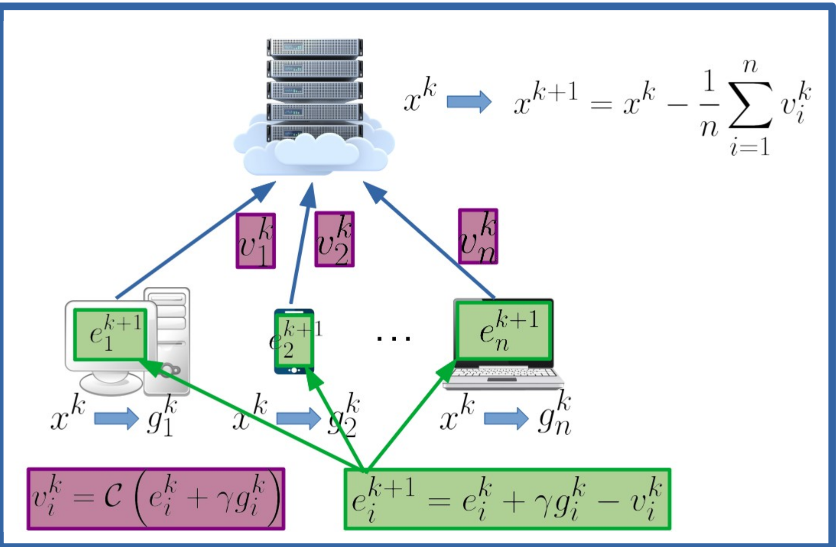
Moreover, if workers compute **full gradients**, then the rate of convergence is linear

$$\mathcal{O}\left(\left(\omega + \frac{L}{\delta\mu}\right)\log\frac{1}{\varepsilon}\right)$$

5.3. New method: EC-LSVRG-DIANA

EC-LSVRG-DIANA

$$g_i^k = \hat{g}_i^k - h_i^k + h^k$$



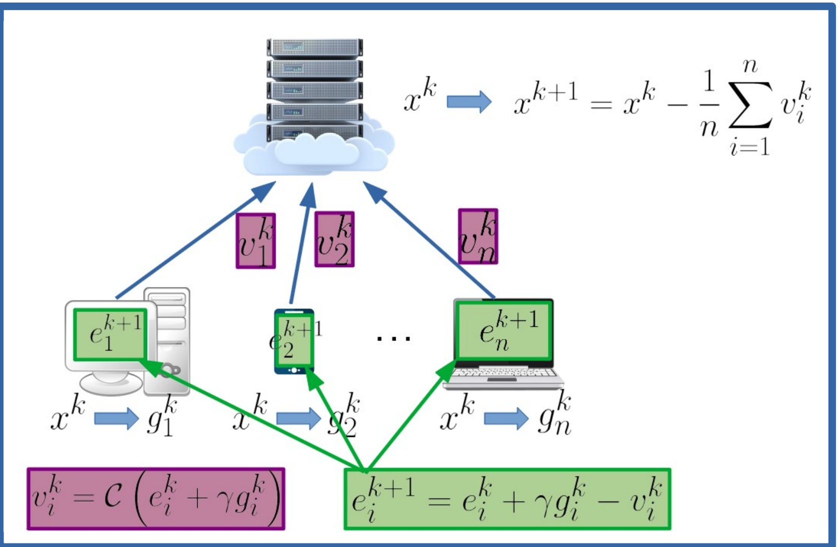
Works for the case:

- $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

EC-LSVRG-DIANA

$$g_i^k = \boxed{\hat{g}_i^k} - h_i^k + h^k \qquad \boxed{l} \sim [m] \text{ uniformly at random}$$

$$\boxed{\hat{g}_i^k} = \nabla f_{i\boxed{l}}\left(x^k\right) - \nabla f_{i\boxed{l}}\left(w_i^k\right) + \nabla f_i\left(w_i^k\right)$$



Works for the case:

- $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

EC-LSVRG-DIANA

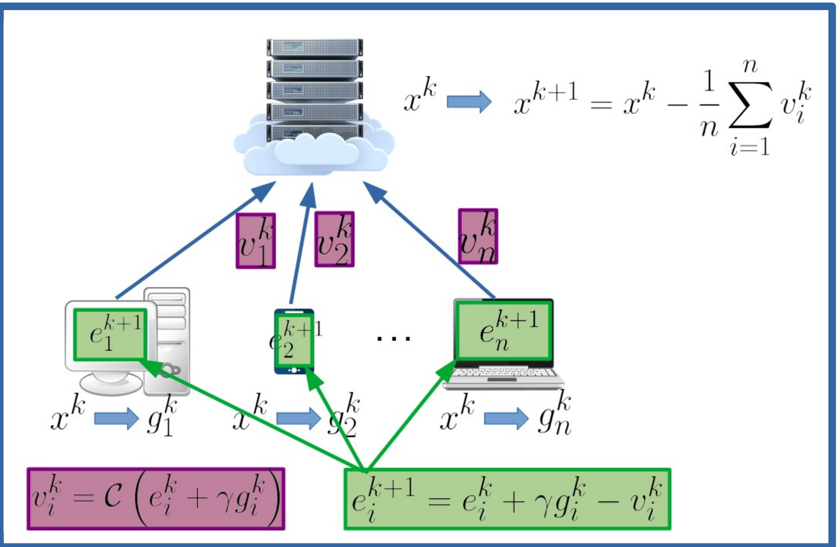
$$g_i^k = \boxed{\hat{g}_i^k} - h_i^k + h^k$$

$\boxed{l} \sim [m]$ uniformly at random

$$\boxed{\hat{g}_i^k} = \nabla f_{i\boxed{l}}\left(x^k\right) - \nabla f_{i\boxed{l}}\left(\boxed{w_i^k}\right) + \nabla f_i\left(\boxed{w_i^k}\right)$$

Reduction of the variance introduced due to the stochasticity of the gradients

$$\boxed{w_i^{k+1}} = \begin{cases} x^k, & \text{with probability } p \\ \boxed{w_i^k}, & \text{with probability } 1-p \end{cases}$$



Works for the case:

- $$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x)$$

EC-LSVRG-DIANA: Rate of Convergence

EC-LSVRG-DIANA finds such $\hat{x}$ that $\mathbb{E}[f(\hat{x})] - f(x^*) \leq \varepsilon$ after

$$\mathcal{O}\left(\left(\omega + m + \frac{L}{\delta\mu}\right)\log\frac{1}{\epsilon}\right)$$ iterations

6. Unified Convergence Analysis of Methods with Error Compensation

Key Assumption

$$g^k = \frac{1}{n}\sum_{i=1}^{n} g_i^k, \quad \mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right) \quad \bar{g}_i^k = \mathbb{E}\left[g_i^k \mid x^k\right]$$

$$\frac{1}{n}\sum_{i=1}^{n} \left\|\bar{g}_i^k\right\|^2 \leq 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1\sigma_{1,k}^2 + B_2\sigma_{2,k}^2 + D_1$$

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\left\|g_i^k - \bar{g}_i^k\right\|^2 \mid x^k\right] \leq 2\widetilde{A}\left(f\left(x^k\right) - f\left(x^*\right)\right) + \widetilde{B}_1\sigma_{1,k}^2 + \widetilde{B}_2\sigma_{2,k}^2 + \widetilde{D}_1$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \leq 2A'\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1'\sigma_{1,k}^2 + B_2'\sigma_{2,k}^2 + D_1'$$

$$\mathbb{E}\left[\sigma_{1,k+1}^2 \mid \sigma_{1,k}^2, \sigma_{2,k}^2\right] \leq \left(1 - \rho_1\right)\sigma_{1,k}^2 + 2C_1\left(f\left(x^k\right) - f\left(x^*\right)\right) + G\rho_1\sigma_{2,k}^2 + D_2$$

$$\mathbb{E}\left[\sigma_{2,k+1}^2 \mid \sigma_{2,k}^2\right] \leq \left(1 - \rho_2\right)\sigma_{2,k}^2 + 2C_2\left(f\left(x^k\right) - f\left(x^*\right)\right)$$

Key Assumption

$$g^k = \frac{1}{n}\sum_{i=1}^n g_i^k, \quad \mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right) \quad \bar{g}_i^k = \mathbb{E}\left[g_i^k \mid x^k\right]$$

$$\frac{1}{n}\sum_{i=1}^n \left\|\bar{g}_i^k\right\|^2 \leq 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1\sigma_{1,k}^2 + B_2\sigma_{2,k}^2 + D_1$$

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|g_i^k - \bar{g}_i^k\right\|^2 \mid x^k\right] \leq 2\widetilde{A}\left(f\left(x^k\right) - f\left(x^*\right)\right) + \widetilde{B}_1\sigma_{1,k}^2 + \widetilde{B}_2\sigma_{2,k}^2 + \widetilde{D}_1$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \leq 2A'\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1'\sigma_{1,k}^2 + B_2'\sigma_{2,k}^2 + D_1'$$

$$\mathbb{E}\left[\sigma_{1,k+1}^2 \mid \sigma_{1,k}^2, \sigma_{2,k}^2\right] \leq \left(1-\rho_1\right)\sigma_{1,k}^2 + 2C_1\left(f\left(x^k\right) - f\left(x^*\right)\right) + G\rho_1\sigma_{2,k}^2 + D_2$$

$$\mathbb{E}\left[\sigma_{2,k+1}^2 \mid \sigma_{2,k}^2\right] \leq \left(1-\rho_2\right)\sigma_{2,k}^2 + 2C_2\left(f\left(x^k\right) - f\left(x^*\right)\right)$$

Reflects smoothness properties of the problem and noises introduced by compressions and stochastic gradients

Key Assumption

$$g^k = \frac{1}{n}\sum_{i=1}^n g_i^k, \quad \mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right) \quad \bar{g}_i^k = \mathbb{E}\left[g_i^k \mid x^k\right]$$

$$\frac{1}{n}\sum_{i=1}^n \left\|\bar{g}_i^k\right\|^2 \leq 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1\sigma_{1,k}^2 + B_2\sigma_{2,k}^2 + D_1$$

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|g_i^k - \bar{g}_i^k\right\|^2 \mid x^k\right] \leq 2\widetilde{A}\left(f\left(x^k\right) - f\left(x^*\right)\right) + \widetilde{B}_1\sigma_{1,k}^2 + \widetilde{B}_2\sigma_{2,k}^2 + \widetilde{D}_1$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \leq 2A'\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1'\sigma_{1,k}^2 + B_2'\sigma_{2,k}^2 + D_1'$$

$$\mathbb{E}\left[\sigma_{1,k+1}^2 \mid \sigma_{1,k}^2, \sigma_{2,k}^2\right] \leq \left(1-\rho_1\right)\sigma_{1,k}^2 + 2C_1\left(f\left(x^k\right) - f\left(x^*\right)\right) + G\rho_1\sigma_{2,k}^2 + D_2$$

$$\mathbb{E}\left[\sigma_{2,k+1}^2 \mid \sigma_{2,k}^2\right] \leq \left(1-\rho_2\right)\sigma_{2,k}^2 + 2C_2\left(f\left(x^k\right) - f\left(x^*\right)\right)$$

Reflects smoothness properties of the problem and noises introduced by compressions and stochastic gradients

Describes the process of variance reduction of the variance coming from compressions

Key Assumption

$$g^k = \frac{1}{n}\sum_{i=1}^n g_i^k, \quad \mathbb{E}\left[g^k \mid x^k\right] = \nabla f\left(x^k\right) \quad \bar{g}_i^k = \mathbb{E}\left[g_i^k \mid x^k\right]$$

$$\frac{1}{n}\sum_{i=1}^n \left\|\bar{g}_i^k\right\|^2 \le 2A\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1\sigma_{1,k}^2 + B_2\sigma_{2,k}^2 + D_1$$

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|g_i^k - \bar{g}_i^k\right\|^2 \mid x^k\right] \le 2\widetilde{A}\left(f\left(x^k\right) - f\left(x^*\right)\right) + \widetilde{B}_1\sigma_{1,k}^2 + \widetilde{B}_2\sigma_{2,k}^2 + \widetilde{D}_1$$

$$\mathbb{E}\left[\left\|g^k\right\|^2 \mid x^k\right] \le 2A'\left(f\left(x^k\right) - f\left(x^*\right)\right) + B_1'\sigma_{1,k}^2 + B_2'\sigma_{2,k}^2 + D_1'$$

$$\mathbb{E}\left[\sigma_{1,k+1}^2 \mid \sigma_{1,k}^2, \sigma_{2,k}^2\right] \le \left(1-\rho_1\right)\sigma_{1,k}^2 + 2C_1\left(f\left(x^k\right) - f\left(x^*\right)\right) + G\rho_1\sigma_{2,k}^2 + D_2$$

$$\mathbb{E}\left[\sigma_{2,k+1}^2 \mid \sigma_{2,k}^2\right] \le \left(1-\rho_2\right)\sigma_{2,k}^2 + 2C_2\left(f\left(x^k\right) - f\left(x^*\right)\right)$$

Reflects smoothness properties of the problem and noises introduced by compressions and stochastic gradients

Describes the process of variance reduction of the variance coming from compressions

Describes the process of variance reduction of the variance coming from stochastic gradients

Main Theorem

Some quantity depending only on the
starting point and stepsize

$$\mathbb{E}\left[f\left(\bar{x}^K\right) - f\left(x^*\right)\right] \leq (1-\eta)^K \frac{\Psi\left(x^0, \gamma\right)}{\gamma} + \gamma \Phi\left(D_1, \widetilde{D}_1, D_1', D_2\right)$$

Linear function

$$\eta = \min\left\{\frac{\gamma\mu}{2}, \frac{\rho_1}{4}, \frac{\rho_2}{4}\right\}$$

Methods with Error Compensation Covered by Our Framework

Problem	Method	Alg #	Citation	Sec #	Rate (constants ignored)
(1)+(3)	EC-SGDsr	Alg 3	new	H.1	$\widetilde{\mathcal{O}}\left(\frac{\mathcal{L}}{\mu}+\frac{\sqrt{L\mathcal{L}}}{\delta\mu}+\frac{\mathrm{Var}}{n\mu\varepsilon}+\frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-SGD	Alg 4	[45]	H.2	$\widetilde{\mathcal{O}}\left(\frac{\kappa}{\delta}+\frac{\mathrm{Var}}{n\mu\varepsilon}+\frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-GDstar	Alg 5	new	H.3	$\mathcal{O}\left(\frac{\kappa}{\delta}\log\frac{1}{\varepsilon}\right)$
(1)+(2)	EC-SGD-DIANA	Alg 6	new	H.4	Option I: $\widetilde{\mathcal{O}}\left(\omega+\frac{\kappa}{\delta}+\frac{\sigma^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta}+\frac{\kappa}{\delta}+\frac{\sigma^2}{n\mu\varepsilon}+\frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(3)	EC-SGDsr-DIANA	Alg 7	new	H.5	Option I: $\widetilde{\mathcal{O}}\left(\omega+\frac{\mathcal{L}}{\mu}+\frac{\sqrt{L\mathcal{L}}}{\delta\mu}+\frac{\mathrm{Var}}{n\mu\varepsilon}+\frac{\sqrt{L\mathrm{Var}}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta}+\frac{\mathcal{L}}{\mu}+\frac{\sqrt{L\mathcal{L}}}{\delta\mu}+\frac{\mathrm{Var}}{n\mu\varepsilon}+\frac{\sqrt{L\mathrm{Var}}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-GD-DIANA[†]	Alg 6	new	H.4	$\mathcal{O}\left(\left(\omega+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG	Alg 8	new	H.6	$\widetilde{\mathcal{O}}\left(m+\frac{\kappa}{\delta}+\frac{\sqrt{L\zeta_*^2}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-LSVRGstar	Alg 9	new	H.7	$\mathcal{O}\left(\left(m+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG-DIANA	Alg 10	new	H.8	$\mathcal{O}\left(\left(\omega+m+\frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$

Methods with Error Compensation Covered by Our Framework

Problem	Method	Alg #	Citation	Sec #	Rate (constants ignored)
(1)+(3)	EC-SGDsr	Alg 3	new	H.1	$\widetilde{\mathcal{O}}\left(\frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-SGD	Alg 4	[45]	H.2	$\widetilde{\mathcal{O}}\left(\frac{\kappa}{\delta} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-GDstar	Alg 5	new	H.3	$\mathcal{O}\left(\frac{\kappa}{\delta}\log\frac{1}{\varepsilon}\right)$
(1)+(2)	EC-SGD-DIANA	Alg 6	new	H.4	Option I: $\widetilde{\mathcal{O}}\left(\omega + \frac{\kappa}{\delta} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{\kappa}{\delta} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(3)	EC-SGDsr-DIANA	Alg 7	new	H.5	Option I: $\widetilde{\mathcal{O}}\left(\omega + \frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L\mathrm{Var}}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L\mathrm{Var}}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-GD-DIANA†	Alg 6	new	H.4	$\mathcal{O}\left(\left(\omega + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG	Alg 8	new	H.6	$\widetilde{\mathcal{O}}\left(m + \frac{\kappa}{\delta} + \frac{\sqrt{L\zeta_*^2}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-LSVRGstar	Alg 9	new	H.7	$\mathcal{O}\left(\left(m + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG-DIANA	Alg 10	new	H.8	$\mathcal{O}\left(\left(\omega + m + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$

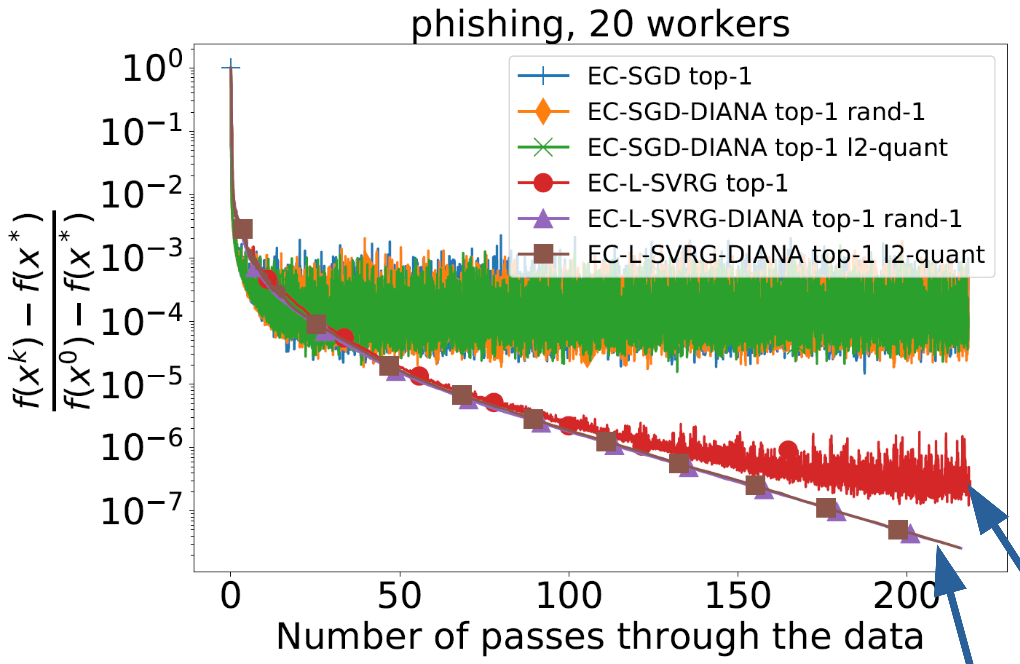
Methods with Error Compensation Covered by Our Framework

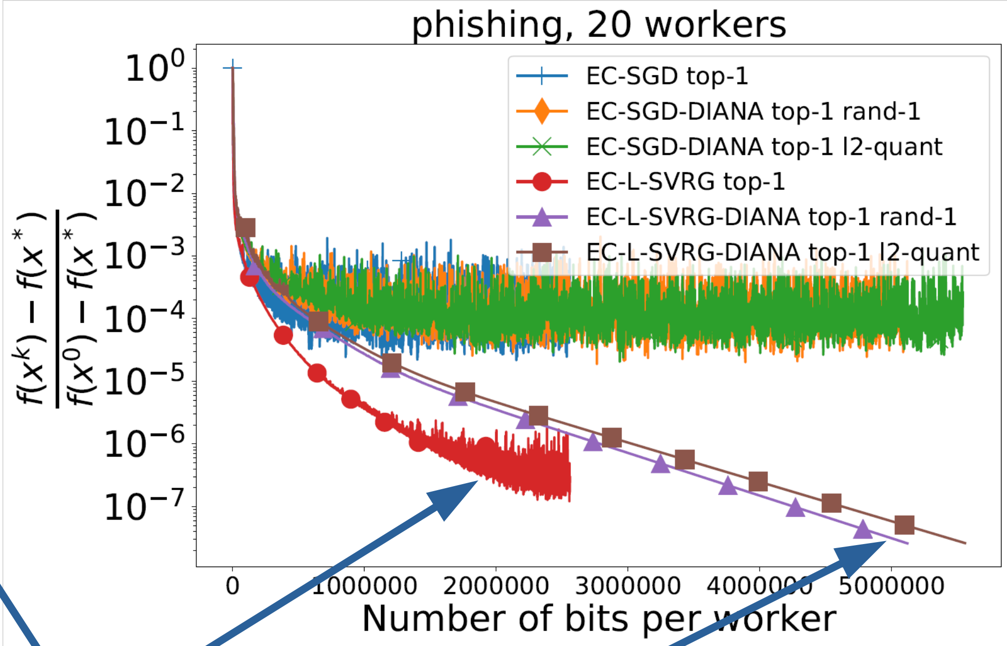
Problem	Method	Alg #	Citation	Sec #	Rate (constants ignored)
(1)+(3)	EC-SGDsr	Alg 3	new	H.1	$\widetilde{\mathcal{O}}\left(\frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-SGD	Alg 4	[45]	H.2	$\widetilde{\mathcal{O}}\left(\frac{\kappa}{\delta} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L(\mathrm{Var}+\zeta_*^2/\delta)}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-GDstar	Alg 5	new	H.3	$\mathcal{O}\left(\frac{\kappa}{\delta}\log\frac{1}{\varepsilon}\right)$
(1)+(2)	EC-SGD-DIANA	Alg 6	new	H.4	Option I: $\widetilde{\mathcal{O}}\left(\omega + \frac{\kappa}{\delta} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{\kappa}{\delta} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{\sqrt{L\sigma^2}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(3)	EC-SGDsr-DIANA	Alg 7	new	H.5	Option I: $\widetilde{\mathcal{O}}\left(\omega + \frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L\mathrm{Var}}}{\delta\mu\sqrt{\varepsilon}}\right)$ Option II: $\widetilde{\mathcal{O}}\left(\frac{1+\omega}{\delta} + \frac{\mathcal{L}}{\mu} + \frac{\sqrt{L\mathcal{L}}}{\delta\mu} + \frac{\mathrm{Var}}{n\mu\varepsilon} + \frac{\sqrt{L\mathrm{Var}}}{\mu\sqrt{\delta\varepsilon}}\right)$
(1)+(2)	EC-GD-DIANA†	Alg 6	new	H.4	$\mathcal{O}\left(\left(\omega + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG	Alg 8	new	H.6	$\widetilde{\mathcal{O}}\left(m + \frac{\kappa}{\delta} + \frac{\sqrt{L\zeta_*^2}}{\delta\mu\sqrt{\varepsilon}}\right)$
(1)+(3)	EC-LSVRGstar	Alg 9	new	H.7	$\mathcal{O}\left(\left(m + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$
(1)+(3)	EC-LSVRG-DIANA	Alg 10	new	H.8	$\mathcal{O}\left(\left(\omega + m + \frac{\kappa}{\delta}\right)\log\frac{1}{\varepsilon}\right)$

Our framework covers even methods without error compensation and methods with delayed updates

7. Experiments

Logistic Regression with l2-regularization





partial variance reduction

full variance reduction

8. More Methods

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

- Methods without error feedback: SGD, SGD-SR (arbitrary sampling), SAGA, SVRG, L-SVRG, QSGD, TernGrad, DQGD, DIANA, **DIANAsr-DQ,** VR-DIANA, JacSketch, SEGA

- Methods with delayed updates: D-SGD, **D-SGD-SR** (arbitrary sampling), **D-QSGD, D-SGD-DIANA, D-LSVRG, D-QLSVRG, D-LSVRG-DIANA**

bold font = new method

More Methods Fitting our Framework

The generality of our approach helps to obtain convergence guarantees for a big number of different stochastic methods (even without error compensation). Here are some examples.

- Methods without error feedback: SGD, SGD-SR (arbitrary sampling), SAGA, SVRG, L-SVRG, QSGD, TernGrad, DQGD, DIANA, **DIANAsr-DQ,** VR-DIANA, JacSketch, SEGA
- Methods with delayed updates: D-SGD, **D-SGD-SR** (arbitrary sampling), **D-QSGD, D-SGD-DIANA, D-LSVRG, D-QLSVRG, D-LSVRG-DIANA**

- ✓ In one theorem, we recover the sharpest rates for all known special cases
- ✓ 16 new methods
- ✓ Our analysis works for weakly convex objectives as well

Thank you for watching!

bold font = new method