

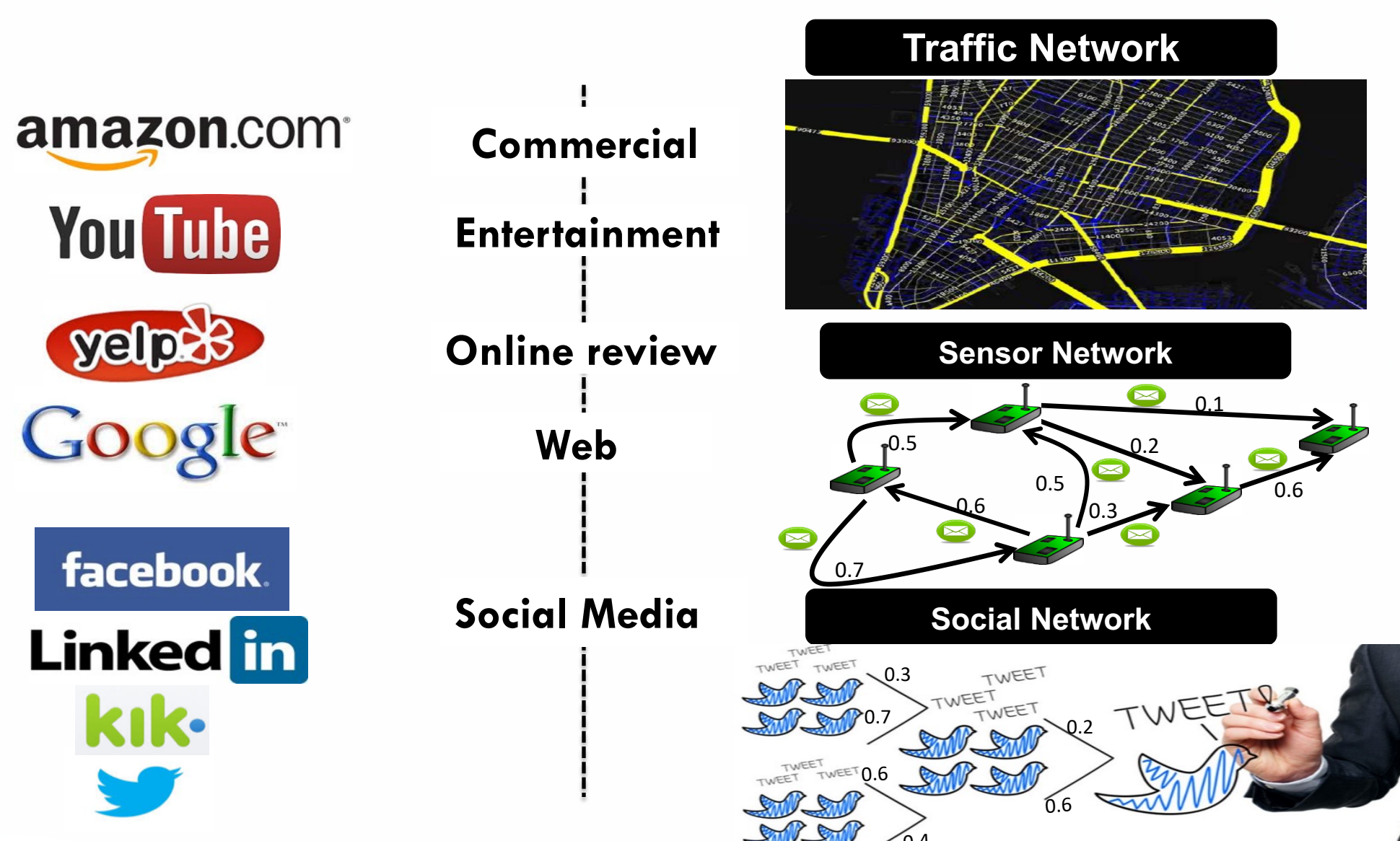
THE UNIVERSITY OF
SYDNEYBrookhaven
National Laboratory

meta

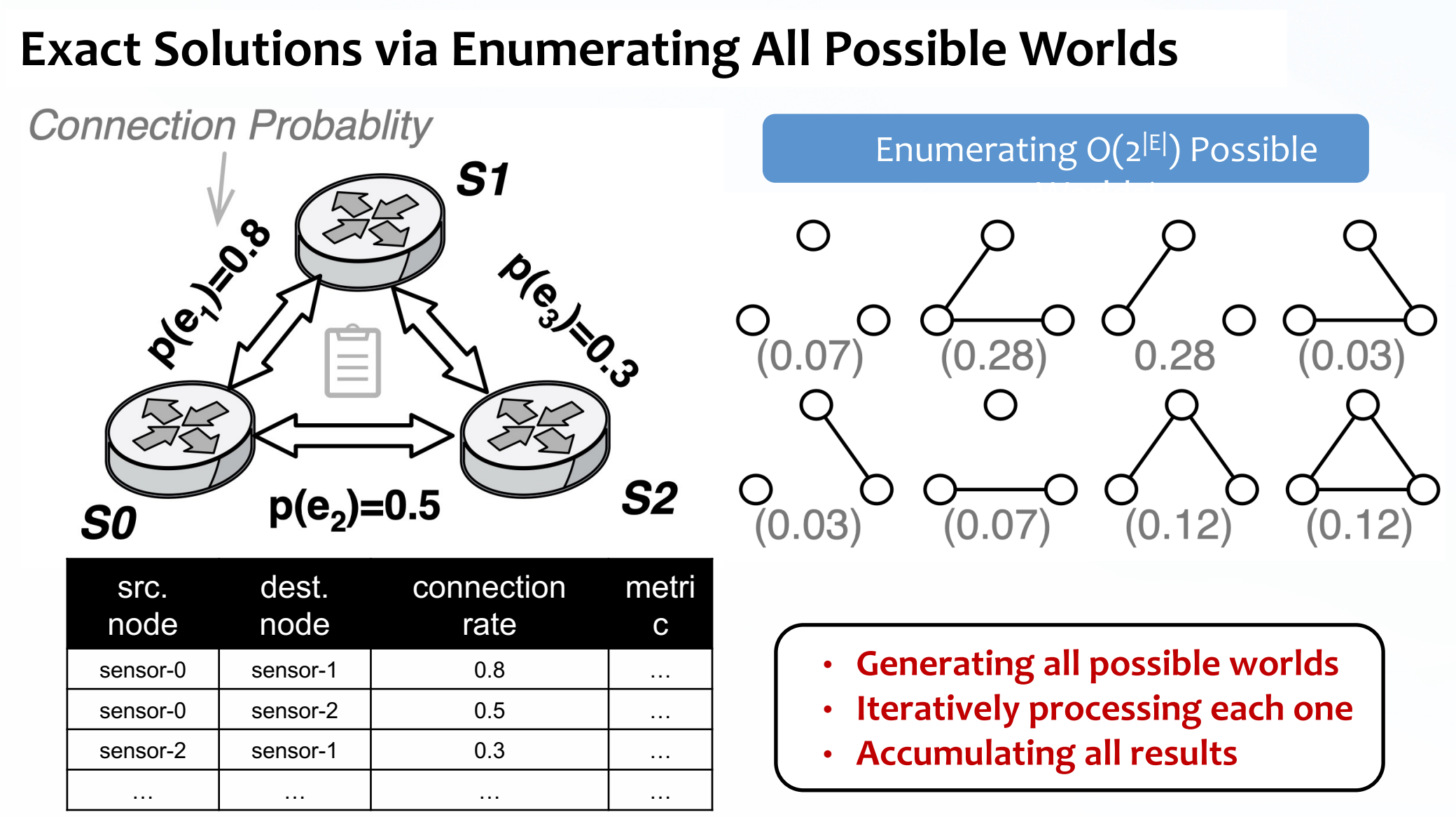
Bring Orders into Uncertainty: Enabling Efficient Uncertain Graph Processing via Novel Path Sampling on Multi-Accelerator Systems

Heng Zhang, Lingda Li, Hang Liu, Donglin Zhuang, Rui Liu, Chengying Huan, Shuang Song, Dingwen Tao, Yongchao Liu, Charles He, Yanjun Wu, Shuaiwen Leon Song
张珩（智能软件研究中心） zhangheng17@iscas.ac.cn, 15652191318

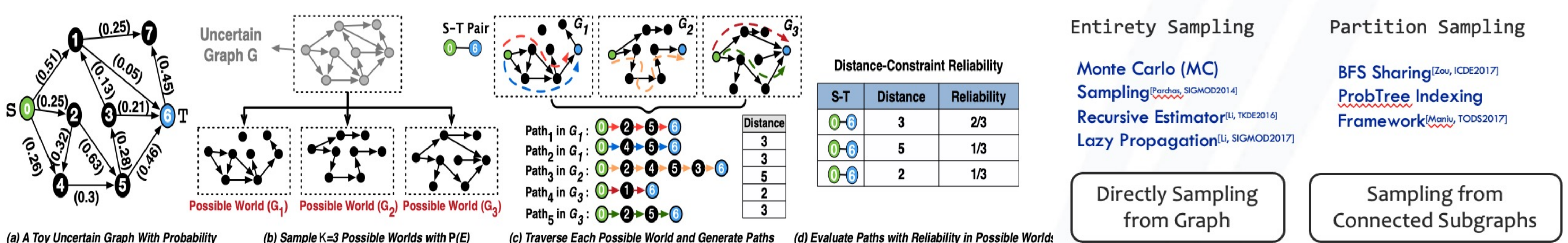
Graphs Are Ubiquitous & Uncertainty Becomes intrinsic to Applications



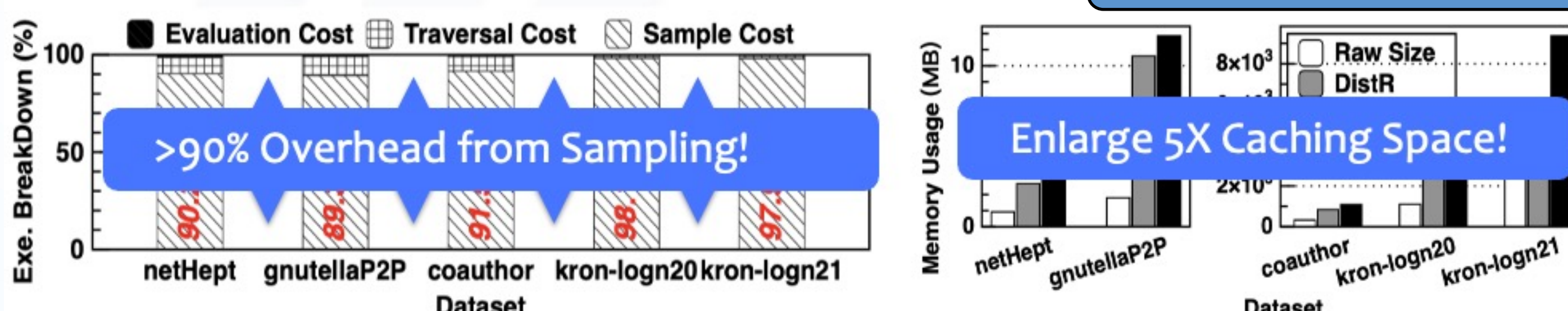
Solutions of Complex Uncertain Graph Processing: A) Exact Solution



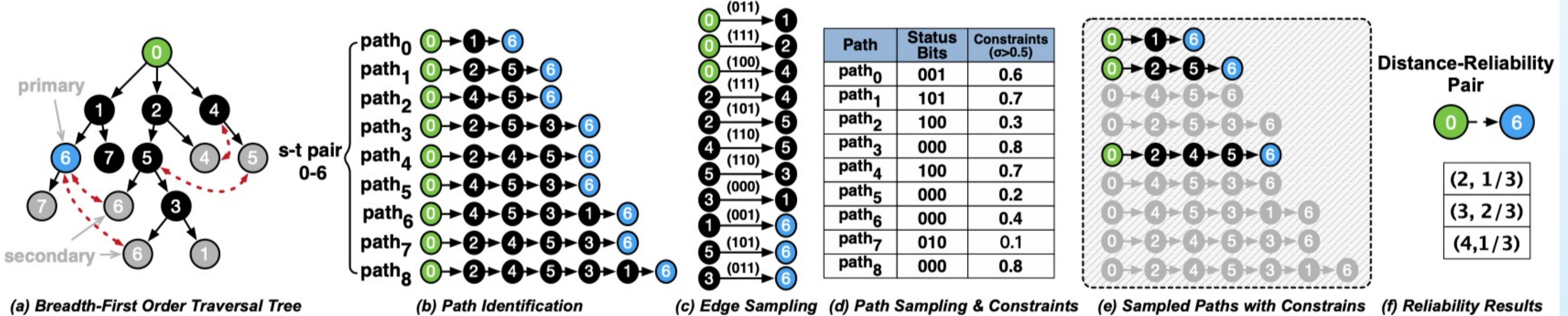
Solutions of Complex Uncertain Graph Processing: B) Approximation Solution



Motivation of Our Work



Our Efficient Uncertain Graph Processing System: Path Sampling Methodology & BPGraph

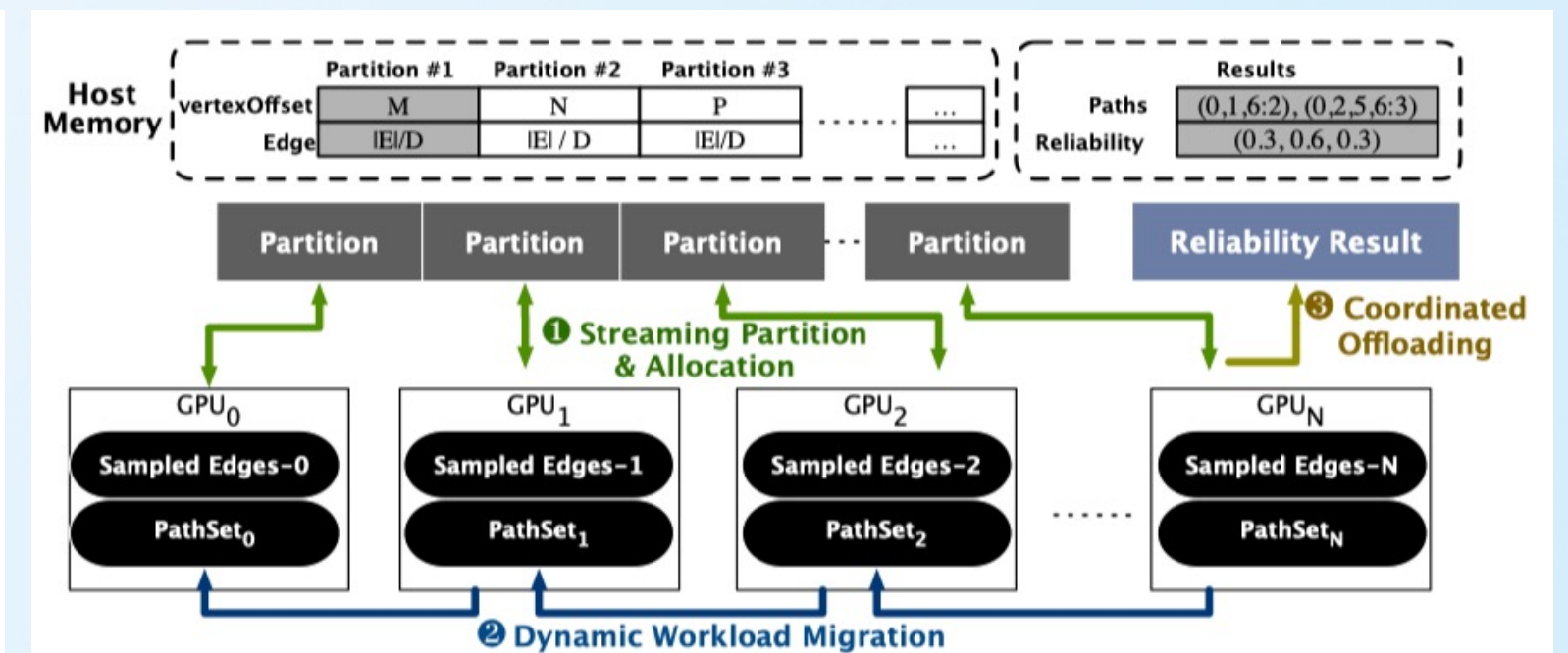
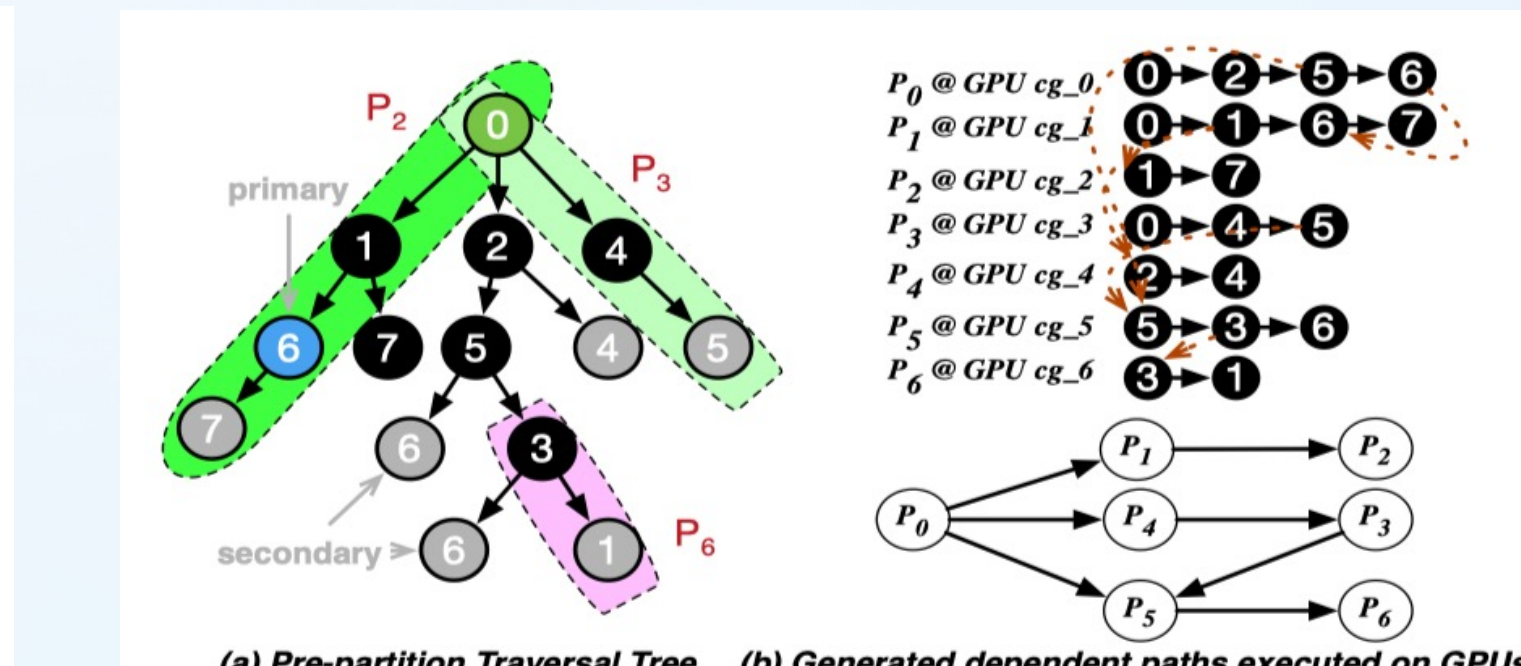


Path-Sampling Centric Programming APIs

GPU-Accelerated Path Identification & Sampling

Scalable Multi-GPU BPGraph Implementation

```
1 void MainProc(Graph g, Vertex srcs[], Vertex tgts[]) {
2   /* Path Identification Stage */
3   ConvertGraphToPaths(g, DispelEntity, srcs, tgts);
4   /* Path Sampling Stage */
5   parallel-for Path p in g.getPaths(srcs, tgts)
6     Initialize(p);
7   // Nested parallel processing edge along paths
8   parallel-for Edge e in p.getEdges()
9     Expand(p, e);
10  synchronize; // Synchronize cooperative threads
11  /* Result Computation: Reduce from all paths */
12  parallel-for Vertex (s, v) in (srcs, tgts)
13    ReduceVertex(g, getPaths(s, v), v);
14  synchronize; // Synchronize cooperative threads
15 }
```



Path Identification
✓ DispelEntity()
Path Sampling
✓ Initialize()
✓ Expand()
Filtering & Result Computation
✓ ReduceVertex()

Asynchronous Path Identify

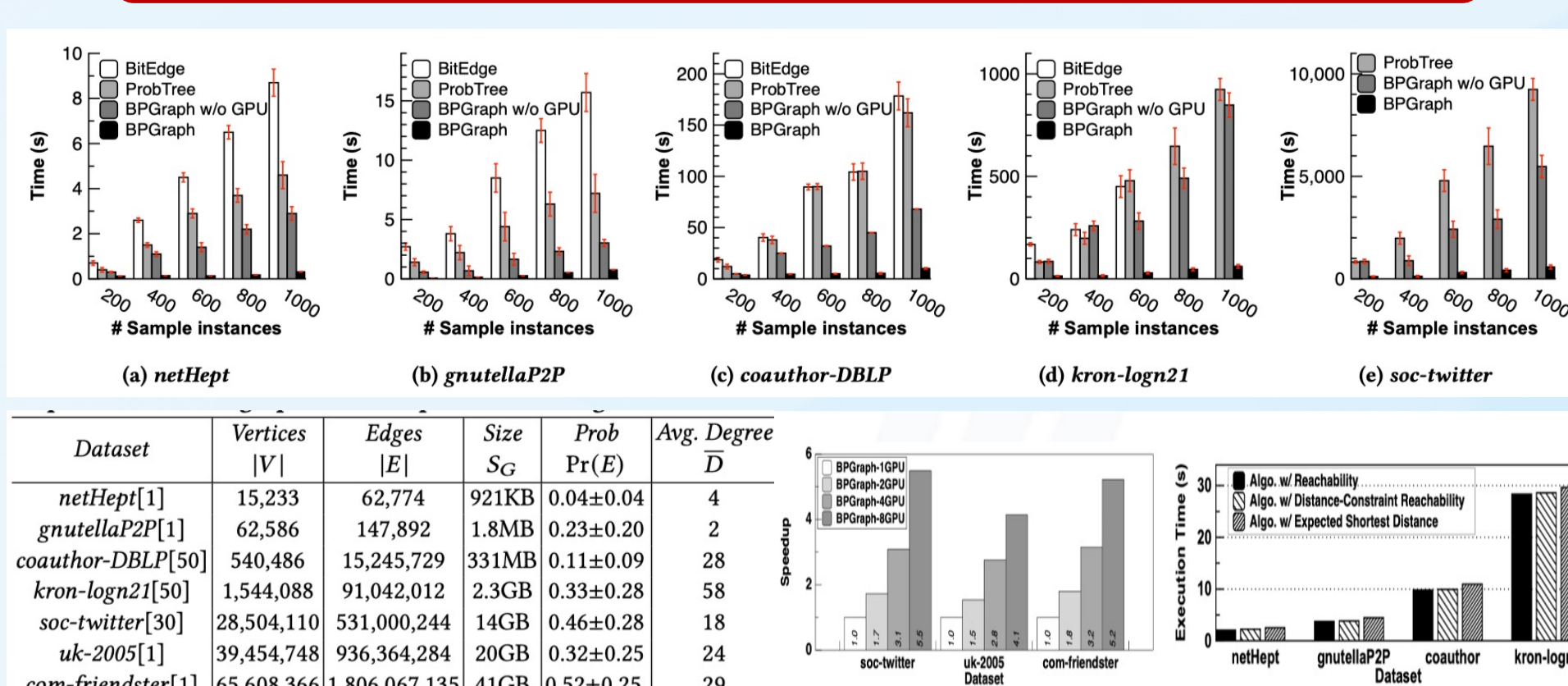
- GPU threads propagate starting vertex along dependency tree
- Asynchronous path identifying
- Compact path storing

Cooperative Group Based Path Sampling

- Tile generation
- Eliminating redundant sampling

- Streaming Uncertain Graph Partition & Allocation
- Dynamic Workload Migration via Dual Buffering
- Coordinated Data Offloading via Collective Operations

Experimental Evaluation



- BPGraph achieves 69×, 43× and 26× speedup on average than CPU-based BE, PT, DR due to well exploited GPUs.
- BPGraph achieves 5.2×, 3.2×, 1.8× speedup using 8 GPUs, 4 GPUs and 2 GPUs than single GPU.

- [1] Heng Zhang, Lingda Li, Hang Liu, et al. 2022. Bring orders into uncertainty: enabling efficient uncertain graph processing via novel path sampling on multi-accelerator systems. In **Proceedings of the 36th ACM International Conference on Supercomputing (ICS '22)**. Association for Computing Machinery, New York, NY, USA, Article 11, 1–14.
- [2] Heng Zhang, Lingda Li, et al. An Efficient Uncertain Graph Processing Framework for Heterogeneous Architectures[C]. **ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)**, 2021. PPoPP'21.