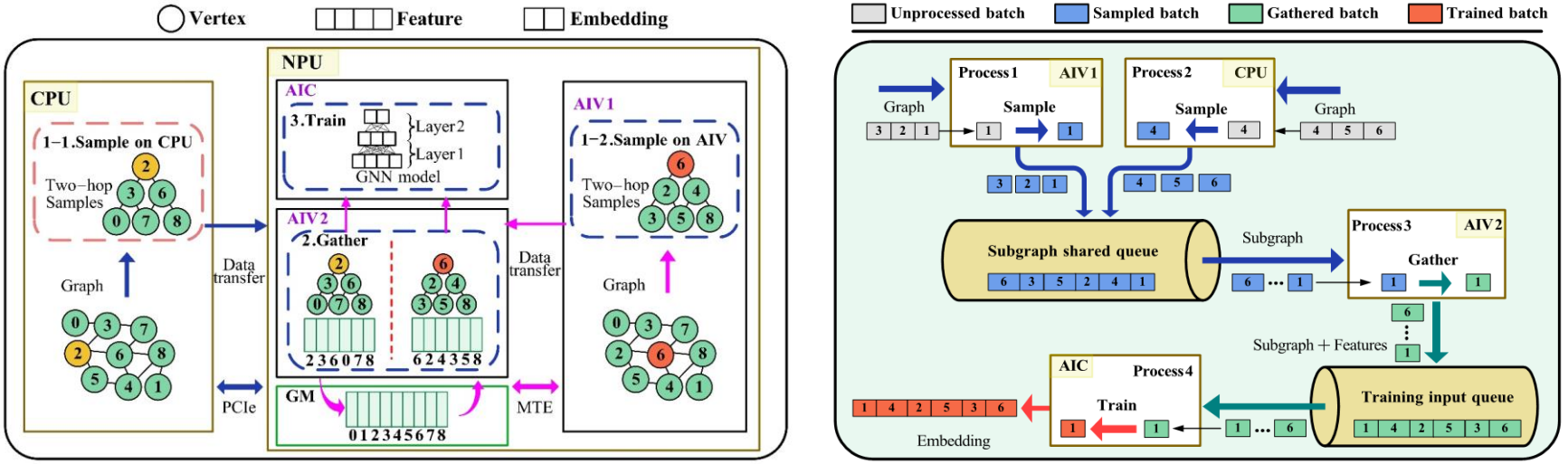
AcOrch: Accelerating Sampling-based GNN Training under CPU-NPU Heterogeneous Environments

Kefu CHEN, Xin AI, Qiange WANG, Yanfeng ZHANG, Ge YU

Frontiers of Computer Science, DOI: 10.1007/s11704-025-50893-0

Problems & Ideas

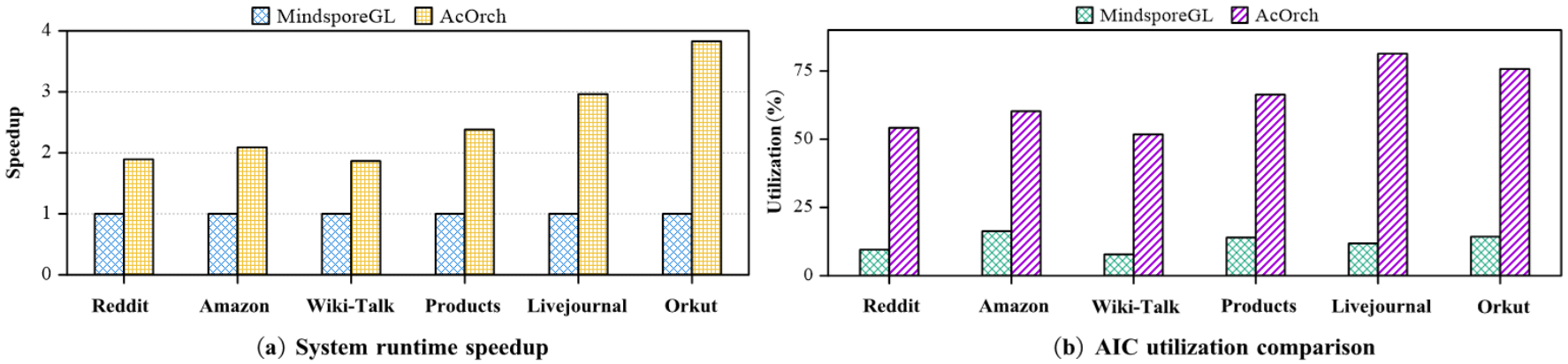
- Problems of step-based task orchestration on CPU-NPU:
  - Step-based orchestration for GNN sampling on CPU–NPU platforms causes load imbalance and idle AIC units.
  - Existing GPU-oriented scheduling frameworks cannot adapt to the multi-compute-unit architecture of NPUs.
- Ideas: A sample-driven orchestration and pipelined execution framework that balances workloads and improves AIC units utilization for efficient GNN training.



Task orchestration and pipeline optimization in AcOrch. Left: Computation-aware task orchestration dynamically partitions sampling workloads between CPU and AIV for dual-path sampling, while feature gathering and training are mapped to AIV and AIC, respectively. Right: Two-level pipelined execution overlaps CPU–NPU and AIV–AIC computations through shared queues, forming a fine-grained producer–consumer pipeline that maximizes concurrency and reduces idle time.

Main Contributions

- Contributions:
  - A sample-driven orchestration framework that dynamically balances workloads between CPU and AIV to mitigate sampling bottlenecks on CPU–NPU platforms;
  - A two-level pipelined execution model that overlaps sampling, feature gathering, and training across CPU–NPU and AIV–AIC to maximize concurrency;
  - A shared-queue mechanism that decouples stages asynchronously and improves throughput stability.



Performance comparison between AcOrch and MindSporeGL. Left: AcOrch achieves an average 2.31× speedup over MindSporeGL across six real-world graph datasets. Right: With finer-grained load partitioning and coordinated scheduling across heterogeneous compute units, AcOrch improves AIC utilization by 4.28× on average, achieving a 52.63% increase over MindSporeGL.