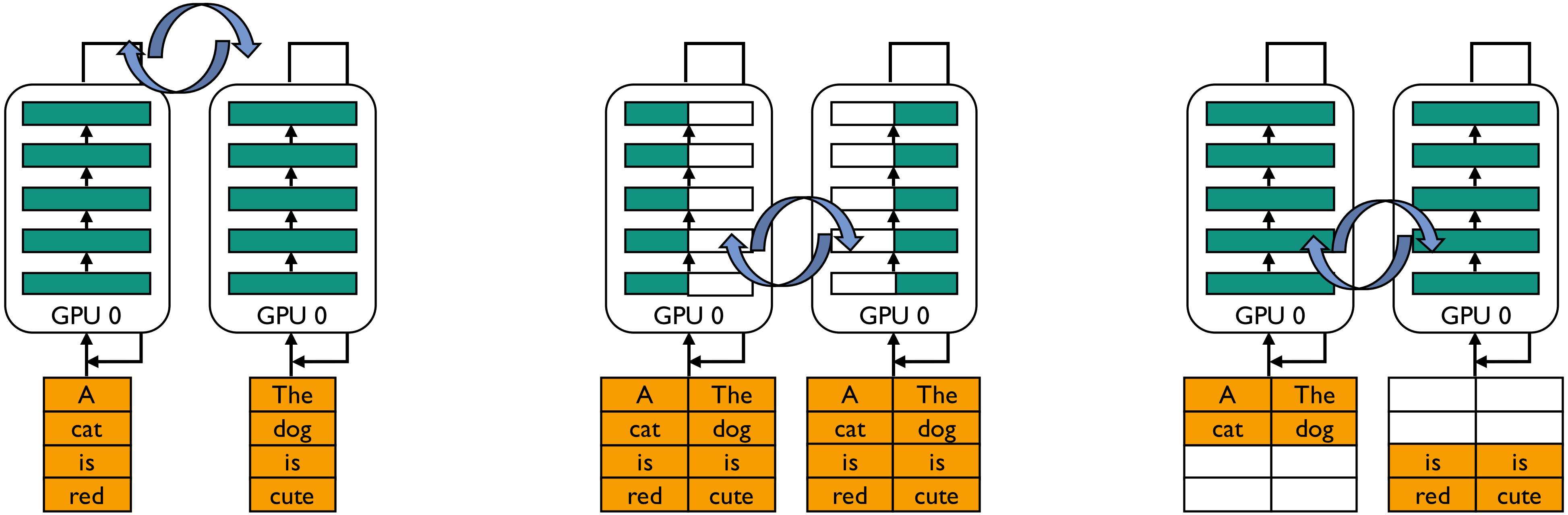
AutoCCL: Automated Collective Communication Tuning for Accelerating Distributed and Parallel DNN Training

Guanbin Xu[1], Zhihao Le[1], Yinhe Chen[1], Zhiqi Lin[1], Zewen Jin[1]
Youshan Miao[2], Cheng Li[1 3]

University of Science and Technology of China[1], Microsoft Research[2]
Hefei Comprehensive National Science Center[3]

1 2 3

Distributed DNN Training



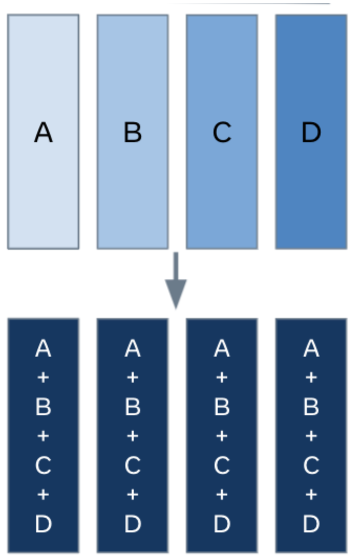
Data Parallelism (DP) | Tensor Parallelism (TP) | Sequence Parallelism (SP)

AllReduce | **AllReduce** | **ReduceScatter, AllGather**

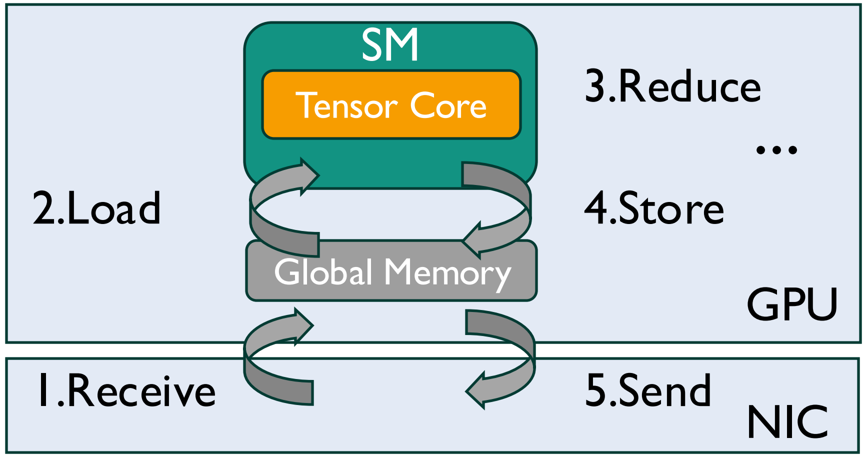
Collective Communication widely adopted

What's Collective Communication

AllReduce:Tree-, Ring-, …



…



Sync with all processes ***Receive-Load-Reduce-Store-Send* on each process**

Infrastructure for DNN Communication

PyTorch TensorFlow …

Share the Collective Communication APIs

NCCL[1]	HCCL[2]	RCCL[3]	…
NVIDIA GPU	HUAWEI NPU	AMD GPU	

- **Foundational library: almost every distributed DNN job**
- **Active community: 3.7k star, 2015 - now**
- **Highly optimized: 57 releases, 900 forks**

[1] NVIDIA Collective Communication Library, https://github.com/nvidia/nccl
[2] Huawei Collective Communication Library, https://gitee.com/ascend/cann-hccl
[3] ROCm Collectives Communication Library, https://github.com/ROCm/rccl
[4] NCCL community data as of Apr. 23, 2025

Collective Communication is still Expensive

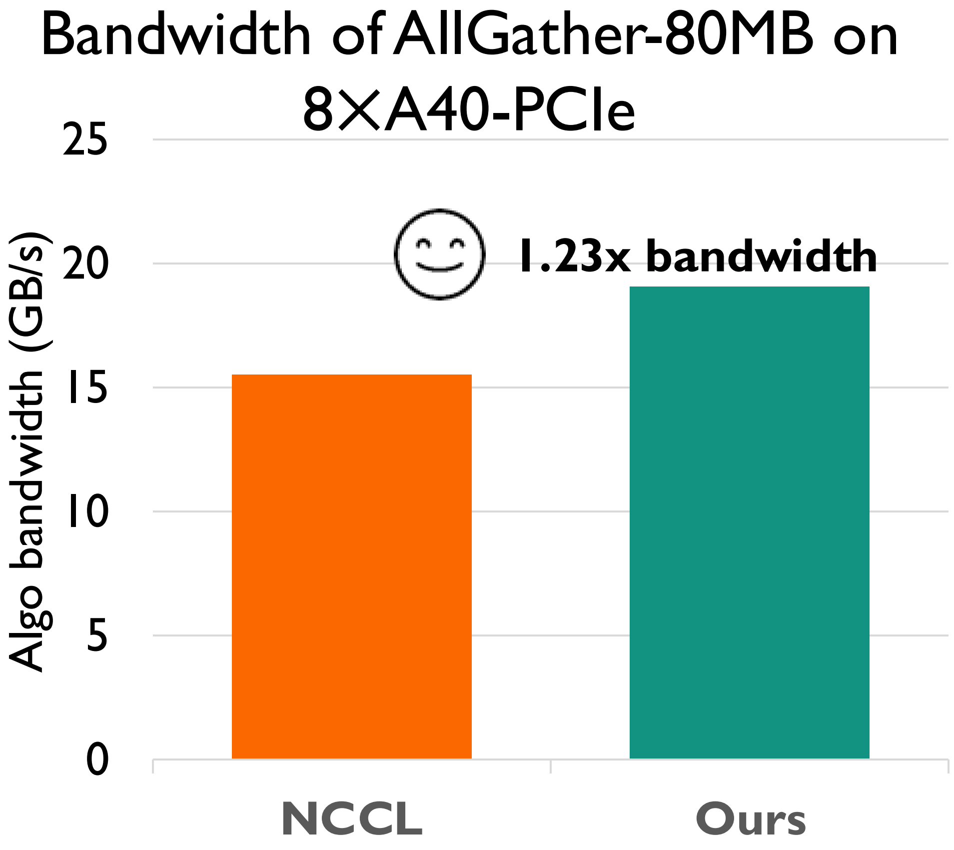
Communication is the bottleneck: up to 42% of time cost [1,2]

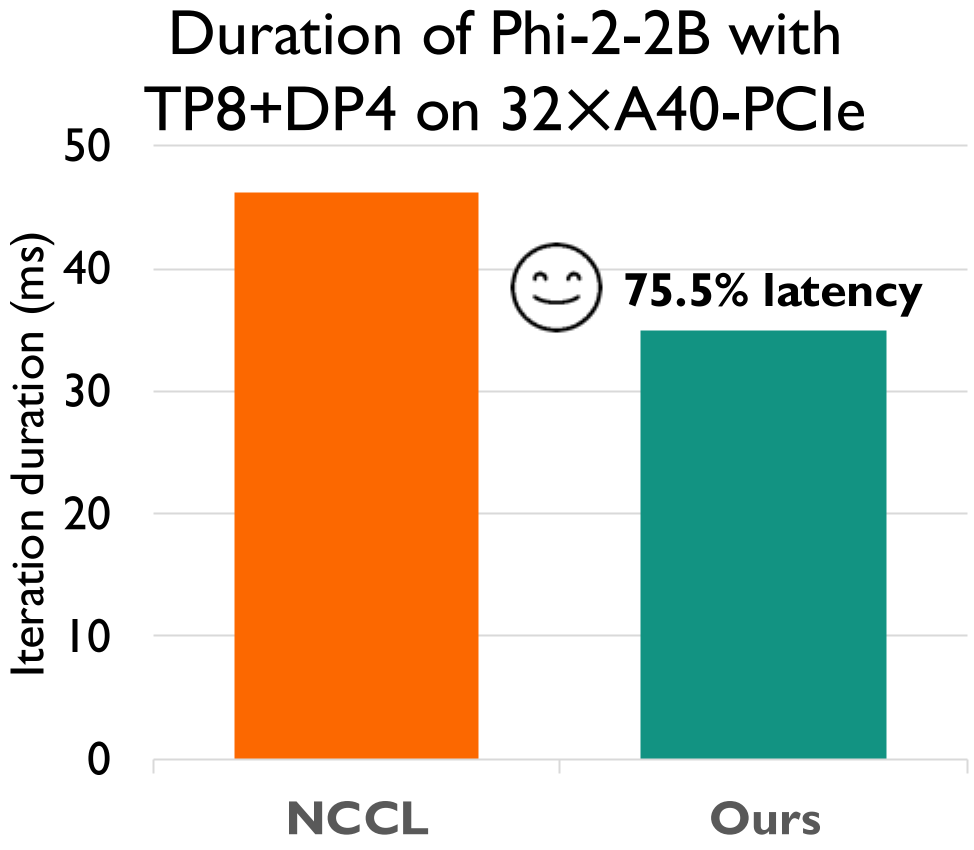
SOTA Models	Training Cost (USD)
GPT3	1.13 million
OPT-175B	1.65 million
Megatron-Turing NLG 530B	3.04 million

Cost of training DNN models[3]

[1] Wang S, et al. Overlap communication with dependent computation via decomposition in large deep learning models ASPLOS22
[2] Wang G, et al. Domino: Eliminating Communication in LLM Training via Generic Tensor Slicing and Overlapping arXiv, 2024
[3] https://epoch.ai/blog/trends-in-the-dollar-training-cost-of-machine-learning-systems

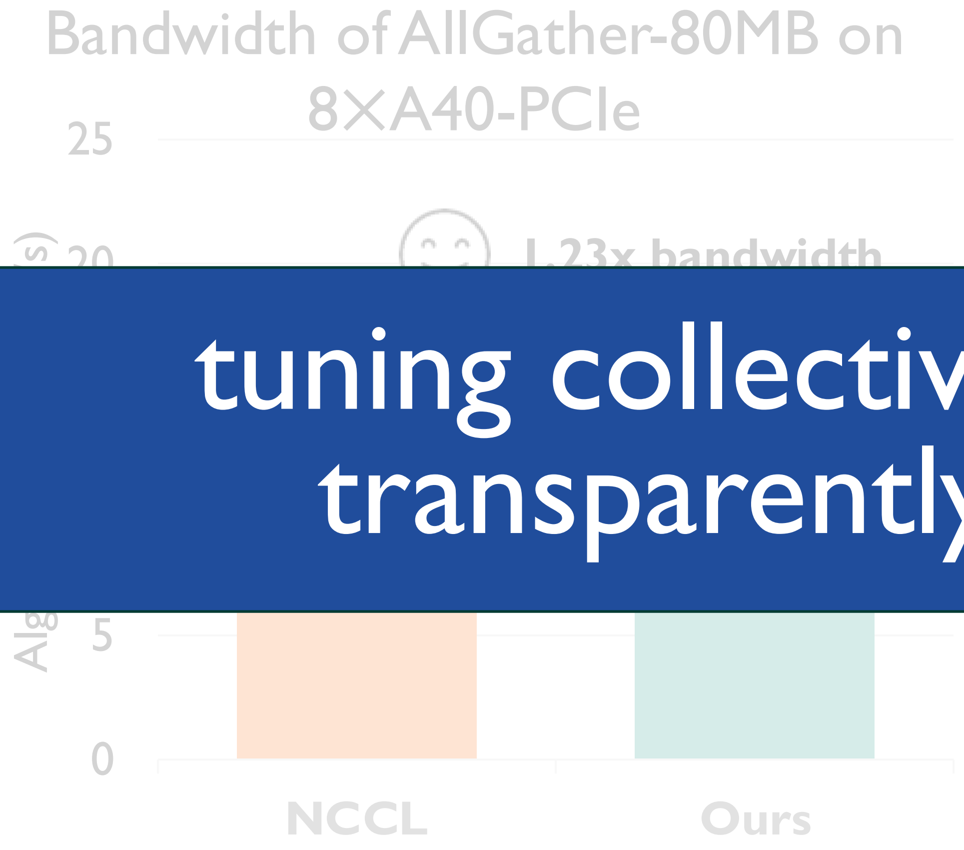
Huge Tuning Opportunities





Huge Improvement after tuning NCCL

Goal

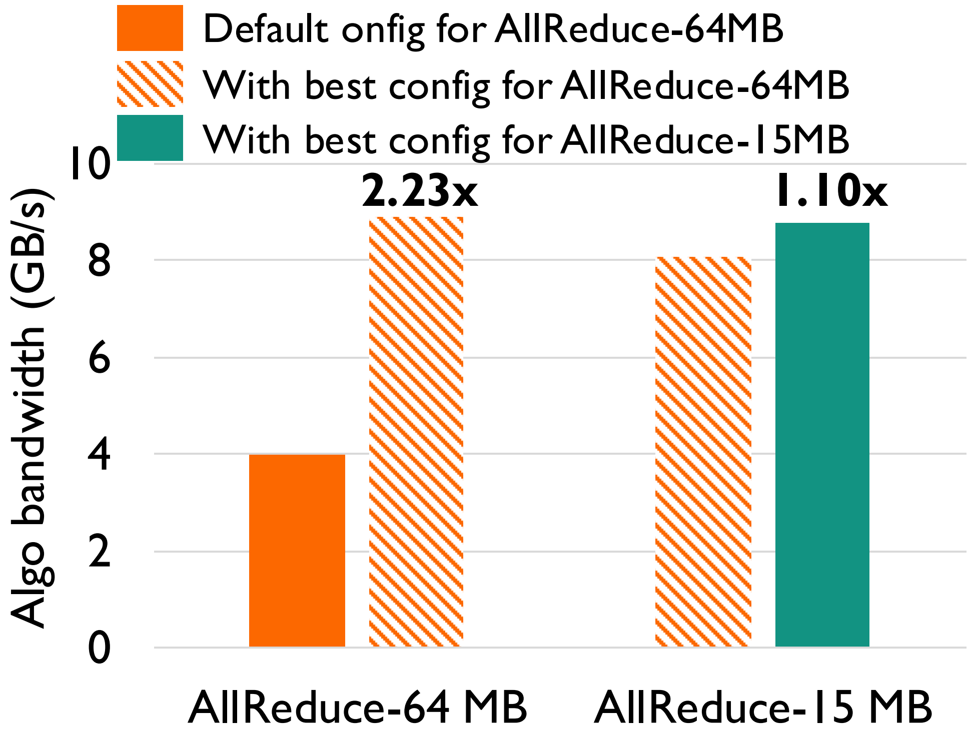


tuning collective communication transparently and efficiently

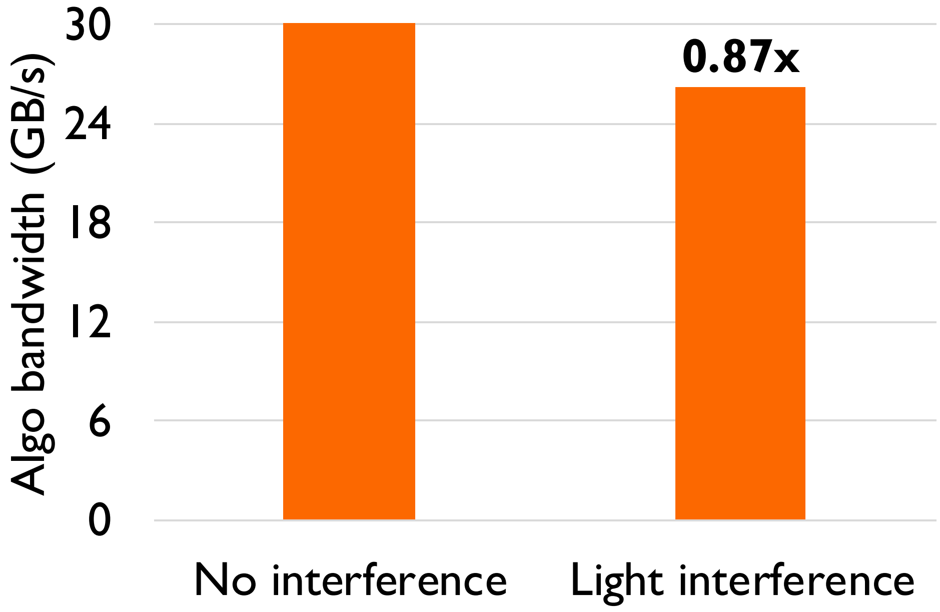
Huge Improvement after tuning NCCL

Tuning is Not Easy

Bandwidth of various tasks with different configs on 8×A40-PCIe



Bandwidth of AllGather(80MB) under various interference on 8×A40-NVlink



No One-Config-Fits-All

Computational tension

Questions Before Tuning

- ❑What low-level parameters are most performance-sensitive?
- ❑What rules could guide the effective tuning?
- ❑How to mitigate the dynamic tension of computation?
- ❑How to support workloads transparently with minimal overhead?

What is the tuning space?

Q1: Build Tuning Space

Original Parameters		Abstracted Parameters	
# of parameters	# of key parameters	Categories	# of Choices
158	28	1 for Algorithm (A)	2
		3 for Protocol (P)	3
		3 for Transport (T)	2
		11 for Nchannel (NC)	128
		3 for Nthread (NT)	20
		7 for Chunk size (C)	8192

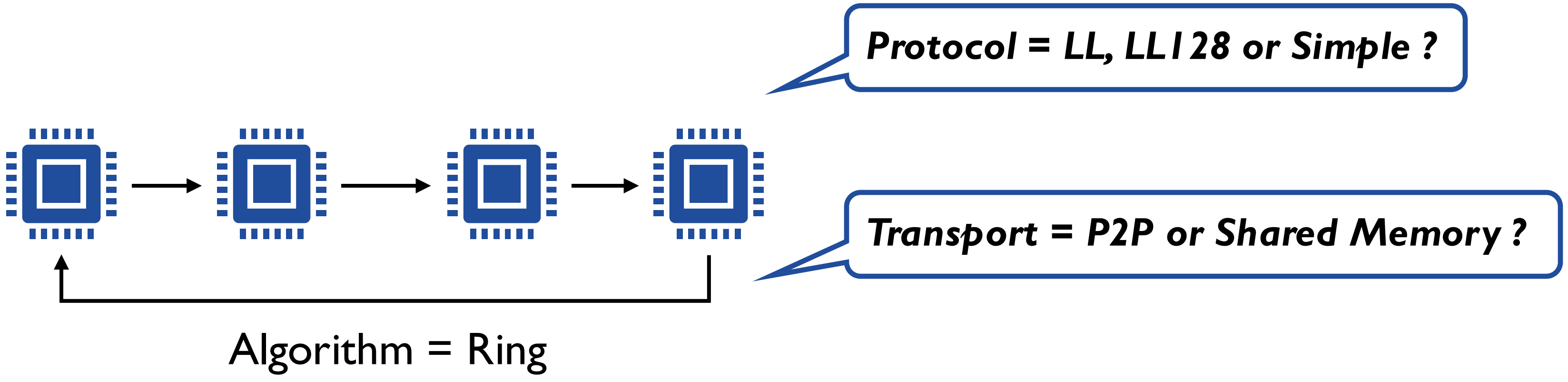
$2\times3\times2\times128\times20\times8192$ > 1 millions

How to find the optimal ?

Q2: Rule1 - Divided into model-able subspaces

Parameter	Choices
Algorithm (A)	Tree, Ring
Protocol (P)	LL, LL128, Simple
Transport (T)	P2P, SHM
Nchannel (NC)	
Nthread (NT)	
Chunk size (C)	

Implementation-related parameters
- ***hard to model but small space***

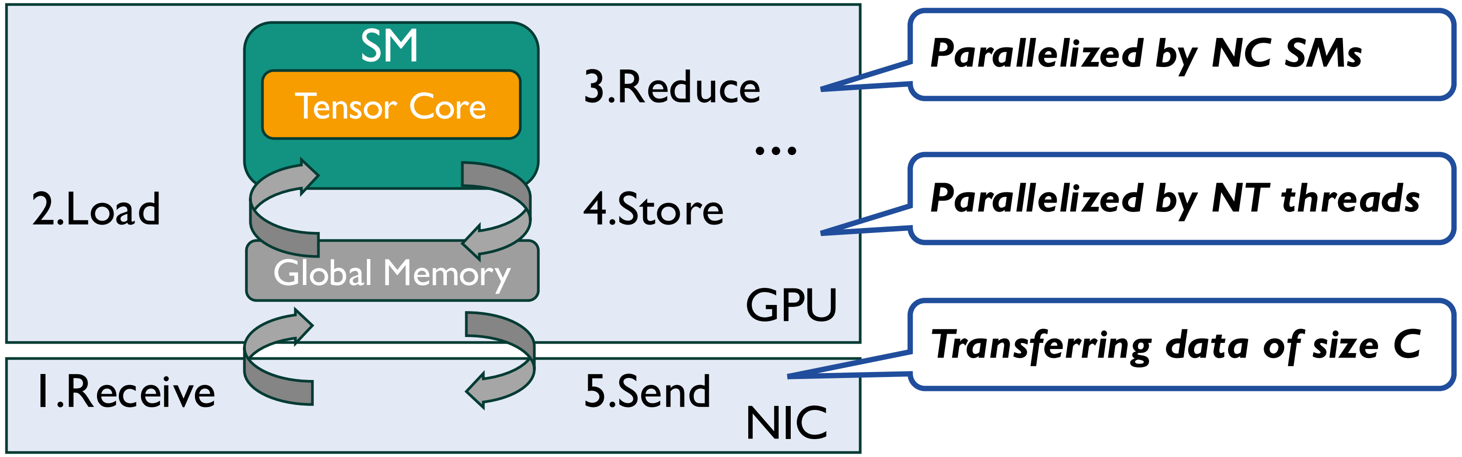


Q2: Rule1 - Divided into model-able subspaces

Parameter	Choices
Algorithm (A)	Tree, Ring
Protocol (P)	LL, LL128, Simple
Transport (T)	P2P, SHM
Nchannel (NC)	$1 \le n \le 128, n \in \mathbb{N}$
Nthread (NT)	$n = 32 \times i, i \in \{1,2,\dots,20\}$
Chunk size (C)	$n = 256 \times i, \quad i \in \{1,2,3\dots,8K\}$

Resource allocation parameters

- ***huge space but model-able***



Q2: Rule1 - Divided into model-able subspaces

Parameter	Choices
Algorithm (A)	Tree, Ring
Protocol (P)	LL, LL128, Simple
Transport (T)	P2P, SHM
Nchannel (NC)	$1 \le n \le 128, n \in \mathbb{N}$
Nthread (NT)	$n = 32\times i, i \in \{1,2,\dots,20\}$
Chunk size (C)	$n = 256\times i, \quad i \in \{1,2,3 \dots, 8K\}$

Implementation-related parameters

- ***hard to model but small space***

Resource allocation parameters

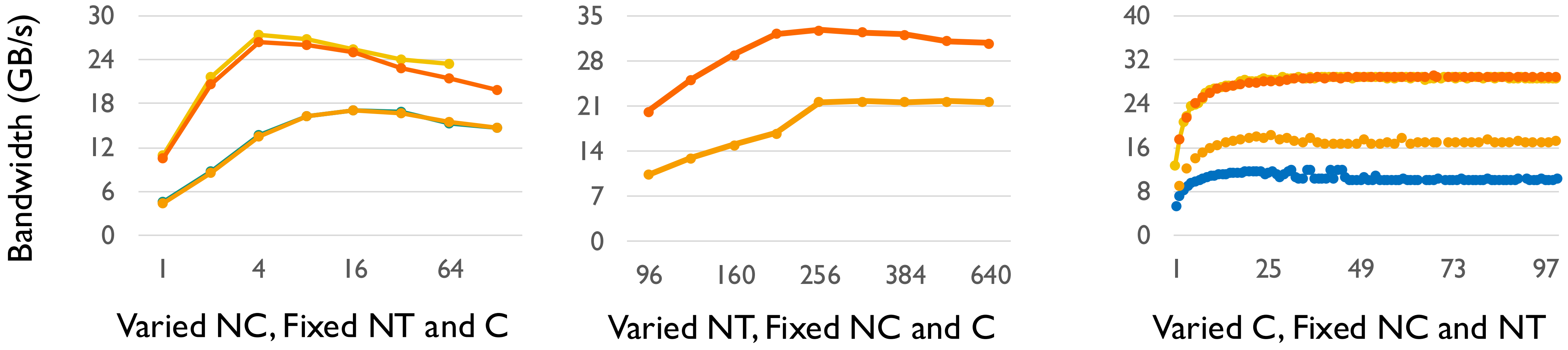
- ***huge space but model-able***

Each subspace contains large various combinations of resource-allocation parameters

A small number of subspaces composed of implementation-related parameters

Q2: Rule2 - Coordinate Descent Search in subspaces

The trends of AllGather(80MB) with config <A, P, T, *, *, *> on 8×A40-NVLink

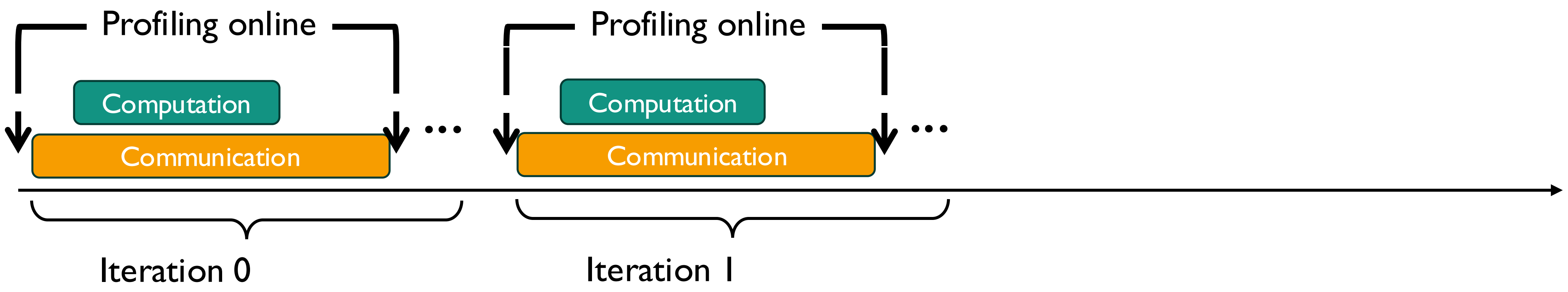


Unimodal functions in every resource-parameter

How to mitigate tension?

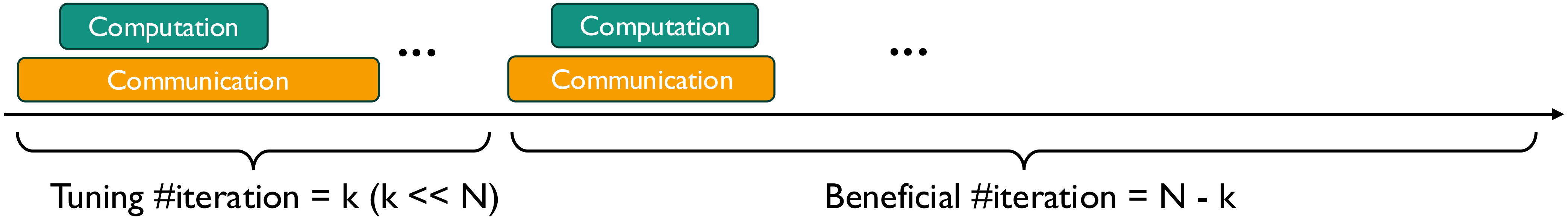
Q3: Tension-aware Tuning in Repetends

Model	Training Time
Megatron-355M[1]	300 K iteraions
DeepSeek-V3[2]	2 months
MegaScale[3]	70 days



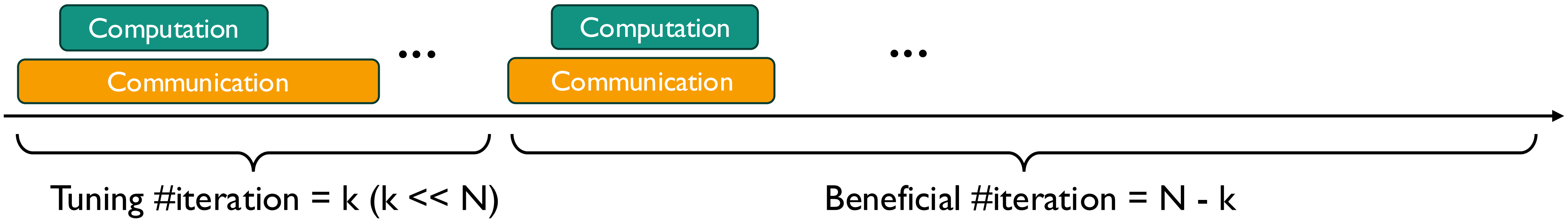
[1]: Shoeybi, Mohammad, et al. "Megatron-lm: Training multi-billion parameter language models using model parallelism." arXiv preprint arXiv:1909.08053
[2]: Liu, Aixin, et al. "Deepseek-v3 technical report." arXiv preprint arXiv:2412.19437
[3]: Jiang Z, et al. MegaScale: Scaling large language model training to more than 10,000 GPUs. NSDI 24

Q4: Embed Tuning into early DNN training

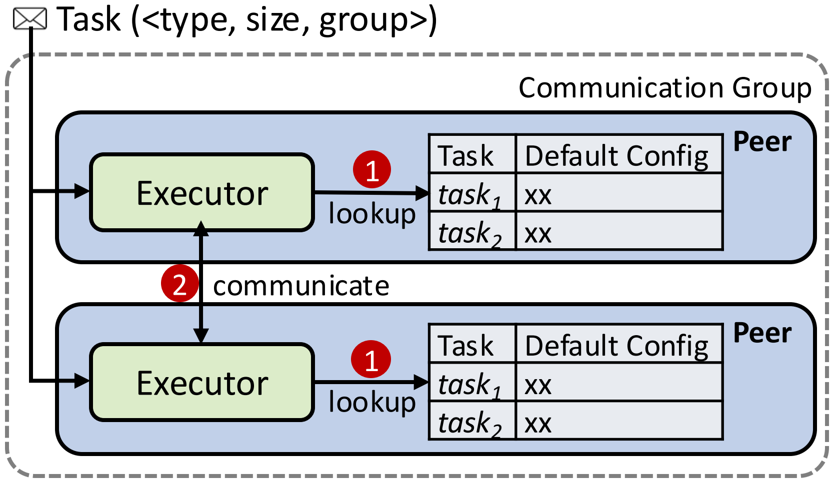


Hide overhead within early iterations

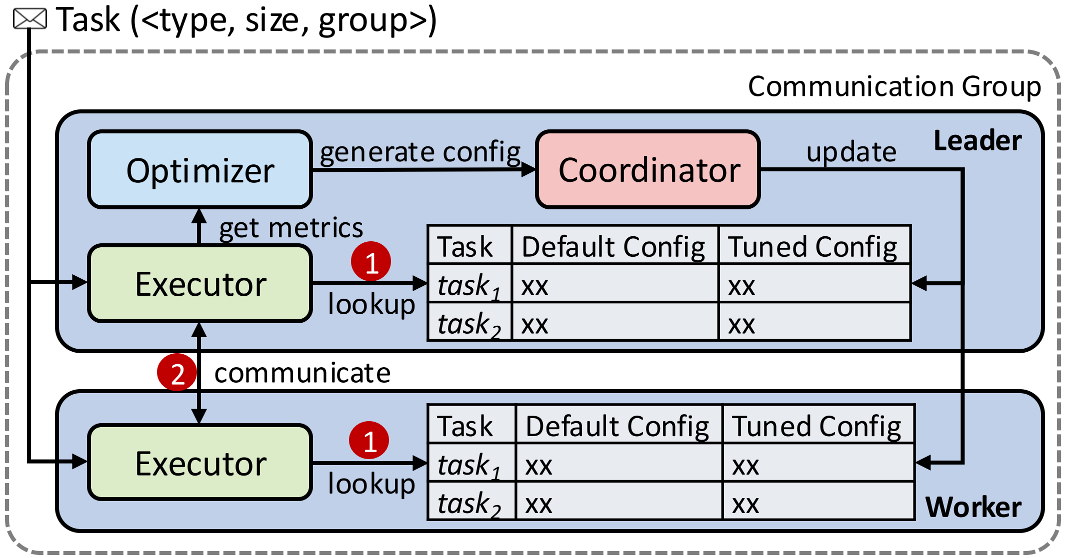
Q4: Embed Tuning into early DNN training



Hide overhead within early iterations



The architecture of NCCL



The architecture of AutoCCL

Share the same APIs

Takeaways

Huge Space for each task

Computation interference

Parameter Classification

Unimodal function

Iterative task

Divide-Conquer

Coordinate Descent Search

Online Tuner

Challenges **Study** **Design**

Tuner	Method	Dynamic	Tension-aware	Accuracy	Overhead
NCCL-tuner	Empirical heuristics	Yes	No	No	Small
AFNFA[1]	Offline Profiling + Fitting	No	No	Depends	Large
Ours	Online Profiling + Search	Yes	Yes	Yes	Small

[1] Wang, Zibo, et al. "AFNFA: An Approach to Automate NCCL Configuration Exploration." APNet2023

Experimental Setup

❑Clusters:

- 2 nodes. Each one has 8×A40-NVLink and 2×400Gbps IB;
- 4 nodes. Each one has 8×A40-PCIe and 100 Gbps IB.

	Type	Size
w/o interference	AllGather, ReduceScatter, AllReduce	1MB – 1GB
w/ interference	AllGather, ReduceScatter, AllReduce	1MB – 1GB

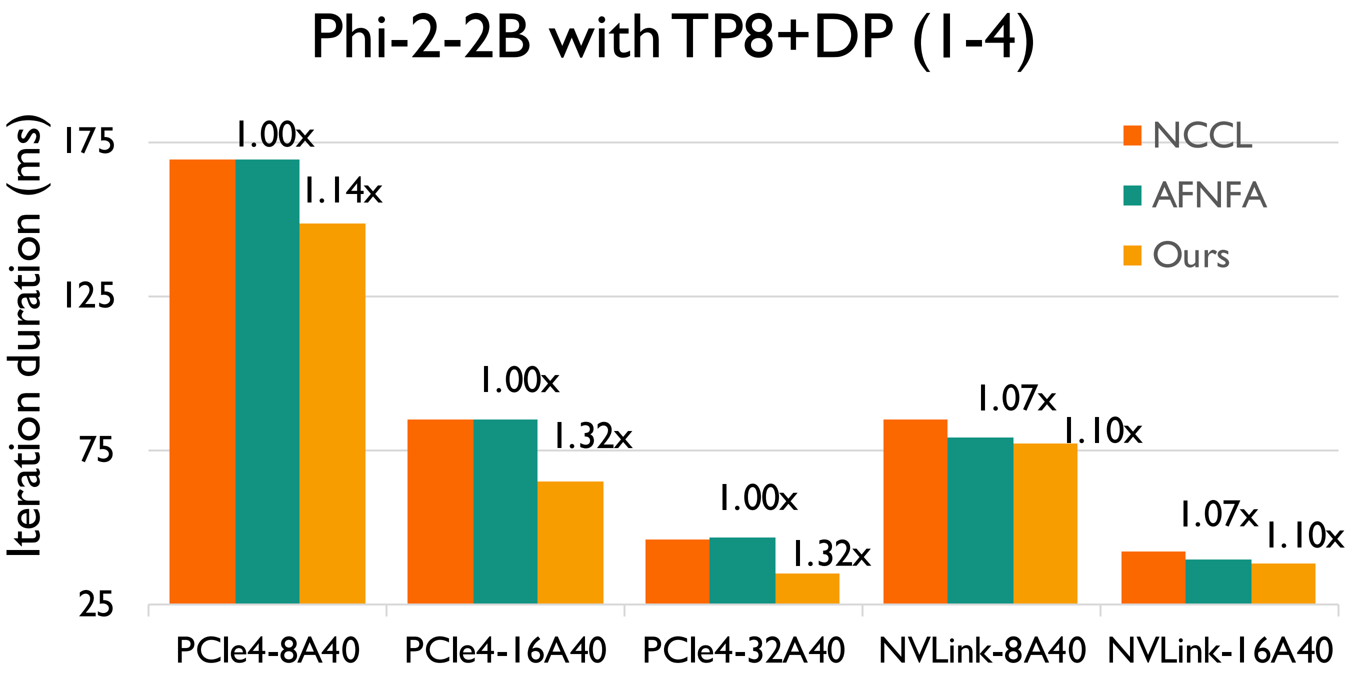
NCCL, AFNFA and AutoCCL

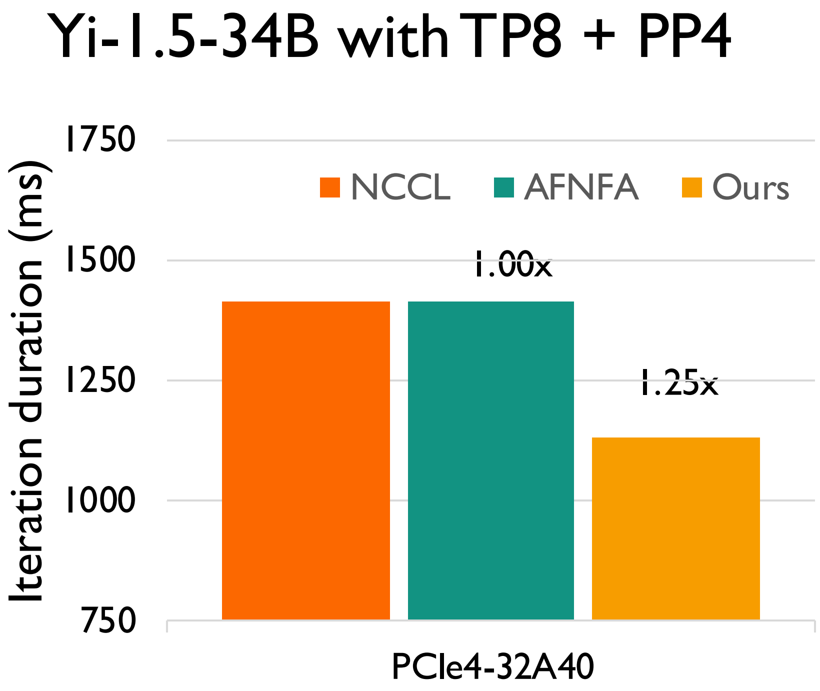
Model	TP	PP	DP
Phi-2-2B	8	1	1-4
Llama-3.1-8B	8	1	1-4
Yi-1.5-34B	8	4	1
VGG-19-0.14B	1	1	8-32

MegatronLM with NCCL, AFNFA and AutoCCL

Note: data parallism (DP), tensor parallelism (TP), pipeline parallelism(PP)

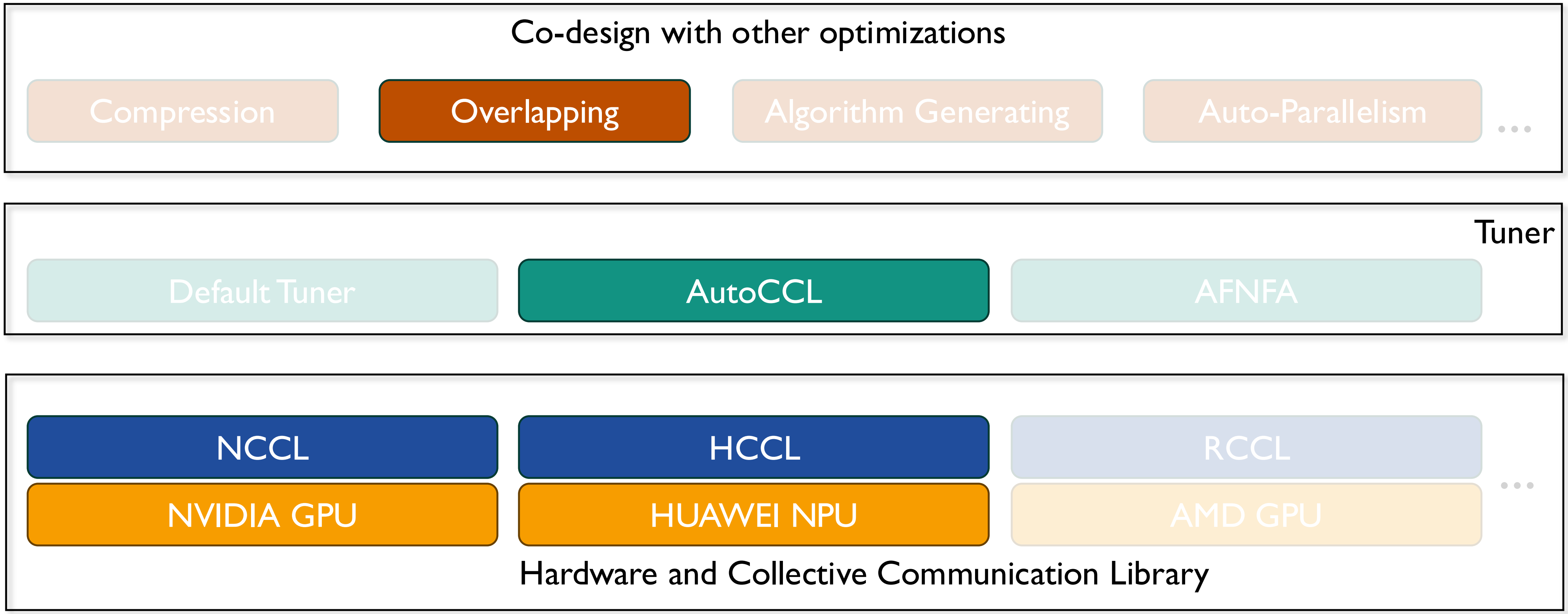
End-to-End Training (Lower is Better)



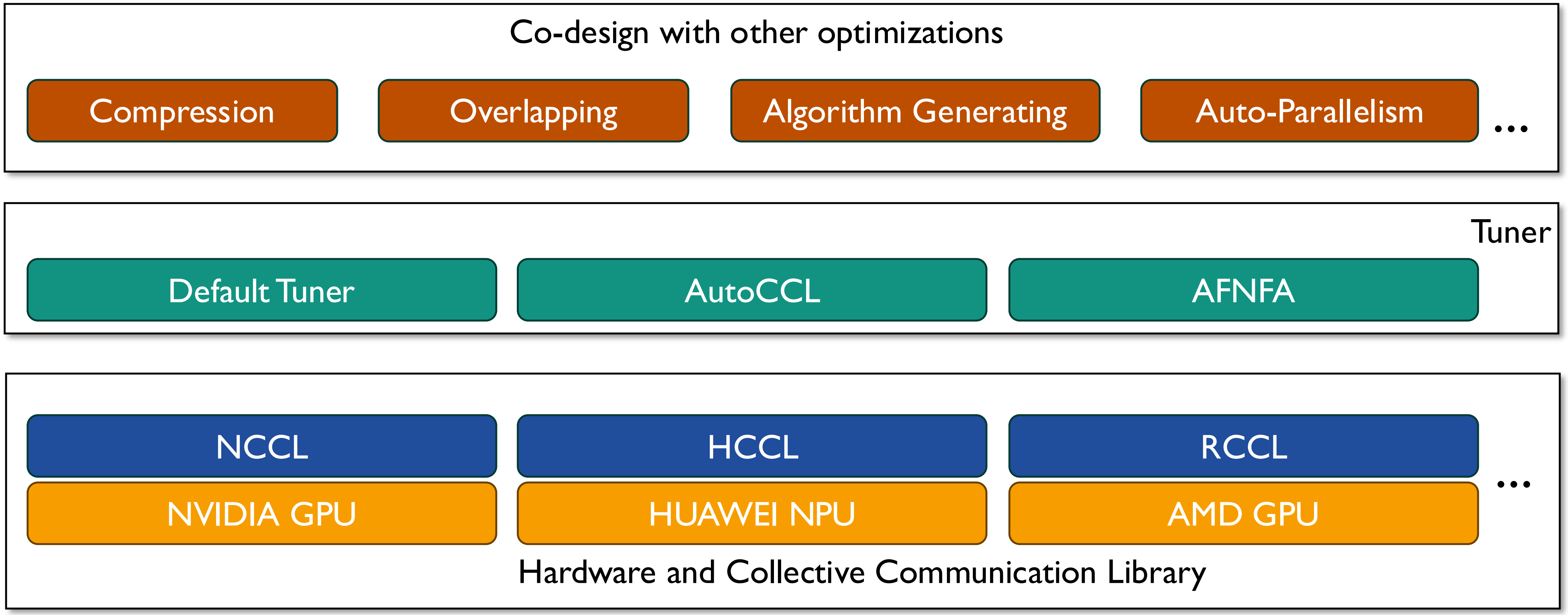


Up to 1.32× throughput within 2 DNN training iterations

Current Supported Works



Landscape of Extension



AutoCCL + X: a foundation for future optimizations

AutoCCL: Automated Collective Communication Tuning for Accelerating Distributed and Parallel DNN Training

Open source: https://github.com/gbxu/autoccl

Contact with me

Join ADSL Lab