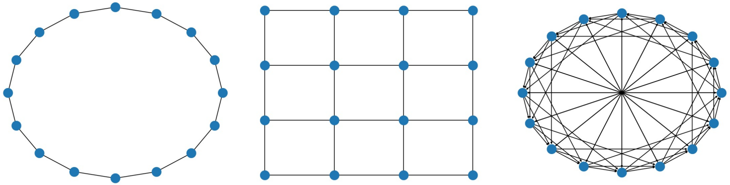
Decentralized SGD: topology

- Assume we connect all nodes with some topology (n=16)



- Communication is only allowed between neighbors
- No global synchronization is allowed

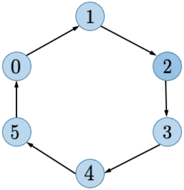
Decentralized SGD: weight matrix

- The weight matrix associated with the topology is defined as

$$w_{ij} \begin{cases} > 0 & \text{if node } j \text{ is connected to } i, \text{ or } i = j; \\ = 0 & \text{otherwise.} \end{cases}$$

- Throughout the lecture we assume the row and column sums of W to be 1

- An example:



$$W = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Figure: A directed ring topology and its associated combination matrix W.

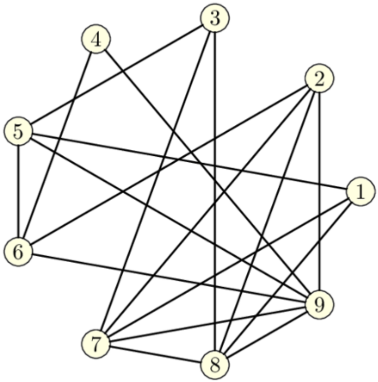
Decentralized SGD (D-SGD): partial averaging

- D-SGD is based on partial-averaging within neighborhood

$$\boxed{\text{Partial averaging:} \quad x_i^+ \qquad \sum_{j \in \mathcal{N}_i} w_{ij} x_j. \quad \forall i \in [n]}$$

- $\mathcal{N}_i$ is the set of neighbors of node i
- Each node only communicates with neighbors; no global sync
- Incurs $\Omega(d_{\max})$ comm. overhead ($d_{\max}$: maximum degree)

Maximum degree[6]



$$d_1 = 3$$
$$d_2 = 4$$
$$d_3 = 3$$
$$\vdots$$
$$d_9 = 6$$

$$d_{\max} = \max_i \{d_i\} = 6$$

[6] Image source:

Decentralized SGD (D-SGD): recursions

- D-SGD $=$ local SGD update$+$ paritial averaging (Loizou and Richtárik, 2020; Nedic and Ozdaglar, 2009; Chen and Sayed, 2012)

$$
\begin{aligned}
x_i^{(k+\frac{1}{2})} &= x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) && \text{(Local update)} \\
x_i^{(k+1)} &= \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k+\frac{1}{2})} && \text{(Partial averaging)}
\end{aligned}
$$

- Per-iteration communication: $\Omega(d_{\max}) \ll \Omega(n)$ when topology is sparse
- Incurs $\Omega(1)$ comm. overhead on sparse topology (ring or grid)

Decentralized SGD is more communication efficient

Model	Ring-Allreduce	Partial average
ResNet-50	278 ms	150 ms
Bert	1469 ms	567 ms

Table: Comparison of per-iter comm. in terms of runtime with 256 GPUs

- ResNet-50 has 25.5M parameters; Bert has 300M parameters
- Partial average saves more communication for larger model

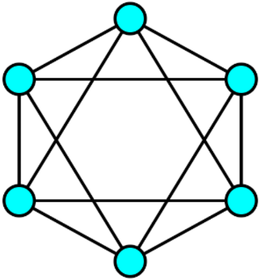
However, D-SGD has slower convergence

- The efficient communication comes with a cost: slow convergence
- Partial averaging is less effective to aggregate information
- The average effectiveness can be evaluated by spectral gap:

$$\rho = \|W - \frac{1}{n}\mathbb{1}\mathbb{1}^T\|_2$$

  - Assume W is doubly-stochastic, it holds that $\rho \in (0, 1)$.
  - Well-connected topology has $\rho \to 0$, e.g. fully-connected topology
  - Sparsely-connected topology has $\rho \to 1$, e.g., ring has $\rho = O(1 - \frac{1}{n^2})$

Weight-matrix of the fully-connected topology



$$W = \frac{1}{5}\mathbb{1}\mathbb{1}^T = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

Decentralized SGD convergence

Recall the assumptions of P-SGD:

Assumption

(A1) Each local loss function $F(x;\xi_i)$ is L-smooth in terms of x;

(A2) Each local stochastic gradient is unbiased, and has bounded variance σ^2:

$$\mathbb{E}[g_i^{(k)}] = \nabla f_i(x^{(k)}), \quad \mathbb{E}\|g_i^{(k)} - \nabla f_i(x^{(k)})\|^2 \le \sigma^2$$

(A3) Each local stochastic gradient $g_i^{(k)}$ is independent of each other

We further introduce another data-heterogeneity assumption

Assumption

(A4) The data heterogeneity is bounded, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f(x)\|^2 \le b^2, \quad \forall x \in \mathbb{R}^d$$

When D_i is identical, we have $\nabla f_i(x) = \nabla f(x)$ for any i and hence $b^2 = 0$

Decentralized SGD convergence

- (Lian et al., 2017; Assran et al., 2019; Koloskova et al., 2020) show that

Theorem (Decentralized SGD convergence)

Under Assumptions (A1)-(A4), and let $\gamma = O(1/\sqrt{T})$*, we have*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla f(x^{(k)})\|^2 = O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}} + \frac{\rho^{2/3}b^{2/3}}{T^{2/3}(1-\rho)^{2/3}}\right)$$

where $T \geq 1$ *is the number of iterations, and* n *is the number of nodes.*

- When topology is fully connected ($\rho = 0$), D-SGD reduces to P-SGD.
- When $\rho = 0$ and $n = 1$, D-SGD reduces to single-node SGD

Convergence rate: P-SGD v.s. D-SGD

- Convergence comparison (i.i.d data distribution, i.e., $b^2 = 0$):

$$\text{P-SGD}: \quad \frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 = O\Big(\frac{\sigma}{\sqrt{nT}}\Big)$$

$$\text{D-SGD}: \quad \frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 = O\Big(\frac{\sigma}{\sqrt{nT}} + \underbrace{\frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}}}_{\text{extra overhead}}\Big)$$

where σ^2 is the gradient noise, and T is the number of iterations.

- D-SGD can asymptotically converge as fast as P-SGD when $T \to \infty$; the first term dominates; reach linear speedup asymptotically

- But it requires more iteration (i.e., T has to be large enough) to reach that stage due to the extra overhead caused by partial averaging

Transient iterations

- **Definition** (Pu et al., 2020): number of iterations before D-SGD achieves linear speedup
- Transient iterations measure the converg. gap between P-SGD and D-SGD
- Longer tran. iters. $\Longrightarrow$ slower convergence than P-SGD
- The transient iteration complexity of D-SGD is

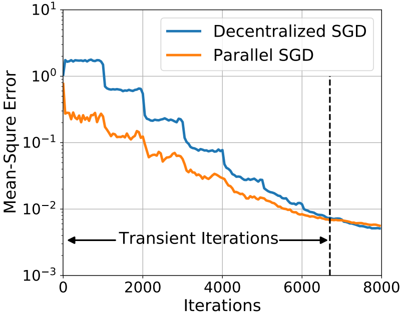
$$\text{iid data}: \quad \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}} \le \frac{\sigma}{\sqrt{nT}} \quad \Longrightarrow \quad T = \Omega\Big(\frac{\rho^4 n^3}{(1-\rho)^2}\Big)$$

$$\text{non-iid data}: \quad \frac{\rho^{2/3}b^{2/3}}{T^{2/3}(1-\rho)^{2/3}} \le \frac{\sigma}{\sqrt{nT}} \quad \Longrightarrow \quad T = \Omega\Big(\frac{\rho^4 n^3}{(1-\rho)^4}\Big)$$

- Sparse topology ($\rho \to 1$) incurs large tran. iters. complexity

Transient iterations: illustration

Illustration of the tran. iters. on D-SGD over ring (logistic regression)



If the transient stage is too long, we may not be able to achieve linear speedup given the limited time/resource budget

Slower convergence will compensate comm. efficiency

- ImageNet dataset; ResNet-50; 256 V100 GPUs

Method	Epoch	Acc.%	Time(hrs.)
P-SGD	120	76.26	2.22
D-SGD	120	75.34	1.55

- D-SGD finishes the same epochs faster because it is more comm. efficient
- D-SGD achieves worse accuracy because it converges slower than P-SGD

Slower convergence will compensate comm. efficiency

- ImageNet dataset; ResNet-50; 256 V100 GPUs

Method	Epoch	Acc.%	Time(hrs.)
P-SGD	120	76.26	2.22
D-SGD	240	76.18	3.03

- When training with more epochs, D-SGD catch up with P-SGD in accuracy; but it takes more wall-clock time than PSGD

- Slower convergence compensates its comm. efficiency

Accelerate D-SGD and make it practical for deep learning

- Recall the transient iteration complexity of D-SGD

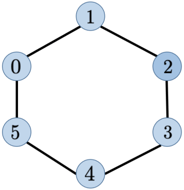
$$\text{iid data}: \quad T = \Omega(\frac{\rho^4 n^3}{(1-\rho)^2})$$
$$\text{non-iid data}: \quad T = \Omega(\frac{\rho^4 n^3}{(1-\rho)^4})$$

- Reducing tran. iter. complexity is the key to accelerating D-SGD

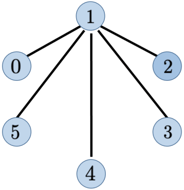
Trade-off between comm. efficiency and convergence rates

- Recall per-iter comm. $\Omega(d_{\max})$ and trans. iters. $\Omega(n^3/(1-\rho)^2)$ (iid data)
- Dense topology: expensive comm. but faster convergence
- Sparse topology: cheap comm. but slower convergence
- What topology shall we use to organize all GPUs?

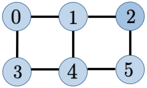
Common topologies



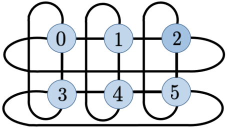
(a) ring



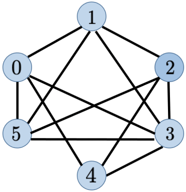
(b) star



(c) 2D-grid



(d) 2D-torus



(e) $\frac{1}{2}$-random graph (one realization)

Common topologies: comm. cost and tran. iters

- According to (Nedić et al., 2018), we have

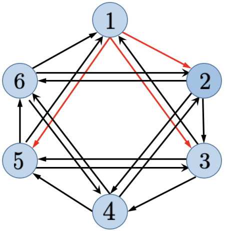
Topology	Per-iter. Comm.	Trans. Iters. (iid scenario)
Ring	$\Omega(2)$	$\Omega(n^7)$
Star	$\Omega(n)$	$\Omega(n^7)$
2D-Grid	$\Omega(4)$	$\Omega(n^5 \log_2^2(n))$
2D-Torus	$\Omega(4)$	$\Omega(n^5)$
$\frac{1}{2}$-RandGraph	$\Omega(\frac{n}{2})$	$\Omega(n^3)$

- These topologies either have expensive comm. cost or longer tran. iters.

- What topology can enable both cheap comm. and fast convergence?

Static exponential graph

- Static exponential graph (Lian et al., 2017, 2018; Assran et al., 2019) is widely-used in deep training
- Empirically successful but less theoretically understood
- Each node links to neighbors that are $2^0, 2^1, \cdots, 2^{\lfloor \log_2(n-1) \rfloor}$ hops away
- In the figure, node 1 connects to $2, 3$ and 5.

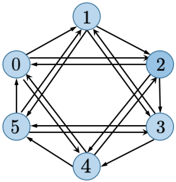


Weight matrix associated with static exponential graph

- The weight matrix W associated with static exp. graph is defined as

$$w_{ij}^{\exp} = \begin{cases} \frac{1}{\lceil \log_2(n) \rceil + 1} & \text{if } \log_2(\mathrm{mod}(j-i, n)) \text{ is an integer or } i = j \\ 0 & \text{otherwise.} \end{cases}$$

- An illustrating example



$$W = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Figure: A 6-node static exponential graph and its associated weight matrix.

Weight matrix over static exponential graph: spectral gap

- Each node has $\lceil \log_2(n) \rceil$ neighbors; per-iter comm. cost is $\Omega(\log_2(n))$
- The following theorem[1] clarifies that $\rho(W^{\exp}) = O(1 - 1/\log_2(n))$; highly non-trivial proofs; requires smart utilization of Fourier transform.

Theorem (Ying et.al., 2021)

Let $\tau = \lceil \log_2(n) \rceil$, and $\rho = \|W - \frac{1}{n}\mathbb{1}\mathbb{1}^T\|_2$ be the spectral gap. It holds that

$$\rho(W^{\exp}) \begin{cases} = 1 - \dfrac{2}{\tau + 1}, & \text{when } n \text{ is even} \\ < 1 - \dfrac{2}{\tau + 1}, & \text{when } n \text{ is odd} \end{cases}$$

[1] B. Ying*, K. Yuan*, Y. Chen*, H. Han, P. Pan, and W. Yin, "Exponential graph is provably efficient for deep training", submitted, 2021

Spectral gap: numerical illustration

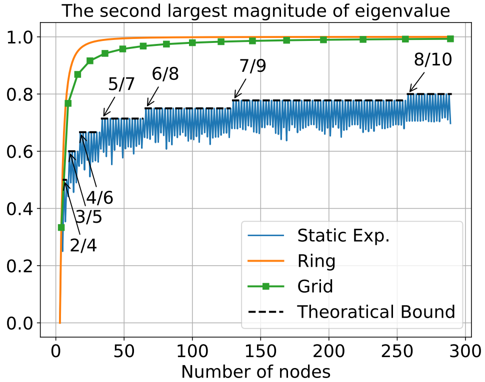


Figure: Illustration of the spectral gaps for ring, grid and static exp. graphs.

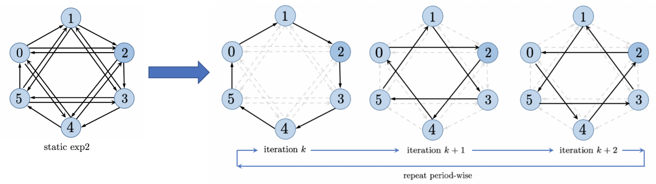
Static exponential graph v.s. other topologies

- Recall D-SGD has tran. iters. $\Omega(n^3/(1-\rho)^2)$
- With $1-\rho = O(1/\log_2(n))$, static exp has tran. iters. $\Omega(n^3 \log_2^2(n))$
- Per-iter comm. and tran. iter. of static exp are nearly best (up to $\log_2(n)$)

Topology	Per-iter. Comm.	Trans. Iters. (iid scenario)
Ring	$\Omega(2)$	$\Omega(n^7)$
Star	$\Omega(n)$	$\Omega(n^7)$
2D-Grid	$\Omega(4)$	$\Omega(n^5 \log_2^2(n))$
2D-Torus	$\Omega(4)$	$\Omega(n^5)$
$\frac{1}{2}$-RandGraph	$\Omega(\frac{n}{2})$	$\Omega(n^3)$
Static Exp	$\tilde{\Omega}(1)$	$\tilde{\Omega}(n^3)$

One-peer exponential graph

- Static exponential graph has $\Omega(\log_2(n))$ per-iteration comm.
- Such overhead is still more expensive than ring or grid
- Split exponential graph into a sequence of one-peer realizations (Assran et al., 2019)



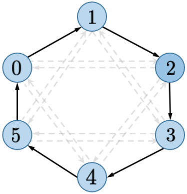
- Each realization has $\Omega(1)$ per-iteration communication

One-peer exponential graph: weight matrix

- We let $\tau = \lceil \log_2(n) \rceil$. The weight matrix $W^{(k)}$ is time-varying

$$w_{ij}^{(k)} = \begin{cases} \frac{1}{2} & \text{if } \log_2(\mathrm{mod}(j-i, n)) = \mathrm{mod}(k, \tau) \\ \frac{1}{2} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

- An illustrating example



$$W = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Decentralized SGD over one-peer exponential graph

- The D-SGD recursion over one-peer exponential graph:

$$\text{Sample } W^{(k)} \text{ over one-peer exponential graph}$$

$$x_i^{(k+\frac{1}{2})} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) \qquad \text{(Local update)}$$

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij}^{(k)} x_j^{(k+\frac{1}{2})} \qquad \text{(Partial averaging)}$$

- One-loop algorithm; each node has one neighbor; per-iter comm. is $\Omega(1)$
- Since each realization is sparser than static exp., will it enable DSGD with longer transient iterations?

One-peer exp. graphs can achieve periodic exact average

Theorem (Periodic Global-averaging)

Suppose $\tau = \log_2(n)$ is a positive integer. It holds that

$$W^{(k+\ell)} W^{(k+\ell-1)} \cdots W^{(k+1)} W^{(k)} = \frac{1}{n} \mathbb{1}\mathbb{1}^T$$

for any integer $k \geq 0$ and $\ell \geq \tau - 1$.

While each realization of one-peer graph is sparser, a sequence of one-peer graphs will enable effective global averaging.

One-peer exp. graphs can achieve periodic exact average

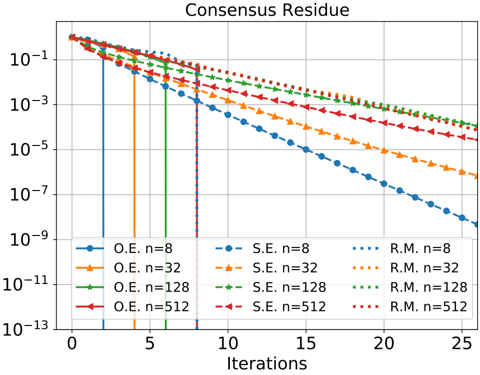


Figure: O.E. graph has periodic global averaging when $\tau = \log_2(n)$ is an integer.

Applying one-peer exp. graphs to DSGD

Assumption

(1) Each $f_i(x)$ is L-smooth; (2) Each gradient noise is unbiased and has bounded variance σ^2; (3) Each local distribution D_i is identical (iid)

Theorem (DSGD convergence with one-peer exp.)

Under the above assumptions and with $\gamma = O(1/\sqrt{T})$, let $\tau = \log_2(n)$ be an integer, DSGD with one-peer exponential graph will converge at

$$\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 = O\Big(\frac{\sigma}{\sqrt{nT}} + \underbrace{\frac{\sigma^{2/3}\log_2^{2/3}(n)}{T^{2/3}}}_{\text{extra overhead}}\Big)$$

Convergence rate for decentralized momentum SGD (DmSGD) with non-iid data distributions is also established in (Ying et al., 2021).

Static exp. v.s. one-peer exp.

- Convergence rate for DSGD over static and one-peer exp. graphs

$$\text{Static exp.} \quad O\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}}\Big) \quad \text{(where } 1-\rho = O(1/\log_2(n))\text{)}$$

$$\text{One-peer exp.} \quad O\Big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{2/3}\log_2^{2/3}(n)}{T^{2/3}}\Big)$$

- DSGD with one-peer exp. converges as fast as static exp. in terms of the established bounds; a surprising result.
- DSGD with both graphs are with the same tran. iters. $O(n^3\log_2^2(n))$
- The same results hold for heterogeneous data scenario, and for DmSGD.

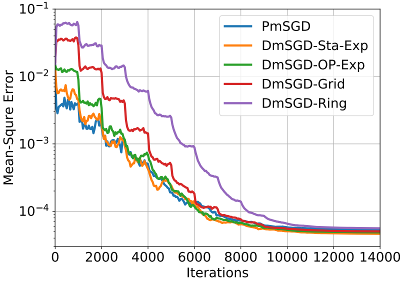
One-peer graph is the state-of-the-art topology

Topology	Per-iter. Comm.	Trans. Iters. (iid scenario)
Ring	$\Omega(2)$	$\Omega(n^7)$
Star	$\Omega(n)$	$\Omega(n^7)$
2D-Grid	$\Omega(4)$	$\Omega(n^5 \log_2^2(n))$
2D-Torus	$\Omega(4)$	$\Omega(n^5)$
$\frac{1}{2}$-RandGraph	$\Omega(\frac{n}{2})$	$\Omega(n^3)$
Static Exp.	$\tilde{\Omega}(1)$	$\tilde{\Omega}(n^3)$
One-peer Exp.	$\Omega(1)$	$\tilde{\Omega}(n^3)$

- Since one-peer exp. incurs less per-iter comm., it is recommended for DL.

Exponential graphs have shorter transient iterations

Illustration of the tran. iters. on DmSGD for logistic regression.



DmSGD over both exp. graphs converge roughly the same; they are faster than other topologies with 32 nodes.

Experimental results: two metrics

- Wall-clock time to finish 90 epochs of training; measures per-iter comm.
- Validation accuracy after 90 epochs of training; measures convgt. rate

Image Classification

- ImageNet-1K dataset
- 1.3M training images
- 50K test images
- 1K classes
- DNN Model: ResNet-50 ($\sim$25.5M parameters)
- GPU: Tesla V100 clusters
- Framework: Pytorch DDP

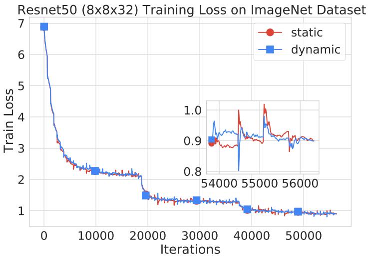
D-SGD achieves better linear speedup

Table: Comparison of top-1 validation accuracy(%) and training time (hours).

nodes	4(4x8 GPUs)		8(8x8 GPUs)		16(16x8 GPUs)		32(32x8 GPUs)	
topology	acc.	time	acc.	time	acc.	time	acc.	time
P-SGD	76.32	11.6	76.47	6.3	76.46	3.7	76.25	2.2
Ring	76.16	11.6	76.14	6.5	76.16	3.3	75.62	1.8
one-peer exp.	76.34	11.1	76.52	5.7	76.47	2.8	76.27	1.5

Convergence curves: one-peer exp. v.s. static exp.

Image classification: ResNet-50 for ImageNet; $8 \times 8 = 64$ GPUs.



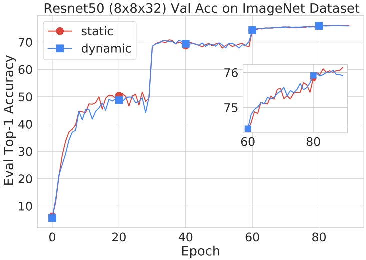


Figure: DmSGD over one-peer exp. converges as fast as over static exp.

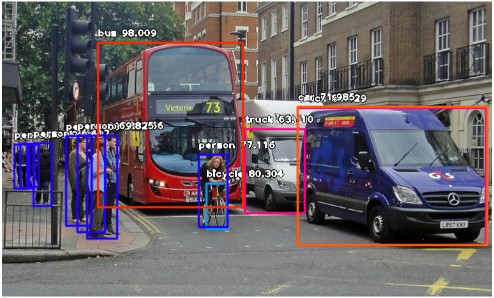
Comparing different models/methods: one-peer v.s. static

MODEL	RESNET-50			MOBILENET-V2			EFFICIENTNET		
TOPOLOGY	STATIC	ONE-PEER	DIFF	STATIC	ONE-PEER	DIFF	STATIC	ONE-PEER	DIFF
PARALLEL SGD	76.21	-	-	70.12	-	-	77.63	-	-
VANILLA DMSGD	76.14	76.06	-0.08	69.98	69.81	-0.17	77.62	77.48	-0.14
DMSGD	76.50	76.52	+0.02	69.62	69.98	+0.36	77.44	77.51	+0.07
QG-DMSGD	76.43	76.35	-0.08	69.83	69.81	-0.02	77.60	77.72	+0.12

- setting: ImageNet; $8 \times 8 = 64$ GPUs; diff $=$ o.e - s.e.
- both topo. achieve similar accuracy across different models and algorithms
- accuracy difference is minor (except for MobileNet with DmSGD)
- QG-DmSGD (Lin et al., 2021) and DmSGD can outperform PSGD in ResNet-50 in accuracy

Object Detection

- Dataset: PASCAL/COCO
- GPU: Tesla V100 clusters
- Framework: Pytorch DDP; BlueFog



Comparing different tasks: one-peer exp. v.s. static exp.

Dataset	PASCAL VOC				COCO			
Model	RetinaNet		Faster RCNN		RetinaNet		Faster RCNN	
topology	static	one-peer	static	one-peer	static	one-peer	static	one-peer
Parallel SGD	79.0	-	80.3	-	36.2	-	37.2	-
Vanilla DmSGD	79.0	79.1	80.7	80.5	36.3	36.1	37.3	37.2
DmSGD	79.1	79.0	80.4	80.5	36.4	36.4	37.1	37.0
QG-DmSGD	79.2	79.1	80.8	80.4	36.3	36.2	37.2	37.1

- setting: object detection; $8 \times 8 = 64$ GPUs;
- both topo. achieve similar accuracy across different algorithms in detection

Summary

- Both per-iter comm. and tran. iter. of exp. graphs are nearly best (up to $\log_2(n)$ factors) among known topologies
- While one-peer exp. is sparser, it can converge as fast as staic exp.
- One-peer exponential graph is recommend for decentralized DL

D-SGD transient iteration complexity review

- Recall the convergence rate of D-SGD for non-convex and non-iid scenario:

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla f(x^{(k)})\|^2 = O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}} + \frac{\rho^{2/3}b^{2/3}}{T^{2/3}(1-\rho)^{2/3}}\right)$$

  where $b^2 > 0$ deteriorates the dependence on network topology $1-\rho$

- The transient iteration complexity of D-SGD is summarized as

scenario	**iid data**	**non-iid data**
strongly-convex	$\Omega(\frac{n}{1-\rho})$	$\Omega(\frac{n}{(1-\rho)^2})$
generally-convex	$\Omega(\frac{n^3}{(1-\rho)^2})$	$\Omega(\frac{n^3}{(1-\rho)^4})$
non-convex	$\Omega(\frac{n^3}{(1-\rho)^2})$	$\Omega(\frac{n^3}{(1-\rho)^4})$

D-SGD transient iteration complexity review

- Can we improve the dependence on topology for non-iid scenario?
- Main idea: remove the influence of b^2 from the convergence rate (Koloskova et al., 2020; Huang and Pu, 2021; Yuan et al., 2020; Yuan and Alghunaim, 2021)[2]
- Suppose a decentralized method for non-iid scenario can converge as

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla f(x^{(k)})\|^2 = O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{T^{2/3}(1-\rho)^{1/3}}\right)$$

it will improve the transient iteration complexity as follows

$$\Omega(\frac{\rho^4 n^3}{(1-\rho)^4}) \quad \Longrightarrow \quad \Omega(\frac{\rho^4 n^3}{(1-\rho)^2})$$

[2] K. Yuan and S. A. Alghunaim, "Removing data heterogeneity influence enhances network topology dependence of decentralized SGD", arXiv:2105.08023

How does D-SGD suffer from data heterogeneity?

- For simplicity, we consider the deterministic convex decentralized GD:

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} \Big(x_j^{(k)} - \gamma \nabla f_j(x_j^{(k)}) \Big), \quad \forall i \in [n]$$

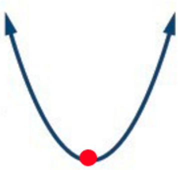
- Suppose $x_i^{(k)} = x^\star$ at iteration k for any $i \in [n]$, it holds that

$$\begin{aligned} x_i^{(k+1)} &= \sum_{j \in \mathcal{N}_i} w_{ij} \Big(x^\star - \gamma \nabla f_j(x^\star) \Big) \\ &= x^\star - \gamma \sum_{j \in \mathcal{N}_i} w_{ij} \nabla f_j(x^\star) \neq x^\star \end{aligned}$$

  where the last inequality holds because $f_i(x) \neq f(x)$ (data-heterogeneous)

- D-GD cannot stay at $x^\star$; data heterogeneity incurs oscillation.

How does D-SGD suffer from data heterogeneity?



$$x_i^{(k)} = x^\star$$

$$x_i^{(k+1)} = x^\star - \gamma \sum_{j \in \mathcal{N}_i} w_{ij} \nabla f_j(x^\star) \neq x^\star$$

Remove the influence of data-heterogeneity

- EXTRA (Shi et al., 2015) is the first decentralized method to remove the influence of data heterogeneity
- Exact-Diffusion (Yuan et al., 2019) (also known as NIDS (Li et al., 2019) or D^2 (Tang et al., 2018)) improves EXTRA on learning rate stability range
- Gradient-tracking based methods (Xu et al., 2015; Di Lorenzo and Scutari, 2016; Nedic et al., 2017; Qu and Li, 2018; Pu et al., 2020b; Xin and Khan, 2018) remove data heterogeneity, and can be used in more relaxed settings (e.g., asymmetric/directed/time-varying weight matrix)
- All these algorithms can be unified into one decentralized framework (Alghunaim et al., 2020; Xu et al., 2021; Xin et al., 2020a)

Exact-Diffusion

- For Exact-Diffusion, each node run the following recursion in parallel

$$
\boxed{
\begin{aligned}
\psi_i^{(k+1)} &= x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) && \text{(local SGD)} \\
\phi_i^{(k+1)} &= \psi_i^{(k+1)} + x_i^{(k)} - \psi_i^{(k)} && \text{(bias correction)} \\
x_i^{(k+1)} &= \sum_{j \in \mathcal{N}_i} w_{ij}\, \phi_j^{(k+1)} && \text{(partial averaging)}
\end{aligned}
}
$$

- When correction term $x_i^{(k)} - \psi_i^{(k)}$ is removed from the correction step, Exact-Diffusion reduces to standard D-SGD

- The weight matrix W needs to be symmetric, and satisfies $\lambda_n(W) > -\frac{1}{3}$

How is Exact-Diffusion immune to data heterogeneity?

- Combining all recursions, we achieve the deterministic version

$$x_i^{(k+1)} = \sum_{j\in\mathcal{N}_i} w_{ij} \left(2x_i^{(k)} - x_i^{(k-1)} + \gamma(\nabla f(x_i^{(k)}) - \nabla f(x_i^{(k-1)}))\right)$$

- Assume $x_i^{(k-1)} = x_i^{(k)} = x^\star$ for any $i \in [n]$, at iteration $k+1$ we have

$$x_i^{(k+1)} = \sum_{j\in\mathcal{N}_i} w_{ij}(2x^\star - x^\star) = x^\star$$

- When initialized from the minimum, Exact-Diffusion can stay there in spite of the data heterogeneity $\nabla f_i(x) \neq \nabla f_j(x)$

Exact-Diffusion convergence

Assumption

(A1) Each local loss function $F(x;\xi_i)$ is L-smooth in terms of x;

(A2) Each local stochastic gradient is unbiased, and has bounded variance σ^2

(A3) Each local stochastic gradient $g_i^{(k)}$ is independent of each other

(A4) W is positive semi-definite

Theorem (Yuan and Alghunaim (2021))

Under the above assumptions and with appropriate γ, Exact-Diffusion will converge at (S.C. is for strongly-convex and G.C. is for generally-convex)

$$\frac{1}{T+1}\sum_{k=0}^{T}\left(\mathbb{E}f(\bar{x}^{(k)}) - f(x^\star)\right) = O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{(1-\rho)^{1/3}T^{2/3}}\right) \qquad \textit{(G.C.)}$$

$$\frac{1}{H_T}\sum_{k=0}^{T}h_k\left(\mathbb{E}f(\bar{x}^{(k)}) - f(x^\star)\right) = \tilde{O}\left(\frac{\sigma^2}{nT} + \frac{\rho^2\sigma^2}{(1-\rho)T^2}\right) \qquad \textit{(S.C.)}$$

where h_k is some positive weight and $H_T = \sum_{k=0}^{T} h_k$.

Convergence comparison: Exact-Diffusion v.s. D-SGD

In the strongly-convex setting,

- The convergence rate comparison:

$$\text{D-SGD}: \quad \tilde{O}\left(\frac{\sigma^2}{nT} + \frac{\rho^2\sigma^2}{(1-\rho)T^2} + \frac{\rho^2 b^2}{(1-\rho)^2 T^2}\right)$$

$$\text{Exact-Diffusion}: \quad \tilde{O}\left(\frac{\sigma^2}{nT} + \frac{\rho^2\sigma^2}{(1-\rho)T^2}\right)$$

- The transient iteration complexity comparison (Huang and Pu, 2021; Yuan and Alghunaim, 2021):

$$\text{D-SGD}: \ \Omega\left(\frac{\rho^2 n}{(1-\rho)^2}\right) \qquad \text{Exact-Diffusion}: \ \Omega\left(\frac{\rho^2 n}{1-\rho}\right)$$

Convergence comparison: Exact-Diffusion v.s. D-SGD

In the generally-convex setting,

- The convergence rate comparison:

$$\text{D-SGD}: \quad O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{(1-\rho)^{1/3}T^{2/3}} + \frac{\rho^{2/3}b^{2/3}}{(1-\rho)^{2/3}T^{2/3}}\right)$$

$$\text{Exact-Diffusion}: \quad O\left(\frac{\sigma}{\sqrt{nT}} + \frac{\rho^{2/3}\sigma^{2/3}}{(1-\rho)^{1/3}T^{2/3}}\right)$$

- The transient iteration comparison (Yuan and Alghunaim, 2021):

$$\text{D-SGD}: \ \Omega\left(\frac{\rho^4 n^3}{(1-\rho)^4}\right) \qquad \text{Exact-Diffusion}: \ \Omega\left(\frac{\rho^4 n^3}{(1-\rho)^2}\right)$$

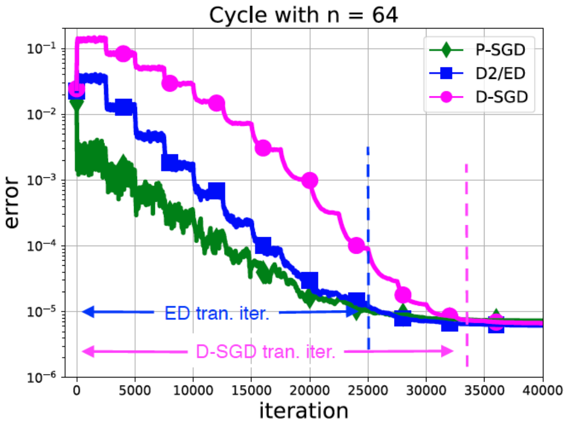
Convergence comparison: Exact-Diffusion v.s. D-SGD

In the non-convex setting,

- Exact-Diffusion can remove data heterogeneity (Tang et al., 2018), but no improved result on network topology dependence was shown
- Gradient-tracking can remove data heterogeneity (Xin et al., 2020b; Zhang and You, 2019; Lu et al., 2019), but no improved result on network topology dependence was shown
- It is still an open question whether data-heterogeneity-corrected methods (such as EXTRA, Exact-Diffusion, and Gradient tracking) can have an improved network topology dependence than P-SGD

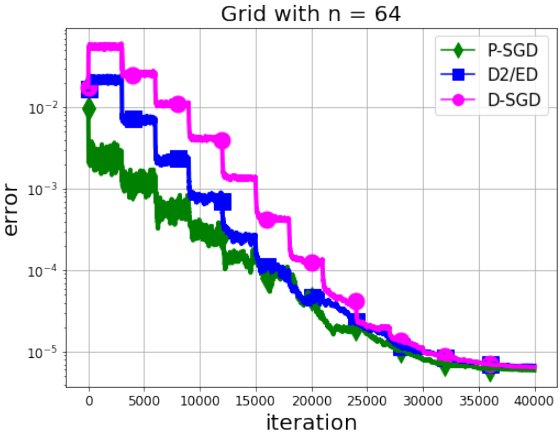
Experiments: Exact-Diffusion v.s. D-SGD

Convex setting: logistic regression problem; non-iid scenario



Convergence comparison: Exact-Diffusion v.s. D-SGD

Strongly-convex setting: least-square problem; non-iid scenario



Summary

- The data heterogeneity b^2 in D-SGD deteriorates the topology dependence
- EXTRA/Exact-Diffusion/Gradient-tracking can remove the influence of b^2
- Exact-Diffusion improves the topology dependence when b^2 exists.

non-iid scenario	**Exact-Diffusion**	**D-SGD**
strongly-convex	$\Omega(\frac{\rho^2 n}{1-\rho})$	$\Omega(\frac{\rho^2 n}{(1-\rho)^2})$
generally-convex	$\Omega(\frac{\rho^4 n^3}{(1-\rho)^2})$	$\Omega(\frac{\rho^4 n^3}{(1-\rho)^4})$
non-convex	N.A.	$\Omega(\frac{\rho^4 n^3}{(1-\rho)^4})$

Motivation

- Recall non-convex D-SGD suffers from additional transient iterations

$$\text{homogeneous (iid) data:} \quad \Omega\Big(\frac{\rho^4 n^3}{(1-\rho)^2}\Big)$$

$$\text{heterogeneous (non-iid) data:} \quad \Omega\Big(\frac{\rho^4 n^3}{(1-\rho)^4}\Big)$$

- $\rho \to 1$ will significantly enlarge the transient iteration stage

- Unfortunately, most topologies have $\rho \to 1$ as n grows
  - Ring: $1-\rho = O(1/n^2)$;
  - Grid: $1-\rho = O(1/n)$;
  - Exp.: $1-\rho = O(1/\log_2(n))$

- We have to alleviate the influence of $1/(1-\rho)$ in trans. iters. complexity

Per-iteration communication cost

Model	Ring-Allreduce	Partial average
ResNet-50	278 ms	150 ms
Bert	1469 ms	567 ms

Table: Comparison of per-iter comm. in terms of runtime with 256 GPUs

- While global average takes longer comm. time, it is not too bed
- We can mix partial average with global average (Chen et al., 2021)[3].
- In a period of H iterations: run $H - 1$ partial average and 1 global average

[3] Y. Chen*, K. Yuan*, Y. Zhang, P. Pan, Y. Xu, W. Yin, "Accelerating Gossip SGD with Periodic Global Averaging", ICML 2021

DSGD-PGA: DSGD with Periodic Global Averaging

- DSGD-PGA: accelerate D-SGD with periodic global averaging

$$\boldsymbol{x}_i^{(k+\frac{1}{2})} = \boldsymbol{x}_i^{(k)} - \gamma \nabla F(\boldsymbol{x}_i^{(k)}; \xi_i^{(k+1)})$$

$$\boldsymbol{x}_i^{(k+1)} = \begin{cases} \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j^{(k+\frac{1}{2})} & \text{If } \mathrm{mod}(k+1, H) = 0 \\ \sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{x}_j^{(k+\frac{1}{2})} & \text{If } \mathrm{mod}(k+1, H) \neq 0 \end{cases}$$

  where H is the global averaging period.

- DSGD-PGA is expected to converge faster than D-SGD.

- DSGD-PGA reduces to D-SGD when $H \to \infty$

- Similar idea also appeared in topology-changing D-SGD (Koloskova et al., 2020) and SlowMo (Wang et al., 2019)

DSGD-PGA: Transient iteration complexity

- PGA significantly improves the transient stage of D-SGD in the non-convex setting (Chen et al., 2021):

scenario	DSGD-PGA	D-SGD
iid data	$\Omega(\rho^4 n^3 H^2)$	$\Omega(\frac{\rho^4 n^3}{(1-\rho)^2})$
non-iid data	$\Omega(\rho^4 n^3 H^4)$	$\Omega(\frac{\rho^4 n^3}{(1-\rho)^4})$

- PGA bounds $1/(1-\rho)$ with H; benefits most for sparse topology

Numerical experiments: D-SGD v.s. DSGD-PGA

Problem: logistic regression problem with non-iid data

Cyclic Topology

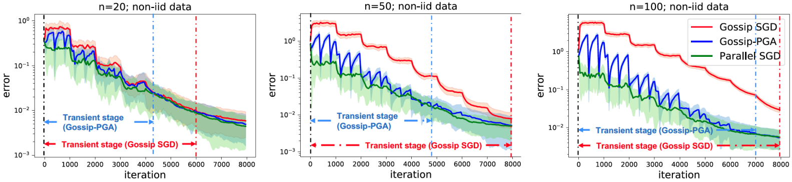


Figure: Transient stage comparison.

DSGD-AGA: D-SGD with Adaptive Global Averaging

- Gossip-AGA avoids the burden of turning parameters
- An effective period strategy: more frequent GA in initial stages
- Intuition: lower consensus variance can speedup convergence

$$\frac{1}{n(T+1)}\sum_{k=0}^{T}\sum_{i=1}^{n}\mathbb{E}\|\boldsymbol{x}_i^{(k)}-\bar{\boldsymbol{x}}^{(k)}\|^2 \le \frac{d_1\gamma^2}{T+1}\sum_{k=0}^{T}\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^{(k)})\|^2 + d_2\gamma^2$$

  Consensus variance gets decreased as $\gamma \to 0$ and $\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^{(k)})\|^2 \to 0$

- Adaptive rule: $H^{(\ell)} = \left(\frac{\mathbb{E}f(\bar{\boldsymbol{x}}^{(0)})}{\mathbb{E}f(\bar{\boldsymbol{x}}^{(T_{\ell-1})})}\right)^{\frac{1}{4}} H^{(0)}$;

Experiments on Large-scale Deep Training

Language Modeling:

- Model: BERT-Large ($\sim$330M parameters)
- Dataset: Wikipedia (2500M words) and BookCorpus (800M words)
- Hardware: 64 GPUs

Image Classification

Method	Final Loss	Wall-clock Time (hrs)
P-SGD	1.75	59.02
D-SGD	2.17	29.7
D-SGD $\times 2$	1.81	59.7
DSGD-PGA	1.82	35.4
DSGD-AGA	1.77	30.4

Table: Comparison of training loss and training time of BERT training.

- DSGD-AGA acheives similar final loss with $2\times$ speedup

Summary

- Periodic global averaging can improve the transient iteration stage:

$$\Omega(\frac{\rho^4 n^3}{(1-\rho)^4}) \quad \Longrightarrow \quad \Omega(\rho^4 n^3 H^4)$$

- PGA benefits most for sparse topology, i.e., $\rho \to 1$

- Global averaging period H can be adjusted adaptively

Discussion

- We consider deep training within high-performance data-center clusters
- Global averaging conducted by All-reduce has tolerable comm. cost
- For mobile AI or federated learning, global averaging is very expensive
- We can approximate global averaging via multiple partial averaging steps, see [Lu and De Sa, 2021, ICML Outstanding Paper Honorable mention]
- However, multiple partial averaging steps are not recommended for data-center clusters; 3 partial averaging steps may take more wall-clock time than one single global averaging