CO2: Efficient Distributed Training with Full Communication-Computation Overlap

Weigao Sun, Zhen Qin, Weixuan Sun, Shidi Li, Dong Li, Xuyang Shen, Yu Qiao, Yiran Zhong

Shanghai AI Laboratory, Shanghai, China

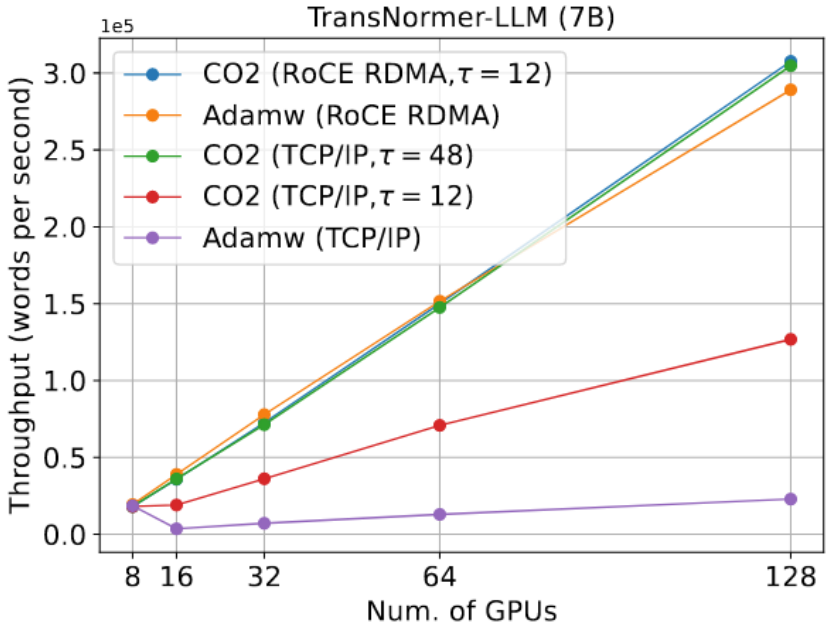
BACKGROUND

- SOTA Distributed Data Parallel (DDP) methods still have communication tail, which degrades the training scalability and efficiency.
- The communication tail is worse in low-bandwidth large clusters.

- Three strategies to improve communication efficiency:
  1. Communication Compression in Single Iteration.
  2. Communication Frequency Reduction.
  3. Communication and Computation Overlapping.

CONTRIBUTIONS

1. Outstanding 100% Scalability (on even low-bandwidth clusters).
2. Good Convergence and Generalization Performance.
3. Theoretical Convergence Guarantees.
4. Compatibility with ZeRO-series Sharding DDP Optimizers.



(a) Scalability of CO2.

METHOD

ICLR

Key points:

1. Local Updating
2. Single-step Asynchronous Communication
3. Staleness Gap Penalty
4. Outer Momentum Clipping

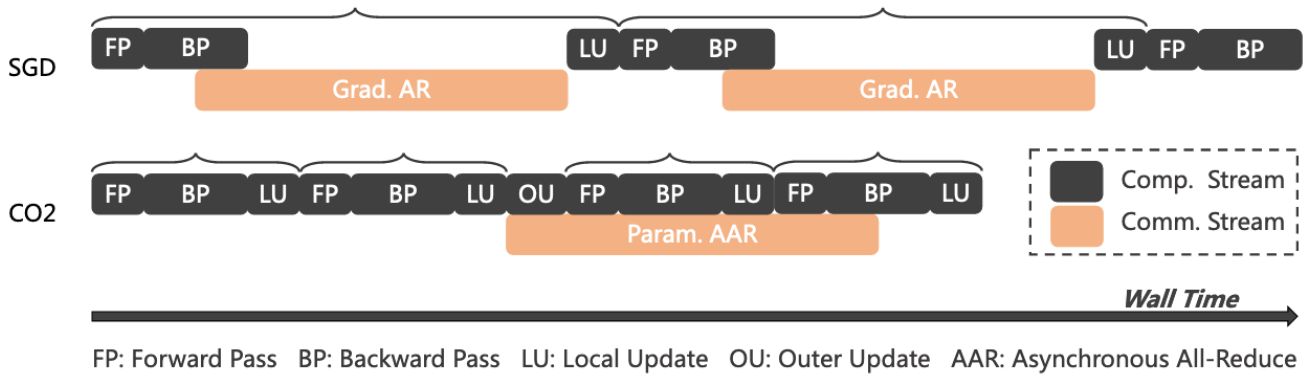


Figure 1: **Visualization of CO2 and SGD**. We exemplify the mechanism of CO2 with a local step count $\tau = 2$. This configuration dictates that the outer update starts after every two local steps, concurrently launching an AAR operation on model parameters. This strategy is made to make the full overlap of AAR communication with local computation possible. CO2 can effectively reduce the wall time required for training compared to the conventional SGD in DDP paradigm.

Algorithm 1: CO2 Algorithm

Input: Data samples $\boldsymbol{\zeta}^{(i)}$ on worker i; Inner learning rate γ_t; Inner loop steps τ; Outer learning rate α; Outer momentum factor β; Outer loop steps T; Initial outer momentum $\boldsymbol{m}_0 = 0$.

for $t \in \{0, 1, \cdots, T-1\}$ **do**

  for $k \in \{0, 1, \cdots, \tau-1\}$ *on worker* i **do**

    FP & BP: $\boldsymbol{g}_{t,k}^{(i)} = \nabla L^{(i)}(\boldsymbol{x}_{t,k}^{(i)}; \boldsymbol{\zeta}_{t,k}^{(i)})$

    LU: $\boldsymbol{x}_{t,k+1}^{(i)} = \boldsymbol{x}_{t,k}^{(i)} - \gamma_t \boldsymbol{g}_{t,k}^{(i)}$

  end

  Launch AAR for $\boldsymbol{x}_{t,\tau}^{(i)}$: $AAR(\boldsymbol{x}_{t,\tau}^{(i)})$

  if *is_completed*$(AAR(\boldsymbol{x}_{t-1,\tau}^{(i)}))$ *is not* $True$ **then**

    $\boldsymbol{x}_{t-1,\tau} = \text{wait}(AAR(\boldsymbol{x}_{t-1,\tau}^{(i)}))$

  end

  Update staleness gap:

  $\boldsymbol{\Lambda}_t = \frac{\|\boldsymbol{x}_{t,0} - \boldsymbol{x}_{t-1,0}\|}{\tau \|\boldsymbol{x}_{t-1,1} - \boldsymbol{x}_{t-1,0}\|} + \mathbf{1}^n$

  Update one-step stale outer momentum:

  $\boldsymbol{m}_t = \beta \boldsymbol{m}_{t-1} + \frac{1}{\boldsymbol{\Lambda}_t} \cdot (\boldsymbol{x}_{t-1,0} - \boldsymbol{x}_{t-1,\tau})$

  Update outer iterates:

  $\boldsymbol{x}_{t+1,0} = \boldsymbol{x}_{t,0} - \alpha \cdot \text{Clip}(\boldsymbol{m}_t, \phi)$

end

THEORICAL CONVERGENCE GUARANTEE

ICLR

Convergence Theorem

Assumption 3.2 *For all* $i \in \{1, 2, \ldots, G\}$, *there exists a finite positive constant* σ^2 *such that* $\mathbb{E}_{\zeta \sim D_i} \left\| \nabla L^{(i)}(\boldsymbol{x}; \zeta) - \nabla f_i(\boldsymbol{x}) \right\|^2 \leq \sigma^2$, *i.e., the variance of* $f_i(\boldsymbol{x})$ *is bounded.*

Assumption 3.3 *There exists a finite positive constant* V *such that* $\mathbb{E} \left\| \boldsymbol{g}_{t,k} - \mathbb{E}\left[\boldsymbol{g}_{t,k}\right] \right\|^2 \leq V$.

Under these assumptions, our convergence results are given as below, where the detailed proof can be found in Appendix A.4.

Theorem 1 *If we take* $\lambda = 1/\Lambda_t, \gamma_t = \gamma$ *and ignore the clip operation, such that* $\bar{\lambda} = \alpha\lambda$, $\frac{\bar{\lambda}\gamma}{1-\beta} = \sqrt{\frac{G}{T\tau}}$ *and* $T\tau \geq GL^2 \left(1 + \sqrt{3}\max\left\{\frac{3\tau(1-\beta-\alpha)}{\alpha}, \frac{4\tau\beta}{1-\beta}, 1\right\}\right)$, *then under Assumptions 3.1, 3.2, 3.3 and* $\frac{1}{G}\sum_{i=1}^{G} \|\nabla f(\mathbf{x}) - \nabla f_i(\mathbf{x})\|^2 \leq \delta^2$, *where* δ *is a positive finite constant, we have:*

$$\frac{1}{T\tau}\sum_{t=0}^{T-1}\sum_{k=0}^{\tau-1} \mathbb{E} \left\| \nabla f\left(\boldsymbol{x}_{t,k}\right) \right\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{GT\tau}}\right) + \mathcal{O}\left(\frac{G\tau}{T}\right), \tag{4}$$

where $f = \frac{1}{G}\sum_{i=1}^{G} f_i(x)$. The theorem indicates that, when the total steps $T\tau$ is sufficiently large, i.e, $T \gg G^3\tau^3$, the RHS is dominated by $\mathcal{O}\left(\frac{1}{\sqrt{GT\tau}}\right)$. So, when the number of workers is G times more, we only need G times less total steps to achieve the same error.

Comprehensive Experiment Design

Table 5: **Tasks, Models, and Datasets implemented in the experiments.** IC: image classification, SS: semantic segmentation, PC: Point Cloud, ALM: autoregressive language modeling, BLM: bidirectional language model. TN-LLM: TransNormer-LLM.

Field	Task	Model	Parameters	Model Type	Dataset
CV	IC	ResNet-50	25.6M	ConvNet	ImageNet-1k
		ViT (Base)	86.6M	Transformer	
		VVT (Large)	61.8M	Linear Transformer	
	SS	VVT (Large)	65.5M	Linear Transformer	ADE20K
	PC	Point-MAE	10.2M	Transformer	ShapeNet
NLP	ALM	GPT-2 (Small)	125M	Transformer (Decoder)	OpenWebText
		GPT-2 (Medium)	355M	Transformer (Decoder)	
		GPT-2 (Large)	770M	Transformer (Decoder)	
		TN-LLM (7B)	6.7B	Linear Transformer	WikiText-103
	BLM	RoBERTa (Large)	355M	Transformer (Encoder)	Wikipedia+BookCorpus

CV Results

Table 1: **Convergence performance on CV tasks.** CO2 performs better than other local-updating methods with a clear margin and is comparable to standard optimizers such as SGD/Adamw.

Task	Model	SGD (Adamw)	Local-SGD(Adamw)	Overlap-Local-SGD(Adamw)	SlowMo	CO2
IC	ResNet-50	76.92 (± 0.05)	75.57 (± 0.76)	76.28 (± 0.18)	77.12 (± 0.11)	77.14 (± 0.09)
	ViT (Base)	81.33 (± 0.04)	78.43 (± 0.22)	78.04 (± 0.35)	79.83 (± 0.16)	80.95 (± 0.08)
	VVT (Large)	83.64 (± 0.06)	81.09 (± 1.15)	80.33 (± 0.49)	82.75 (± 0.27)	83.38 (± 0.06)
SS	VVT (Large)	47.82 (± 0.05)	44.25 (± 2.24)	45.21 (± 1.36)	47.51 (± 0.12)	47.80 (± 0.11)
PC	Point-MAE	68.56 (± 0.08)	64.25 (± 2.11)	63.78 (± 1.92)	68.69 (± 0.32)	68.89 (± 0.39)

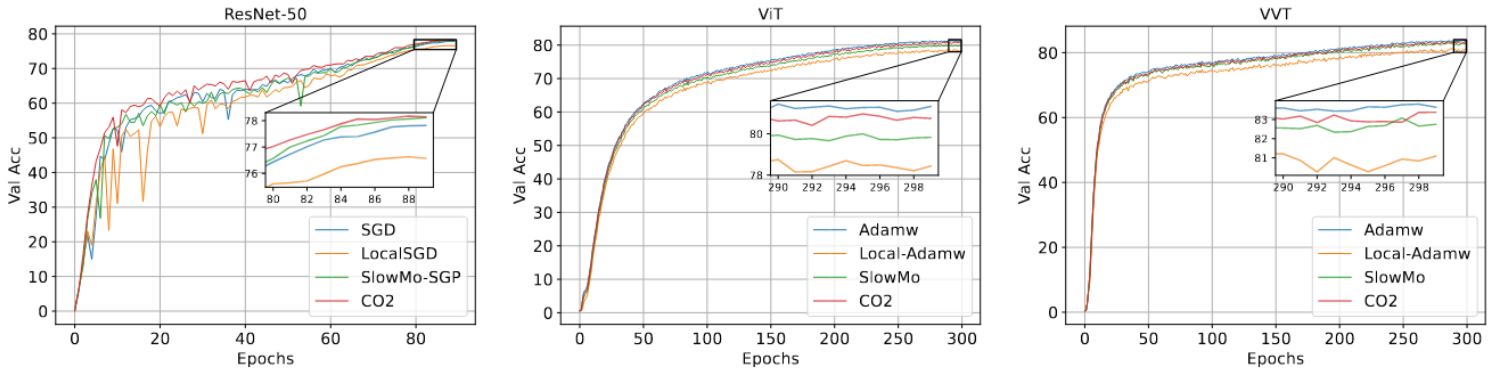


Figure 2: **Validation curves for image classification tasks.** Three models ResNet-50, ViT, and VVT are trained on ImageNet-1K for 90, 300 and 300 epochs, respectively. Our CO2 exhibits robust convergence and good generalization performance when compared to other existing methods.

NLP Results

Table 2: **Convergence performance on NLP tasks.** Quantitative perplexity (PPL) results for GPT-2, TransNormer-LLM and RoBERTa are presented. CO2 shows lower perplexity (lower is better) than the baseline Adamw in all these experiments.

Task	Model	Adamw	Local-Adamw	Overlap-Local-Adamw	SlowMo	CO2
ALM	GPT-2 (Small)	7.44 (± 0.36)	7.95 (± 2.04)	8.11 (± 1.03)	7.34 (± 0.89)	7.37 (± 0.73)
	GPT-2 (Medium)	6.61 (± 0.53)	7.49 (± 1.87)	7.26 (± 1.44)	6.41 (± 0.65)	6.36 (± 0.66)
	GPT-2 (Large)	6.26 (± 0.58)	7.00 (± 1.91)	7.18 (± 0.98)	6.29 (± 0.61)	6.13 (± 0.52)
	TN-LLM (7B)	16.82 (± 0.86)	18.63 (± 3.13)	17.83 (± 2.95)	16.95 (± 1.01)	16.78 (± 0.95)
BLM	RoBERTa (Large)	3.96 (± 0.37)	4.38 (± 0.83)	4.52 (± 1.42)	3.98 (± 0.85)	3.95 (± 0.96)

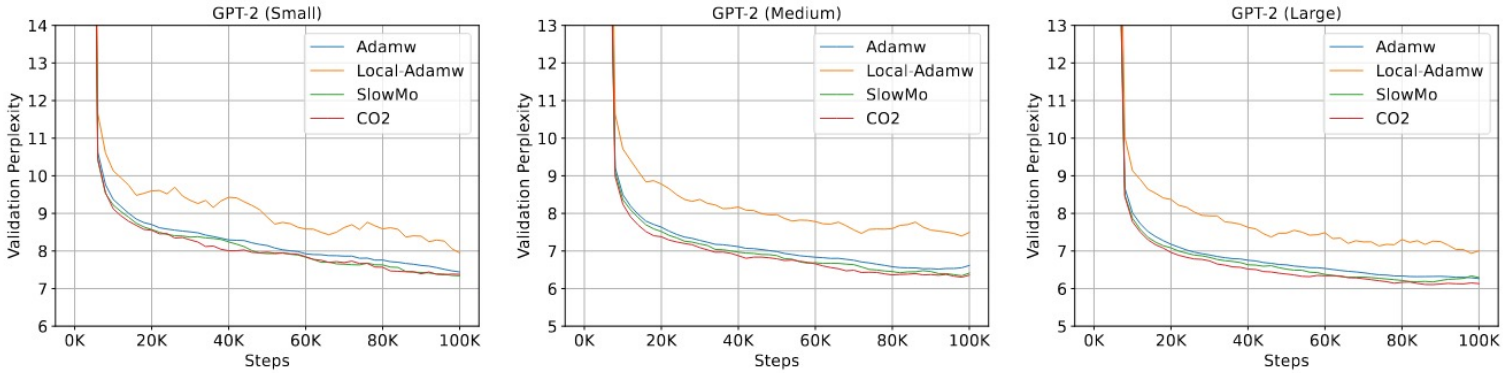
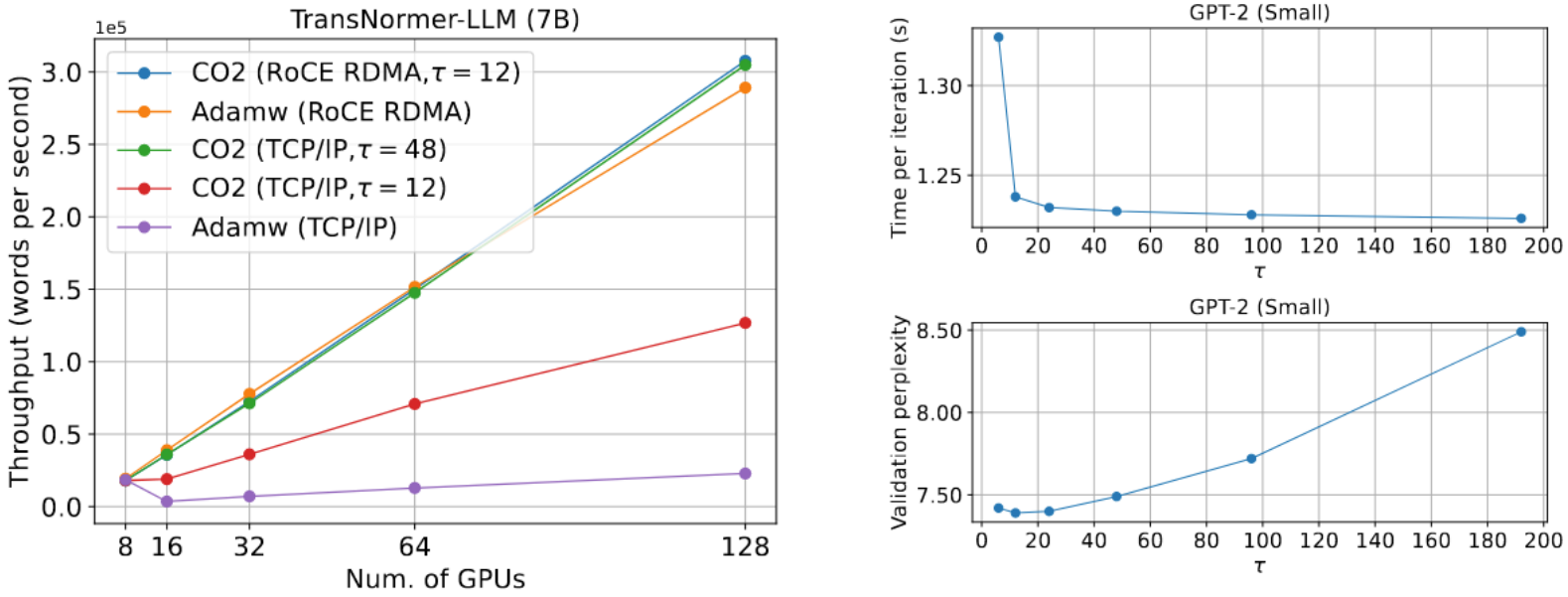


Figure 3: **Validation curves for autoregressive language tasks.** We train GPT-2 on OpenWebText for 100K steps in three sizes: 125M (Small), 355M (Medium), and 770M (Large). CO2 exhibits robust convergence and the best generalization performance when compared to other existing methods.

Scalability Results



(a) Scalability of CO2.

(b) Effects of τ.

Figure 4: (a): **Scalability of CO2.** Throughput (words/sec) results on distinctive inter-node network configurations are presented. CO2 exhibits pecfect 100% scalability on both configurations. (b): **Effects of τ.** Training speed and generalization performance results w.r.t. τ are presented. A larger value of τ leads higher communication efficiency but worse generalization behaviors.

Scalability Results

Table 3: **Quantitative scalability performance of CO2.** Throughput (words/sec) results of CO2 and Adamw from 1 to 16 DGX-A100 servers are presented. Scalability ratio records the scalability from 2 to 16 nodes to take into account the inter-node connections. With suitable configuration of τ, CO2 can reach the scalability ratio of 1, which outperforms the baseline Adamw.

Ethernet	Method	Throughput					Scalability Ratio ($16 \rightarrow 128$)
		8 GPUs	16 GPUs	32 GPUs	64 GPUs	128 GPUs	
RDMA	CO2 (τ=12)	17980 ($\pm$ 48)	35692 ($\pm$ 126)	72491 ($\pm$ 183)	150073 ($\pm$ 362)	307557 ($\pm$ 617)	1.08
	Adamw	19276 ($\pm$ 59)	38888 ($\pm$ 118)	77782 ($\pm$ 93)	151554 ($\pm$ 209)	289106 ($\pm$ 423)	0.93
TCP/IP	CO2 (τ=48)	18090 ($\pm$ 95)	35969 ($\pm$ 193)	71249 ($\pm$ 179)	147507 ($\pm$ 315)	304736 ($\pm$ 729)	1.06
	CO2 (τ=12)	17995 ($\pm$ 72)	18975 ($\pm$ 108)	36095 ($\pm$ 151)	70839 ($\pm$ 373)	129865 ($\pm$ 564)	0.86
	Adamw	18444 ($\pm$ 49)	3488 ($\pm$ 115)	7077 ($\pm$ 127)	12855 ($\pm$ 308)	22810 ($\pm$ 526)	0.82

Other Ablation Results

Table 4: **Ablation results on staleness gap penalty and outer momentum clipping.** We pre-train GPT-2 (Small) with 100K steps for ablation. Both train and validation perplexity results indicate that staleness gap penalty has a noteworthy improvement for convergence performance.

Model	Steps	Metric	CO2	CO2 w/o Penalty	CO2 w/o Clipping
GPT-2 (Small)	100K	Train PPL	7.36	7.52	7.39
		Validation PPL	7.39	7.56	7.42

Table 10: **Performance of staleness gap penalty on CyclicLR.** We pre-train GPT-2 (Small) with 100K steps for ablation. Both train and validation perplexity results indicate that staleness gap penalty is effective on CyclicLR schedule.

Method	Staleness Gap Penalty	Train PPL	Validation PPL
CO2 with CosineAnnealingLR	No	7.52	7.56
CO2 with CosineAnnealingLR	Yes	7.36	7.39
CO2 with CyclicLR	No	7.58	7.63
CO2 with CyclicLR	Yes	7.45	7.51

EXPERIMENTS

ICLR

More Details Of CV/NLP Results

Table 6: **Convergence performance on CV tasks.** CO2 performs better than other local methods with a clear margin and is comparable to standard optimizers such as SGD/Adamw. For IC and SS tasks, we present the throughput results in images/sec, the image resolution is 224×224; For PC task, we present the throughput results in point clouds/sec, each point cloud has 1024 points.

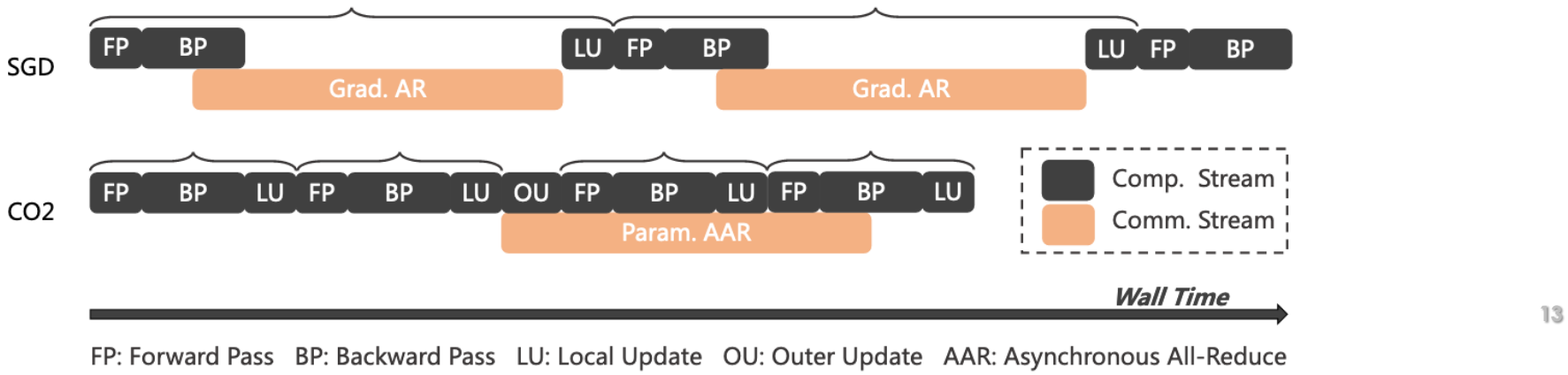
Task	Model	SGD (Adamw)	Local-SGD(Adamw)	Overlap-Local-SGD(Adamw)	SlowMo	CO2
		Acc / Thpt	Acc / Thpt / τ	Acc / Thpt / τ	Acc / Thpt / τ	Acc / Thpt / τ
IC	ResNet-50	76.92 / 108739	75.57 / 108758 / 1	76.28 / 108765 / 1	77.12 / 108741 / 1	77.14 / 108753 / 1
	ViT (Base)	81.33 / 39422	78.43 / 39512 / 3	78.04 / 39511 / 3	79.83 / 39509 / 3	80.95 / 39533 / 3
	VVT (Large)	83.64 / 44375	81.09 / 44390 / 1	80.33 / 44392 / 1	82.75 / 44376 / 1	83.38 / 44387 / 1
SS	VVT (Large)	47.82 / 5384	44.25 / 5528 / 6	45.21 / 5545 / 6	47.51 / 5521 / 6	47.80 / 5562 / 6
PC	Point-MAE	68.56 / 5859	64.25 / 5931 / 3	63.78 / 5950 / 3	68.69 / 5904 / 3	68.89 / 5956 / 3

Table 7: **Convergence performance on NLP tasks.** Quantitative perplexity (PPL) results for GPT-2, TransNormer-LLM and RoBERTa are presented. CO2 shows lower perplexity (lower is better) than the baseline Adamw in all these experiments. For ALM and BLM tasks, we present the throughput results in words/sec. TN-LLM represents TransNormer-LLM.

Task	Model	Adamw	Local-Adamw	Overlap-Local-Adamw	SlowMo	CO2
		Acc / Thpt	Acc / Thpt / τ	Acc / Thpt / τ	Acc / Thpt / τ	Acc / Thpt / τ
ALM	GPT-2 (Small)	7.44 / 6.543e6	7.95 / 6.556e6 / 3	8.11 / 6.556e6 / 3	7.34 / 6.554e6 / 3	7.37 / 6.556e6 / 3
	GPT-2 (Medium)	6.61 / 2.084e6	7.49 / 2.094e6 / 3	7.26 / 2.092e6 / 3	6.41 / 2.091e6 / 3	6.36 / 2.092e6 / 3
	GPT-2 (Large)	6.26 / 1.052e6	7.00 / 1.059e6 / 6	7.18 / 1.057e6 / 6	6.29 / 1.053e6 / 6	6.13 / 1.056e6 / 6
	TN-LLM (7B)	16.82 / 0.281e6	18.63 / 0.303e6 / 12	17.83 / 0.306e6 / 12	16.95 / 0.301e6 / 12	16.78 / 0.308e6 / 12
BLM	RoBERTa (Large)	3.96 / 2262	4.38 / 2815 / 6	4.52 / 2877 / 6	3.98 / 2794 / 6	3.95 / 2892 / 6

SUMMARY

1. With local updating and Single-step asynchronous communication, communication tail can be fully overlapped by multi-step local computation steps, which results 100% scalability.
2. CO2 even performs well on low-bandwidth large clusters, increasing tau to enable more overlap.
3. With staleness gap penalty and outer momentum clipping, CO2 shows good convergence, generalization and training stability.
4. CO2 can integrate with ZeRO-series optimizers to reduce memory usage for large-scale model training.



THANK YOU FOR YOUR TIME!

Check out the CO2 code at:

https://github.com/OpenNLPLab/CO2

https://github.com/weigao266/fairscale-CO2

Connect with me at:

https://twitter.com/sunweigao

https://github.com/weigao266

Add Me at WeChat: