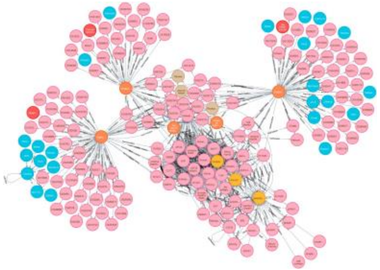
Optimization of GNN Training Through Half-precision

Arnab Kanti Tarafder, Yidong Gong, Pradeep Kumar
William & Mary

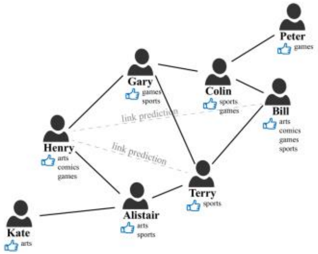
Graph Neural Networks(GNN)

GNN applications are widely used: GCN, GIN, GAT, etc.

Social Network



Biomedical



Recommendation System

Transportation

Financial Risk Detection

Credit: Google Image

1-Page Summary

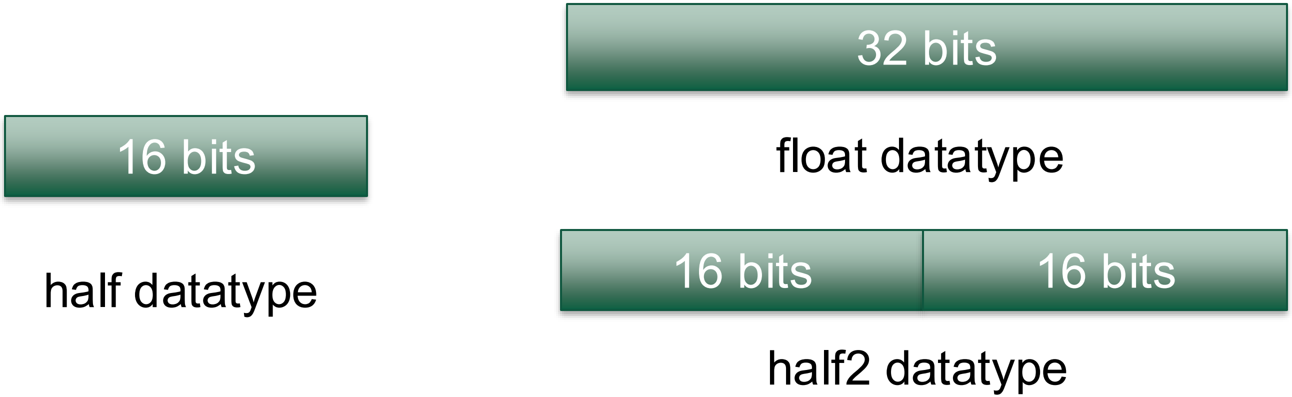
- Half-precision GNN training
  - Suffers from overflow/abnormal accuracies
  - Poor Kernel performance in half-precision
- Not much prior work
  - Training GNN in lower-precision
- We introduce HalfGNN
  - Strong baseline for GNN training & kernels
  - Discretized Non-atomic SpMM for handling overflow
  - Better data load/compute/store using Half2 datatype
  - Proposed Half4/Half8 data-type for even better data-load/store

Outline

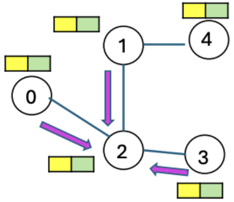
- Background: Half-Precision, SpMM, SDDMM
- Investigation: Accuracy & Performance
- Contributions
  - Discretized SpMM
  - Faster Load/Store/Compute
  - Non-Atomic Kernel Design
- Evaluation

Background

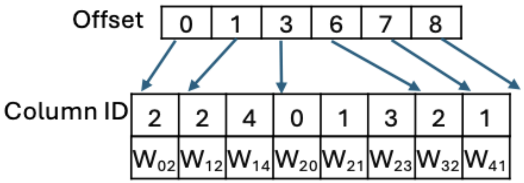
- Float/Single-precision: 32 bits
- Half-precision: 16 bits
- Half2:
    - 2 half-precision data in 32 bits



Background: SpMM ($Y = A_w X$)



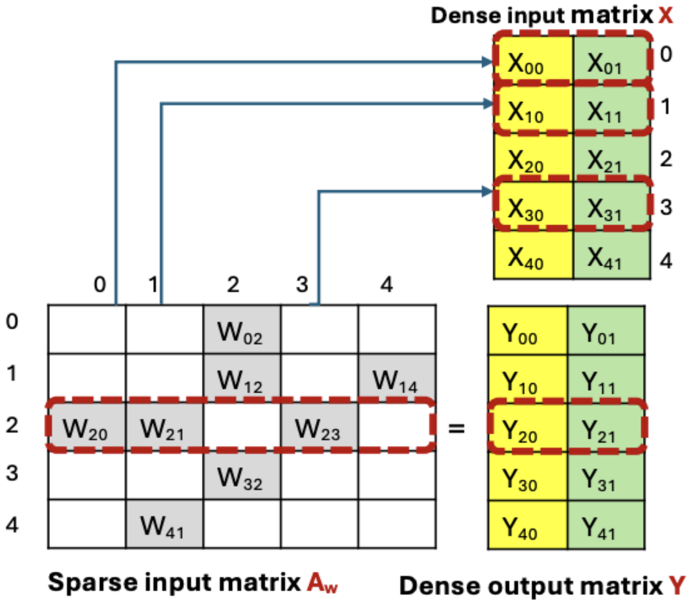
Graph



CSR format

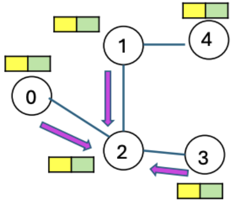
Row ID	0	1	1	2	2	2	3	4
Column ID	2	2	4	0	1	3	2	4
	W_{02}	W_{12}	W_{14}	W_{20}	W_{21}	W_{23}	W_{32}	W_{41}

COO format

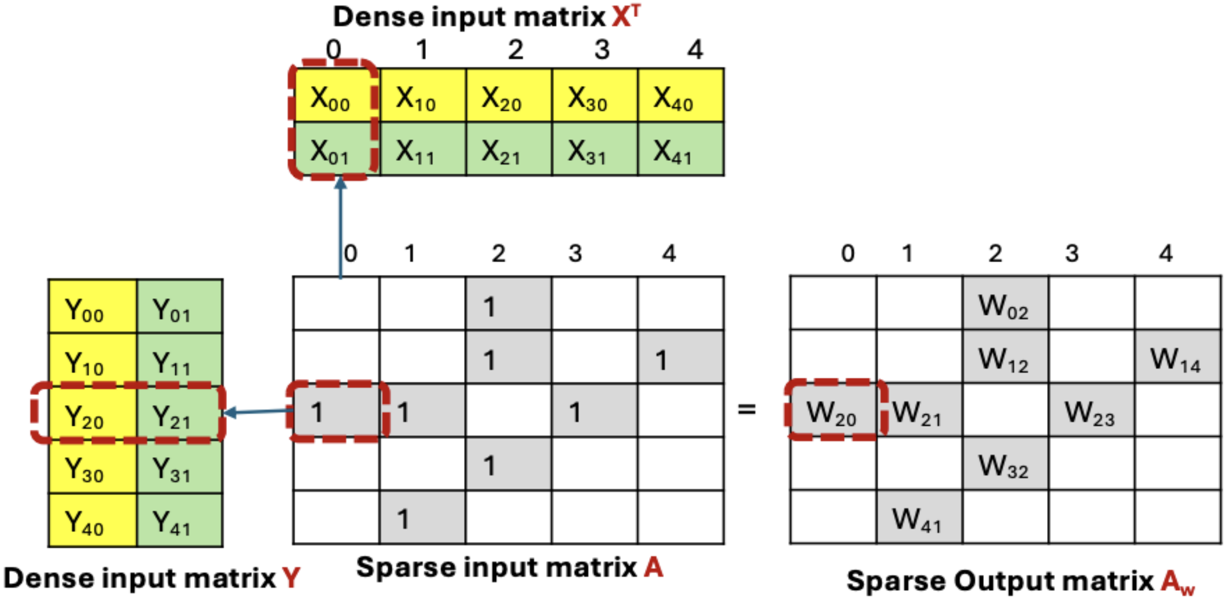


SpMM

Background: SDDMM ($W = A \odot (YX^T)$)



Graph



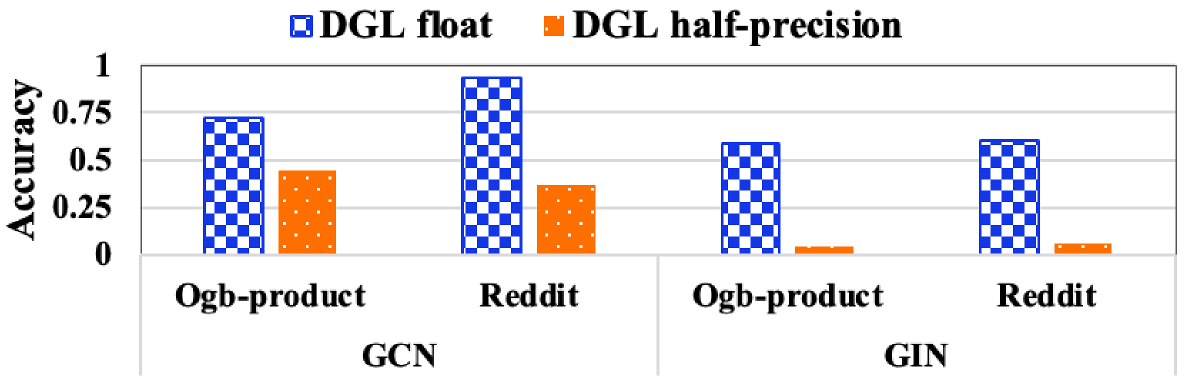
SDDMM

Outline

- Background: Half-Precision, SpMM, SDDMM
- Investigation: Accuracy & Performance
- Contributions
  - Discretized SpMM
  - Faster Load/Store/Compute
  - Non-Atomic Kernel Design
- Evaluation

Investigation: Abnormal Accuracy

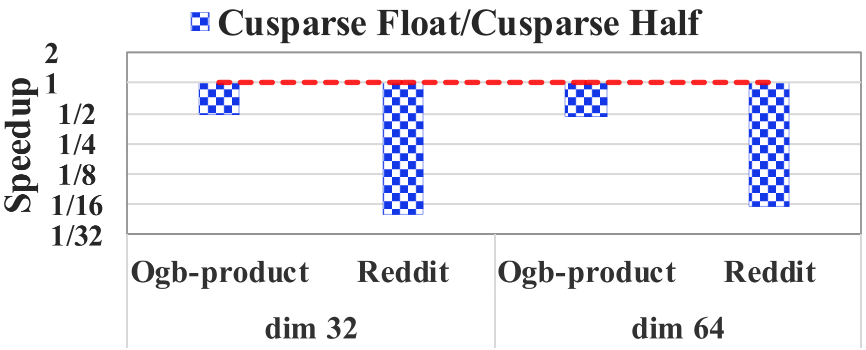
- DGL can not train GCN, GIN in half-precision (16 bits)



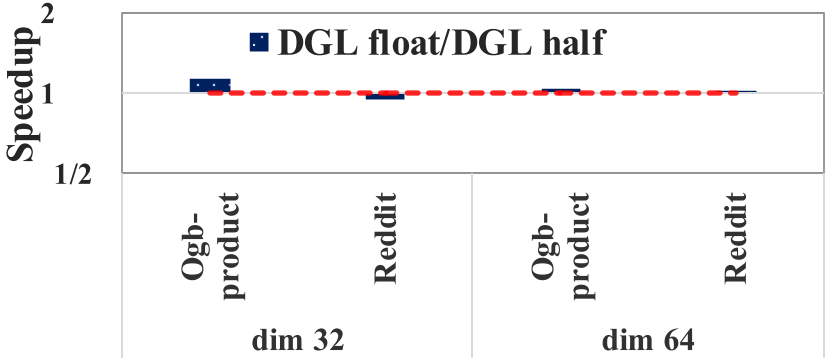
Poor accuracy of DGL in half-precision

Investigation: Poor Performance

- DGL half-precision SpMM kernels are slower than float-based kernels
  - DGL uses Cusparse for SpMM
- DGL SDDMM does not gain any performance for half-precision



Half-precision SpMM is significantly slower



Similar SDDMM runtime

Investigation: Value Overflow

GCN: $\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{H}^{(l)}\mathbf{W}^{(l)})\right)$

GIN: $h_u = \phi\left((1+\epsilon)\cdot x_u + \sum_{v \in N_v} x_v\right)$

- **Overflow**:
  - During aggregation of SpMM for a single vertex before normalization
  - During addition of self-features to SpMM in GIN
  - Even for relatively small sized datasets, half-precision gets out of range

Investigation: Mixed-Precision System API

- Mixed-precision allows half-precision integration in DGL
  - All state tensors in 16 bits, Master weight updates in 32 bits
  - Pytorch backend '*fears numerical instability*' and avoids *half-precision:*
    - *exp, softmax in GAT are invoked in float*

- Once one of these OPs are encountered,
  - The rest of the compute are in float
  - Or one more data-conversion from float to half-precision

https://pytorch.org/docs/stable/amp.html

Outline

- Background: Half-Precision, SpMM, SDDMM
- Investigation: Accuracy & Performance
- Contributions
  - Discretized SpMM
  - Faster Load/Store/Compute
  - Non-Atomic Kernel Design
- Evaluation

Contribution: Discretized SpMM

- Apply normalization after each warps' computation, rather than after the gather of each vertex
  - Normalization after maximum of 128 edges per warp
  - Applied to both GCN, GIN

SpMM *Discretized SpMM*

Contribution: Removal of Overflow Fear for GNN

- GAT Softmax: $m_i = \max_{j \in \mathcal{N}_i}(e_{ij}), e'_{ij} = \exp(e_{ij} - m_i)$

$$\alpha_{ij} = \frac{e'_{ij}}{\sum_{j \in \mathcal{N}_i}(e'_{ij})}$$

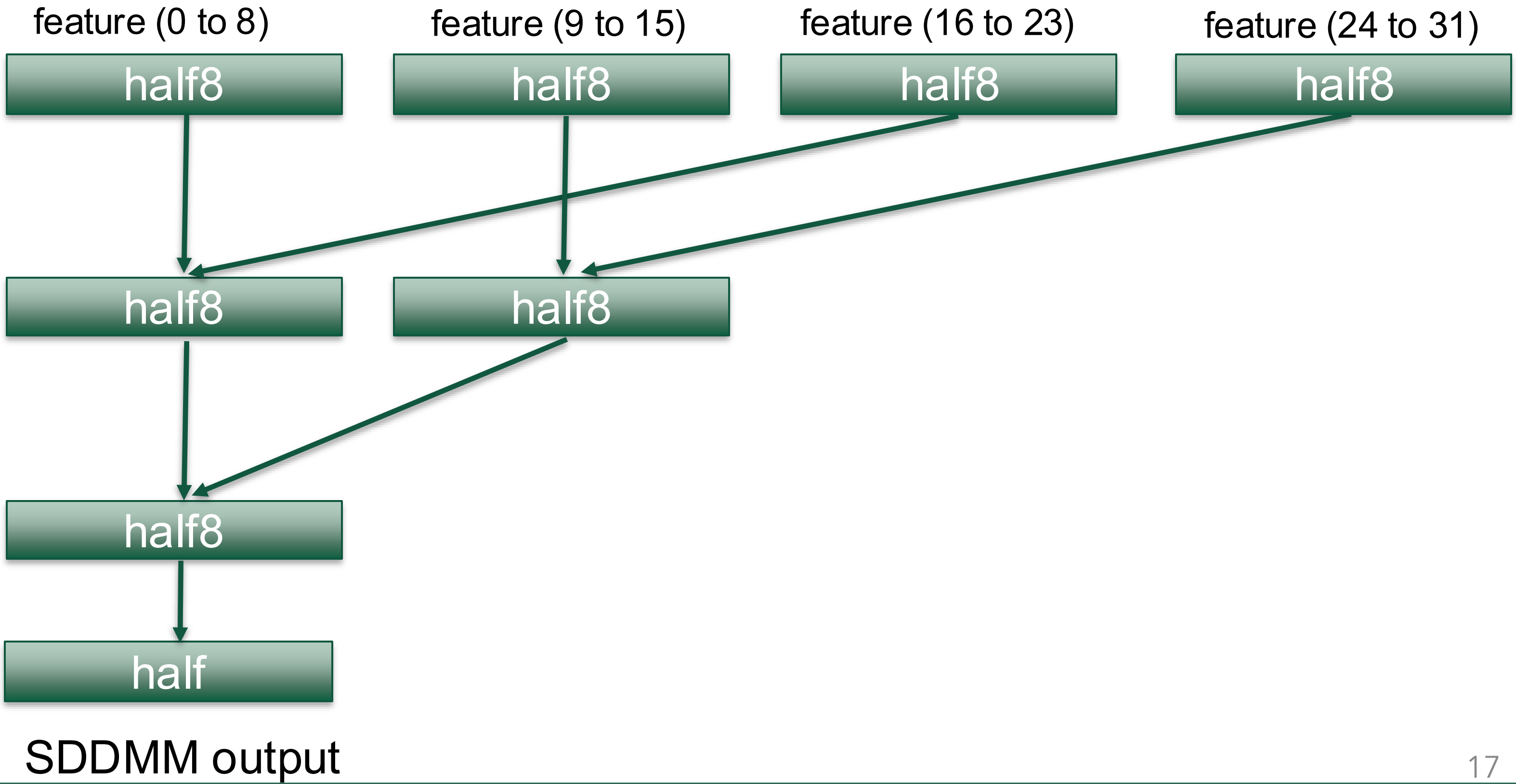
- Impossible to overflow in half-precision
- HalfGNN:
  - Ensures all the kernels in backend run in half-precision
  - Automatic, and integrated in the API

Contribution: Faster Load/Store/Compute

- ***half2, half4, half8***
  - Interchangeable during load/store for float, float2 and float4
  - GNNONE (HPDC'24) exploited float4 load for SDDMM
- Increase per thread throughput
  - e.g., 8 ops (+, -, /, *, max, min) in one op during ***half8***
  - Reduces inter-thread communication during tree-reduce of SDDMM
  - Reduction in synchronization steps for SDDMM

Y. Gong and P. Kumar. GNNONE : A Unified System Optimizations for GNN Kernels. In The 33rd International Symposium on High-Performance Parallel and Distributed Computing (HPDC '24), 2024.

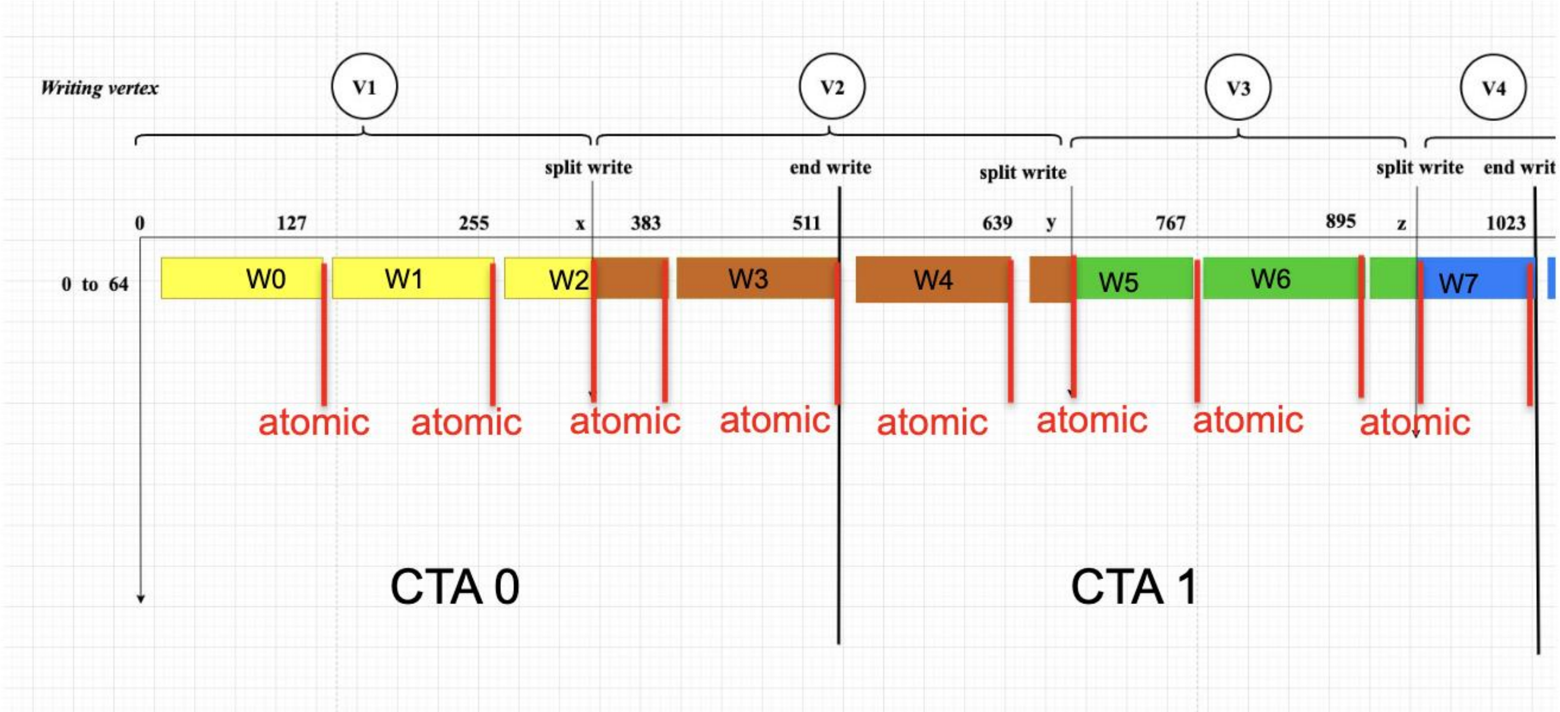
Contribution: Faster Load/Store/Compute



Outline

- Background: Half-Precision, SpMM, SDDMM
- Investigation: Accuracy & Performance
- Contributions
  - Discretized SpMM
  - Faster Load/Store/Compute
  - Non-Atomic Kernel Design
  - Generality of the Solution
- Evaluation

SpMM with Atomic Writes



A typical edge-parallel design

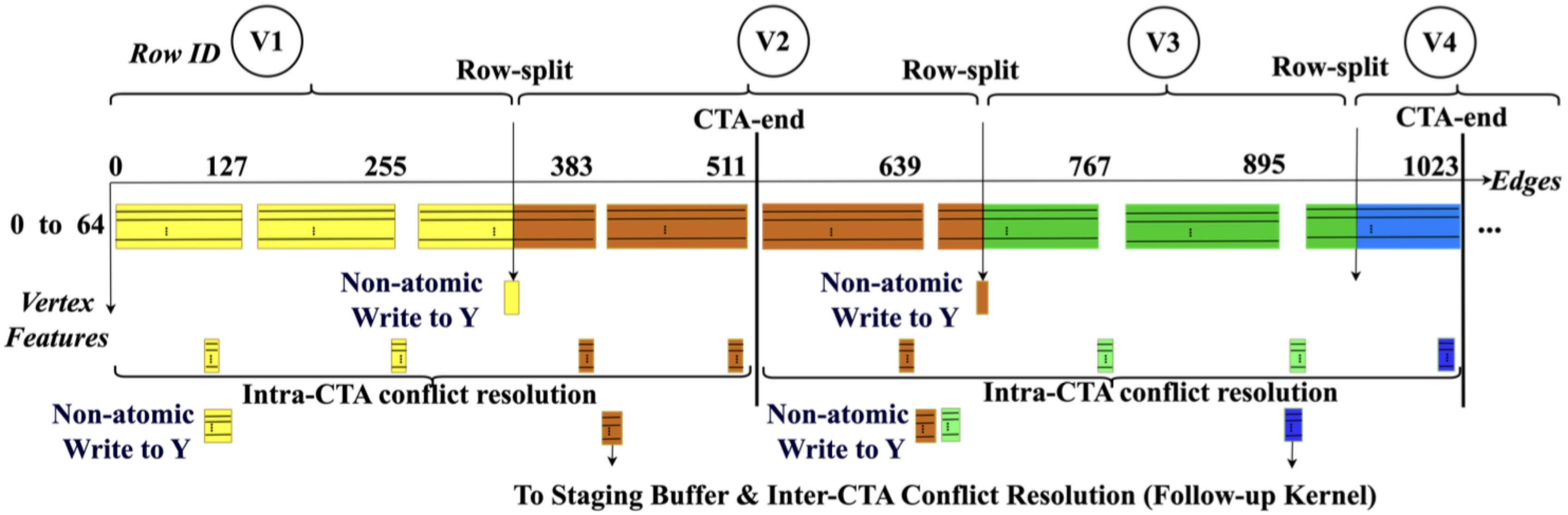
Contribution: Non-atomic Design

2 Stage Edge-Parallel Kernel Design:

- Use the 1st stage to perform
  - Data load, Discretized reduction in SpMM
  - Reduce Intra-CTA & Inter-CTA conflict
  - Use shared memory to communicate among warps
  - Update temporary buffer (**Carryout Buffer**)

- Use stage 2 kernel to write in destination vertices non-atomically
  - Destination vertices can be pre-calculated without overhead in parallel manner

Contribution: Non-atomic Design



Non-atomic Design of SpMM with Carry-out Buffer

Contribution: Generality of the Solution

- Applicable to other systems
  - Vector load/store/compute
  - Non-atomic reduction & write
  - Reduced inter-thread parallelism
  - Improved feature-parallelism through increasing thread operations

- Can Improve Vertex-Parallel Work

Outline

- Background: Half-Precision, SpMM, SDDMM
- Investigation: Accuracy & Performance
- Contributions
  - Discretized SpMM
  - Faster Load/Store/Compute
  - Non-Atomic Kernel Design
- Evaluation

Evaluation

➢ GNN Training:

- GCN : 2 layers
- GIN : 5 layers
- GAT : 3 layers with 1 head
- Hidden dim = 64

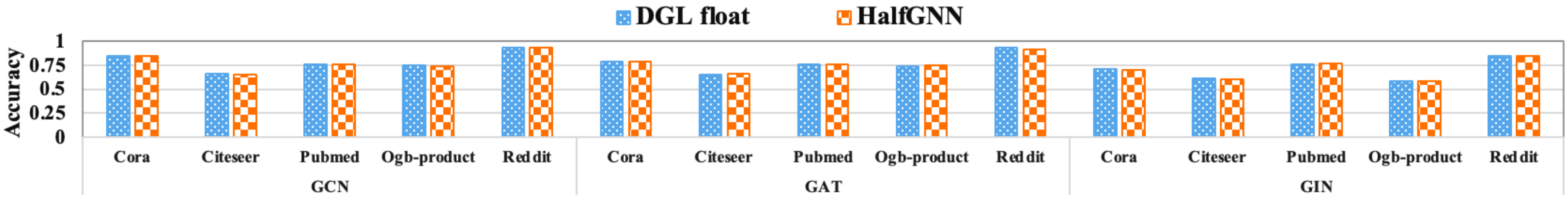
➢ Evaluation Platform:

- All studies are based on Nvidia A100 GPU

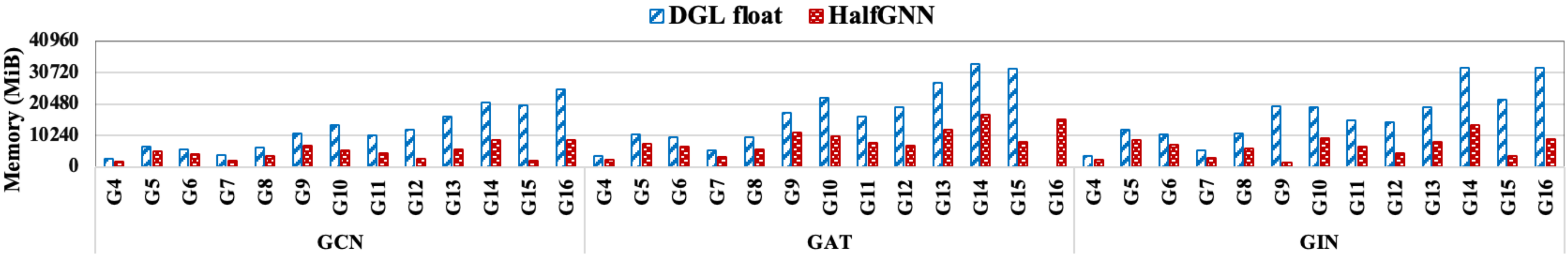
Graph Dataset	Vertex Count	Edge Count	F	C
Cora (G1)*	2,708	10,858	1,433	7
Citeseer (G2)*	3,327	9,104	3,703	6
PubMed (G3)*	19,717	88,648	500	3
Amazon (G4)	400,727	6,400,880	150	7
Wiki-Talk (G5)	2,394,385	10,042,820	150	7
RoadNet-CA (G6)	1,971,279	11,066,420	150	7
Web-BerkStand (G7)	685,230	15,201,173	150	7
As-Skitter (G8)	1,696,415	22,190,596	150	7
Cit-Patent (G9)	3,774,768	33,037,894	150	7
Sx-stackoverflow (G10)	2,601,977	95,806,532	150	7
Kron-21 (G11)	2,097,152	67,108,864	150	7
Hollywood09 (G12)	1,069,127	112,613,308	150	7
Ogb-product (G13)*	2,449,029	123,718,280	100	47
LiveJournal (G14)	4,847,571	137,987,546	150	7
Reddit (G15)*	232,965	114,848,857	602	41
Orkut (G16)	3,072,627	234,370,166	150	7

Benchmarking Graph Datasets

Evaluation: Training Accuracy and Memory consumption

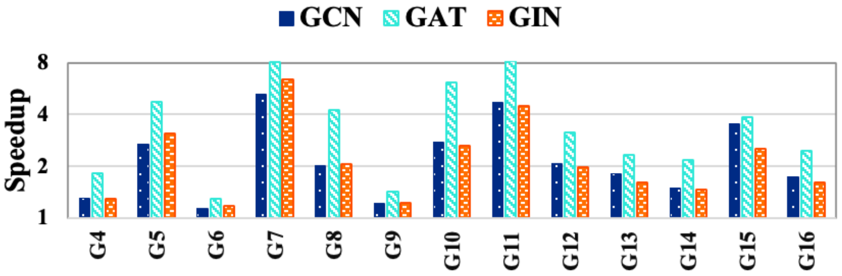


HalfGNN achieves the same accuracy as DGL float

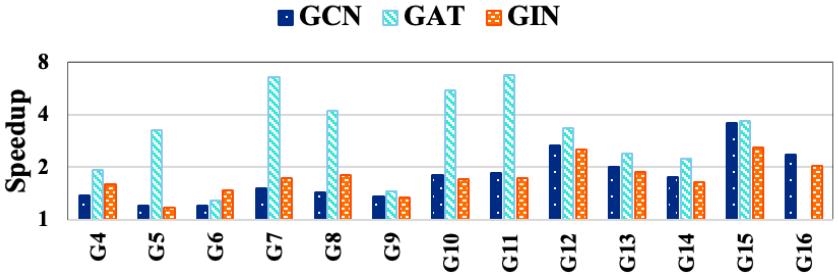


HalfGNN reduces required memory during training

Evaluation: Training Runtime

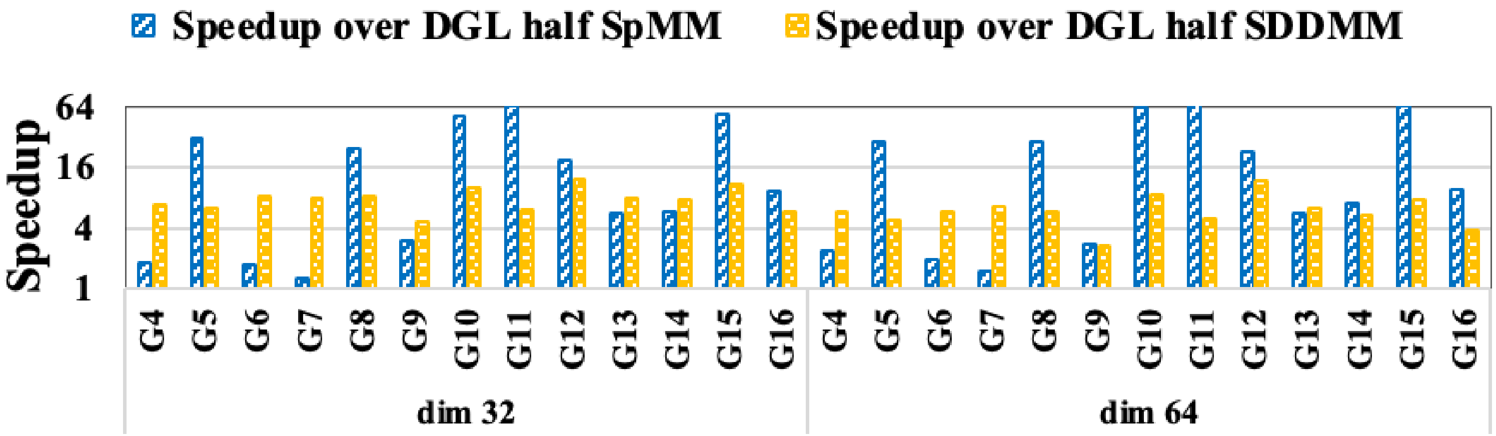


➢ HalfGNN achieves 2.44x, 3.84x, and 2.42x avg. speedup for GCN, GAT, and GIN over DGL-half



➢ HalfGNN achieves 1.85x, 3.55x, and 1.78x avg. speedup for GCN, GAT, and GIN over DGL-float

Evaluation: Kernel Speedup



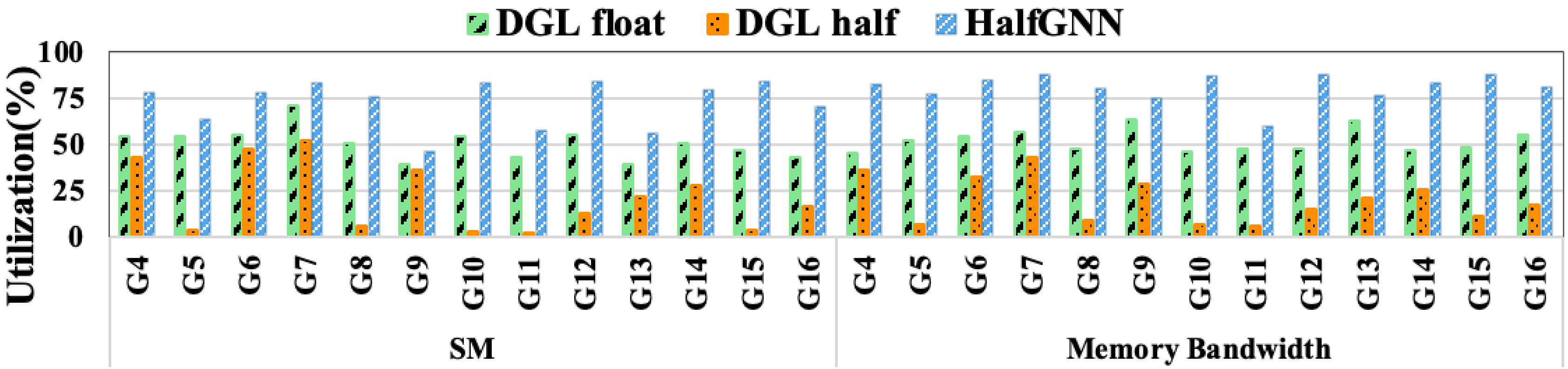
SpMM:

- HalfGNN achieves 22.89x (avg.) speedup over DGL-half

SDDMM:

- HalfGNN achieves 7.12x (avg.) speedup over DGL-half

Evaluation: SpMM Hardware Utilization



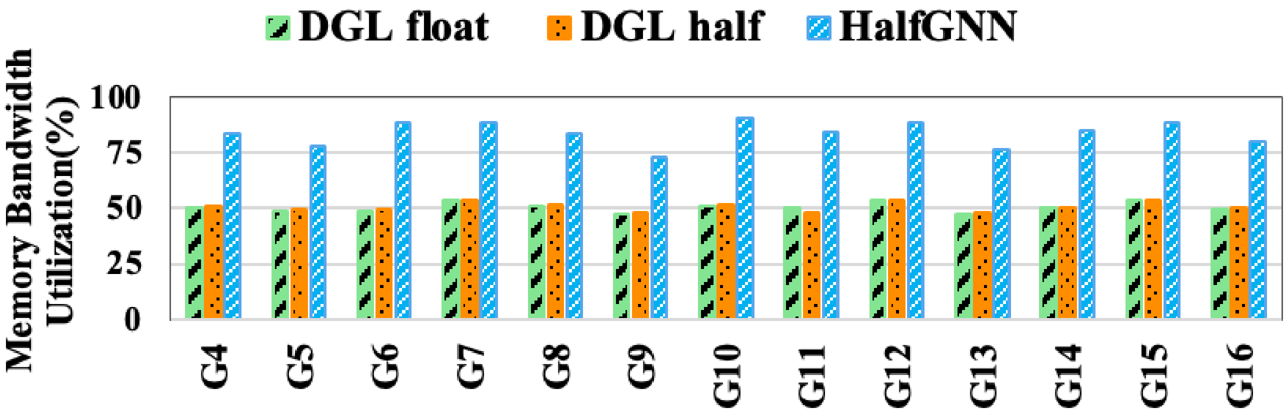
SM utilization:

- HalfGNN : 72.96 %
- DGL float: 50.81 %
- DGL half : 21.58 %

Memory bandwidth utilization:

- HalfGNN : 80.92 %
- DGL float: 51.99 %
- DGL half : 20.22 %

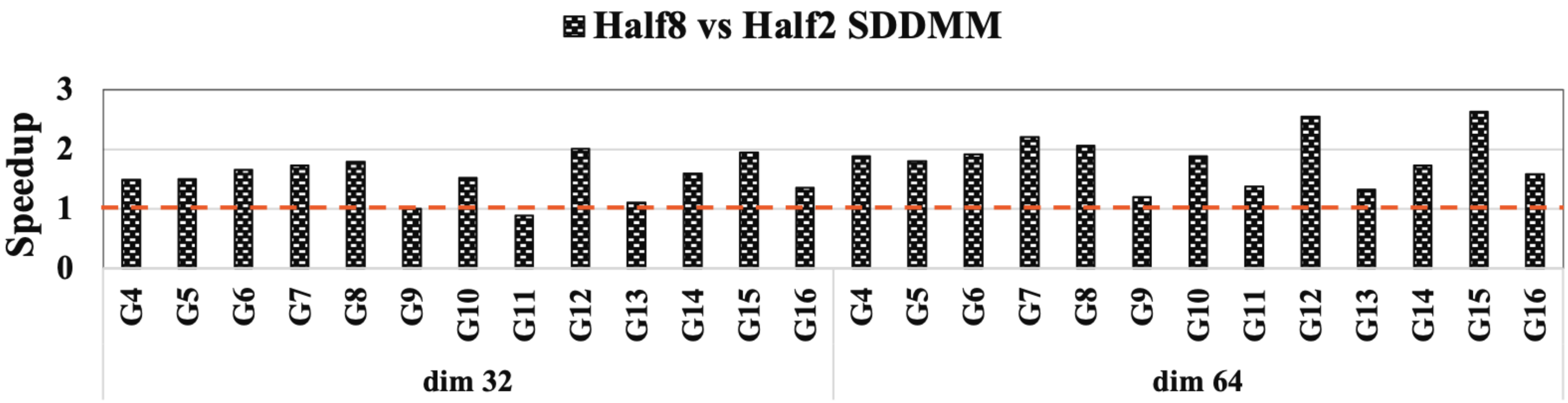
Evaluation: SDDMM Bandwidth Utilization



Memory Bandwidth Utilization:

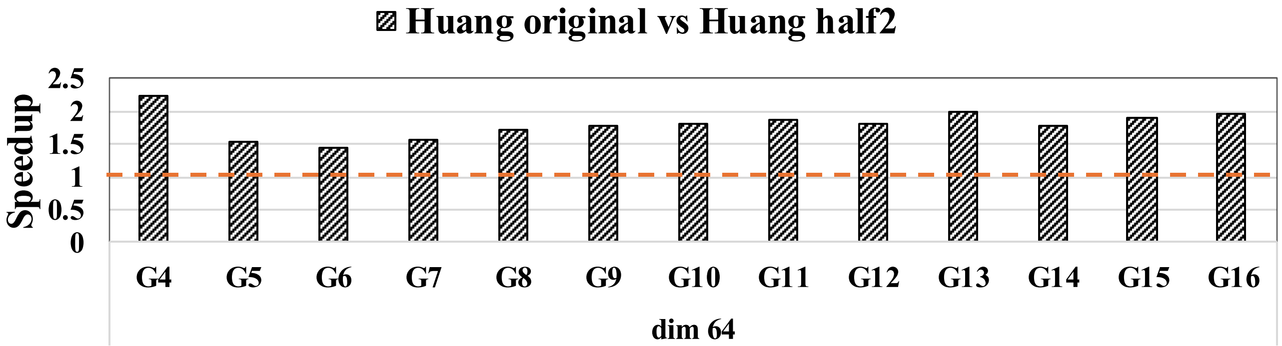
- HalfGNN : 83.71 %
- DGL-float: 50.59 %
- DGL-half : 50.85 %

Evaluation: Half8 vs Half2 for SDDMM



Half8 in SDDMM reduces synchronization bottleneck

Evaluation: Generality of Optimizations



Huang-float vs Huang-half2 speedup: On average gains 1.79x speedup.

K. Huang, J. Zhai, Z. Zheng, Y. Yi, and X. Shen. Understanding and bridging the gaps in current GNN performance optimizations In Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 119–132, 2021.

Take Away

- ✓ Half-precision can provide significantly faster SpMM, SDDMM kernels
- ✓ Strong new baseline in GNN training
- ✓ Optimization techniques applicable to other systems

[Github]: https://github.com/the-data-lab

Thank You

Q & A

WILLIAM & MARY

CHARTERED 1693

NSF