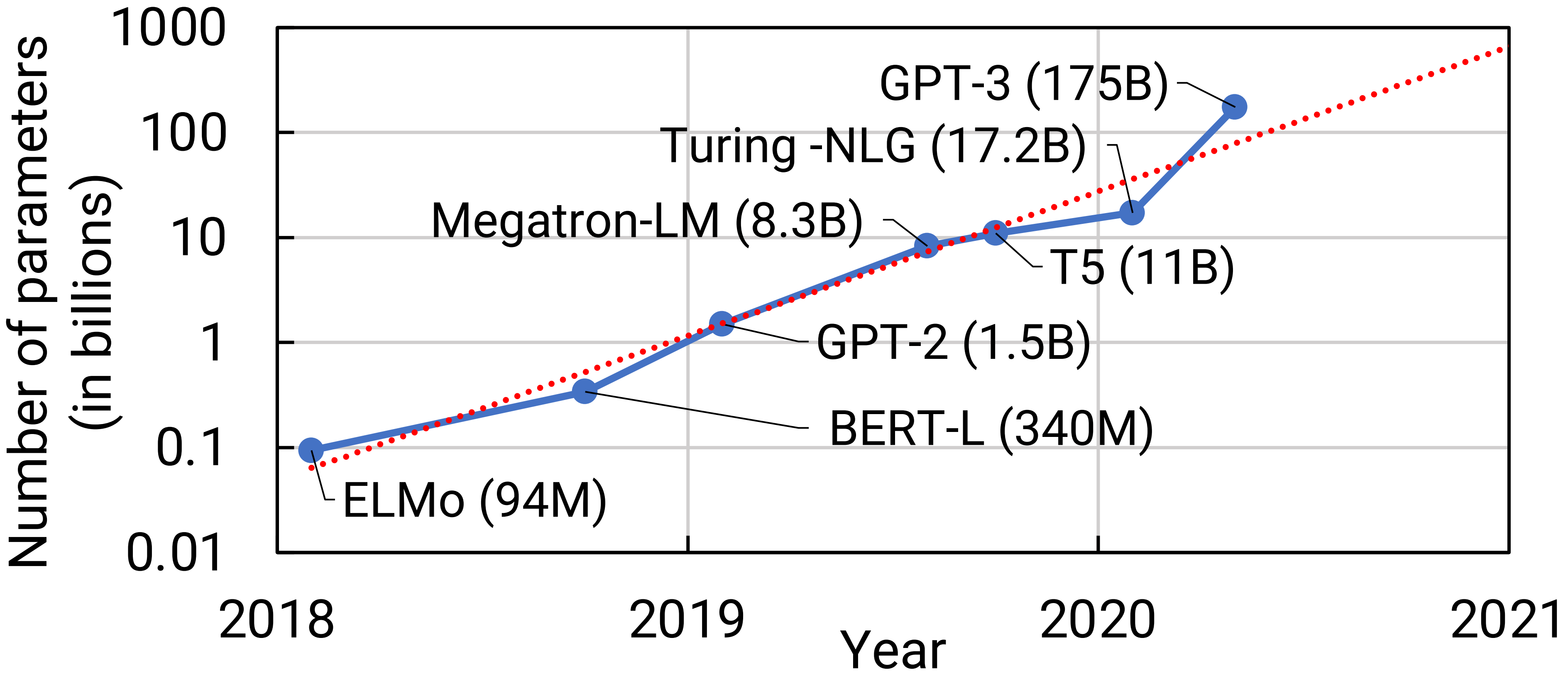
Memory-Efficient Pipeline-Parallel DNN Training

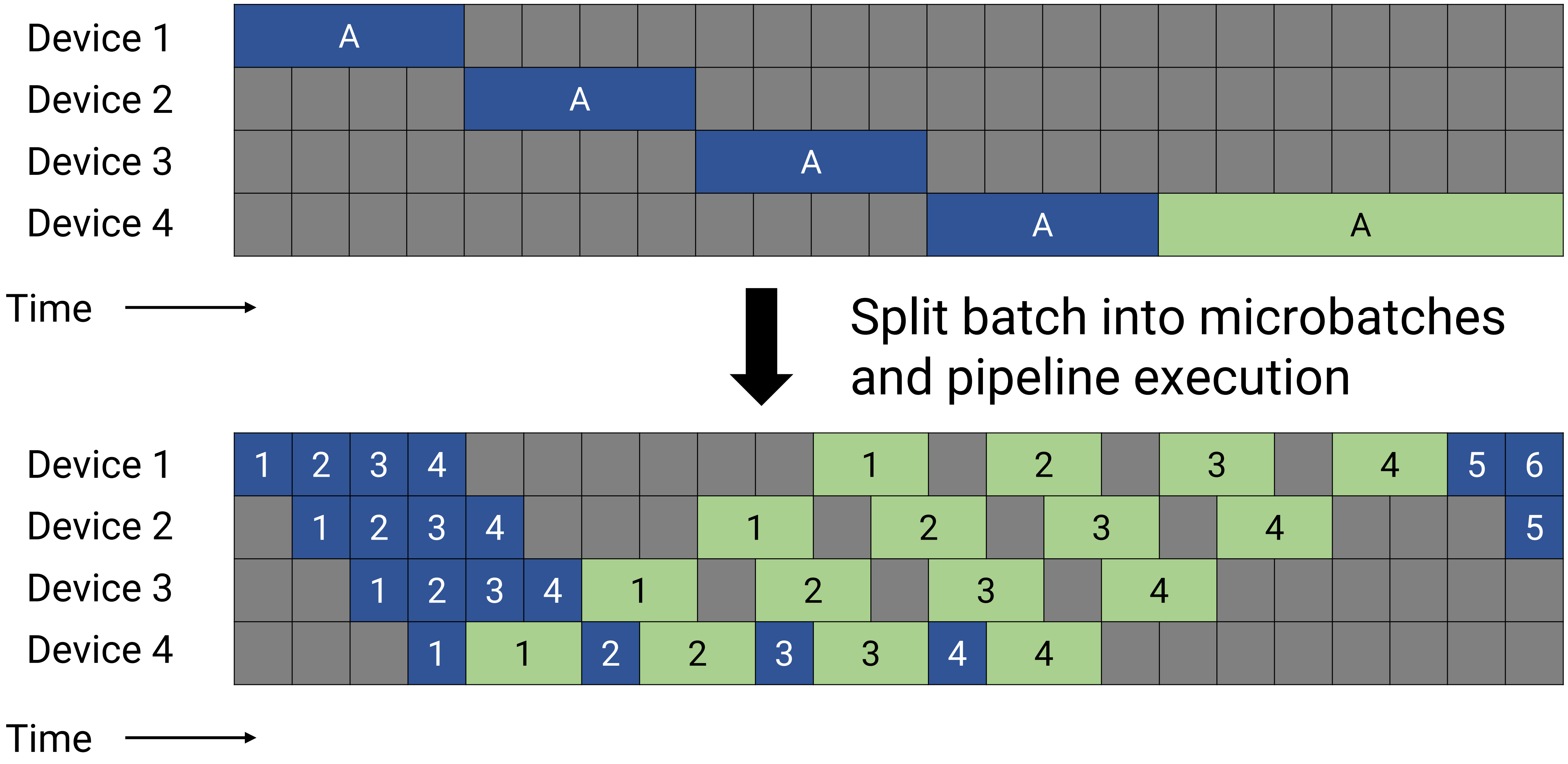
Deepak Narayanan§, Amar Phanishayee★, Kaiyu Shi†, Xie Chen†, Matei Zaharia§

★ **Microsoft Research** † **Microsoft** § **Stanford University**

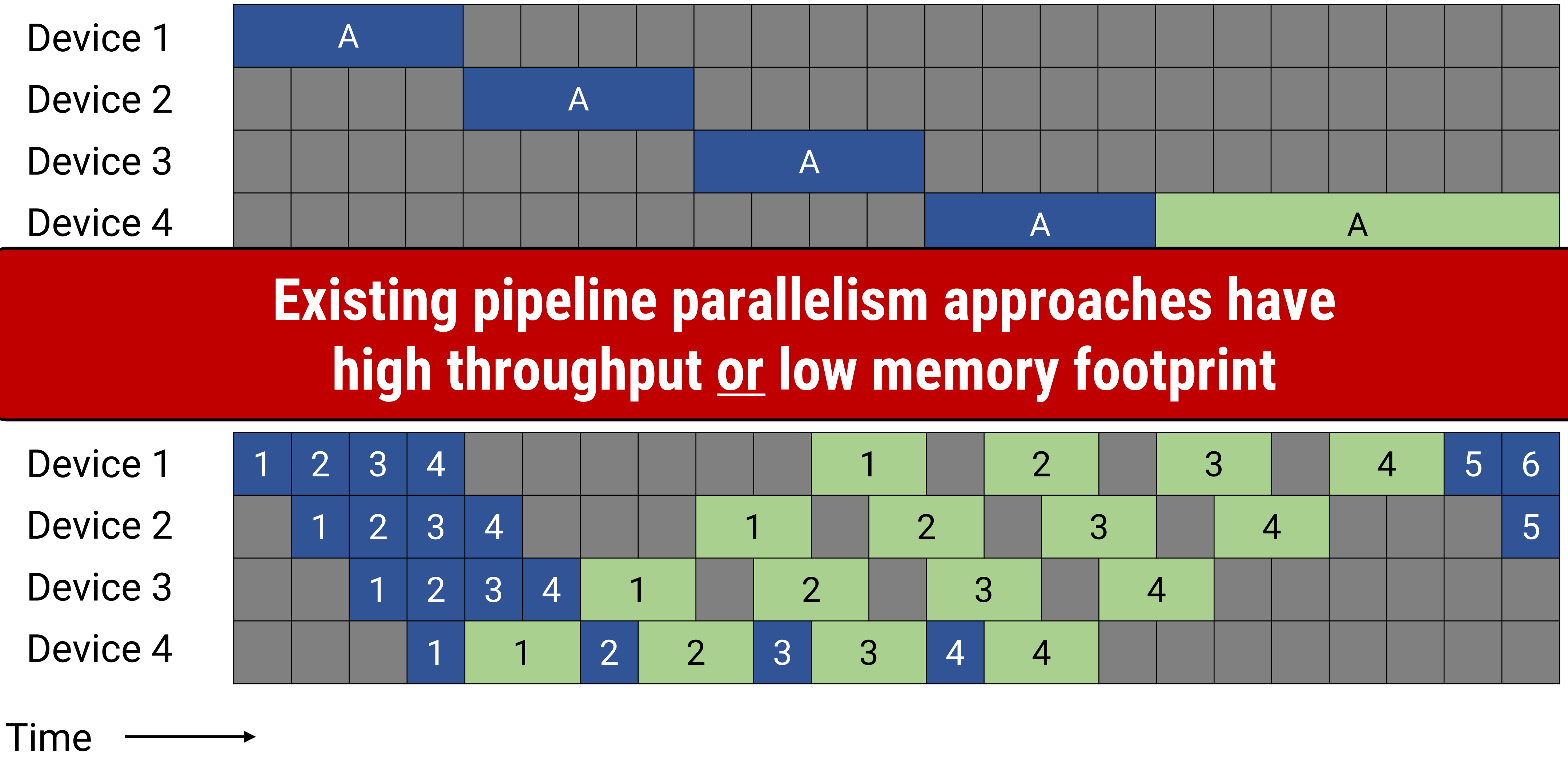
State-of-the-art models are becoming larger!



Model parallelism can alleviate memory pressure



Model parallelism can alleviate memory pressure



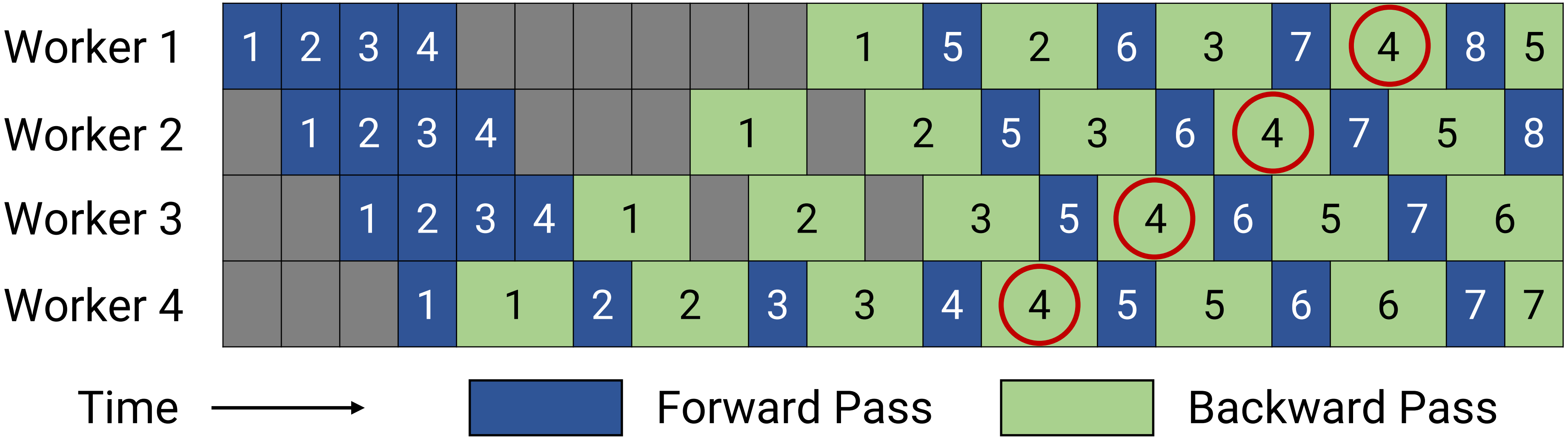
Existing pipeline parallelism approaches have high throughput or low memory footprint

This work: memory-efficient pipeline parallelism

- High throughput
- Low memory footprint
- Strong weight update semantics (same weight version used in both the forward and backward pass for a given batch)

Double-buffered weight updates

Stashed state



$W_1^{(4)}, W_1^{(\text{init.})}$

$W_2^{(4)}, W_2^{(\text{init.})}$

$W_3^{(4)}, W_3^{(\text{init.})}$

$W_4^{(4)}, W_4^{(\text{init.})}$

$W_i^{(j)}$ — (j) → Version number (incorporates gradients from inputs $\leq j$); i → Stage or worker ID

Generate a new weight version every 4 inputs (1→4, 5→8, etc.)

Semantics of double-buffered weight updates

- Vanilla weight update semantics:

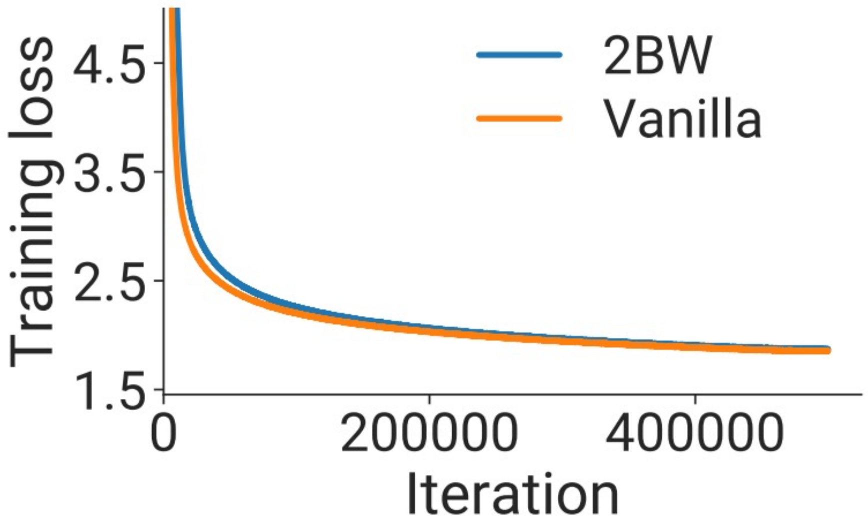
$$W^{(t+1)} = W^{(t)} - \nu \cdot \nabla f(W^{(t)})$$

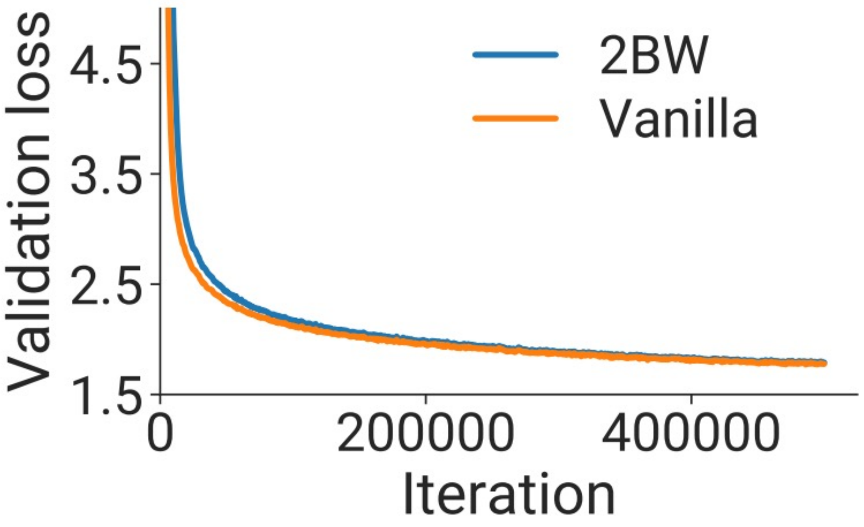
- Weight update semantics with 2BW almost **unchanged** (note additional delay term of 1 in gradient computation):

$$W^{(t+1)} = W^{(t)} - \nu \cdot \nabla f(W^{(t-1)})$$

Evaluation

2BW has weight update semantics similar to vanilla

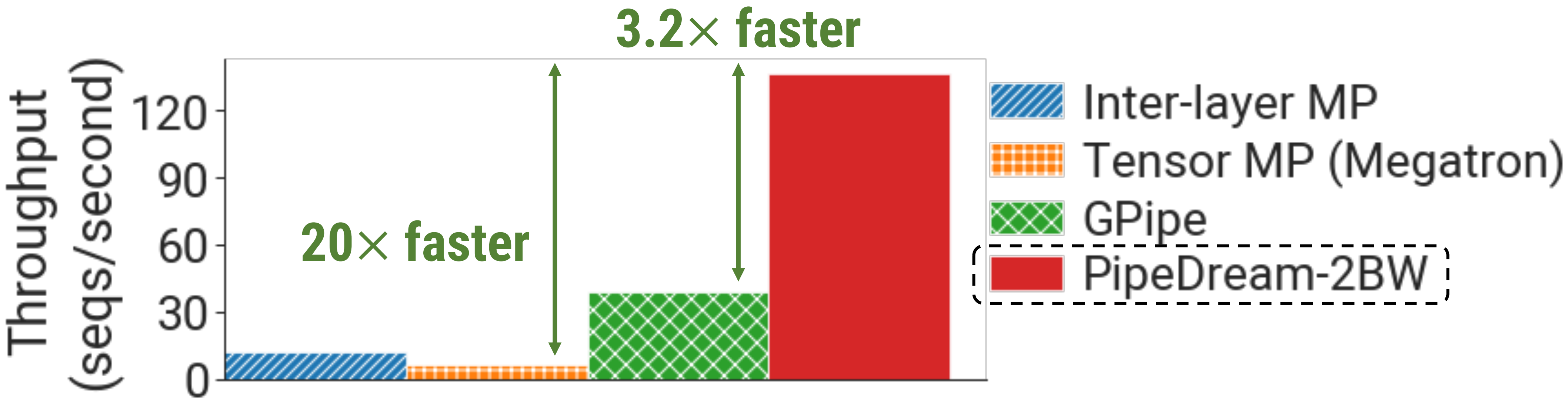




BERT model with 355 million parameters

Accuracy on downstream MNLI and RACE tasks unchanged

PipeDream-2BW is faster than baselines



8 p3.16xlarge instances (64 GPUs) on AWS
3.8-billion parameter GPT model

Conclusion

- Pipeline parallelism can be used to train large models, but can suffer from **low resource utilization** or **high memory footprint**
- PipeDream-2BW accelerates training by up to **3.2x** compared to baselines that use pipelining, and **20x** compared to other baselines

Code open sourced at
https://github.com/msr-fiddle/pipedream/tree/pipedream_2bw

https://cs.stanford.edu/~deepakn/ **deepakn@cs.stanford.edu**