Efficient GPU Kernels for N:M-Sparse Weights in Deep Learning

Bin Lin[12], Ningxin Zheng[1], Lei Wang[1], **Shijie Cao[1]**, Lingxiao Ma[1],

Quanlu Zhang[1], Yi Zhu[1], Ting Cao[1], Jilong Xue[1], Yuqing Yang[1], Fan Yang[1]

[1]Microsoft Research Asia, [2]Tsinghua University

Sparsity in DNN Weights

- Sparsity(redundancy) occurs in Deep Neural Networks
  - Weight pruning aims to find and remove redundant weights
- With the support of sparsity:
  - Storage: reduce model size
  - Computation: skip zero computation

0.2	0.1	0.2	-0.6
0.4	-0.3	0.4	0.1
0.7	-0.1	-0.3	0.1
-0.1	0.6	-0.5	0.3

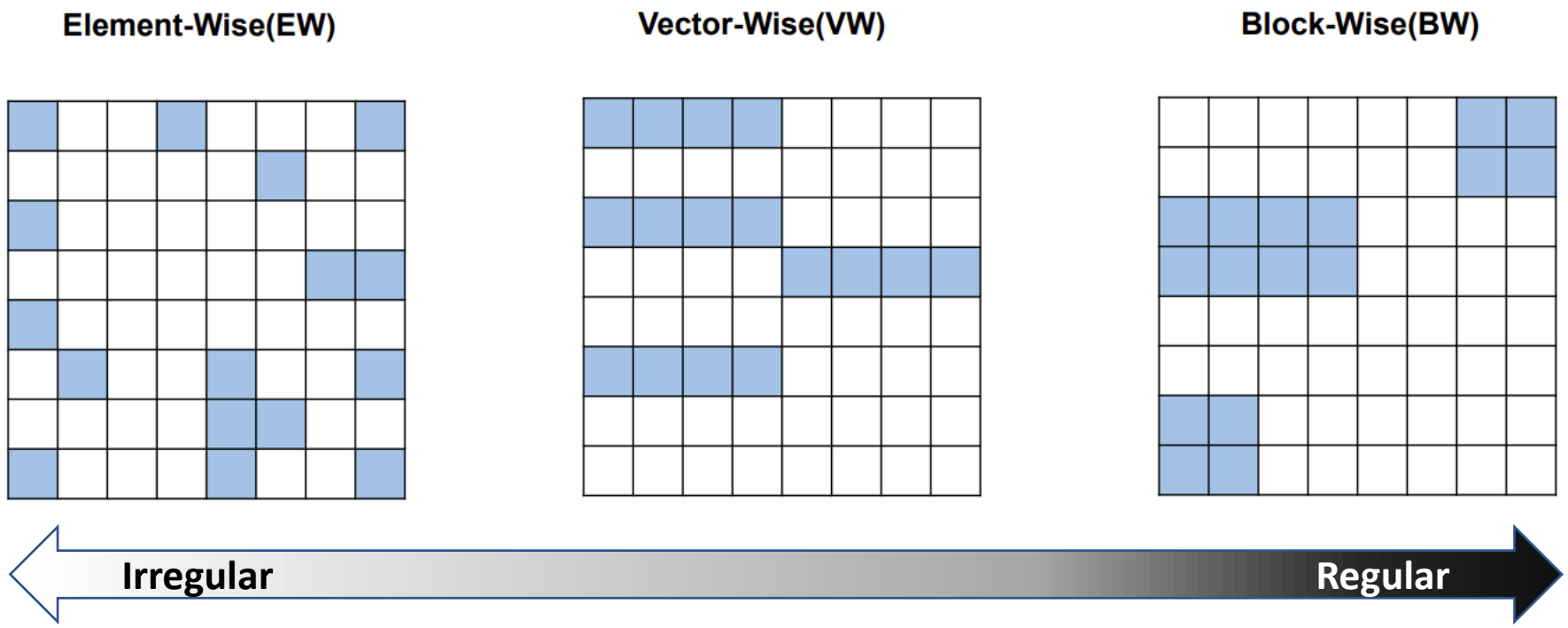
Dense Matrix

0	**0**	**0**	-0.6
0.4	**0**	0.4	**0**
0.7	**0**	**0**	**0**
0	0.6	-0.5	**0**

Pruned Sparse Matrix

Sparsity Patterns

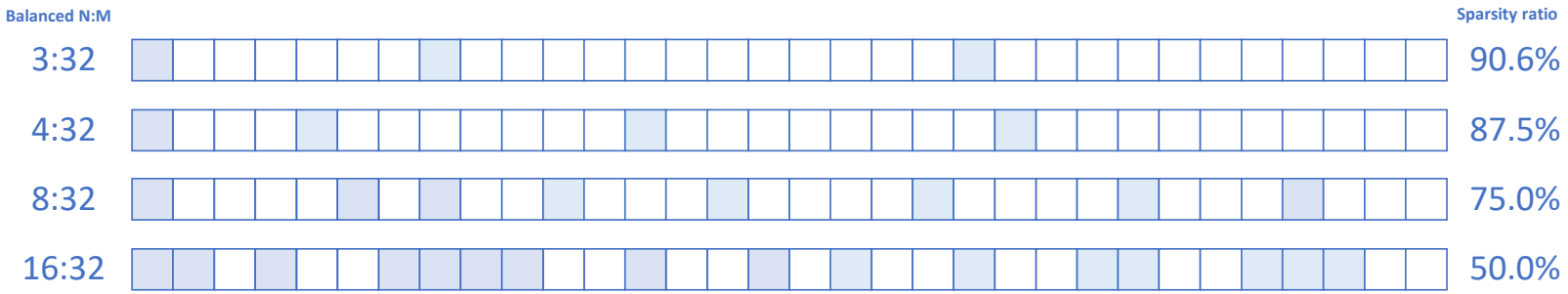
- The trade-off between model accuracy and inference efficiency



Such an accuracy-efficiency trade-off remains a longstanding challenge for sparse DNNs

N:M Sparsity

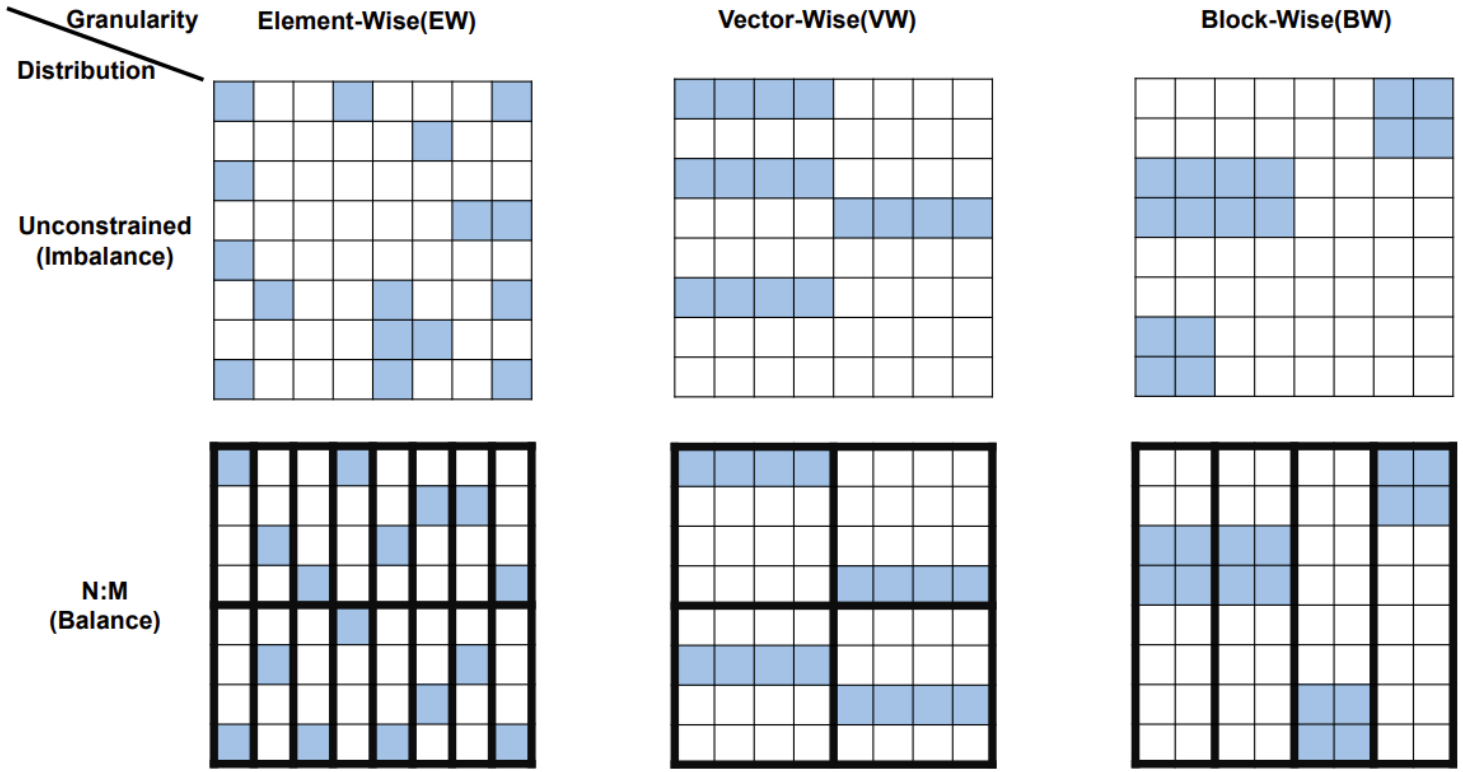
- N:M sparsity emerges as a promising alternative to achieve both high model accuracy and high inference efficiency
- What is N:M sparsity?
  - N no-zero elements in every M elements (**balanced** distribution constraint)



  - Sparse Tensor Core (2:4 sparsity) in NVIDIA Ampere Architecture: A special case of general N:M sparsity

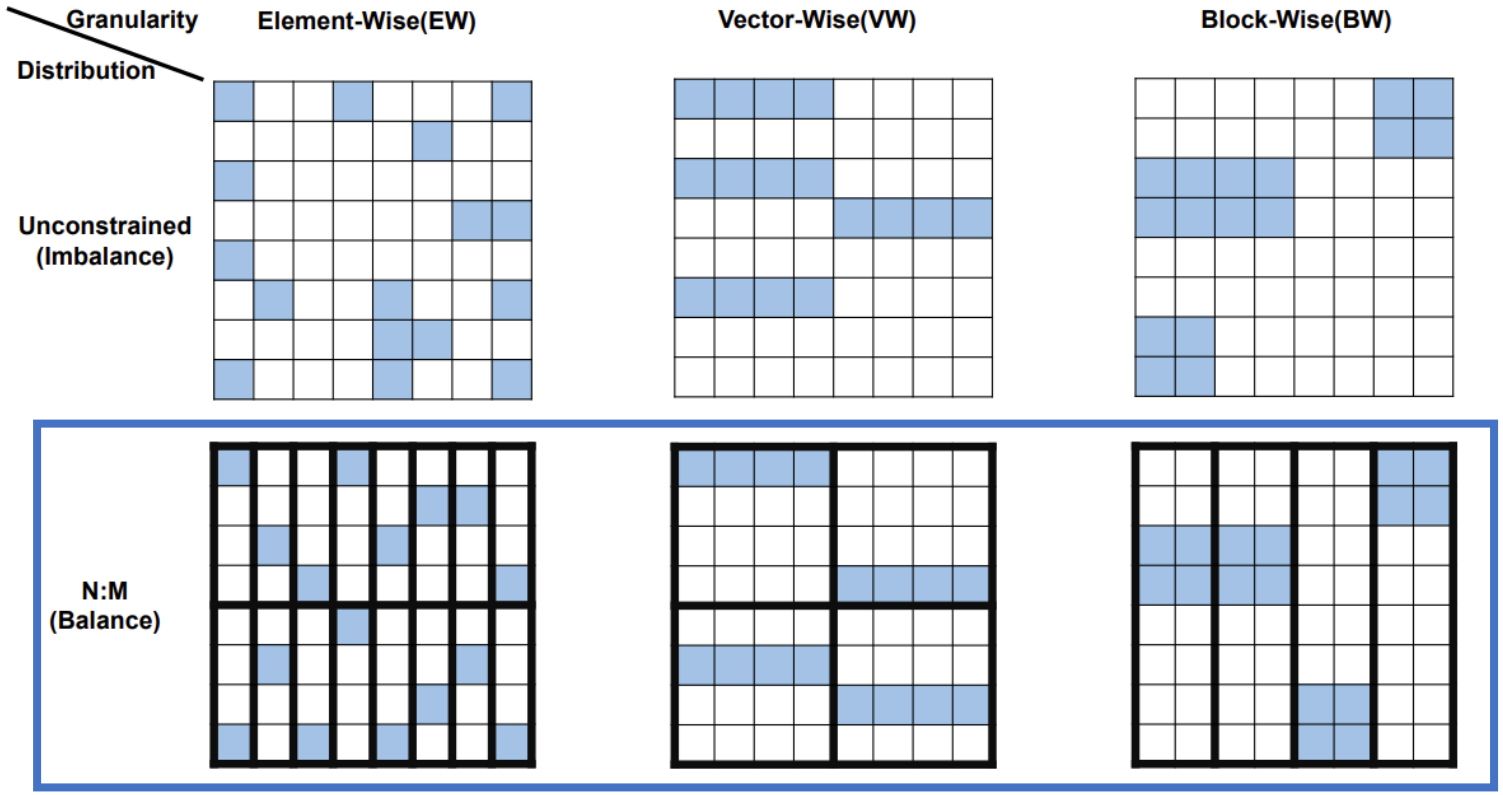
Extend N:M Sparsity to VW/BW Sparsity

- Unified representation of sparsity patterns with **granularity** and **distribution**.



Extend N:M Sparsity to VW/BW Sparsity

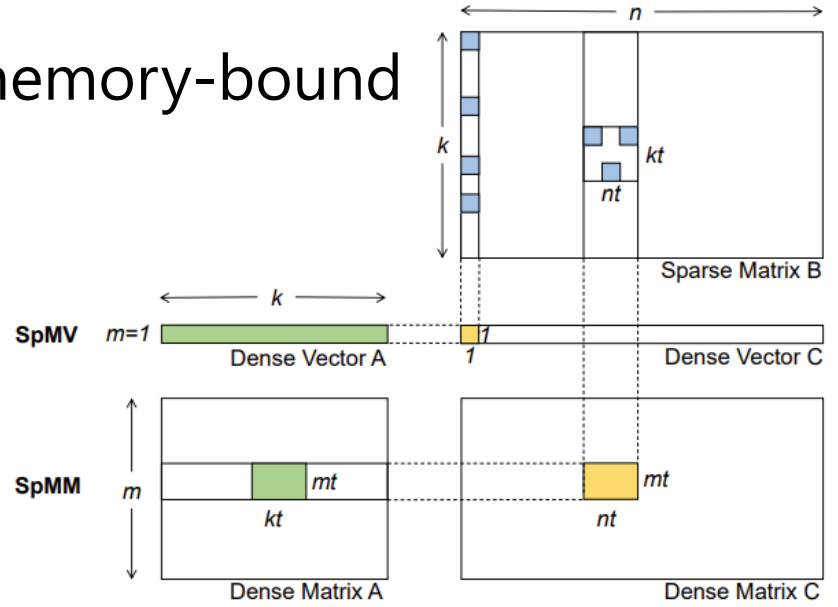
- Unified representation of sparsity patterns with **granularity** and **distribution**.



There is a lack of GPU kernels dedicated to general N:M sparsity with various sparsity ratios

nmSPARSE

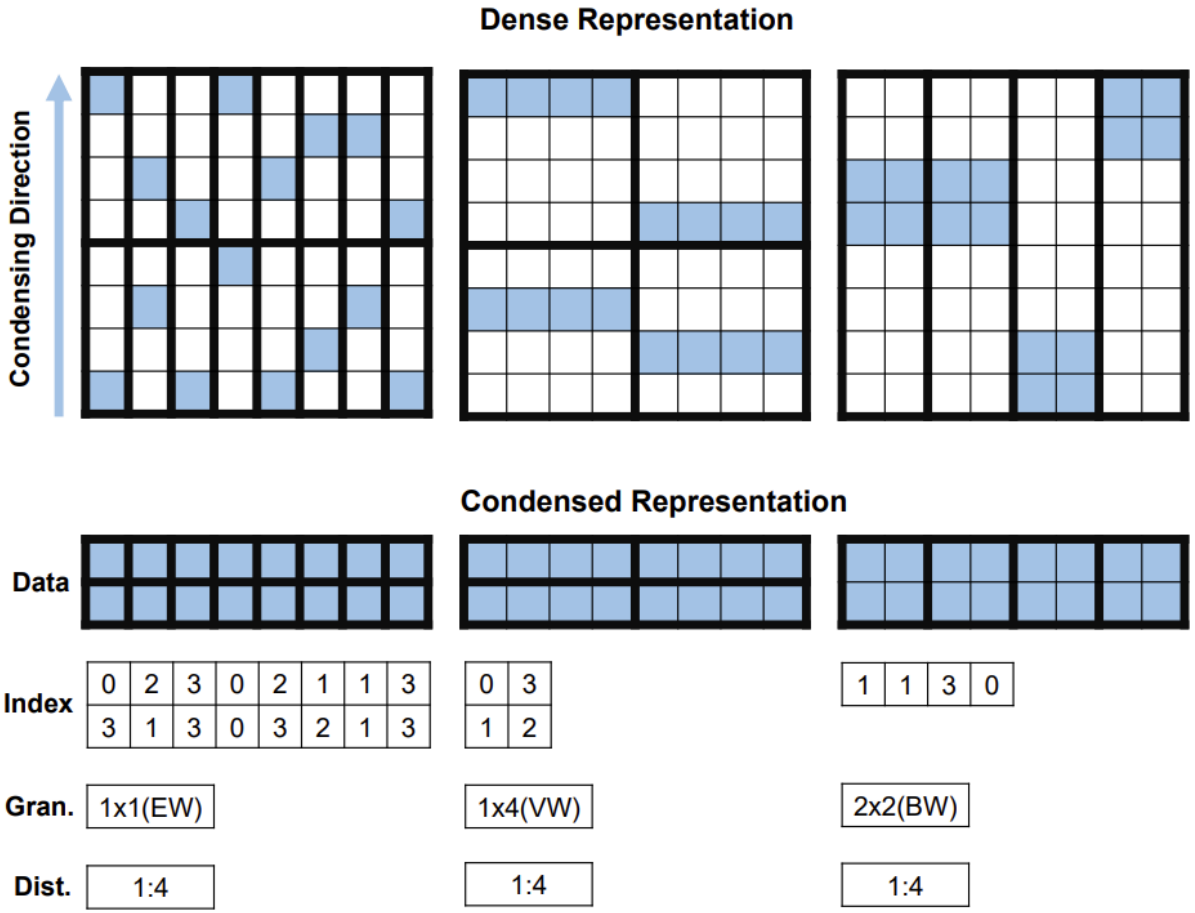
- We present *nmSPARSE*, a GPU library of **SpMV** and **SpMM** kernels for general N:M sparsity with various sparsity ratios.
- *nmSPARSE* rearranges **irregular computation and scattered memory access** into hardware-aligned **regular computation and conflict-free memory access** by leveraging the intrinsic balanced distribution of N:M sparsity.
- Highlight the importance of SpMV because MV is memory-bound and crucial for autoregressive generative models.



nmSPARSE Kernel Design

- Condensed representation of N:M sparsity
  - Reduce the memory footprint and decoding overhead
- Element-Wise N:M sparsity
  - Leverage balanced distribution to eliminate bank conflicts
- Vector-Wise/Block-Wise N:M Sparsity
  - Leverage both balanced distribution and larger granularity to offer superior performance with aligned memory access and Tensor Core support

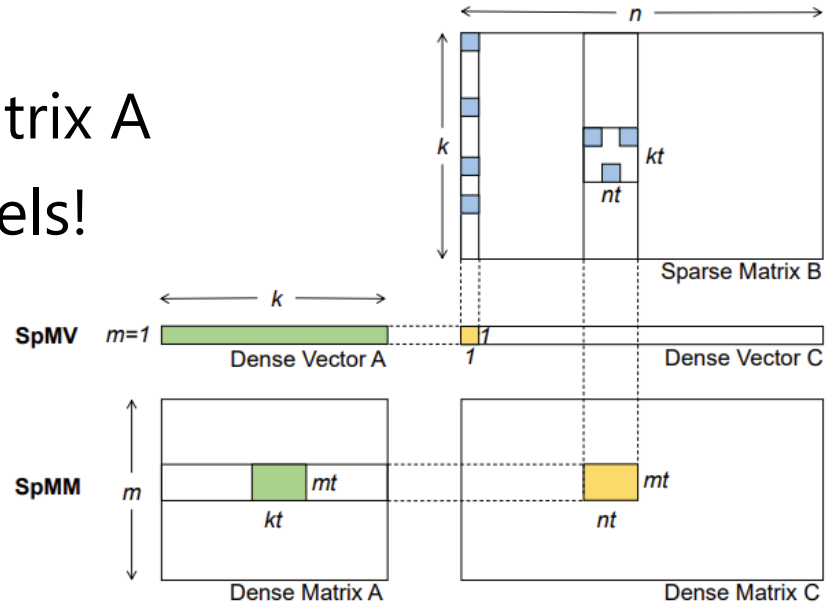
Condensed representation of N:M sparsity



- Efficient non-zero data loading
- Decoding-friendly index

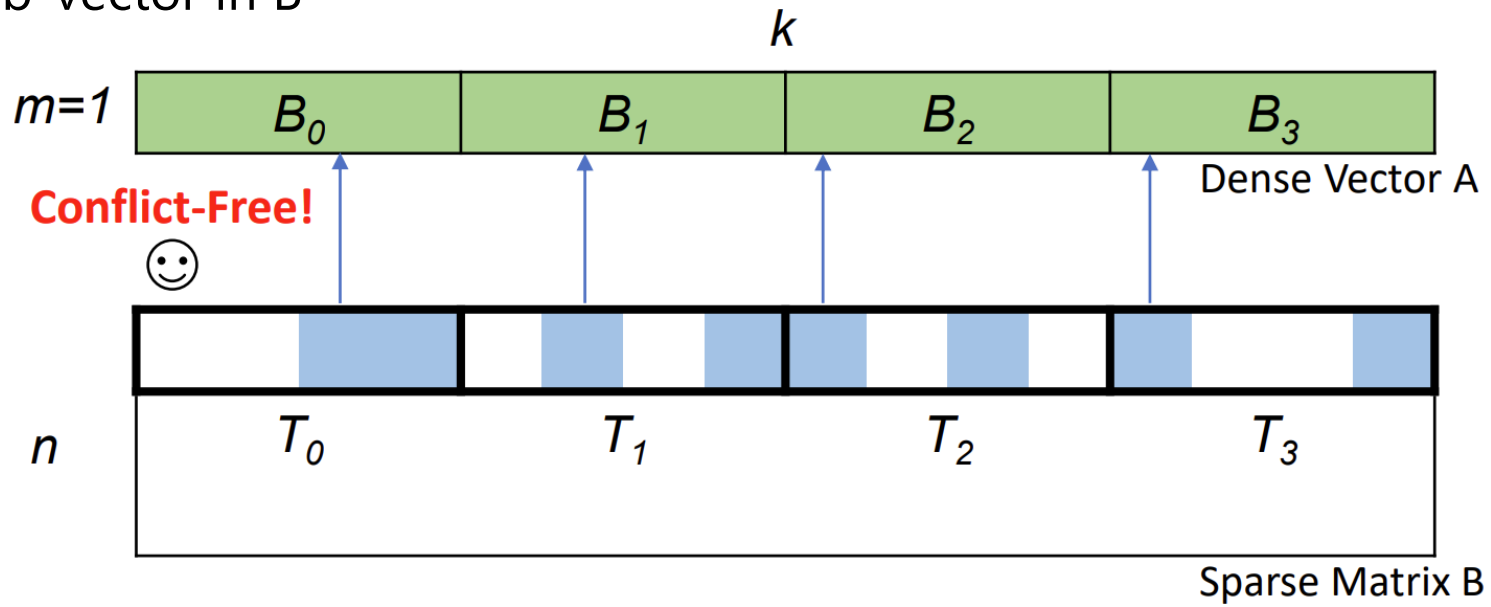
Element-Wise N:M sparsity

- Challenges of SpMV/SpMM on GPU:
  - Decoding overheads of sparse matrix B
  - Irregular and scattered memory accesses to dense vector/matrix A
- New opportunities of N:M sparsity:
  - Intrinsic workload balance
  - Fast decoding and loading of sparse matrix B
  - Locality of memory accesses to dense vector/matrix A
- Leverage N:M to design efficient SpMV/SpMM kernels!



SpMV for Element-Wise N:M sparsity

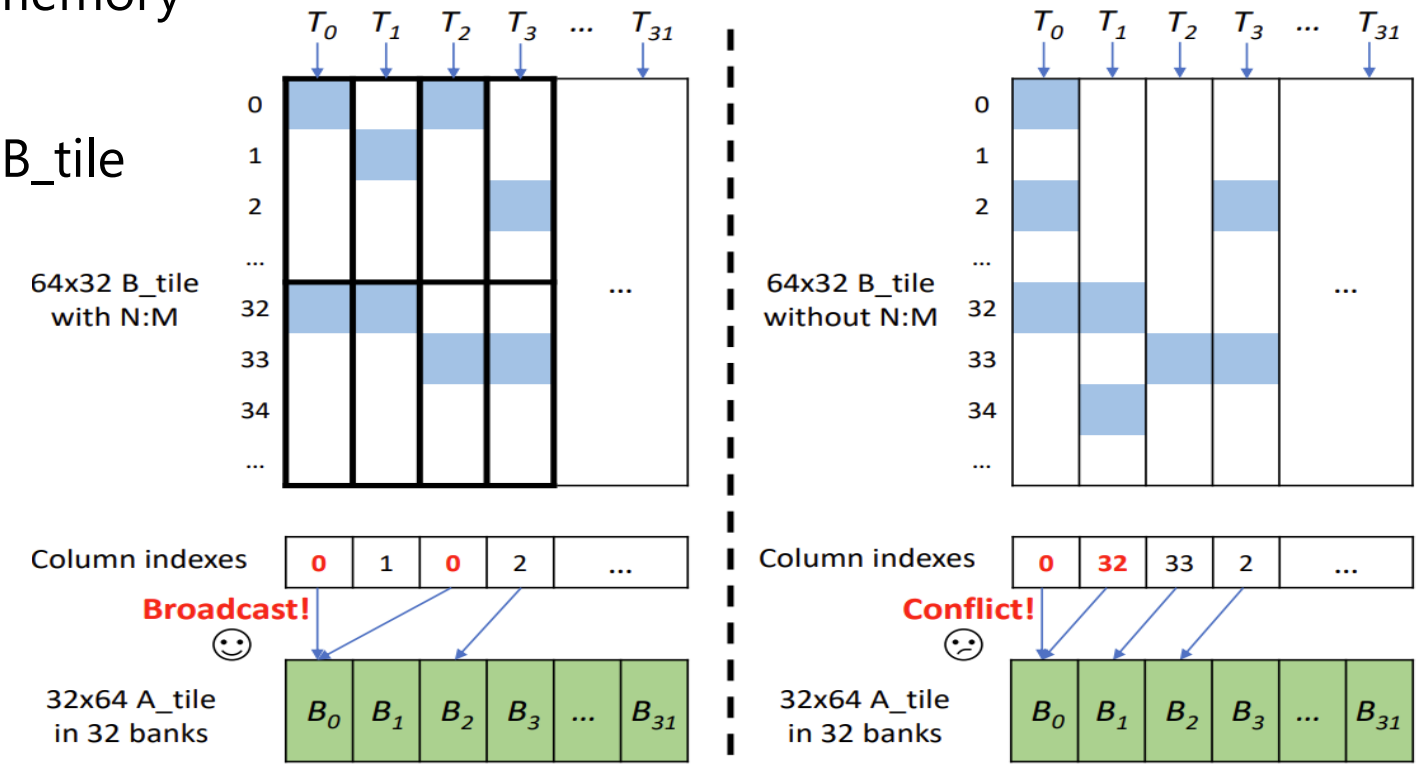
- Data organization
  - Vector A is partitioned to sub-vectors of size M and stored in distinct memory banks
- Thread mapping
  - Each thread for a sub-vector in B



Conflict-free access to shared memory

SpMM for Element-Wise N:M sparsity

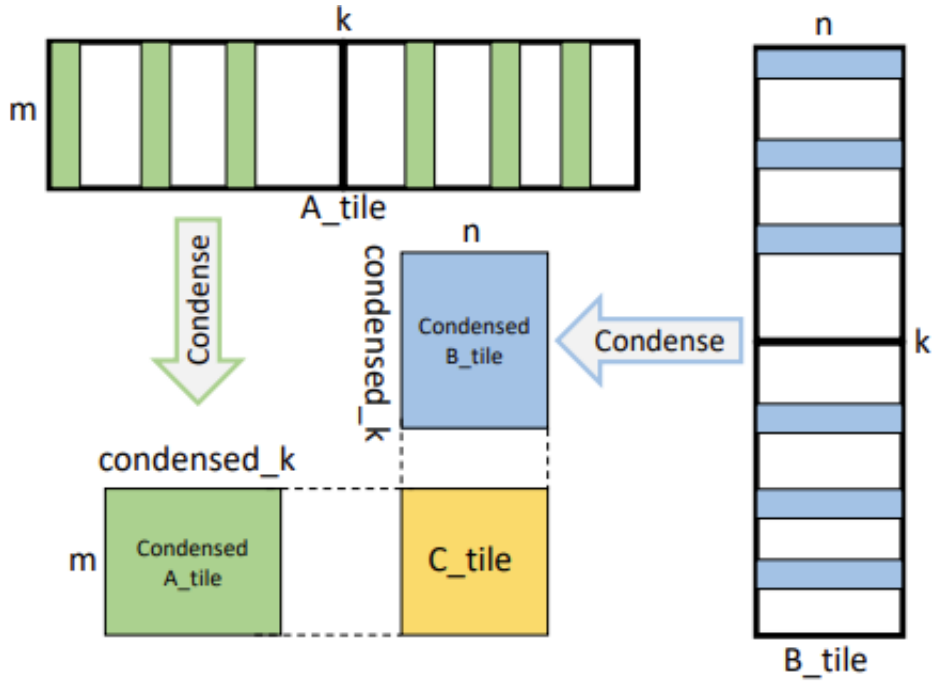
- Data organization
  - Row-major A_tile in shared memory
- Thread mapping
  - Each thread for a column in B_tile



Conflict-free broadcast access to shared memory

Vector-Wise/Block-Wise N:M Sparsity

- Leverage both
  - Balanced distribution of N:M Sparsity
  - Large granularity of VW/BW Sparsity



Aligned memory accesses and Tensor Core support

Implementation

- Pruning algorithm: We extend NVIDIA ASP to support:
  - General N:M settings.(original ASP only supports 2:4
  - Vector-Wise and Block-Wise sparsity pruning

- GPU Kernels:
  - CUDA kernels for different granularities respectively
  - Leverage Tensor Core for granularities larger than 64 or 64x64

- End-to-end model Inference:
  - Integrate nmSPARSE to SparTA.

Evaluation

- Operator benchmarks
  - Baselines: cuBLAS, cuBLASLt, cuSPARSE, cuSPARSELt, Sputnik
- E2E application study on Transformer
  - Baselines: Rammer, TensorRT, SparTA
- Platform:
  - NVIDIA Tesla A100-PCIE-80GB GPU

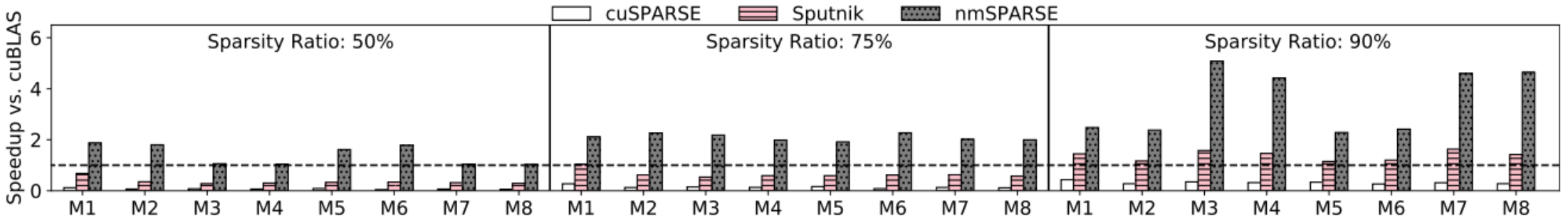
Operator Settings

- Shapes: Synthetic shapes + Bert shapes + OPT shapes

	M1	M2	M3	M4	M5	M6	M7	M8
m	1	1	1	1	1	1	1	1
k	1024	2048	4096	8192	1024	4096	5120	20480
n	1024	2048	4096	8192	4096	1024	20480	5120
	M9	M10	M11	M12	M13	M14	M15	M16
m	256	1024	4096	256	1024	4096	256	1024
k	1024	1024	1024	2048	2048	2048	4096	4096
n	1024	1024	1024	2048	2048	2048	4096	4096
	M17	M18	M19	M20	M21	M22	M23	M24
m	4096	256	1024	4096	256	1024	4096	256
k	4096	8192	8192	8192	1024	1024	1024	4096
n	4096	8192	8192	8192	4096	4096	4096	1024
	M25	M26	M27	M28	M29	M30	M31	M32
m	1024	4096	256	1024	4096	256	1024	4096
k	4096	4096	5120	5120	5120	20480	20480	20480
n	1024	1024	20480	20480	20480	5120	5120	5120

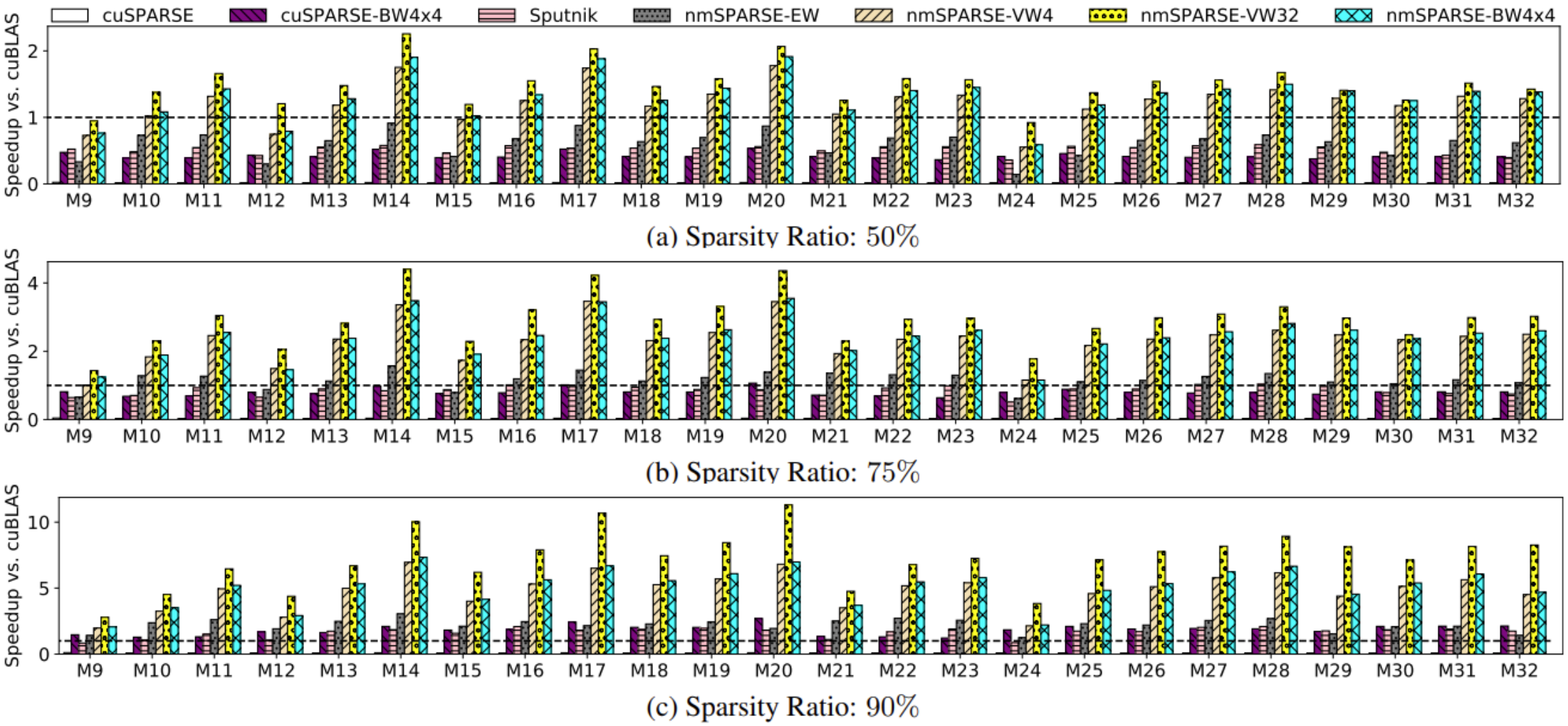
- Sparsity ratios: 50%, 75%, 90%

Operator Benchmarks with CUDA Core



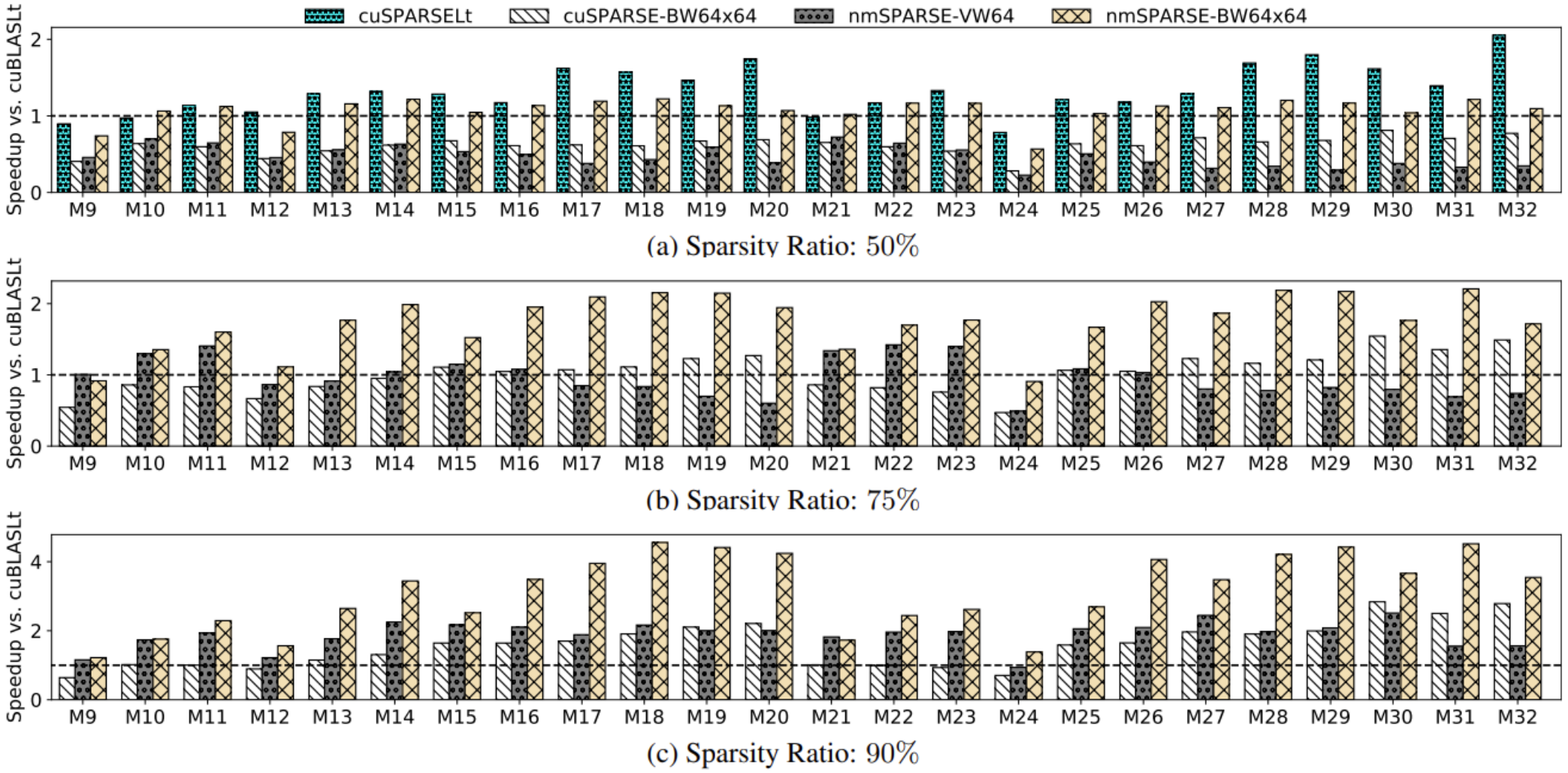
SpMV on CUDA Cores: up to 5.2x speedup

Operator Benchmarks with CUDA Core



SpMM on CUDA Cores: up to 6.0x speedup

Operator Benchmarks with Tensor Core



SpMM on Tensor Cores: up to 2.8x speedup

Application Study on Transformer

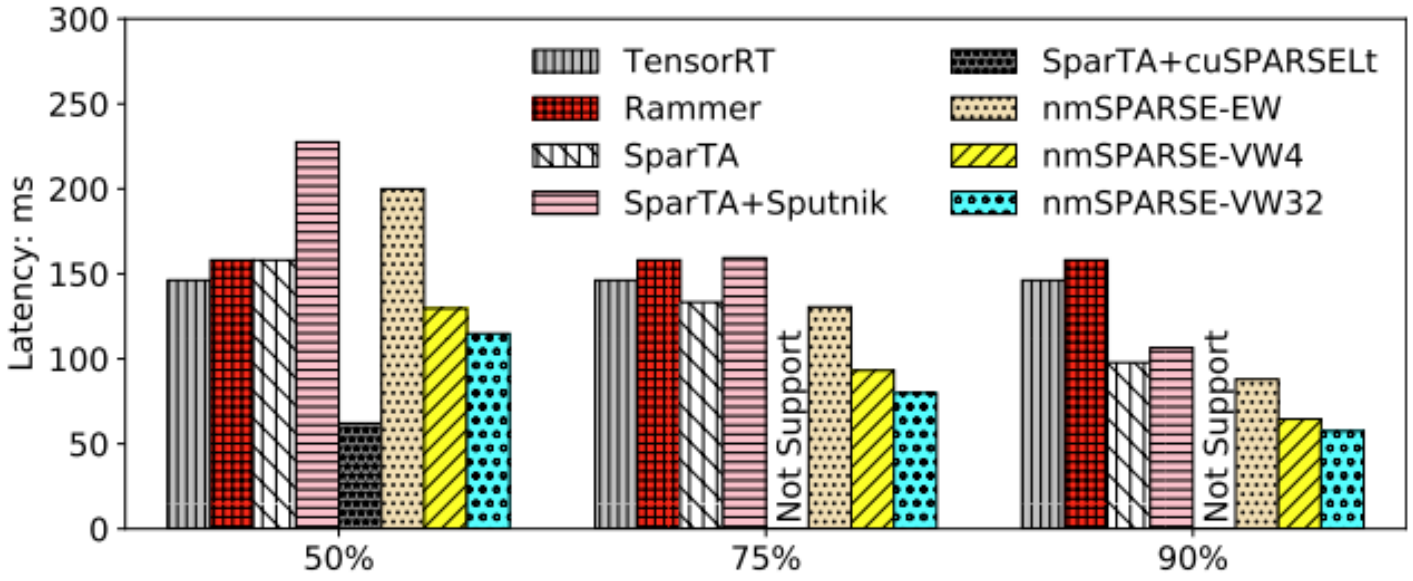
- Experimental Setup:
  - Model: bert-large
  - Dataset: SQuAD-1.1
- Pruning effectiveness of general N:M sparsity

N:M	**2:4**	**4:8**	**8:16**	**16:32**
F1	90.59	90.81	90.76	90.95
N:M	**1:4**	**2:8**	**4:16**	**8:32**
F1	88.80	89.57	89.79	90.09

Sparsity Ratio	50%	75%	90%
EW	90.72	90.10	86.23
EW-N:M	90.95	90.09	82.02
VW4	89.26	83.61	79.88
VW4-N:M	90.432	87.26	79.91
VW32	88.57	80.28	79.53
VW32-N:M	89.52	81.48	79.66
VW64	88.56	80.07	79.49
VW64-N:M	89.09	81.22	78.18
BW4x4	87.77	79.91	79.76
BW4x4-N:M	89.58	81.80	79.72
BW64x64	87.46	80.06	79.85
BW64x64-N:M	87.75	79.82	79.96

Application Study on Transformer

- End-to-end speedup:
  - nmSPARSE-EW outperforms dense baselines from 75% sparsity ratio
  - nmSPARSE-VW4 outperforms dense baselines from 50% sparsity ratio
  - cuSPARSELt with Sparse Tensor Core performs the best at 50% sparsity ratio



Summary

- We presents *nmSPARSE*, a GPU library of **SpMV** and **SpMM** kernels for general N:M sparsity with various sparsity ratios.
- We hope *nmSPARSE* can benefit efficient sparse model inference and motivate new innovations on N:M sparsity in both machine learning and system communities.

- Artifact: https://github.com/microsoft/SparTA/tree/nmsparse_artifact
- Code: https://github.com/microsoft/SparTA/tree/nmsparse