At-Scale Sparse Deep Neural Network Inference With Efficient GPU Implementation

Mert Hidayetoglu, Carl Pearson, Vikram Sharma Mailthody,
Eiman Ebrahimi[*], Jinjun Xiong[†], Rakesh Nagi, and Wen-mei Hwu

University of Illinois at Urbana-Champaign
[*]NVIDIA
[†]IBM Research

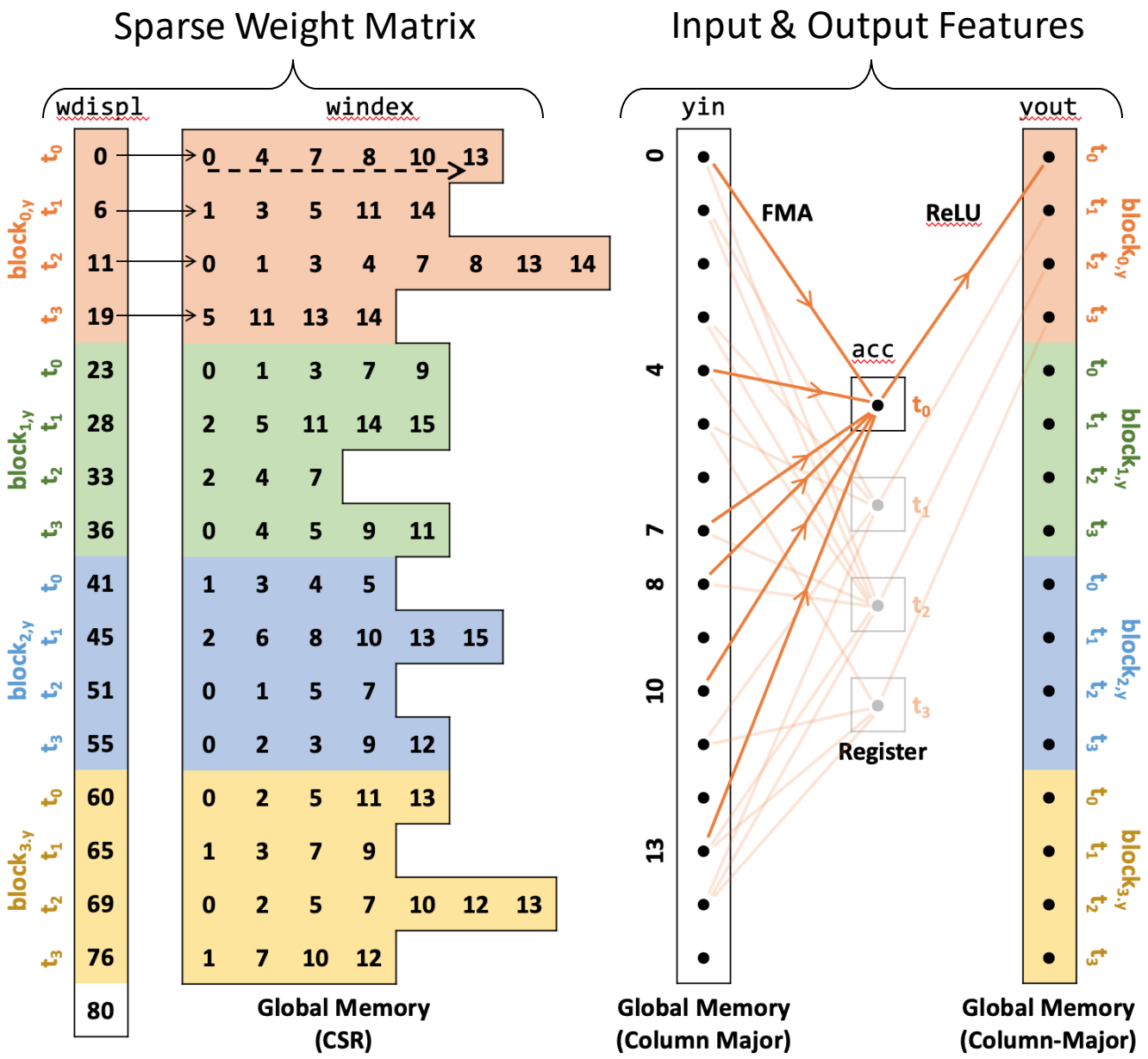
IEEE High Performance Extreme Computing (HPEC)
Wednesday, Sep. 23, 2020, 17:30 ET

ILLINOIS Coordinated Science Laboratory

IBM C3SR center for cognitive computing systems research

Baseline Sparse Layer Implementation

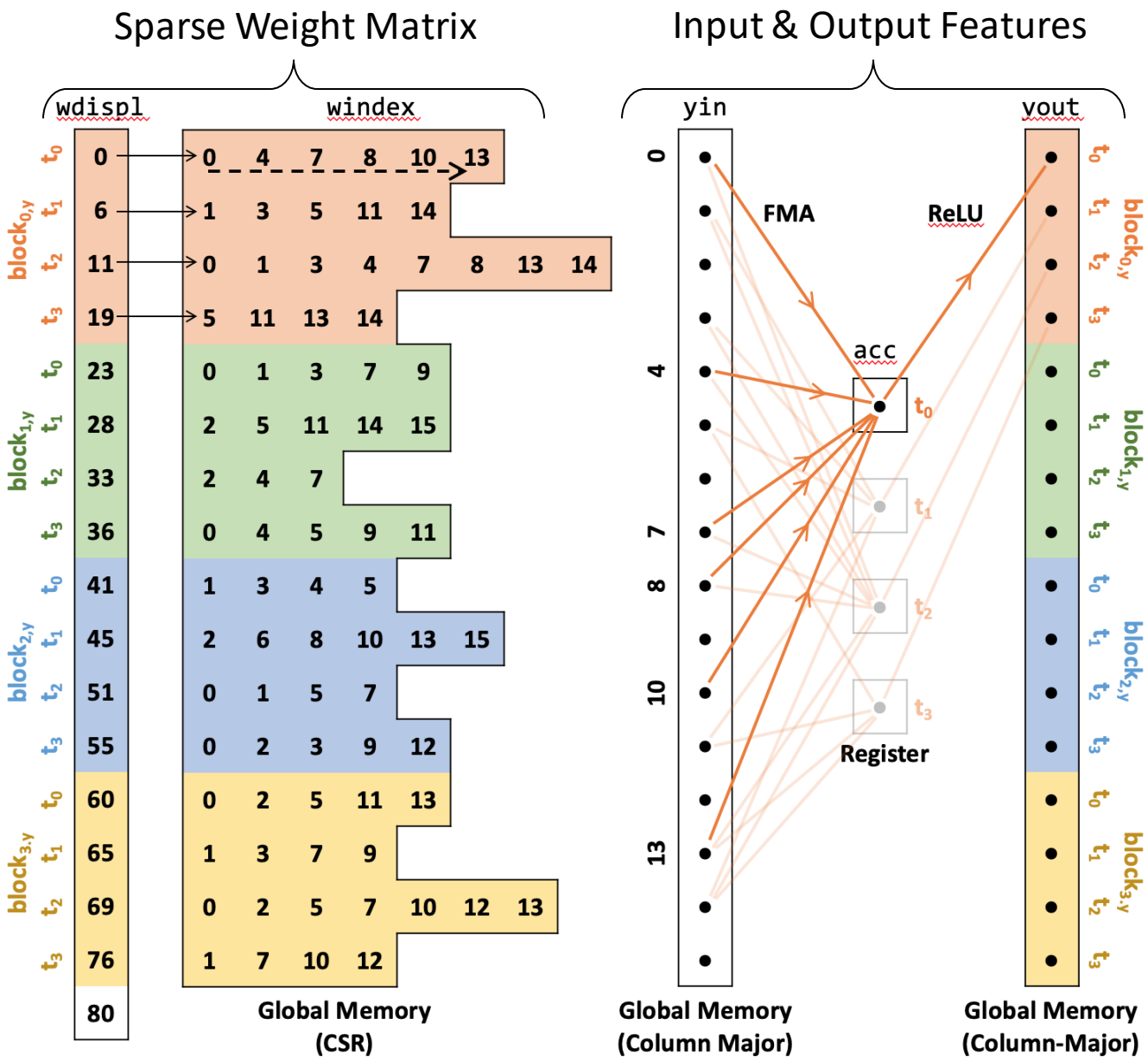
- Sparse weights: CSR
- Activation features: dense (col.-major)
- Gather approach
- Each thread computes a single output
- Inactive features are pruned
- ReLU is fused with SpMM



Baseline Sparse Layer Implementation

- Sparse weights: CSR
- Activation features: dense (col.-major)
- Gather approach
- Each thread computes a single output
- Inactive features are pruned
- ReLU is fused with SpMM

Data Access Redundancies

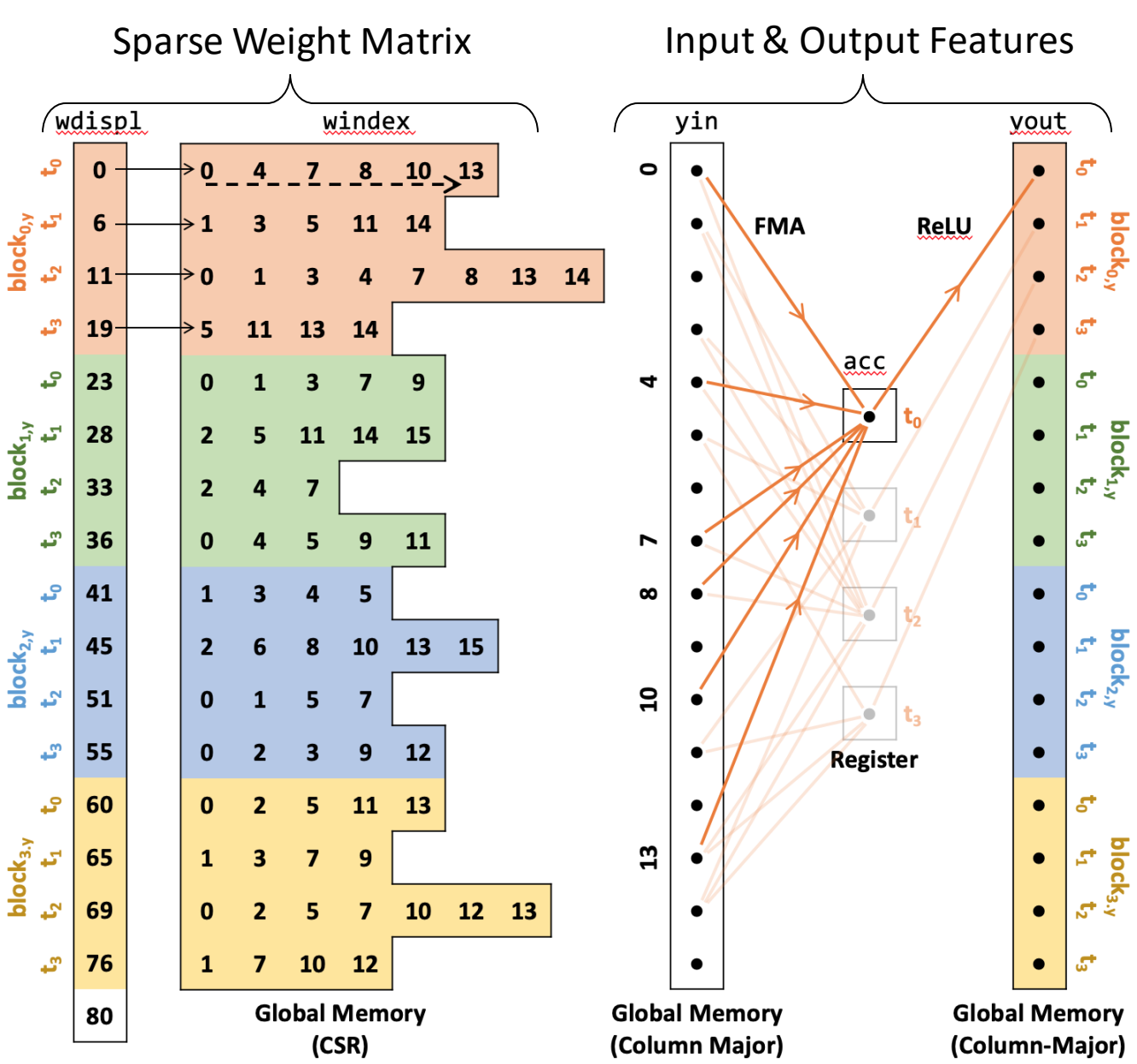


Baseline Sparse Layer Implementation

- Sparse weights: CSR
- Activation features: dense (col.-major)
- Gather approach
- Each thread computes a single output
- Inactive features are pruned
- ReLU is fused with SpMM

Data Access Redundancies

- Weight matrix by all output features
- Input features by different threads



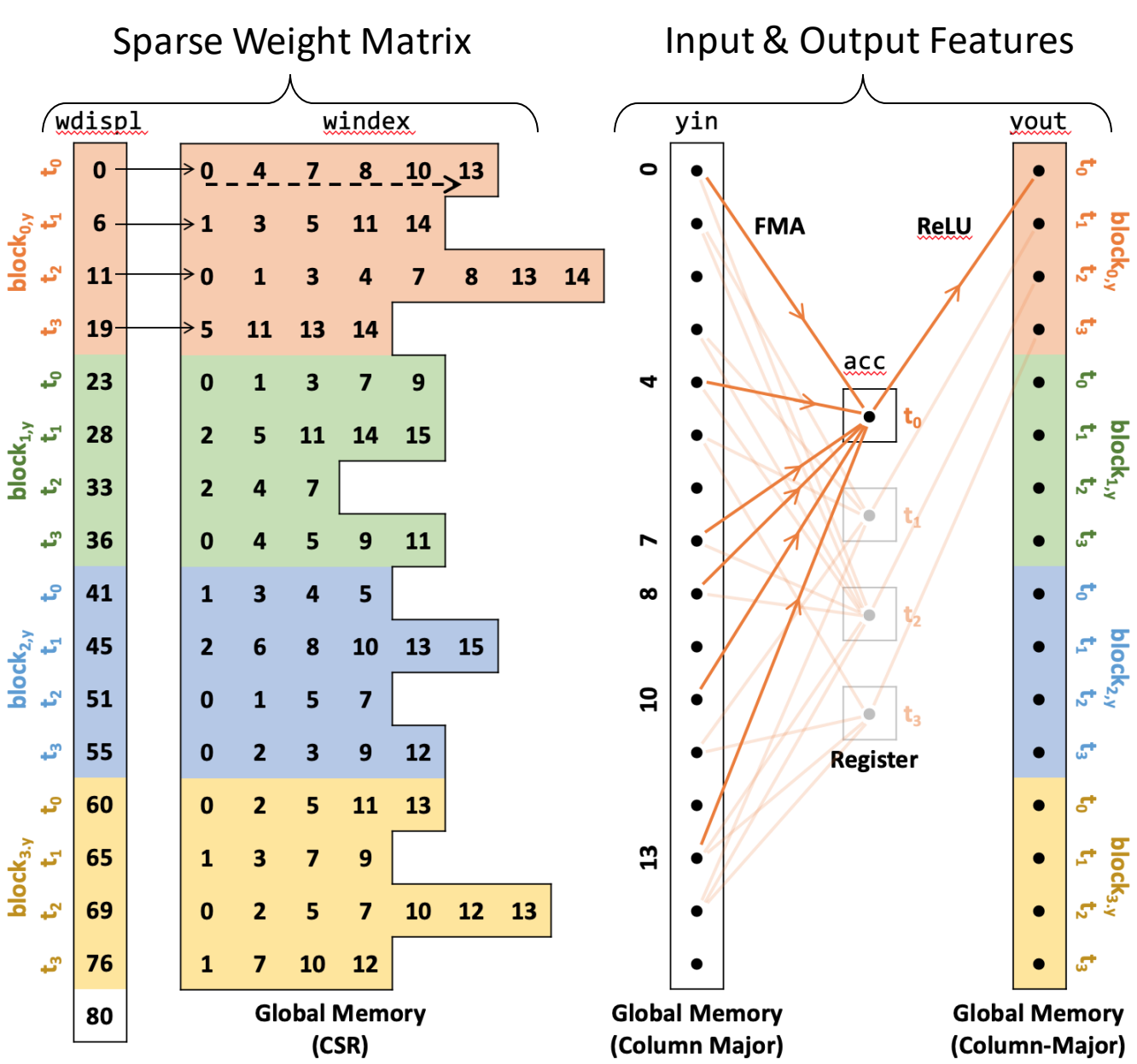
Baseline Sparse Layer Implementation

- Sparse weights: CSR
- Activation features: dense (col.-major)
- Gather approach
- Each thread computes a single output
- Inactive features are pruned
- ReLU is fused with SpMM

Data Access Redundancies

- Weight matrix by all output features
- Input features by different threads

Data Access Latencies



Baseline Sparse Layer Implementation

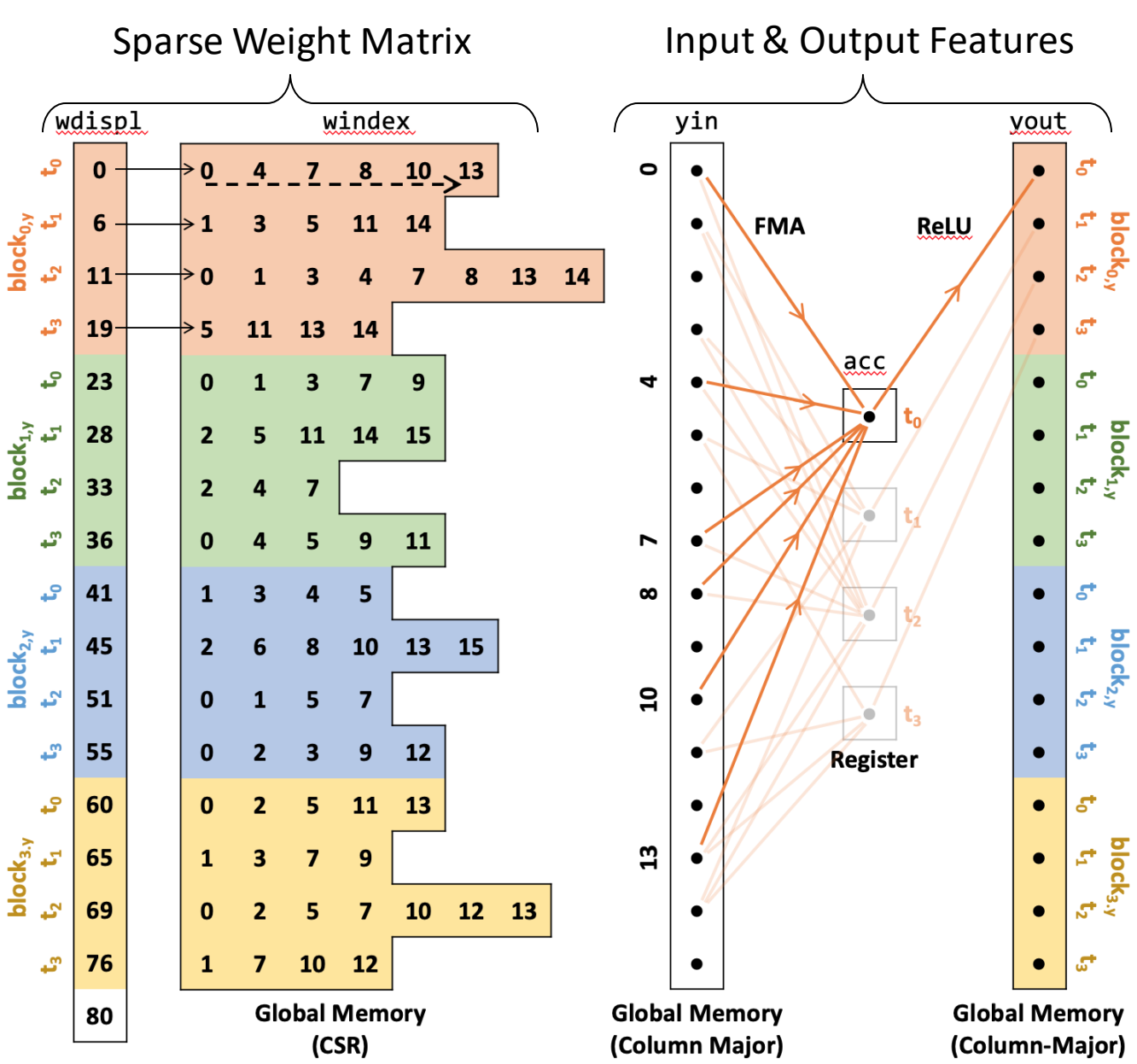
- Sparse weights: CSR
- Activation features: dense (col.-major)
- Gather approach
- Each thread computes a single output
- Inactive features are pruned
- ReLU is fused with SpMM

Data Access Redundancies

- Weight matrix by all output features
- Input features by different threads

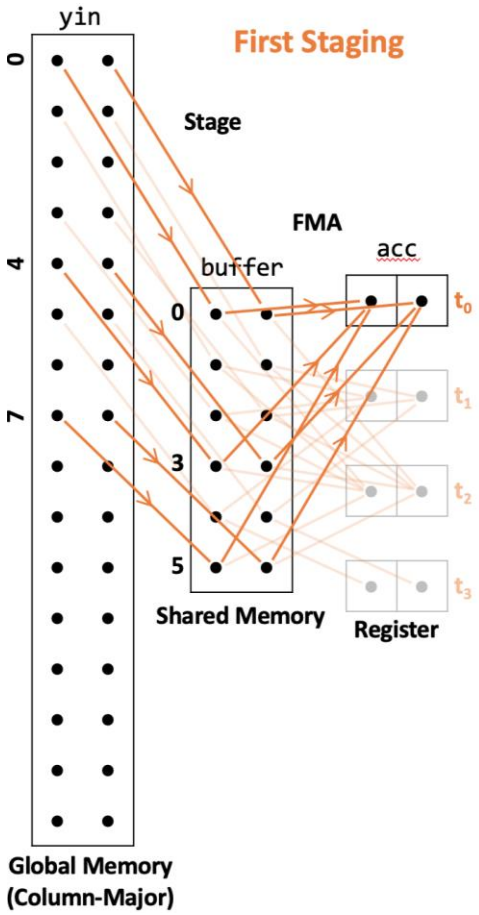
Data Access Latencies

- Irregular access to input features
- Uncoalesced access to weight matrix



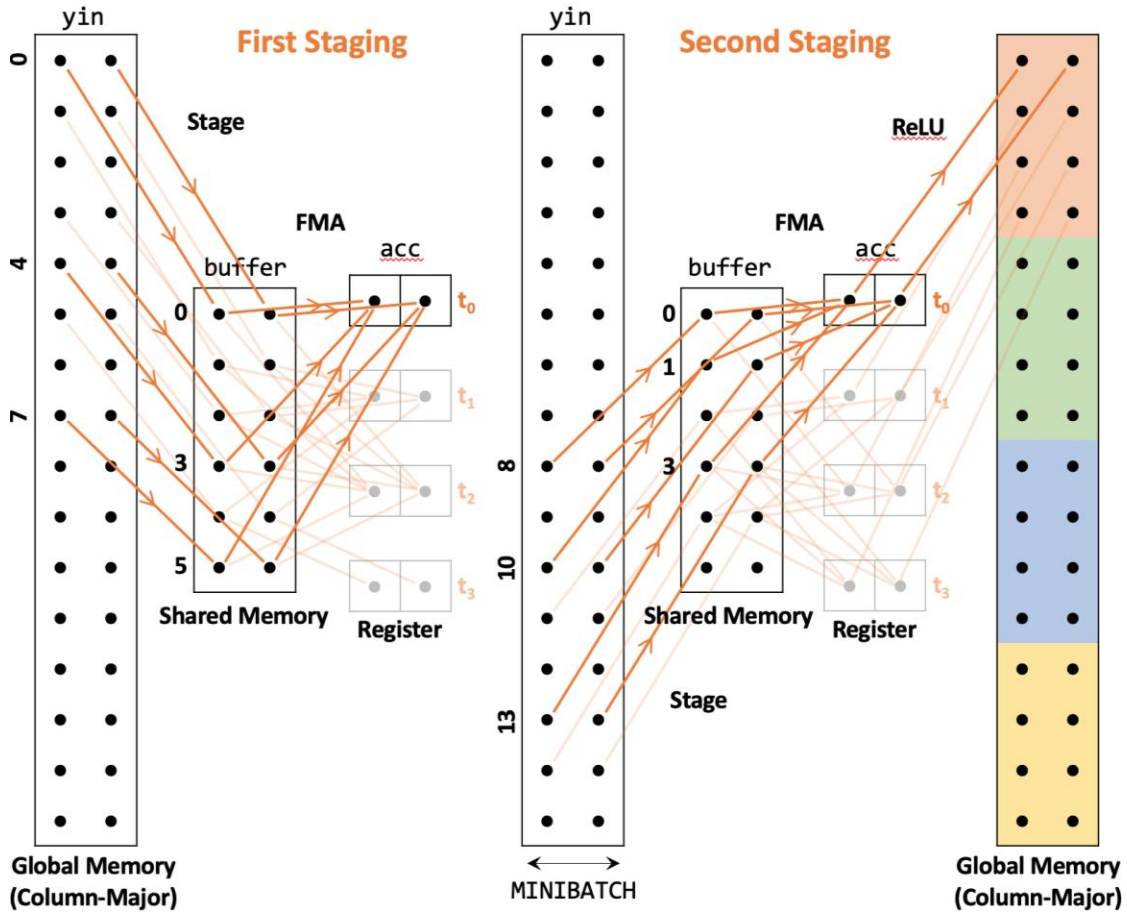
Optimized Sparse Layer Implementation

Multi-Level Input Buffering



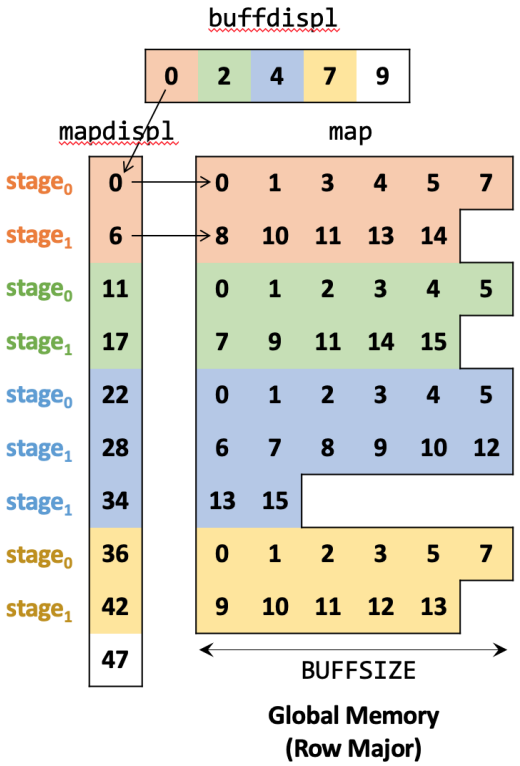
Optimized Sparse Layer Implementation

Multi-Level Input Buffering

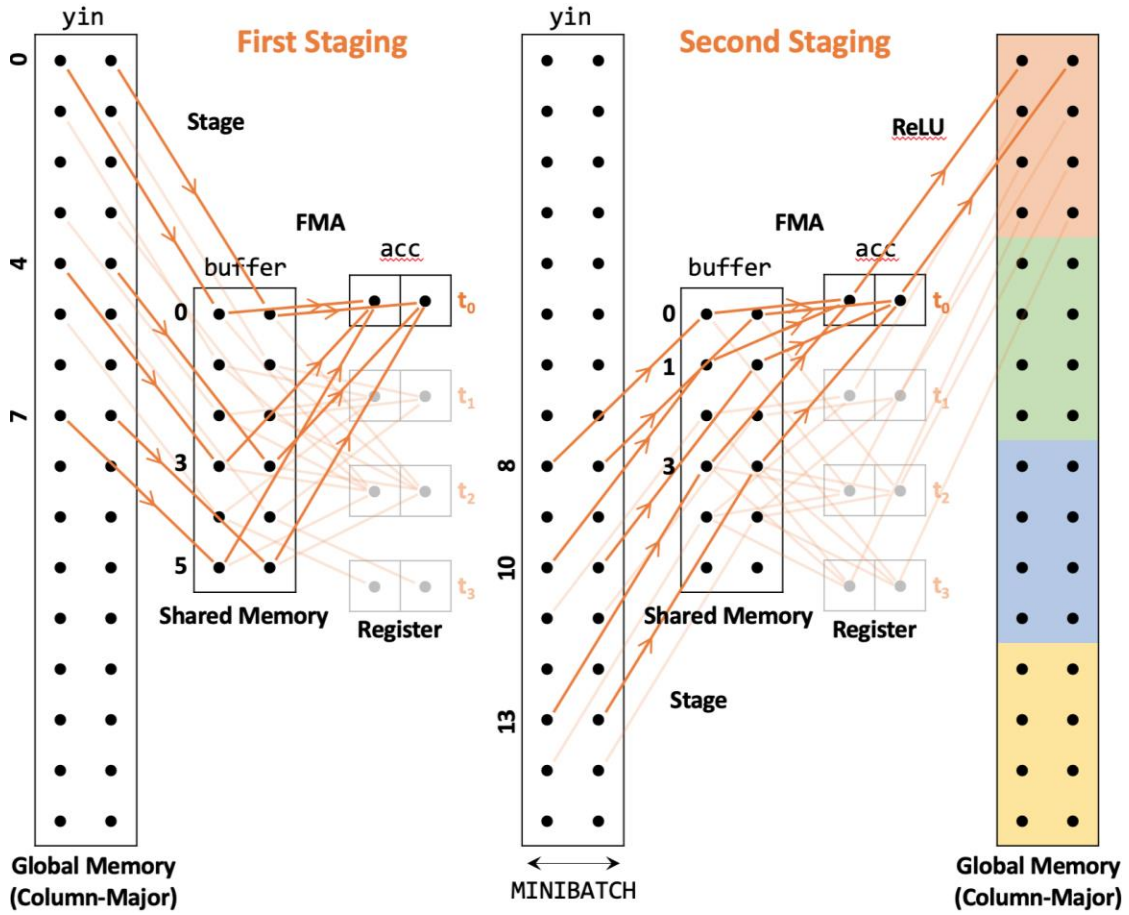


Optimized Sparse Layer Implementation

Intermediate Data Structures

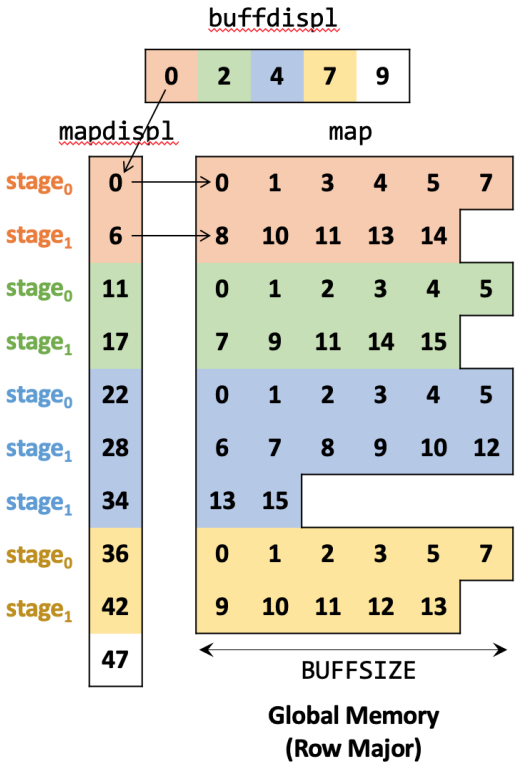


Multi-Level Input Buffering

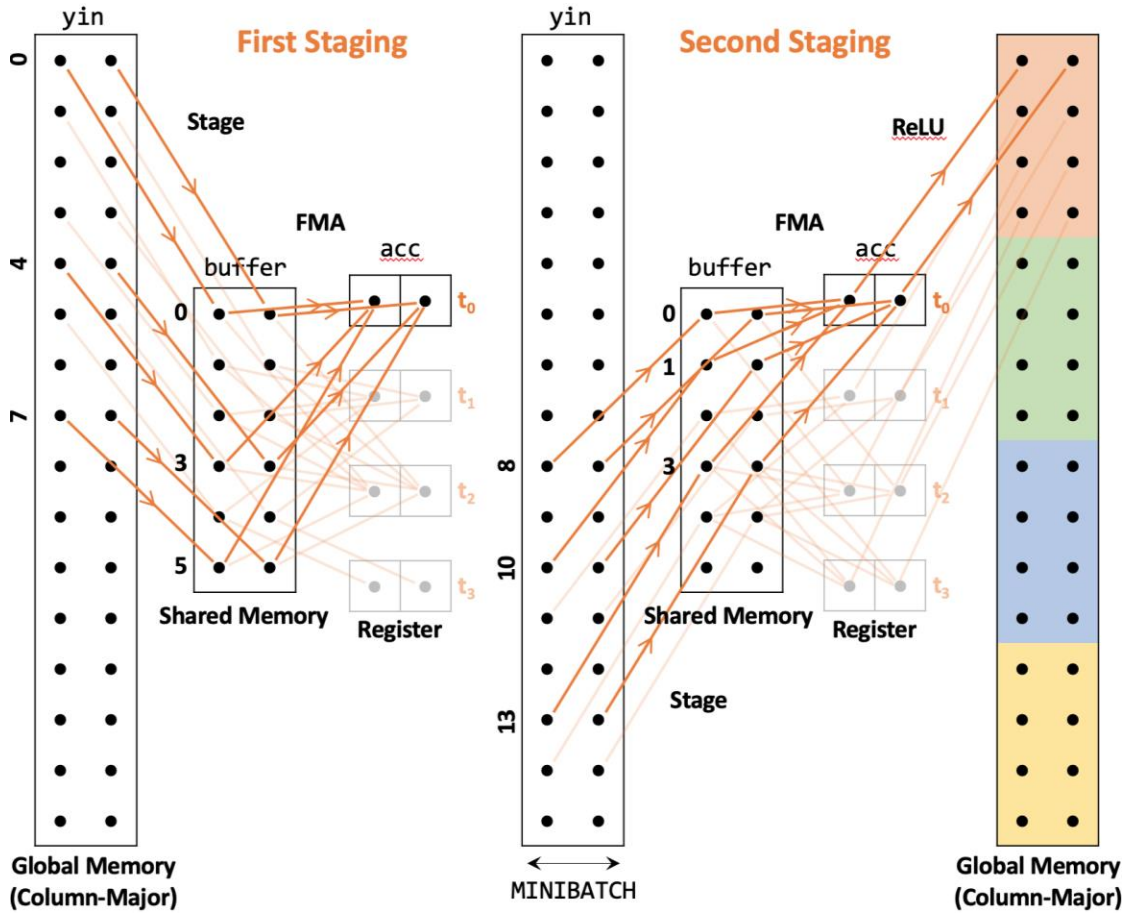


Optimized Sparse Layer Implementation

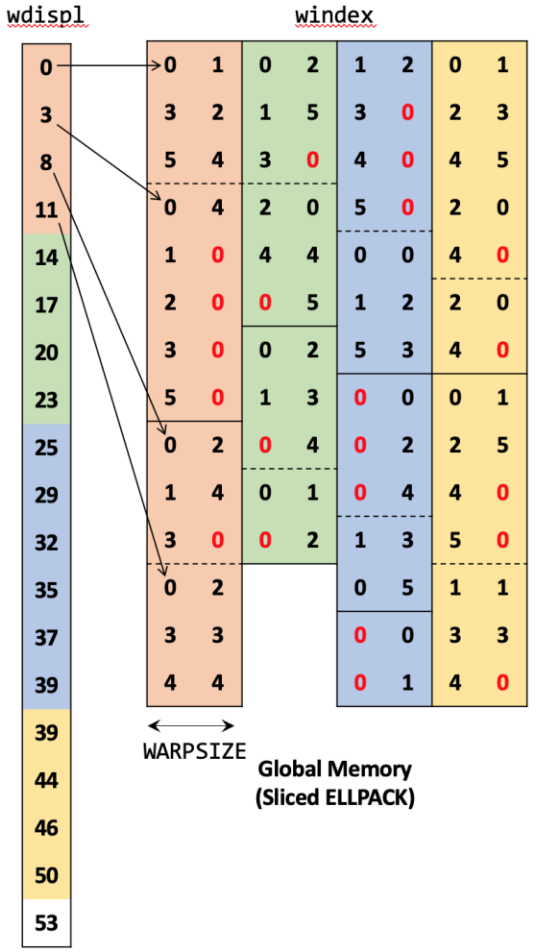
Intermediate Data Structures



Multi-Level Input Buffering



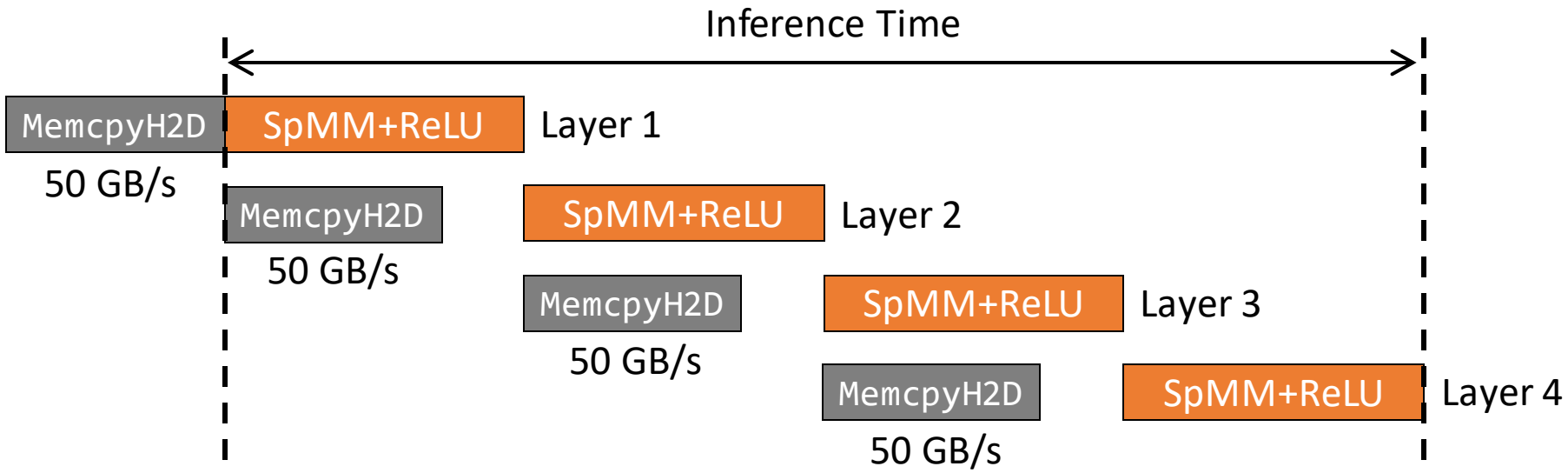
Weights Sliced ELLPACK



Memory Optimizations

- `short` indices
- `half` weights (not used for Graph Challenge)
- Compact `struct` (not used for Graph Challenge)
- Batching for features
- Out-of-core streaming for weights

Out-of-core Streaming



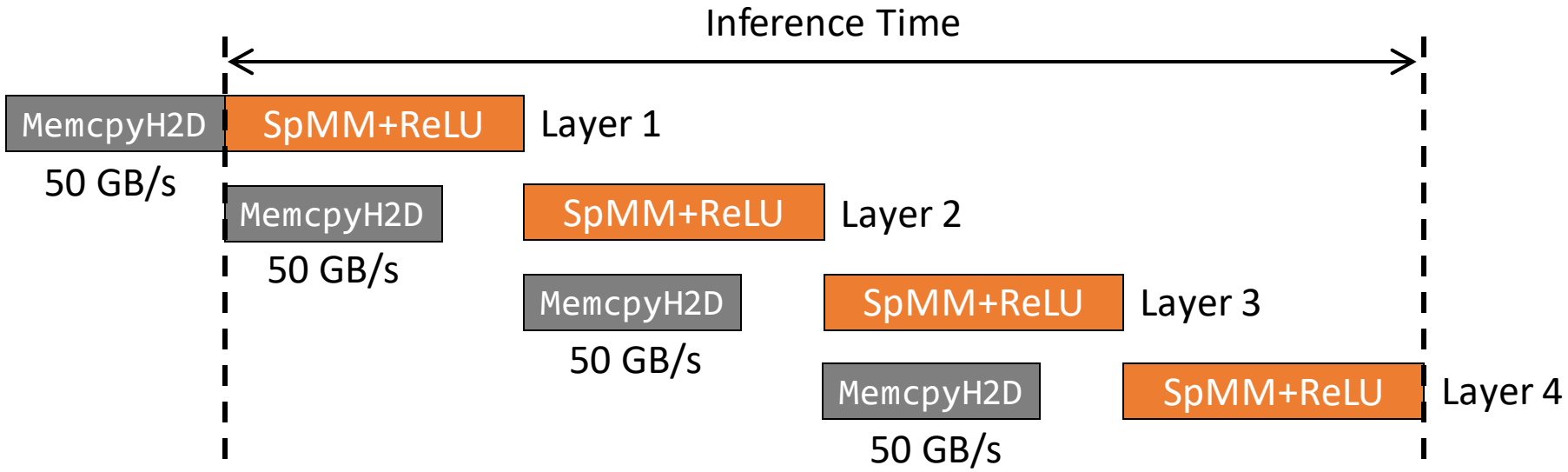
Memory Optimizations

- `short` indices
- `half` weights (not used for Graph Challenge)
- Compact `struct` (not used for Graph Challenge)
- Batching for features
- Out-of-core streaming for weights

Multi-GPU Parallelization

- Batch parallelism
- Weights are duplicated
- Features are partitioned
- Suffers from load imbalancing
- Suffers from low granularity

Out-of-core Streaming



Inference Throughput (TeraEdges/Second)

Neurons	Layers	Single V100	Single A100	Number of V100 GPUs (Six per Node)								
				3	6	12	24	48	96	192	384	768
1024	120	10.51 (0.225s)	16.74 (1.59×)	18.92	22.46	25.52	28.52	27.77	**29.17**	27.89	29.12	29.13
	480	12.87 (0.073s)	20.99 (1.63×)	21.47	24.34	26.92	28.73	28.43	**29.30**	28.80	29.10	23.06
	1920	14.30 (0.264s)	20.68 (1.45×)	22.26	24.77	27.33	28.70	28.58	28.60	28.73	**28.83**	28.83
4096	120	9.45 (0.100s)	14.27 (1.51×)	20.69	31.36	47.82	62.03	70.31	75.81	79.11	81.13	**82.20**
	480	11.74 (0.322s)	18.63 (1.59×)	28.18	40.58	56.54	67.63	73.16	77.27	80.02	79.97	**82.22**
	1920	13.88 (1.08s)	19.86 (1.43×)	30.53	44.48	62.74	72.57	73.72	76.25	79.99	80.67	**82.32**
16384	120	6.15 (0.614s)	11.60 (1.89×)	16.31	28.85	50.74	64.33	89.18	111.44	**146.88**	114.87	111.30
	480	7.45 (2.027s)	14.31 (1.92×)	19.82	32.88	50.83	71.45	95.78	112.61	138.62	138.30	**139.44**
	1920	7.84 (7.704s)	15.27 (1.95×)	20.86	33.62	57.08	77.73	104.83	120.63	146.11	146.30	**146.40**
65536	120	3.47 (4.352s)	8.15 (2.35×)	10.90	18.77	34.20	51.14	73.67	100.72	162.19	173.25	**179.58**
	480	3.83 (15.769s)	9.08 (2.37×)	12.13	20.39	37.63	56.66	75.29	108.06	166.15	**170.26**	169.30
	1920	3.93 (61.474s)	9.33 (2.37×)	12.47	20.88	38.81	58.08	77.55	112.01	167.43	170.06	**171.37**

Inference Throughput (TeraEdges/Second)

Neurons	Layers	Single V100	Single A100	Number of V100 GPUs (Six per Node)								
				3	6	12	24	48	96	192	384	768
1024	120	10.51 (0.225s)	16.74 (1.59×)	18.92	22.46	25.52	28.52	27.77	**29.17**	27.89	29.12	29.13
	480	12.87 (0.073s)	20.99 (1.63×)	21.47	24.34	26.92	28.73	28.43	**29.30**	28.80	29.10	23.06
	1920	14.30 (0.264s)	20.68 (1.45×)	22.26	24.77	27.33	28.70	28.58	28.60	28.73	**28.83**	28.83
4096	120	9.45 (0.100s)	14.27 (1.51×)	20.69	31.36	47.82	62.03	70.31	75.81	79.11	81.13	**82.20**
	480	11.74 (0.322s)	18.63 (1.59×)	28.18	40.58	56.54	67.63	73.16	77.27	80.02	79.97	**82.22**
	1920	13.88 (1.08s)	19.86 (1.43×)	30.53	44.48	62.74	72.57	73.72	76.25	79.99	80.67	**82.32**
16384	120	6.15 (0.614s)	11.60 (1.89×)	16.31	28.85	50.74	64.33	89.18	111.44	**146.88**	114.87	111.30
	480	7.45 (2.027s)	14.31 (1.92×)	19.82	32.88	50.83	71.45	95.78	112.61	138.62	138.30	**139.44**
	1920	7.84 (7.704s)	15.27 (1.95×)	20.86	33.62	57.08	77.73	104.83	120.63	146.11	146.30	**146.40**
65536	120	3.47 (4.352s)	8.15 (2.35×)	10.90	18.77	34.20	51.14	73.67	100.72	162.19	173.25	**179.58**
	480	3.83 (15.769s)	9.08 (2.37×)	12.13	20.39	37.63	56.66	75.29	108.06	166.15	**170.26**	169.30
	1920	3.93 (61.474s)	9.33 (2.37×)	12.47	20.88	38.81	58.08	77.55	112.01	167.43	170.06	**171.37**

Inference Throughput (TeraEdges/Second)

Neurons	Layers	Single V100	Single A100	Number of V100 GPUs (Six per Node)								
				3	6	12	24	48	96	192	384	768
1024	120	10.51 (0.225s)	16.74 (1.59×)	18.92	22.46	25.52	28.52	27.77	**29.17**	27.89	29.12	29.13
	480	12.87 (0.073s)	20.99 (1.63×)	21.47	24.34	26.92	28.73	28.43	**29.30**	28.80	29.10	23.06
	1920	14.30 (0.264s)	20.68 (1.45×)	22.26	24.77	27.33	28.70	28.58	28.60	28.73	**28.83**	28.83
4096	120	9.45 (0.100s)	14.27 (1.51×)	20.69	31.36	47.82	62.03	70.31	75.81	79.11	81.13	**82.20**
	480	11.74 (0.322s)	18.63 (1.59×)	28.18	40.58	56.54	67.63	73.16	77.27	80.02	79.97	**82.22**
	1920	13.88 (1.08s)	19.86 (1.43×)	30.53	44.48	62.74	72.57	73.72	76.25	79.99	80.67	**82.32**
16384	120	6.15 (0.614s)	11.60 (1.89×)	16.31	28.85	50.74	64.33	89.18	111.44	**146.88**	114.87	111.30
	480	7.45 (2.027s)	14.31 (1.92×)	19.82	32.88	50.83	71.45	95.78	112.61	138.62	138.30	**139.44**
	1920	7.84 (7.704s)	15.27 (1.95×)	20.86	33.62	57.08	77.73	104.83	120.63	146.11	146.30	**146.40**
65536	120	3.47 (4.352s)	8.15 (2.35×)	10.90	18.77	34.20	51.14	73.67	100.72	162.19	173.25	**179.58**
	480	3.83 (15.769s)	9.08 (2.37×)	12.13	20.39	37.63	56.66	75.29	108.06	166.15	**170.26**	169.30
	1920	3.93 (61.474s)	9.33 (2.37×)	12.47	20.88	38.81	58.08	77.55	112.01	167.43	170.06	**171.37**

		This Work	Bisson & Fatica [18] 2019 Champion		Davis et al. [20] 2019 Champion		Ellis & Rajamanickam [21] 2019 Innovation		Wang et al. [22] 2019 Student Innov.		Wang et al. [23] 2019 Finalist	
Neurons	**Layers**	**Throughput**	**Throughput**	**Speedup**	**Throughput**	**Speedup**	**Throughput**	**Speedup**	**Throughput**	**Speedup**	**Throughput**	**Speedup**
1024	**120**	2.917E+13	4.517E+12	6.46	1.533E+11	190.28	2.760E+11	105.69	1.407E+11	207.32	8.434E+10	345.88
	480	2.930E+13	7.703E+12	3.80	2.935E+11	99.83	2.800E+11	104.64	1.781E+11	164.51	9.643E+10	303.84
	1920	2.883E+13	8.878E+12	3.25	2.754E+11	104.68	2.800E+11	102.96	1.896E+11	152.06	9.600E+10	300.30
4096	**120**	8.220E+13	6.541E+12	12.57	1.388E+11	592.22	2.120E+11	387.74	1.943E+11	423.06	6.506E+10	1,263.52
	480	8.222E+13	1.231E+13	6.68	1.743E+11	471.72	2.160E+11	380.65	2.141E+11	384.03	6.679E+10	1,230.99
	1920	8.232E+13	1.483E+13	5.55	1.863E+11	441.87	2.160E+11	381.11	2.197E+11	374.69	6.617E+10	1,244.02
16384	**120**	1.469E+14	1.008E+13	14.57	1.048E+11	1,401.53	1.270E+11	1,156.54	1.966E+11	747.10	3.797E+10	3,867.84
	480	1.394E+14	1.500E+13	9.29	1.156E+11	1,206.23	1.280E+11	1,089.38	2.060E+11	676.89	3.747E+10	3,721.66
	1920	1.464E+14	1.670E+13	8.77	1.203E+11	1,216.96	1.310E+11	1,117.56	1.964E+11	745.52	3.750E+10	3,903.72
65536	**120**	1.796E+14	9.388E+12	19.13	1.050E+11	1710.29	9.110E+10	1971.24	1.892E11	949.15	–	–
	480	1.703E+14	1.638E+13	10.40	1.091E+11	1,560.59	8.580E+10	1,984.38	1.799E+11	946.41	–	–
	1920	1.714E+14	1.787E+13	9.59	1.127E+11	1,520.59	8.430E+10	2,032.86	–	–	–	–

Thank You

Q&A

https://github.com/merthidayetoglu/SpDNN_Challenge2020