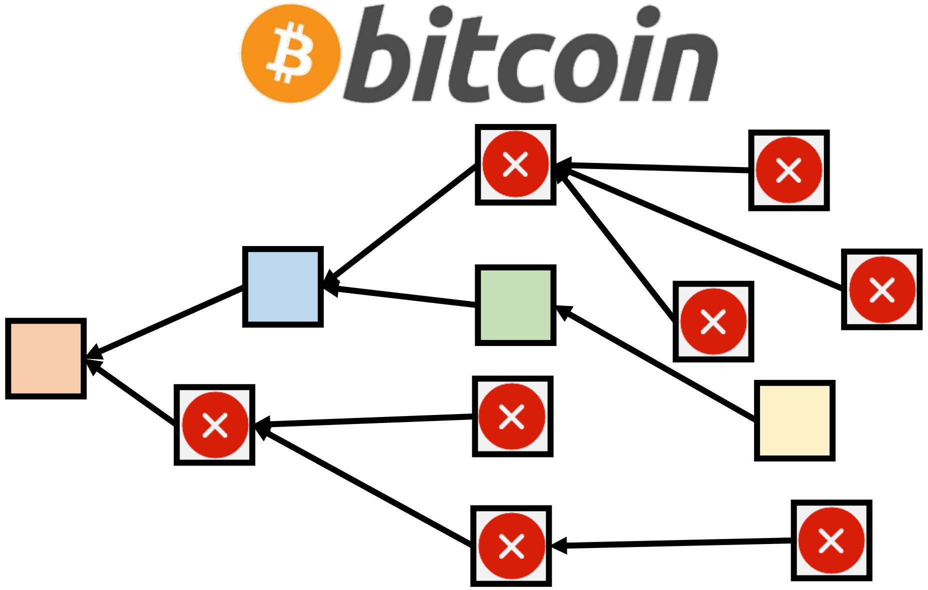
Ladder: A Convergence-based Structured DAG Blockchain for High Throughput and Low Latency

Dengcheng Hu[1], Jianrong Wang[1], Xiulong Liu* [1], Hao Xu[1], Xujing Wu[2], Muhammad Shahzad[3], Guyue Liu[4], Keqiu Li[1]

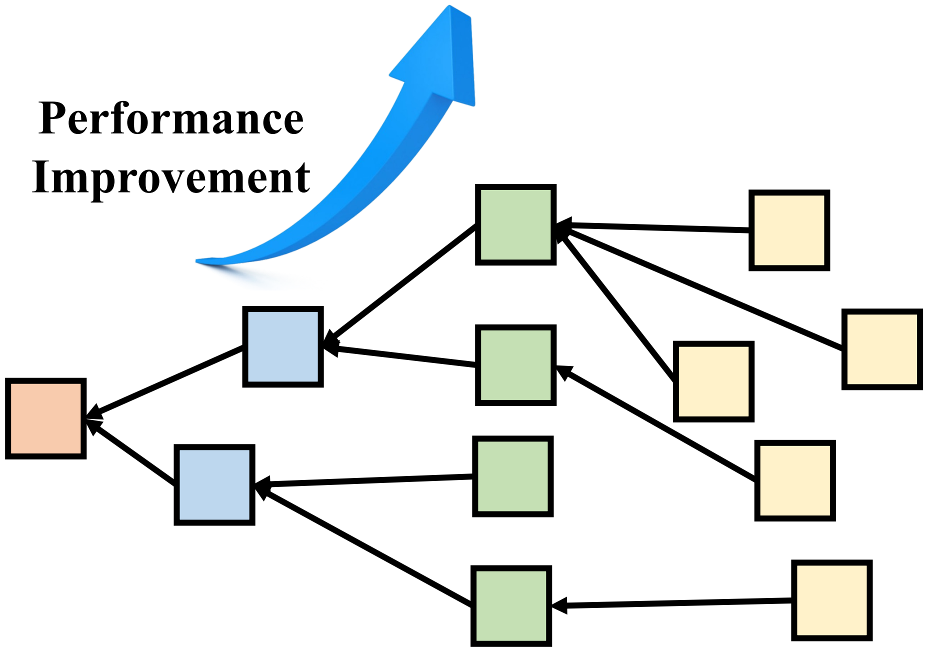
[1]*Tianjin University*
[2]*JD.com*
[3]*North Carolina State University*
[4]*Peking University*

Background

Directed Acyclic Graph (DAG) offers a scalable alternative to chain-style ledgers by enabling concurrent block generation through parallelized topology



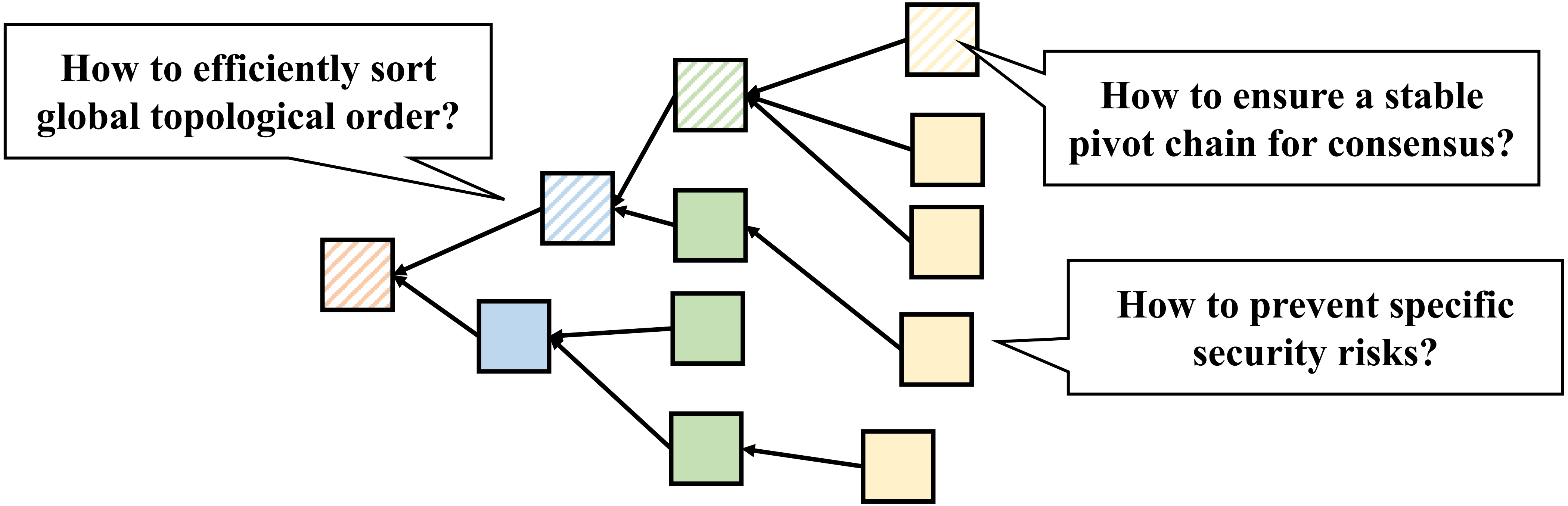
Chain-structure



DAG-structure

Background

Broader DAG structures enhance scalability through parallel block generation but introduce key challenges

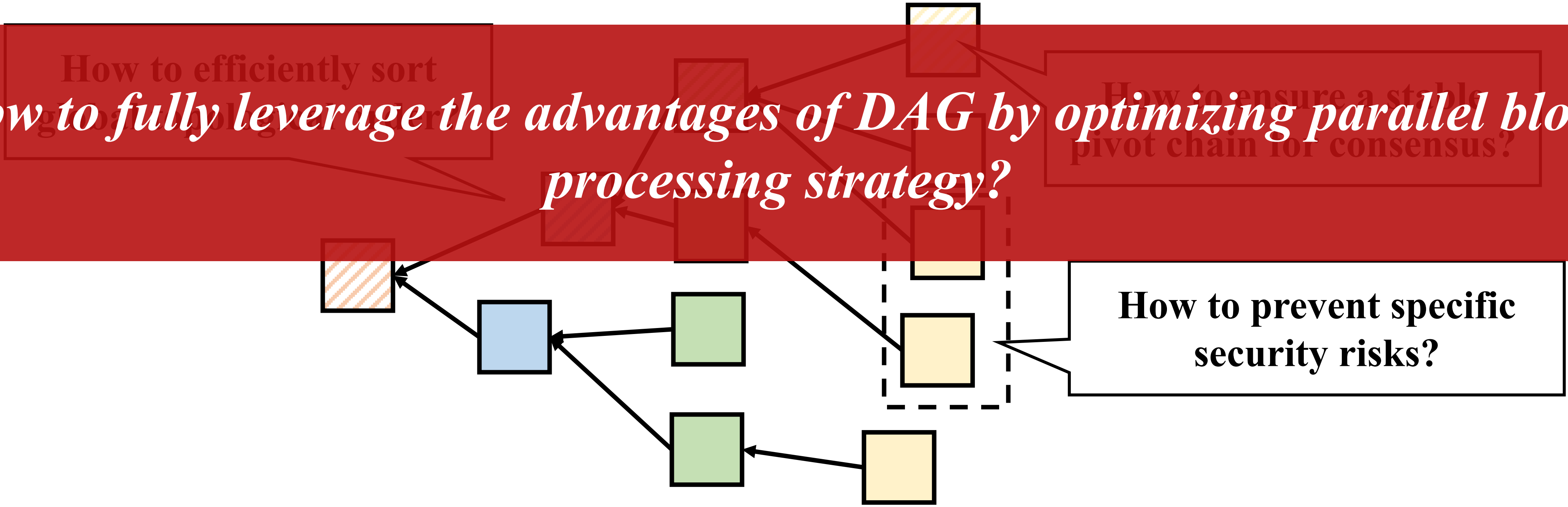


DAG-structure

Background

Broader DAG structures enhance scalability through parallel block generation but introduce key challenges

How to fully leverage the advantages of DAG by optimizing parallel block processing strategy?



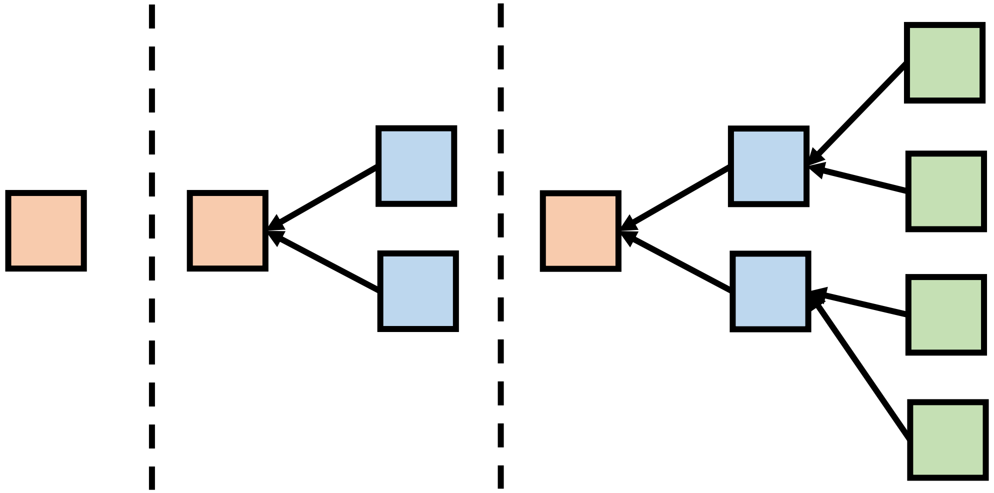
DAG-structure

Limitation of Prior Arts

Solutions	Sorting Methods	Confirmation Logic	Balance Attack Resistance	Performance	
				TPS	Latency
GHOST	Independent Local Sorting with Global Confirmation	Reference-count Global Ordering	No	200	<60min
Inclusive		Partially Ordered Ledger	No	350	<1min
Spectre			-	-	<1min
Phantom		Weight-based Global Ordering	No	40	<1min
Conflux			No	2823	<1min
OHIE		Hierarchical Global Ordering	Yes	2513	<10min
Ladder	Single-Node-Driven Ordering & Confirmation		Yes	4506	<1min

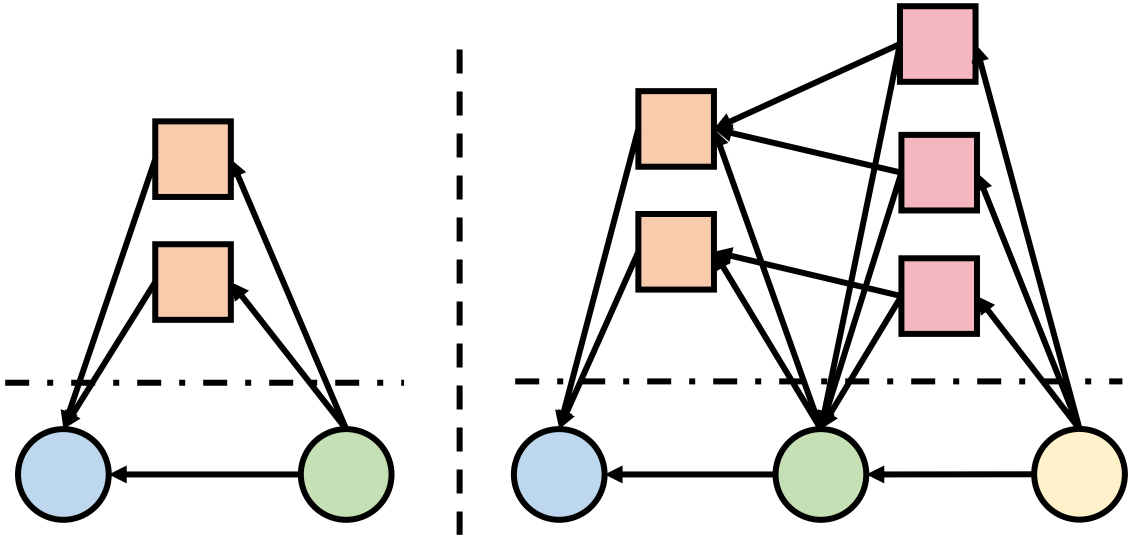
Basic Idea

Traditional DAG



Lacking control over parallel chain topology, leading to **performance limitations** and vulnerability to **balance attacks**.

Our Method



Employing a **dual-chain architecture** where one chain structurally constrains the other through **convergent referencing**.

Technology Challenges

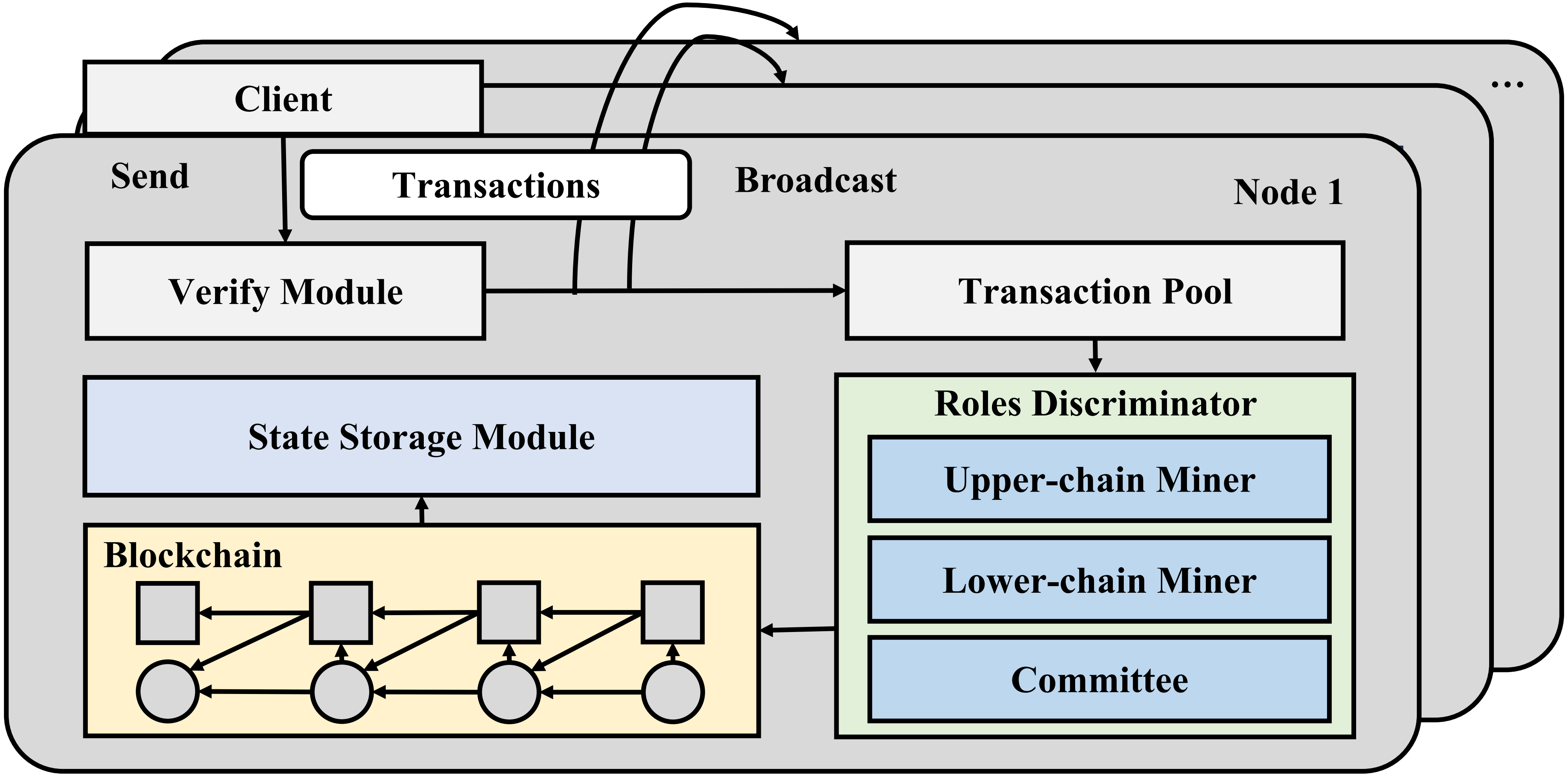
- ***How to use the one-chain to effectively converge generated frok blocks of another chain?***

- ***How to ensure system security by preventing adversaries from becoming convergence nodes?***

Basic Setup

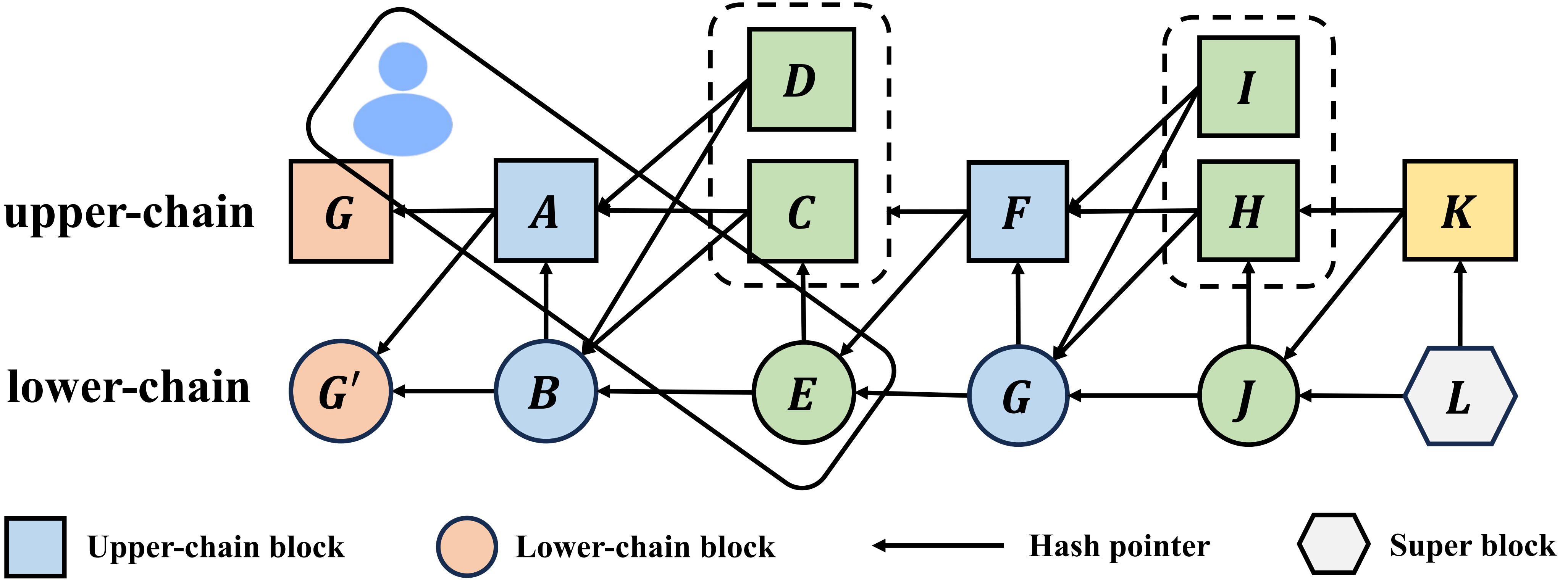
nsdi '25

Ladder assumes a δ-synchronous network and that the adversary contributes less than 30% of the total computational power



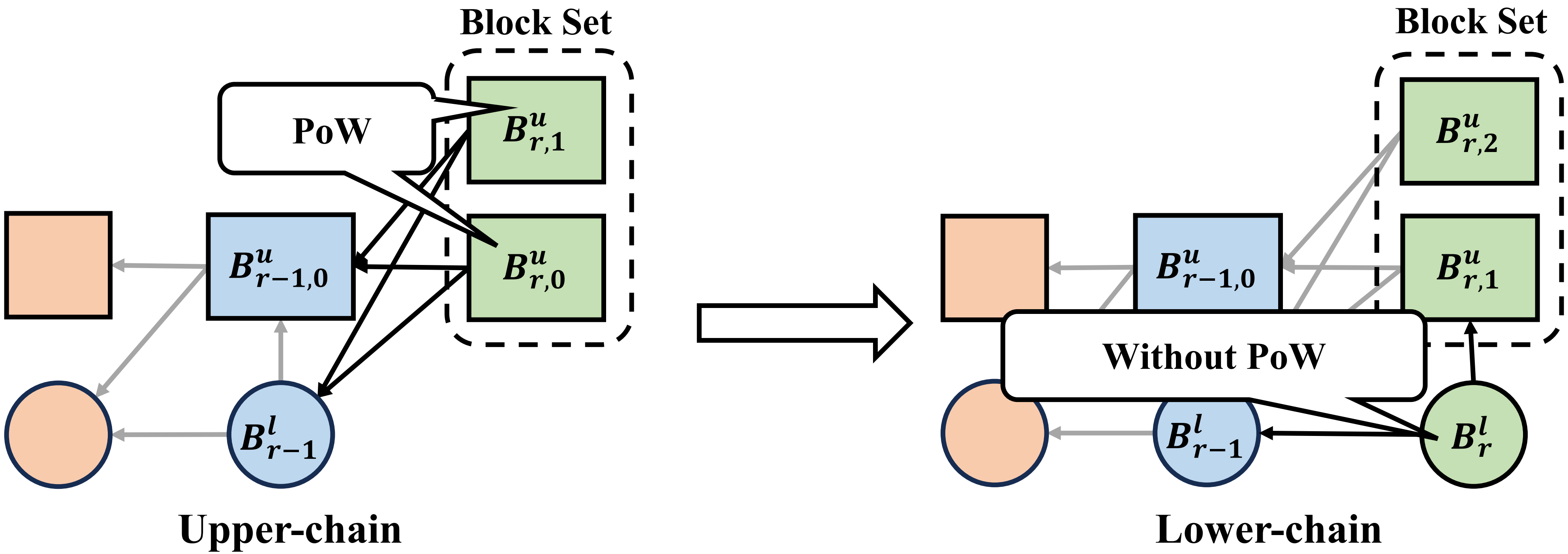
Ladder Overview

Ladder generates DAG in the upper-chain while the lower-chain drives convergence and narrows the DAG structure



Normal Operations of Ladder

Nodes in Ladder utilize Proof-of-Work (PoW) for upper-chain block generation while simultaneously regulating the production of lower-chain blocks



Ladder uses lower-chain blocks to determine the sequence of all forked upper-chain blocks

Hardest Chain Principle

HCP dynamically weights block subtree difficulty (rather than sub-block count) to select the standard upper-chain blocks

$$\mathcal{D}(B_i) := \sqrt{\mathcal{Z}(B_i)} + \sum_{B_j \in S_i} \mathcal{D}(B_j)$$

Leading zeros in the hash of block B_i

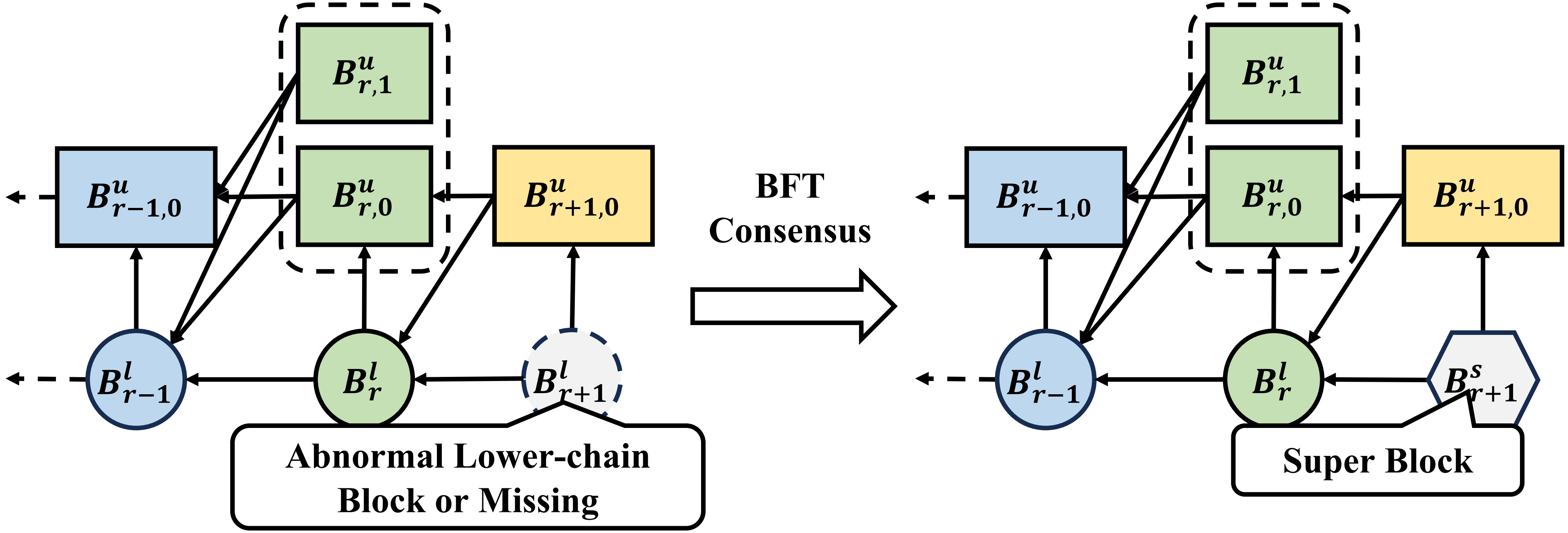
Set of all legal upper-chain blocks pointing to B_i

Difficulty of block B_j

HCP thwarts liveness and balance attacks by requiring prohibitive computational power to override established subtree weights

Super Blocks of Ladder

Ladder forms a committee of recent standard upper-chain block producers to generate Super Blocks resolving lower-chain anomalies

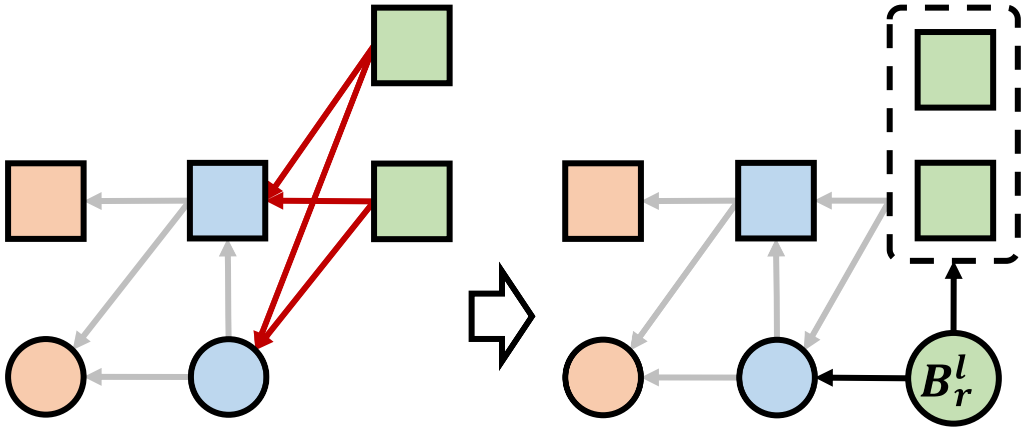


Committee employs deterministic BFT consensus to ensure the finality of lower-chain

Exception Cases in Upper-chain

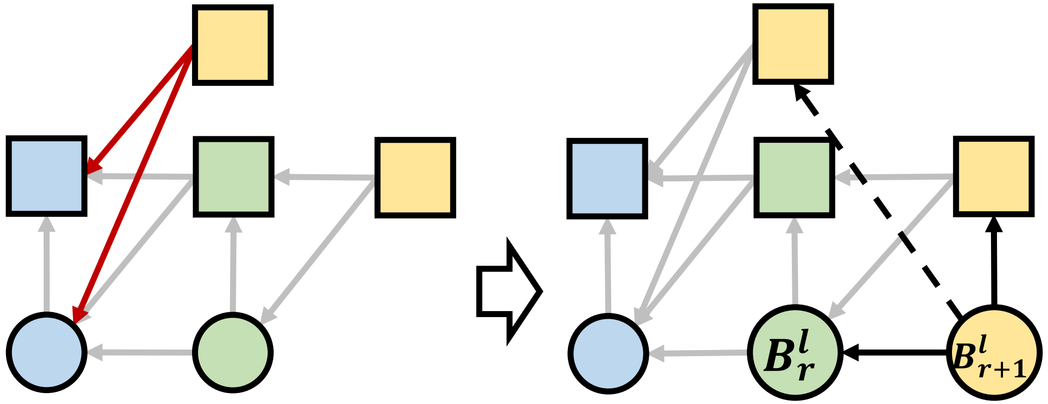
Two upper-chain fork types may occur

Tip Fork



Multiple valid upper-chain blocks in round r, one block generator is selected by convergence node

Chain Fork



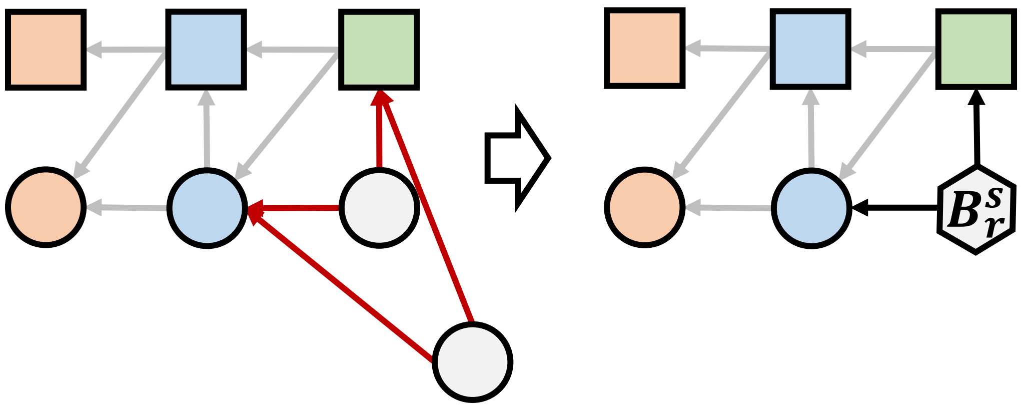
Delayed upper-chain block in round r, referenced by lower-chain block in round $r+1$

Exception Cases in Lower-chain

nsdi '25

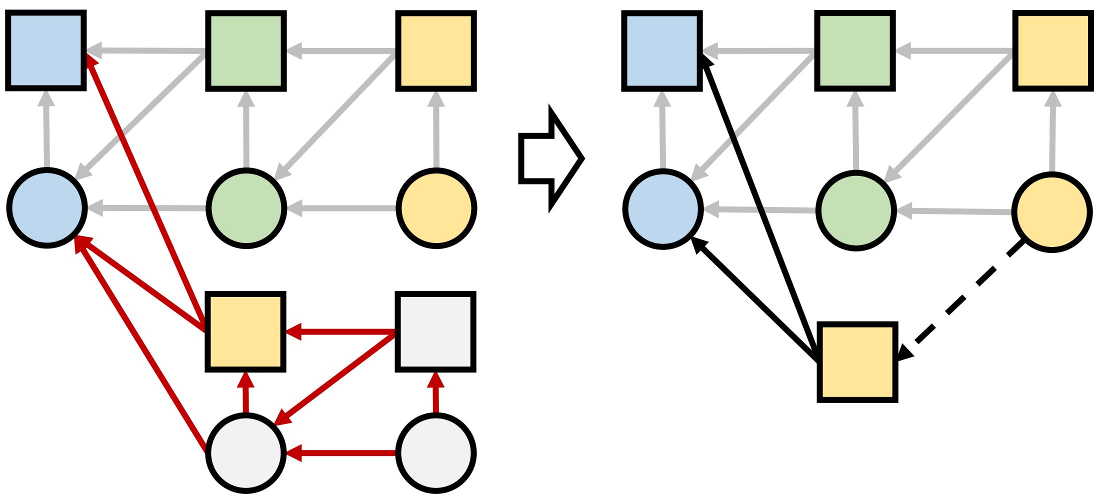
Two lower-chain fork types may occur

Tip Fork



Multiple lower-chain blocks in round r, resolved by BFT to generate a super block

Chain Fork



Two parallel Ladders may arise, reconciled through the Hardest Chain Principle

Parameter Settings

BFT Committee Size

- The BFT committee may introduce security risk: Adversaries exceeding 1/3 when generating the Super Block

Committee Size	120	180	240	300
Byzantine nodes exceeding 1/3 probability	4.9×10^{-3}	1.8×10^{-4}	6×10^{-6}	1.96×10^{-7}
Rounds to reach a cumulative probability of 99%	9.2×10^{2}	2.6×10^{4}	7.7×10^{5}	2.4×10^{7}

Evaluation

nsdi '25

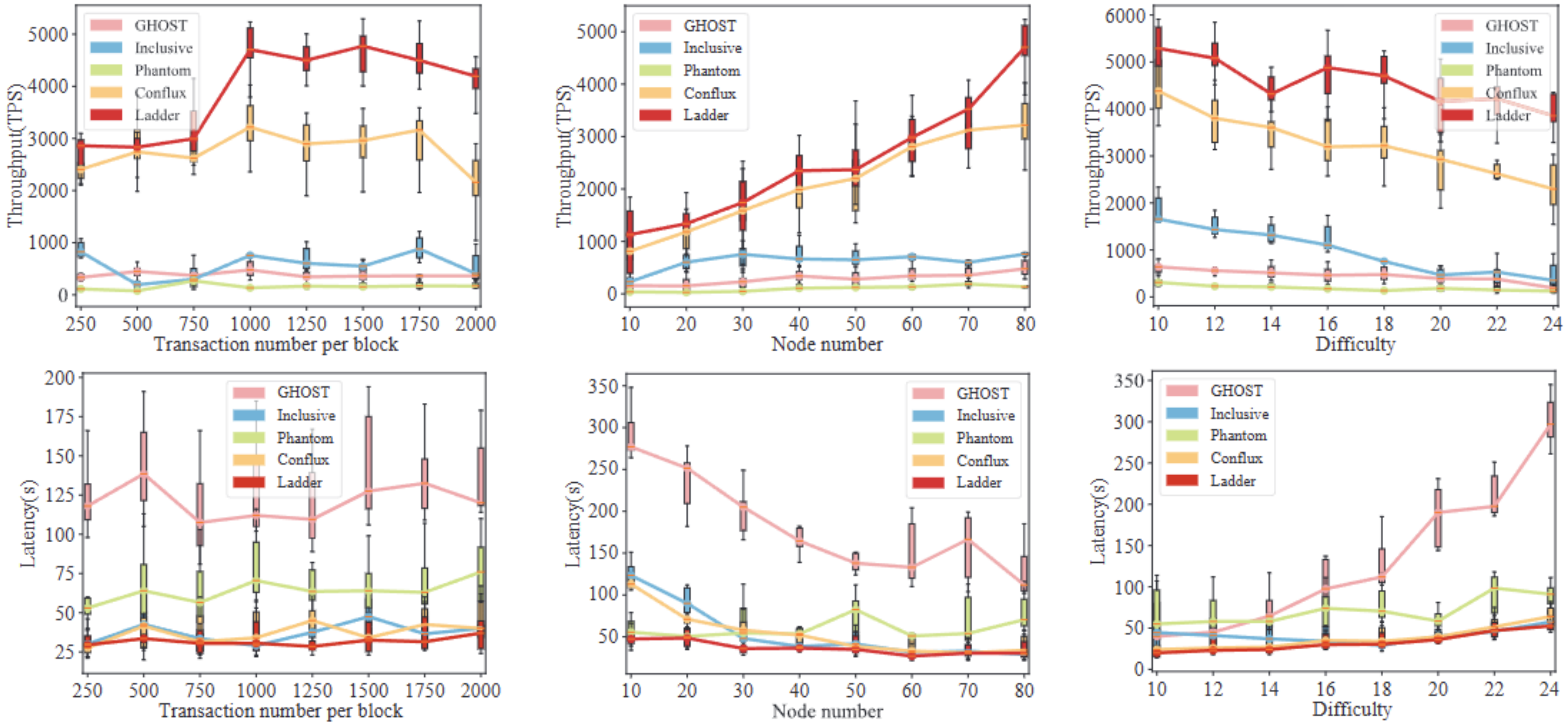
We implement a prototype of Ladder using 80 nodes and compare performance with GHOST, Inclusive, Phantom, and Conflux

Server	Network Delay	Node Number	PoW Difficulty
Intel(R) Core(TM) i5-4590 CPU@3.30 GHz and 8 GB of RAM	80-120ms	80	18

Evaluation

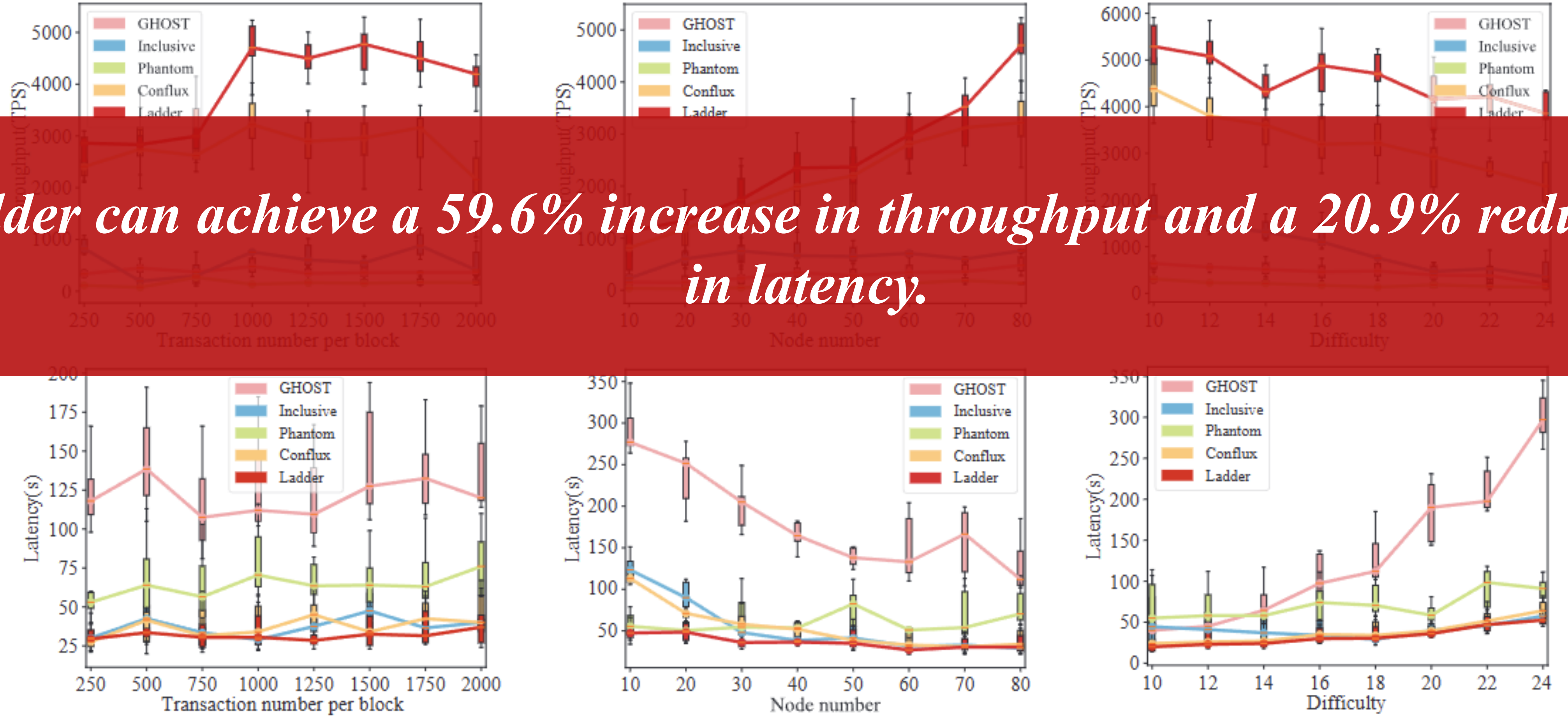
nsdi '25

We use three key variables – transaction number per block, node number, and difficulty level – with Throughput and Latency as performance metrics



Evaluation

We use three key variables – transaction number per block, node number, and difficulty level – with Throughput and Latency as performance metrics



Ladder can achieve a 59.6% increase in throughput and a 20.9% reduction in latency.

Security Analysis

We make the security analysis from two perspectives:

Resistance Against Common Attacks:

- ☐ **Sybil Attack**
- ☐ **Denial of Service (DoS) Attack**
- ☐ **Double-Spending Attack**
- ☐ **Eclipse Attack**

Security and Availability:

- ☐ ***Theorem 1: Any block in the lower-chain of Ladder is a valid block with a high probability.***
- ☐ ***Theorem 2: Ladder satisfies common prefix, finality, and liveness properties.***

Thanks!

Contact Information: hdc@tju.edu.cn