Explaining the Performance of Supervised and Semi-Supervised Methods for Automated Sparse Matrix Format Selection

Sunidhi Dhandhania, Akshay Deodhar, Konstantin Pogorelov, **Swarnendu Biswas**, and Johannes Langguth

International Workshop on Deployment and Use of Accelerators

ICPP Workshops 2021

Sparse Matrix Vector Multiplication (SpMV)

- Important computation kernel
  - PageRank, Conjugate Gradient, and Indirect solvers for systems of linear equations

$$\underset{A}{\begin{bmatrix} 1.0 & - & 2.0 & - & 3.0 \\ - & 4.0 & - & 5.0 & - \\ - & - & 6.0 & - & - \\ 7.0 & - & - & 8.0 & - \\ - & 9.0 & - & - & - \end{bmatrix}} \times \underset{X}{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}} = \underset{Y}{\begin{bmatrix} 22 \\ \dots \\ \\ \\ \end{bmatrix}}$$

J Greathouse and M Daga. Efficient Sparse Matrix-Vector Multiplication on GPUs using the CSR Storage Format. AMD Research 2014.

Performance of SpMV

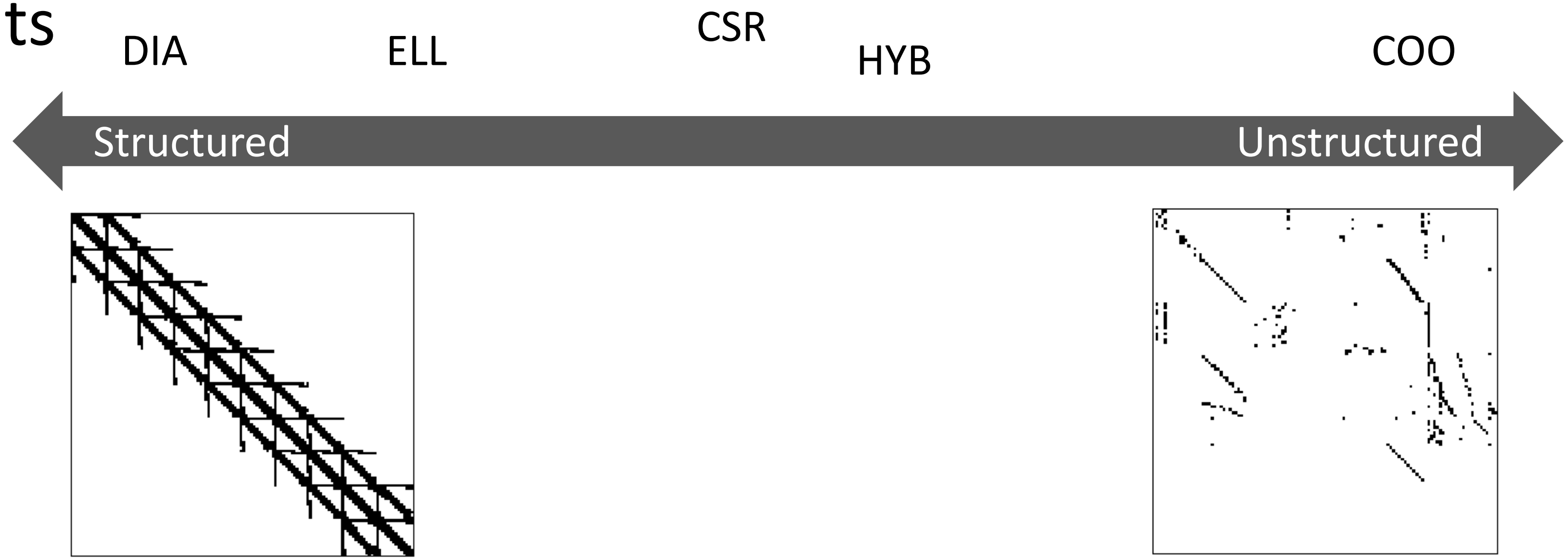
- Memory accesses are unstructured and have irregular access patterns unlike dense matrix operations
- Limited data reuse, FLOPS/byte is very low → memory-bound

Performance is sensitive to

- Sparsity pattern of the input matrix
- Processor microarchitecture and memory hierarchy
- Kernel implementation
- Other aspects like the Compiler and OS

Sparse Matrix Formats

- Representing matrices in a sparse format result in significant memory savings
- MKL from Intel and CUSP and cuSparse from NVIDIA support many popular formats

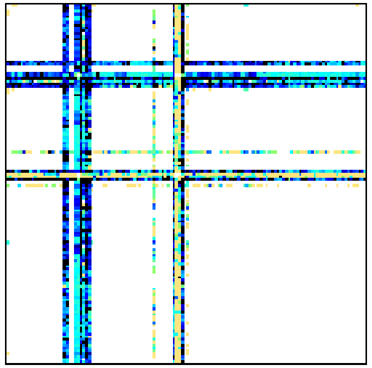


N. Bell and M. Garland. Sparse Matrix-Vector Multiplication on Throughput-Oriented Processors. NVIDIA Research.

Sensitivity to Sparse Formats

	Slowdown (relative)			Optimal Format		
	Pascal	**Turing**	**Volta**	**Pascal**	**Turing**	**Volta**
mawi_201512012345	164.8	194.8	121.3	HYB	HYB	HYB
lp_osa_60	5.2	8.8	7.5	HYB	COO	COO

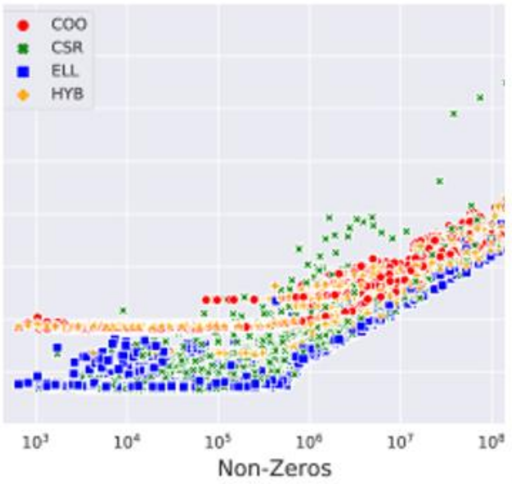
- CSR format is used as the default
- CUSP library was used for benchmarking



mawi_201512012345	
Rows	18571154
Column	18571154
Nonzeros	38040320
Mean nonzeros per row	2.04
Max nonzeros in a row	16399896
Average std dev of nonzeros across rows	3805.8

Sensitivity to Sparse Formats

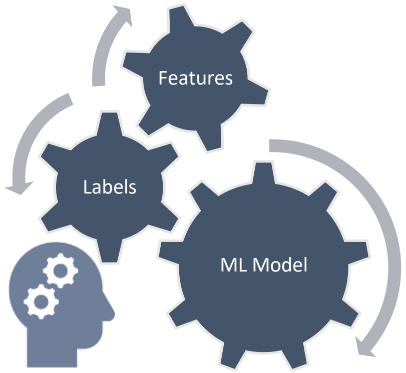
	Slowdown (relative)	Optimal Format
mawi		
lp_		

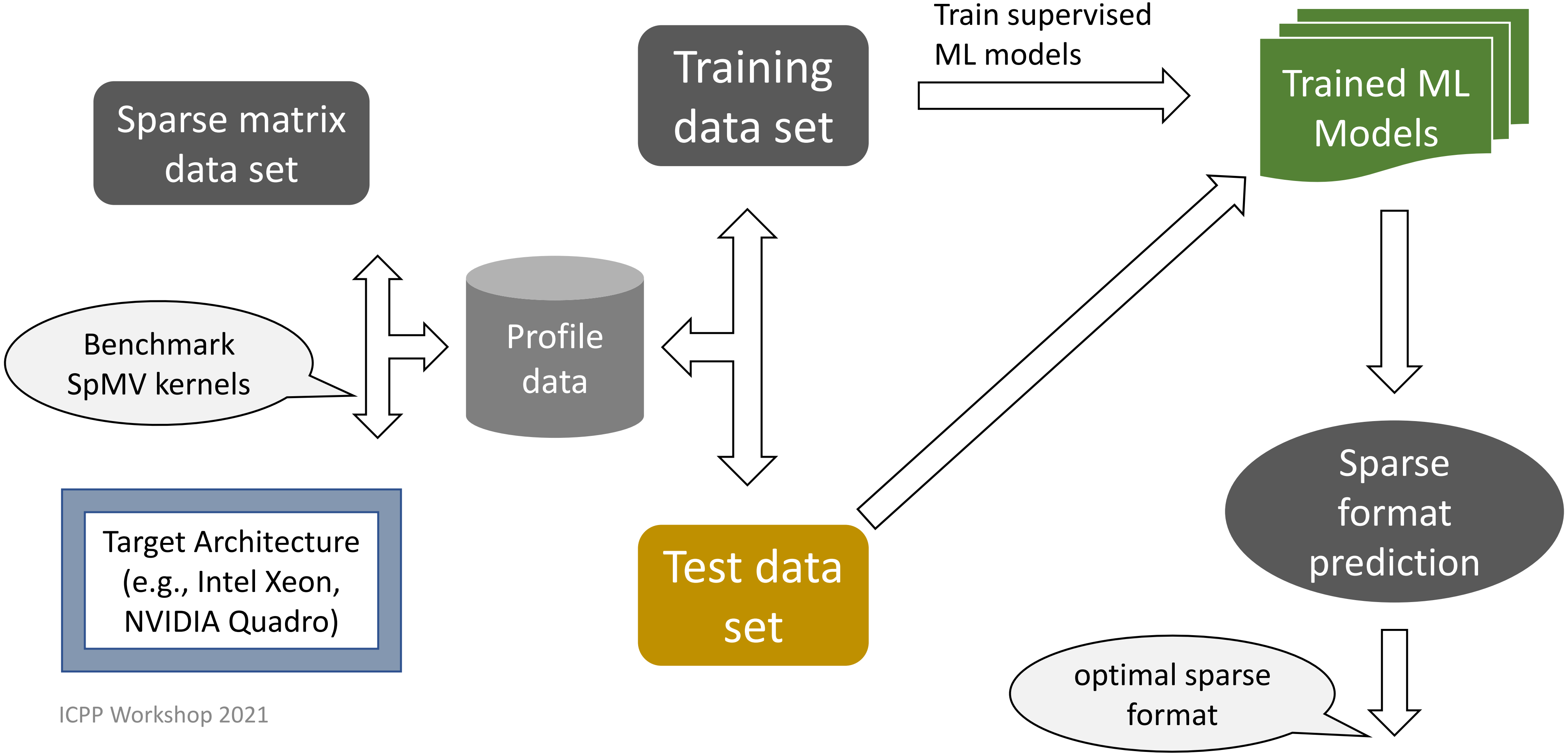


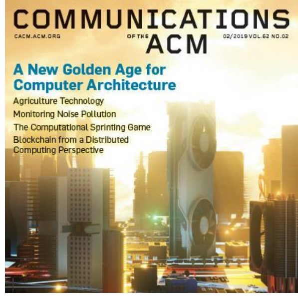
No single format is the best across all input matrices and all target architectures

Average std dev of nonzeros across rows	3805.8

Supervised ML Techniques







Challenges with Supervised Methods

Need representative training data and accurate feature set

- Compare the size of ImageNet and SuiteSparse!

Numerical computations are being parallelized across heterogeneous compute devices

- Trained models are specific to the profiled architecture, need to retrain for all possible target architectures

+

GeForce GTX 1080
- 90.65% accuracy
- 1.07X speedup

Volta V100
- 71% accuracy
- 0.97X speedup

Challenges with Supervised Methods

May need to retrain if new sparsity patterns are found or new sparse formats are proposed

- Several new sparsity formats have been recently proposed (e.g., CVR, CSR5, CSR2, and PELLR)

Training supervised ML models require benchmarking M matrices $\times$ F formats $\times$ N trials, which will often run into days

- Overhead comes from reading matrix files and format conversion

Desired Requirements for Automated Sparse Format Selection

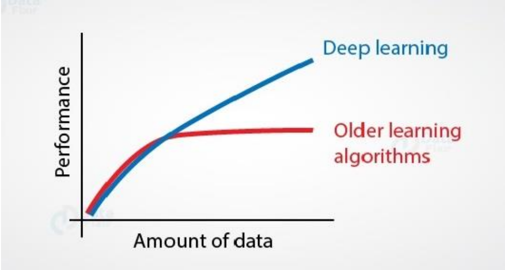
Solution should not be tightly coupled to the target architecture

- Model should be easily portable to different target hardware

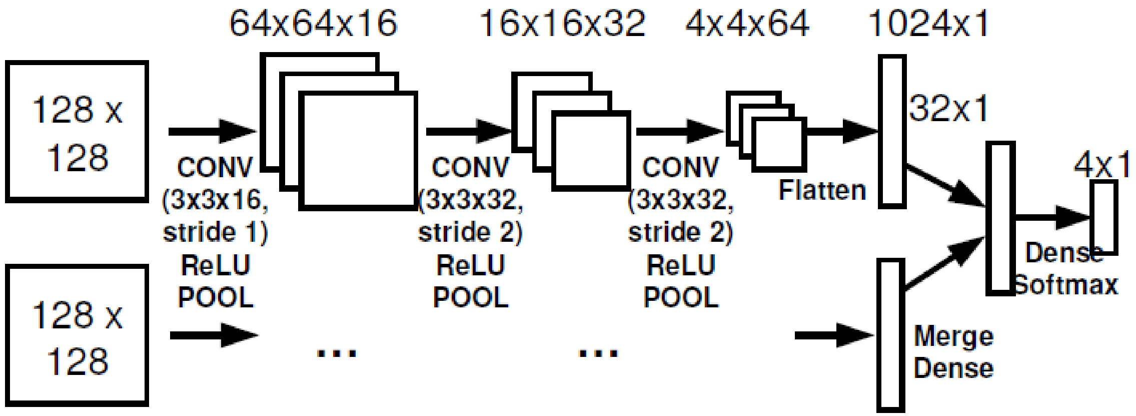
Approach should be flexible to incorporate new data

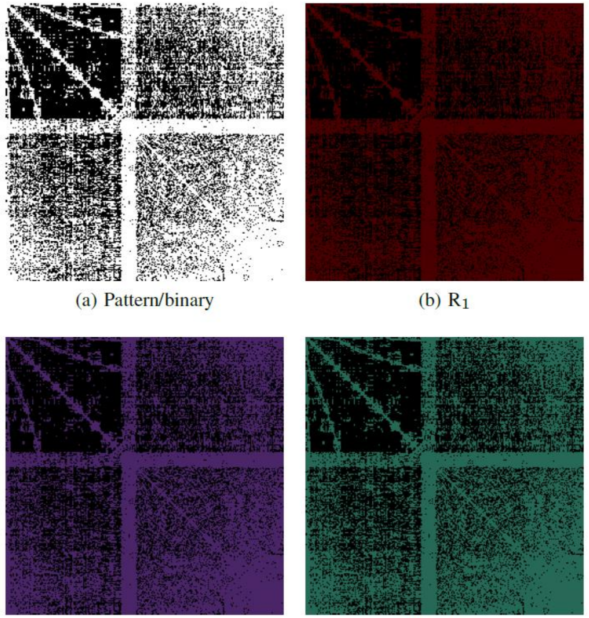
Techniques should aim for the “train once, deploy multiple times” paradigm

DL Techniques for Automated Format Selection



- CNNs have had great success in image classification and computer vision
- Why not use CNNs for classifying matrices?



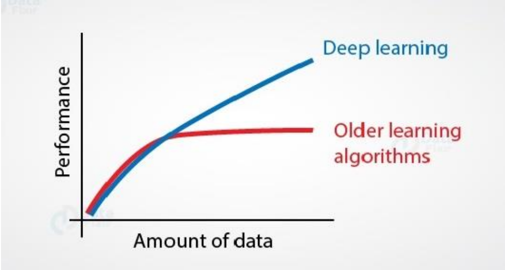


(a) Pattern/binary (b) R_1

(c) $R_1G_2B_3$ (d) $R_2G_3B_4$

Y. Zhao et al. Bridging the Gap between Deep Learning and Sparse Matrix Format Selection. PPoPP 2018.
J. Pichel and B. Pateiro-Lopez. A New Approach for Sparse Matrix Classification Based on Deep Learning Techniques. CLUSTER 2018.

Do the DL Techniques Address the Challenges?



- DL models require even larger datasets to have good accuracy
- Training and inference is very costly compared to non-DL models

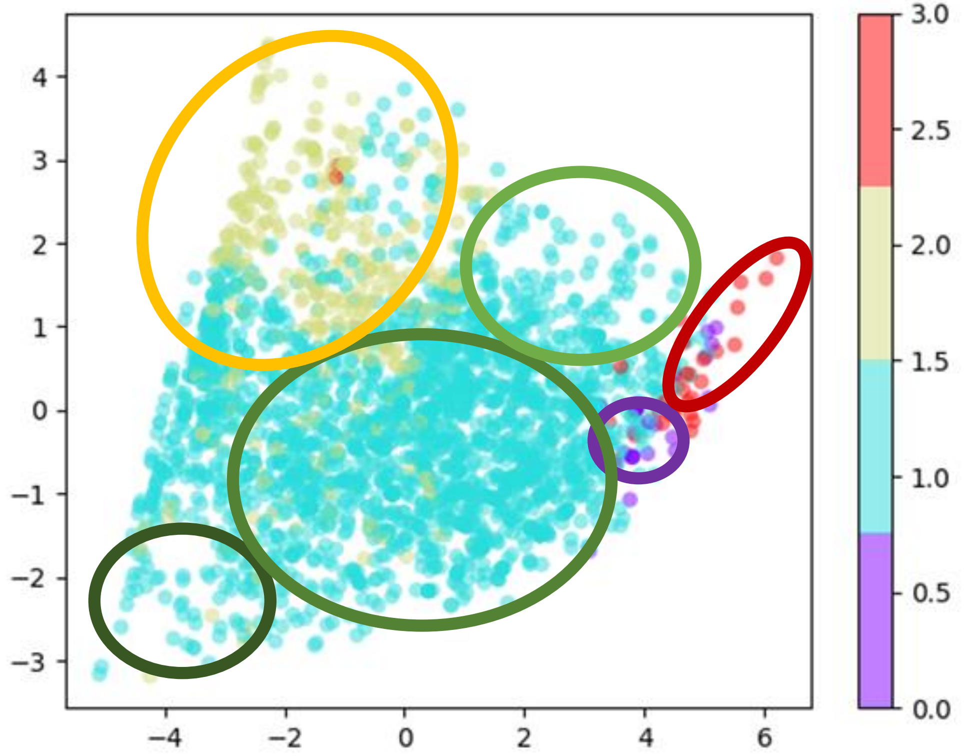
W. Zhou et al. Enabling Runtime SpMV Format Selection through an Overhead Conscious Method. TPDS 2020.

Our Proposal

Semi-Supervised Method for Automated Sparse Matrix Format Selection

Semi-Supervised Format Selection via Clustering

- Create clusters to identify matrices with similar execution characteristics
- Benchmark a **few** matrices from each cluster to assign a label



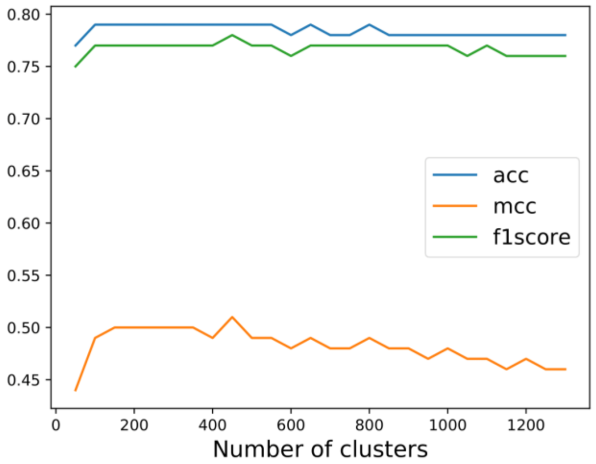
- Quality of cluster C

$$quality(C) = \frac{max_f \; count(C, f)}{|c|}$$

$count(C, f)$ gives the number of matrices in cluster C having format f

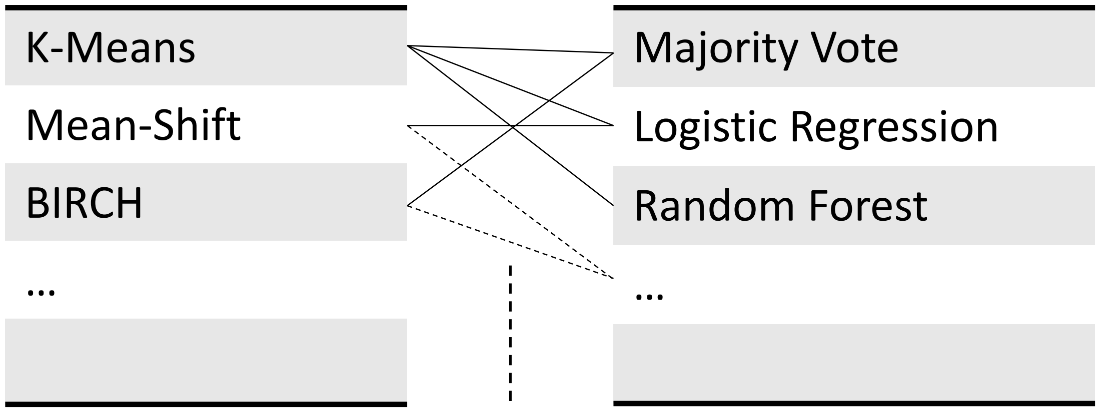
Devising an Accurate Clustering and Labeling Scheme

- Naïve application of K-Means clustering gives poor results
- Our pipeline
  - Apply transformations (log or square root) to the feature set
  - Apply Min-max scaling to scale each feature to [0,1]
  - Use PCA to decompose the features to a vector of size 8
- How to find K?
  - More small clusters will increase accuracy
  - Few large clusters reduces training time and limits overfitting, but can be more inaccurate



Dissecting Clustering-based Format Selection

- Clusters will be invariant across platforms (ideal)
- Assignment of labels to clusters is platform-specific
- Benefits
  - Easy to port the model to a different architecture
  - Easy to include new sparse formats

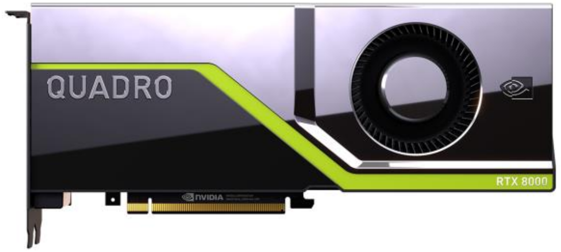


Implementation and Platforms

- Implemented sparse format selection techniques with scikit-learn and TensorFlow libraries
- Used CUDA Toolkit 9.2 and CUSP library from NVIDIA

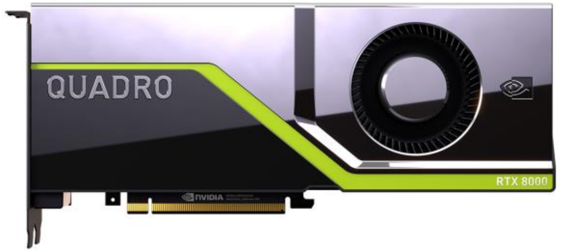
	Pascal	Volta	Turing
Model	GTX 1080	V100 SXM3	RTX 8000
# SMs	20	80	72
Memory (GB)	8 (GDDR5)	32 (HBM2)	48 (GDDR6)
Memory bandwidth	320 GB/s	897 GB/s	672 GB/s

Performance of Semi-Supervised Approaches in Local Setting



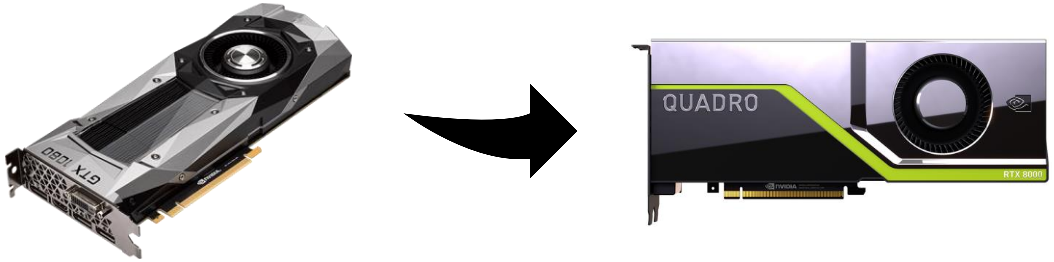
	# Clusters	MCC	ACC (%)	F1
K-Means + VOTE	300	0.629	**88.2**	**0.877**
K-Means + LR	150	0.537	86.0	0.845
K-Means + RF	200	**0.631**	87.5	0.873
Mean-Shift + VOTE	30	0.137	79.2	0.710
Mean-Shift + LR	30	0.111	79.0	0.705
Mean-Shift + RF	30	0.145	79.3	0.713
BIRCH + VOTE	150	0.622	88.1	0.874
BIRCH + LR	100	0.354	82.2	0.777
BIRCH + RF	200	0.628	87.9	0.874

Performance of Supervised Approaches in Local Setting



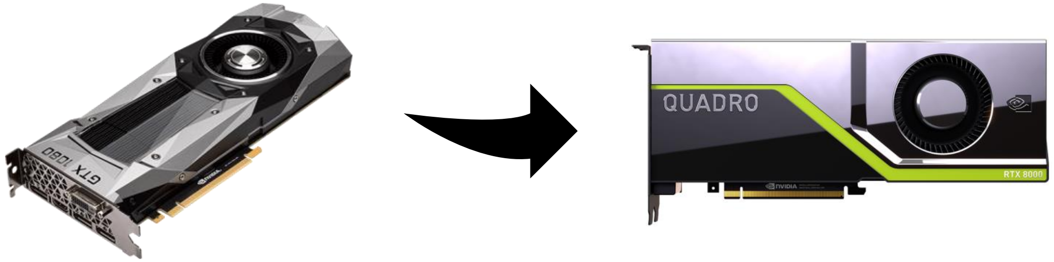
	MCC	ACC (%)	F1	GT	CSR	# Slowdown $\geq$ 1.5X
DT	0.83	94.36	0.94	0.99	1.05	17
RF	0.85	95.04	0.95	1	1.05	**11**
SVM	0.81	93.85	0.94	0.99	1.04	21
KNN	0.85	94.81	0.95	0.99	1.05	15
XGBoost	**0.87**	**95.62**	**0.96**	**1**	**1.05**	**11**
CNN	0.72	90.45	0.94	0.98	1.04	14

Performance of Semi-Supervised Approaches in Transfer Setting



	# Clusters	**0% Training Data**			**25% Training Data**		
		MCC	**ACC (%)**	**F1**	**MCC**	**ACC (%)**	**F1**
K-Means+VOTE	1250	0.605	86.6	0.870	**0.638**	**88.1**	**0.880**
K-Means+LR	125	0.582	87.2	0.861	0.592	87.5	0.863
K-Means+RF	200	**0.630**	**87.3**	**0.872**	0.642	87.3	0.874
BIRCH+VOTE	175	0.593	86.4	0.866	0.610	0.878	0.871
BIRCH+LR	100	0.482	84.9	0.825	0.544	0.862	0.847
BIRCH+RF	200	0.611	87.2	0.869	0.613	0.879	0.870

Performance of Supervised Approaches in Transfer Setting



	0% Training Data					25% Training Data				
	MCC	**ACC (%)**	**F1**	**GT**	**CSR**	**MCC**	**ACC (%)**	**F1**	**GT**	**CSR**
DT	0.55	81.06	0.82	0.97	1.03	0.65	86.99	0.87	0.98	1.04
RF	0.63	84.85	0.86	0.98	1.04	**0.70**	**88.94**	**0.89**	**0.96**	**1.05**
SVM	**0.64**	**85.49**	**0.86**	**0.98**	**1.04**	0.68	88.04	0.88	0.98	1.04
KNN	0.46	76.23	0.78	0.95	1.01	0.54	81.08	0.83	0.96	1.02
XGBoost	0.49	77.47	0.79	0.96	1.02	0.60	83.58	0.85	0.97	1.03

Key Takeaways

- Semi-supervised approaches for sparse format selection can be competitive with supervised approaches
  - Explore additional techniques to improve the performance of semi-supervised methods
  - Provides several desirable benefits including easy model portability, easy to include new data, and extend to a runtime with online learning

Explaining the Performance of Supervised and Semi-Supervised Methods for Automated Sparse Matrix Format Selection

Sunidhi Dhandhania, Akshay Deodhar, Konstantin Pogorelov, **Swarnendu Biswas**, and Johannes Langguth

DUAC, ICPP Workshops 2021