ICML
International Conference
On Machine Learning

TSPipe: Learn from Teacher Faster with Pipelines

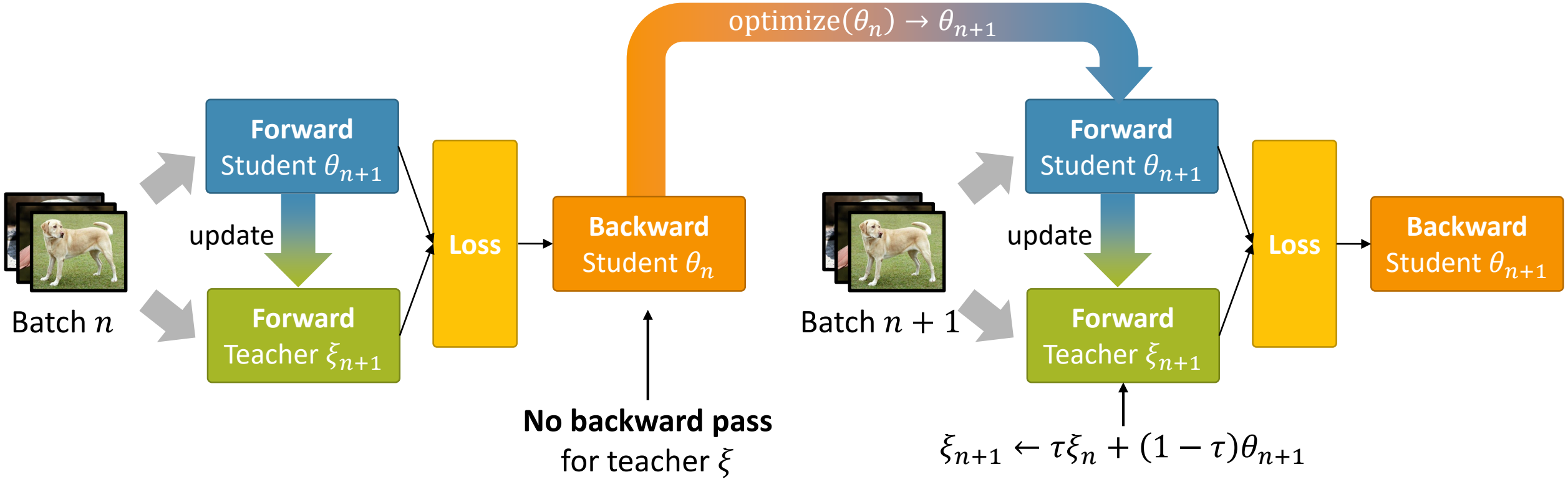
Hwijoon Lim, Yechan Kim, Sukmin Yun,

Jinwoo Shin, Dongsu Han

KAIST

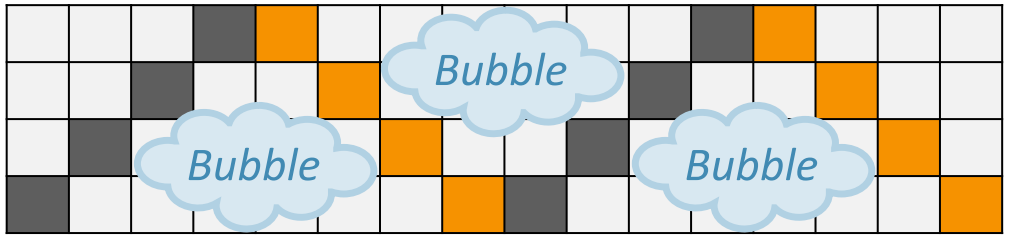
Teacher-Student (TS) Framework

- Teacher-student (TS) framework is commonly adopted in Knowledge Distillation (KD)
- Also adopted by many momentum-based Self-Supervised Learning (SSL) networks
  - Teacher network ξ_n is slowly updated as an exponential moving average of student θ_n

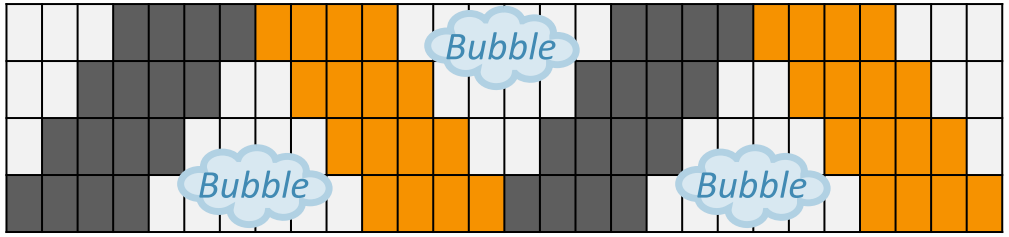


Model Parallelism and Pipeline Parallelism

- Some large models cannot be trained as a whole, even with a cutting-edge GPU
- Model Parallelism
  - **split a model** into muiltiple partitions and train with multiple GPUs
  - **serious GPU under-utilization** due to the dependency between partitions
- Pipeline Parallelism
  - pipelines computation of each batch for better GPU utilization
  - Approaches that preserve training semantics (e.g. GPipe) **fail to fully utilize GPUs**
  - Approaches that achieve higher utilization **incur overheads** (e.g. memory, accuracy)



Inter-Layer Model Parallelism

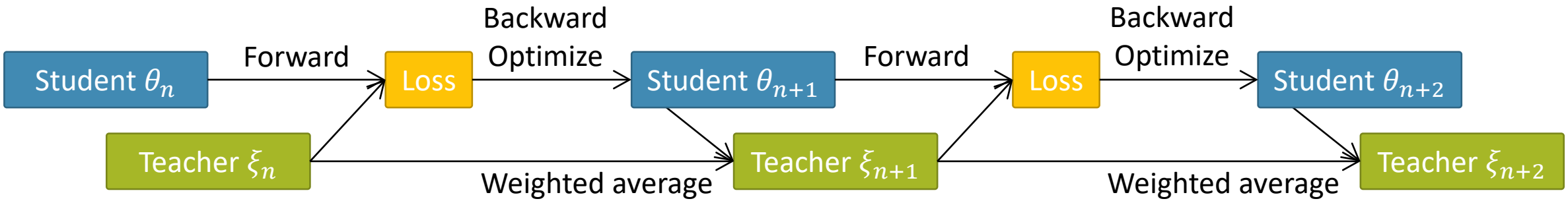


Pipeline Parallelism (GPipe)

Existing pipeline parallelism schemes cannot fully utilize GPU without tradeoffs.

Challenge

- **Can we fully schedule the computations** despite the dependency between them?
  - To compute the teacher ξn+1, we need to wait for student θn+1 to be computed
- **Can we eliminate pipeline bubbles** by inserting computations while GPUs are idle?
  - Reordering computations may require activation stashing for gradient calculation

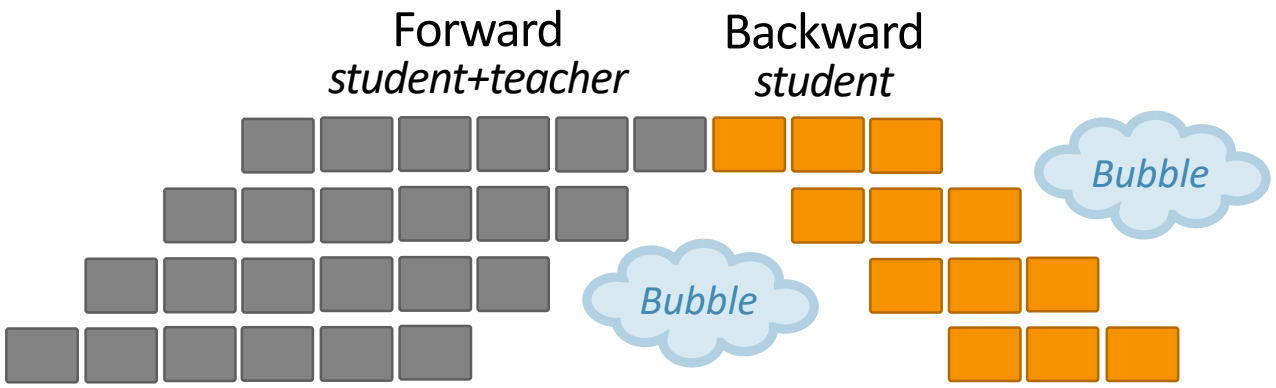


Teacher network ***does not require a backward pass***

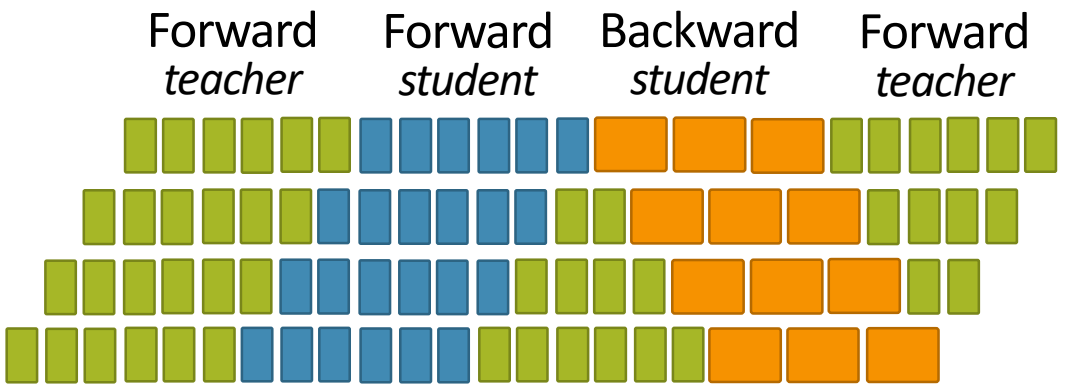
→ Teacher network's forward pass can be **scheduled more leniently** without activation stashing

TSPipe Design

1. Separates the scheduling of student and teacher network from its design
2. Interleaves teacher's forward pass between the computation of student



Gpipe [Huang '19]

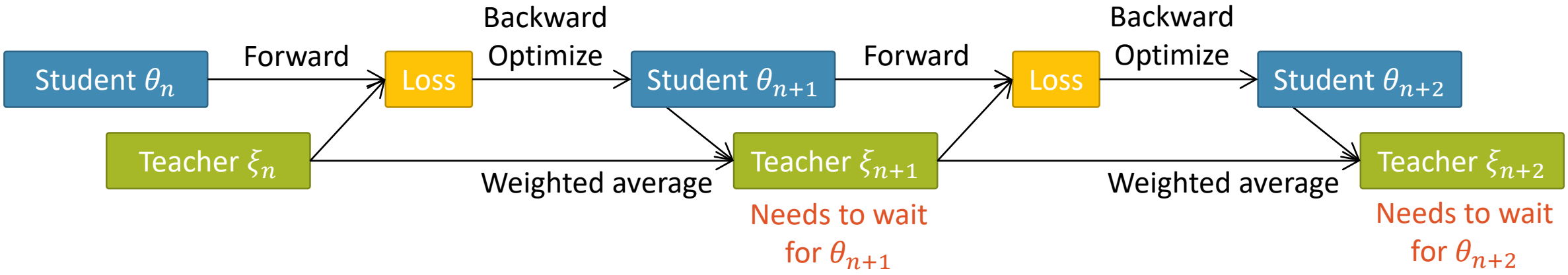


TSPipe (Ours)

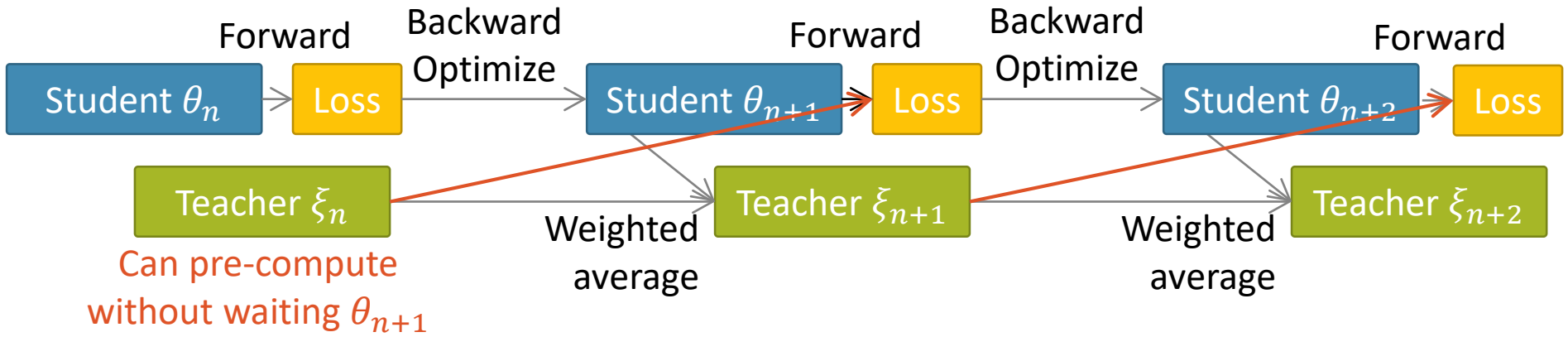
TSPipe fully schedules GPU pipeline by interleaving computation in pipeline bubble.

TSPipe Design - Attaining high model accuracy

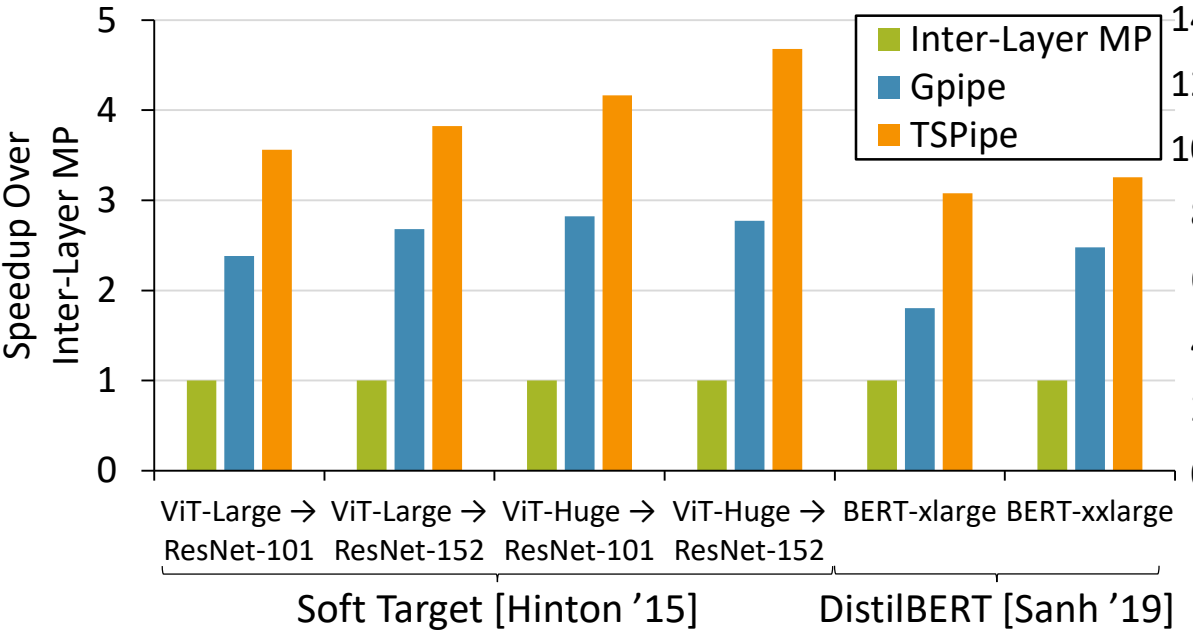
- Original training semantic: $\theta_{n+1} \leftarrow \text{optimizer}(\theta_n, \nabla_{\theta_n}\mathcal{L}_{\theta_n,\boldsymbol{\xi_n}}, \eta)$



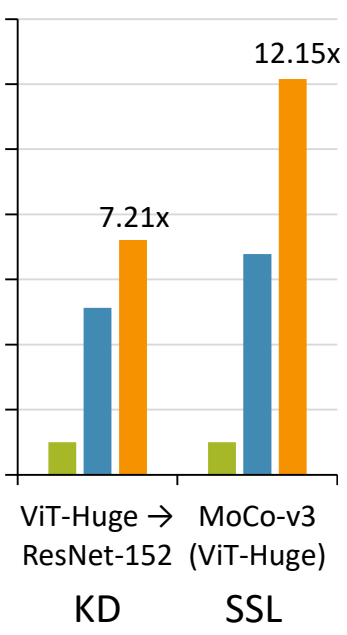
- As $\xi_{n-1} \approx \xi_n$, TSPipe uses: $\theta_{n+1} \leftarrow \text{optimizer}(\theta_n, \nabla_{\theta_n}\mathcal{L}_{\theta_n,\boldsymbol{\xi_{n-1}}}, \eta)$
  - ***Asymmetric parameter update***: Use stale parameter for only teacher network ξ_n



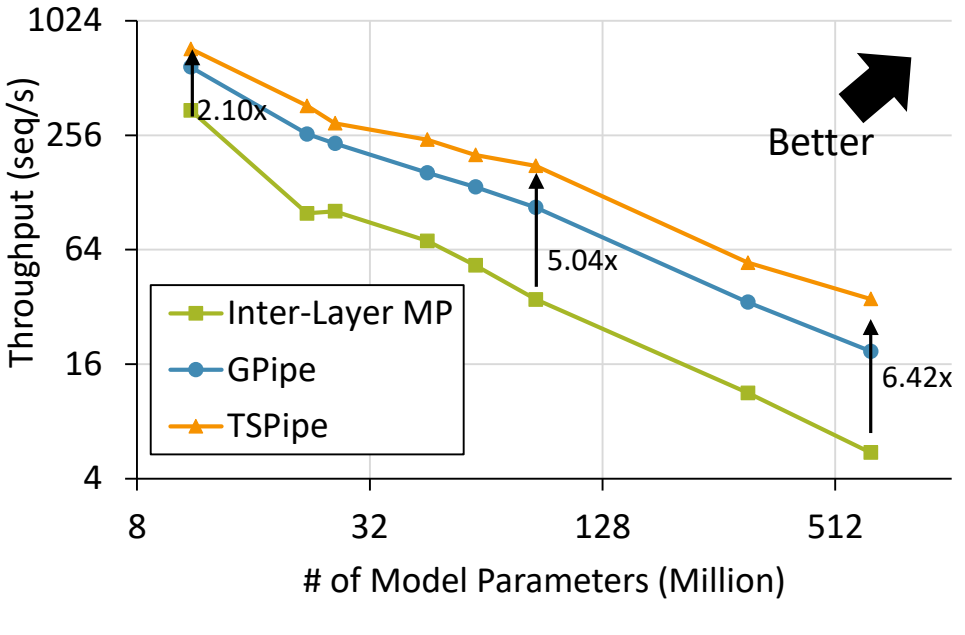
Evaluation – Speedup



Knowledge Distillation with 8 GPUs



16 GPUs

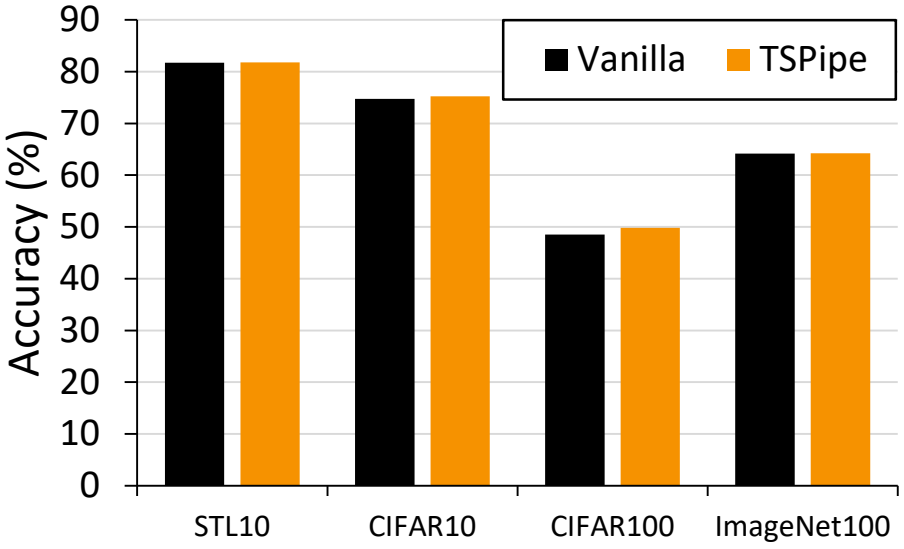


Self-supervised Learning with 8 GPUs

TSPipe achieves up to 6.42x (8 GPUs) and 12.15x (16 GPUs) speedup compared to Inter-layer MP.

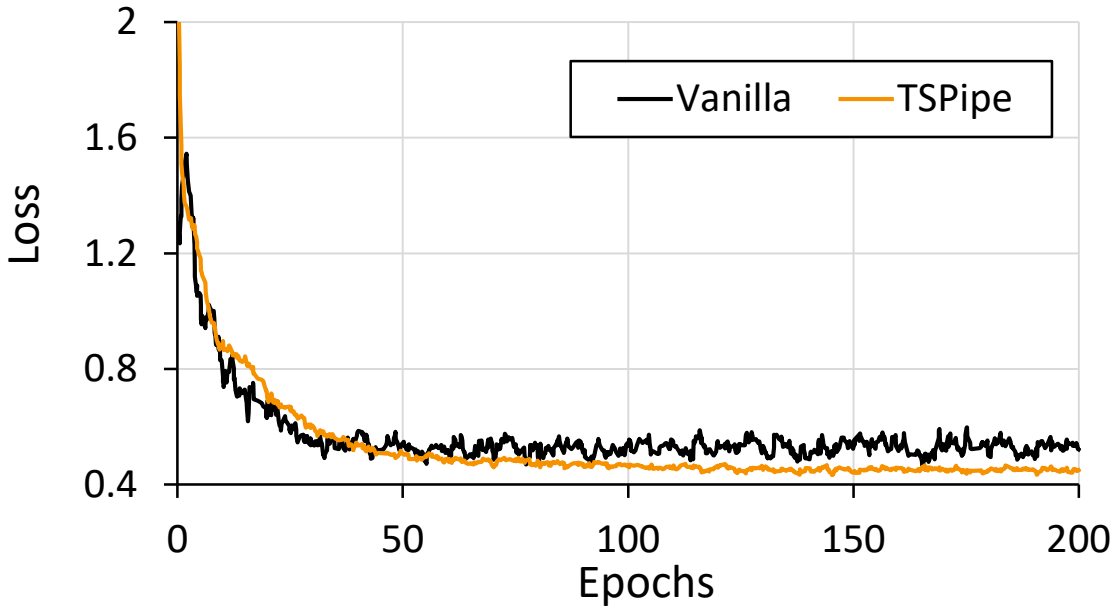
Evaluation – Accuracy

- Asymmetric parameter update mitigates model accuracy drop.
  - Up to 5.9%p accuracy degradation without Asymmetric parameter update



Linear evaluation Accuracy (Top-1)

BYOL with ResNet-18



Training loss curve

BYOL with ResNet-50

TSPipe achieves speedup without loss of model accuracy.

Conclusion

- TSPipe is a framework that enables faster training of large models with the TS framework without risking any performance degradation of the model.
- TSPipe utilizes 100% of GPU pipelines for training KD and SSL with momentum networks, leveraging the properties of the TS framework.