Mila

MuLoCo: Muon is a practical inner optimizer for DiLoCo

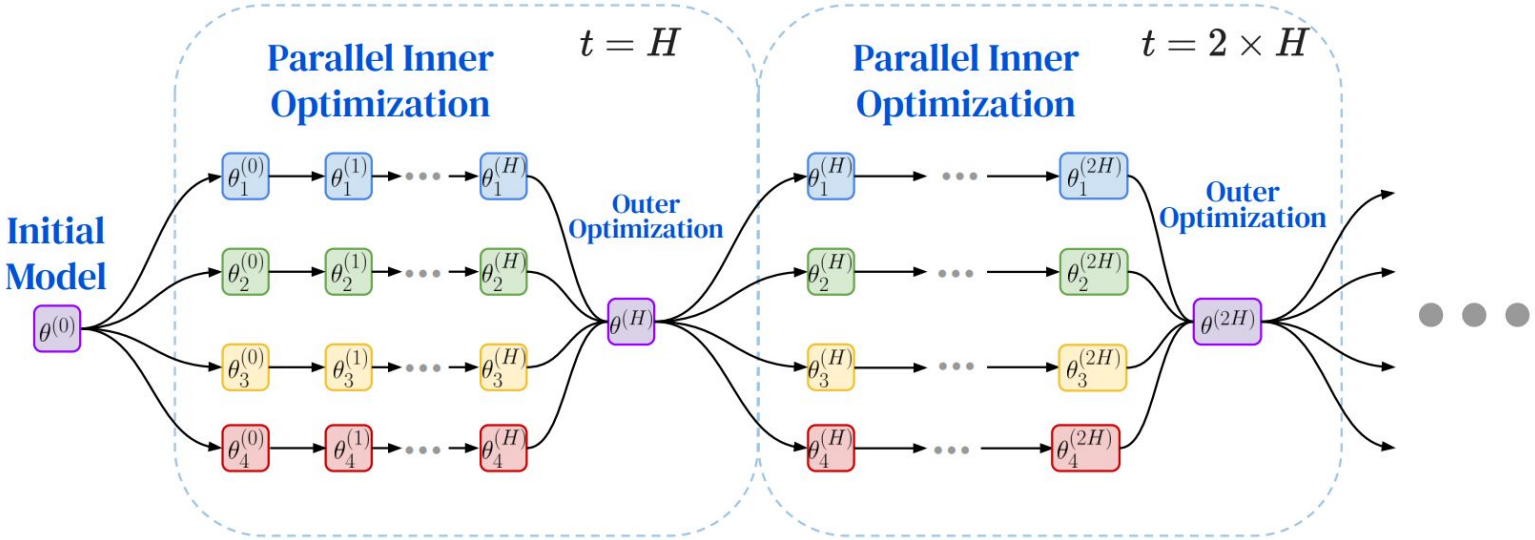
Presented by: Benjamin Thérien

Collaborators: Xiaolong Huang, Aaron Defazio, Irina Rish, Eugene Belilovsky

Background

DiLoCo

(Charles et al. 2025)



- Trains many models in parallel
- Synchronize models after H steps
- **Communication is reduced by a factor H**

DiLoCo's biggest problem

(Charles et al. 2025)

N	DP	DiLoCo			
		$M = 1$	$M = 2$	$M = 4$	$M = 8$
35M	3.485	3.482 (−0.1%)	3.508 (+0.7%)	3.554 (+2.0%)	3.621 (+3.9%)
90M	3.167	3.162 (−0.1%)	3.182 (+0.5%)	3.213 (+1.5%)	3.265 (+3.1%)
180M	2.950	2.943 (−0.2%)	2.957 (+0.3%)	2.981 (+1.1%)	3.019 (+2.3%)

- M is the number of workers
- Performance decreases as M increases

Muon

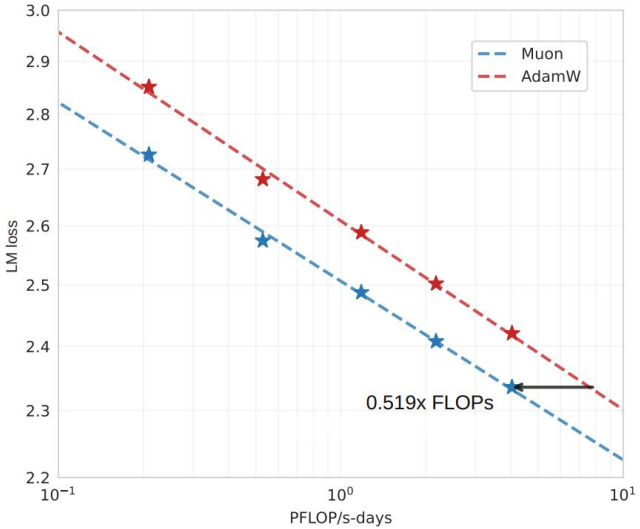
(Jordan et al. 2024)

Algorithm 2 Muon

Require: Learning rate η, momentum μ

1: Initialize $B_0 \leftarrow 0$
2: **for** $t = 1, \ldots$ **do**
3: Compute gradient $G_t \leftarrow \nabla_\theta \mathcal{L}_t(\theta_{t-1})$
4: $B_t \leftarrow \mu B_{t-1} + G_t$
5: $O_t \leftarrow \text{NewtonSchulz5}(B_t)$
6: Update parameters $\theta_t \leftarrow \theta_{t-1} - \eta O_t$
7: **end for**
8: **return** θ_t

(Moonshot AI 2025)



- **(LEFT)** Muon is a second order optimization algorithm
- Muon leverages **NewtonShulz iteration** to orthonormalize the gradient
- **(RIGHT)** Muon scales better than AdamW

Mila

Motivation

Should I use DiLoCo or MuLoCo or DP Muon?

Batch size and communication cost are linked

- Bigger batch = fewer communications
- within datacenter small batch=low comms cost because of fast interconnect

Mila

Empirical study

Model scaling prescription

Model Parameters	Transformer Layers	Attention Heads	QKV Dimension	Hidden Dimension	Token Budget	HP Sweep
150M	6	4	512	1,408	3B	✓
416M	12	8	1,024	2,816	8.16B	✓
914M	18	12	1,536	4224	18.12B	✓
1.76B	24	16	2,048	5,632	35.23B	✓
3.07B	30	20	2,560	7,040	61.4B	✓
15.23B	54	36	4,608	12,672	304.6B	×

- we use a 20 tokens per parameter (TPP) ratio

Mila

Results

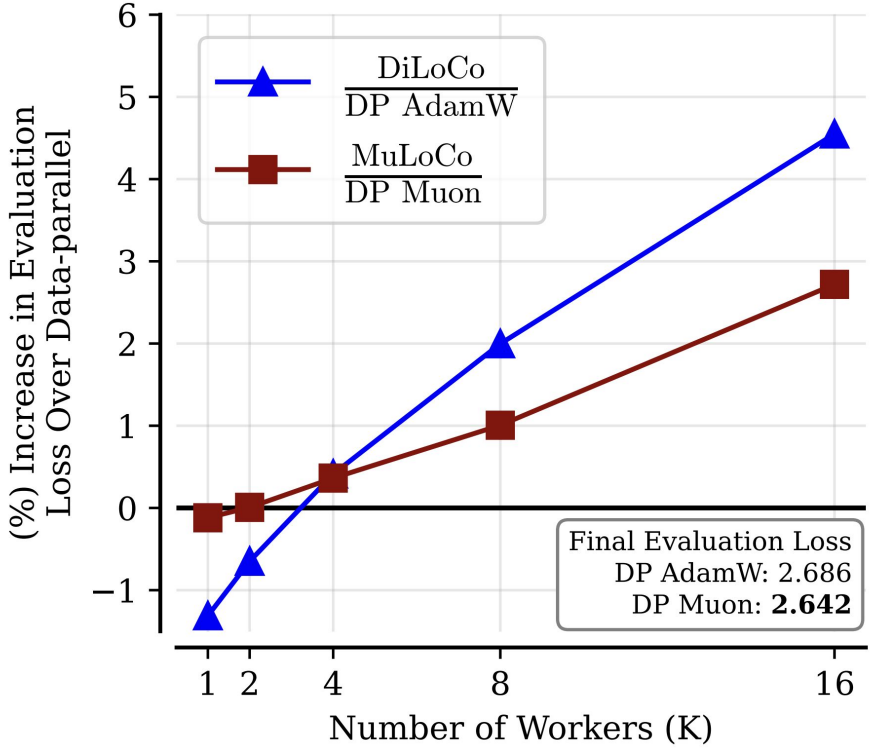
MuLoCo outperforms DiLoCo in absolute performance

	Method	**150M**	**416M**	**914M**	**1.8B**	**3.1B**	**15B** (No Sweep)
DP	Muon	3.124	2.641	2.402	2.246	2.128	**1.875**
	AdamW	3.158	2.682	2.440	2.266	2.145	1.901
K=1	MuLoCo	**3.120** (−0.1%)	**2.638** (−0.1%)	**2.400** (−0.1%)	**2.238** (−0.4%)	**2.122** (−0.3%)	1.878 (+0.2%)
	DiLoCo	3.142 (−0.5%)	2.650 (−1.2%)	2.411 (−1.2%)	2.265 (−0.1%)	2.136 (−0.4%)	1.877 (−1.3%)
K=2	MuLoCo	3.130 (+0.2%)	2.642 (+0.0%)	2.402 (+0.0%)	2.245 (+0.0%)	2.127 (−0.1%)	–
	DiLoCo	3.159 (+0.0%)	2.668 (−0.5%)	2.428 (−0.5%)	2.275 (+0.4%)	2.155 (+0.5%)	–
•••							
K=16	MuLoCo	3.222 (+3.1%)	2.713 (+2.8%)	2.448 (+1.9%)	2.291 (+2.0%)	2.165 (+1.7%)	1.902 (+1.4%)
	DiLoCo	3.326 (+5.3%)	2.808 (+4.7%)	2.522 (+3.4%)	2.348 (+3.6%)	2.215 (+3.3%)	1.910 (+0.4%)

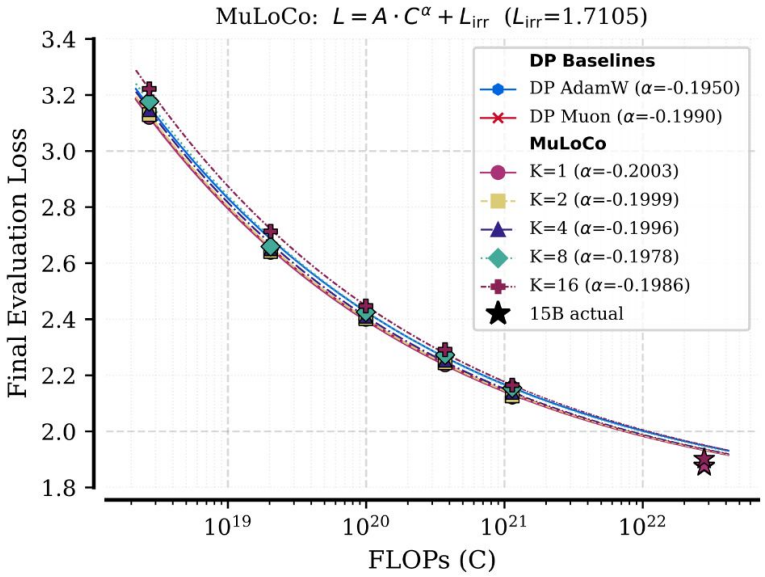
- We pre-train 150M-3B at 20TPP
- Each datapoint is swept across batch size, LR, WD, OLR, ILR
- MuLoCo's absolute performance is always better (**expected**)

MuLoCo improves worker scaling

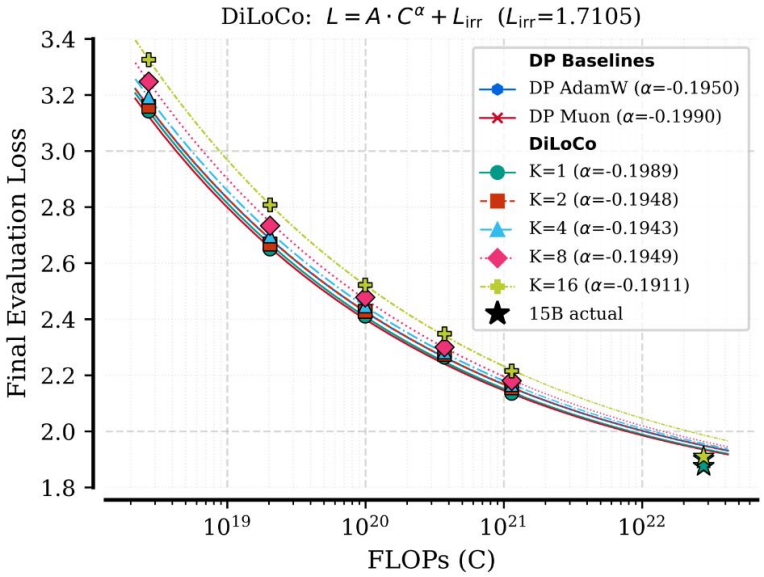
- MuLoCo outperforms DiLoCo relative to their data-parallel baselines as the number of workers (K) increases.



MuLoCo's strong worker scaling is preseved at scale

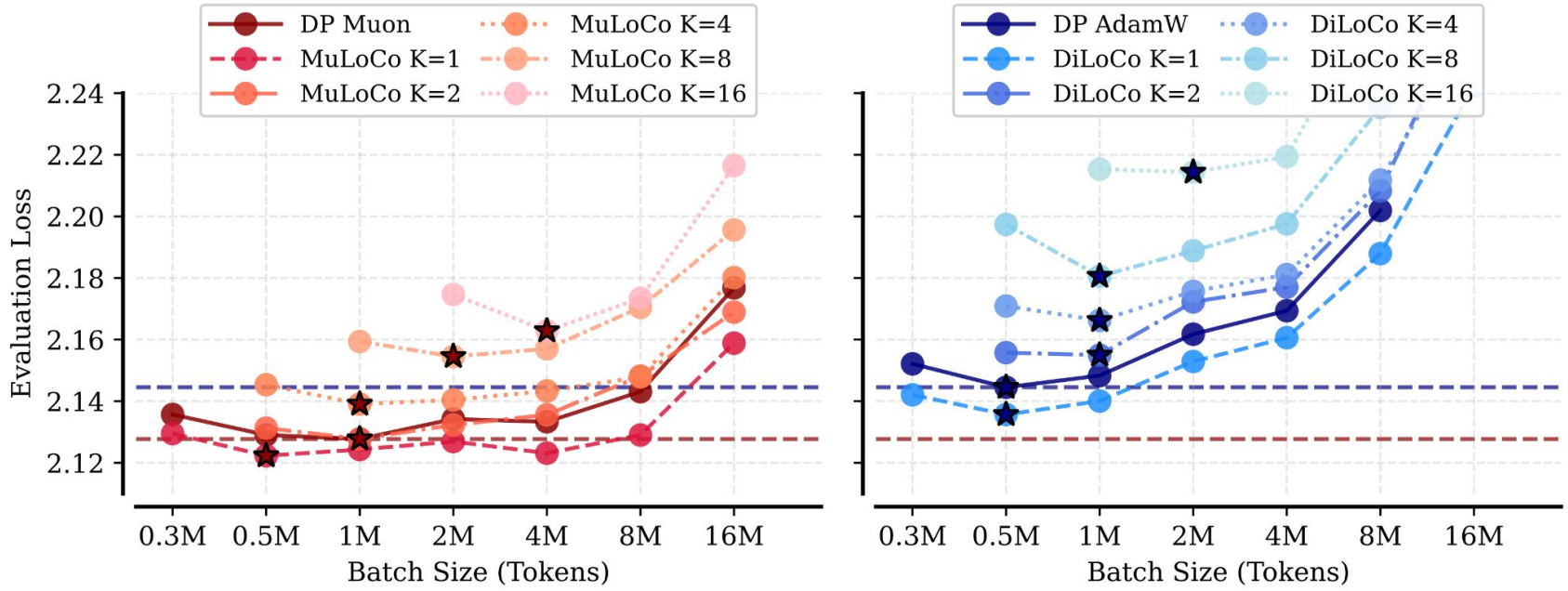


(a) MuLoCo scaling law fit in compute



(b) DiLoCo scaling law fit in compute

MuLoCo's critical batch size is much larger than DiLoCo



- Large batch sizes warrant cross datacenter training

Mila

Conclusion

Conclusion

- MuLoCo scales better than DiLoCo as worker count increases.
- MuLoCo has a much larger critical batch size than DiLoCo, thus warranting cross-datacenter training.
- See out paper for more results!