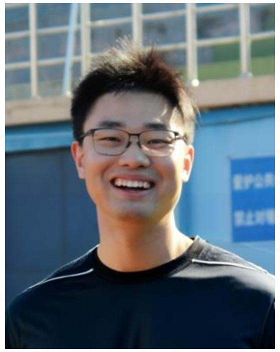
Randomized Point Serialization-Based Efficient Point Transformer in High-Energy Physics Applications

Siqi Miao

Ph.D. Student

ML @ Georgia Tech

Siqi Miao[1]



Zhiyuan Lu[2]

Mia Liu[3]

Javier Duarte[4]

Pan Li[1]

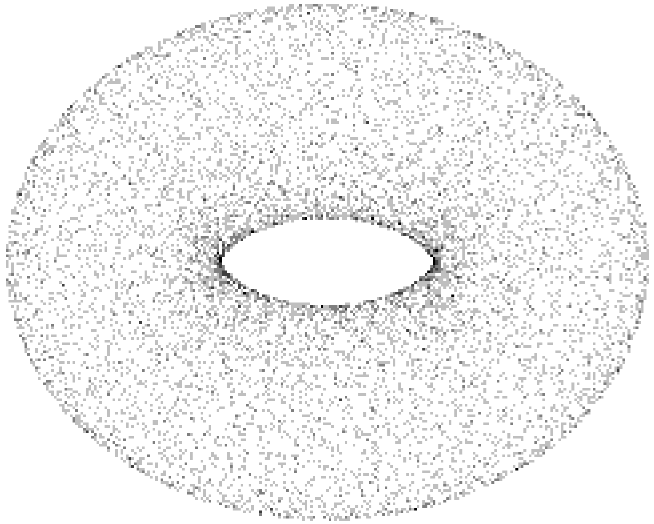
Georgia Tech

北京郵電大學
Beijing University of Posts and Telecommunications

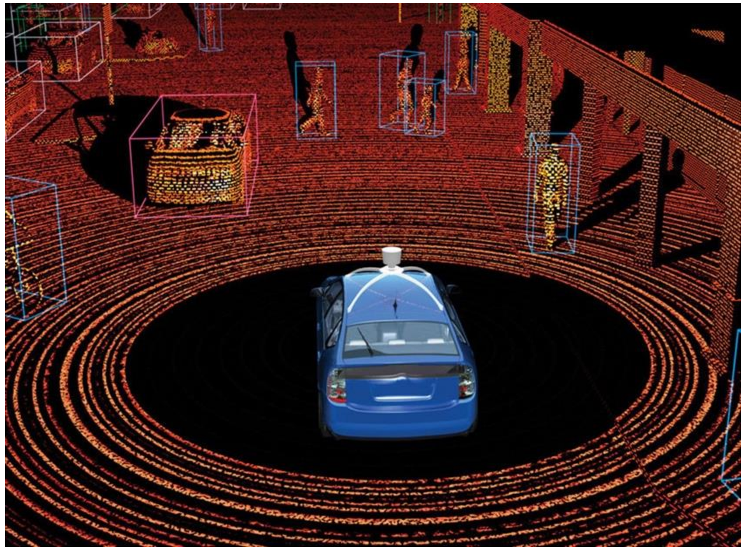
PURDUE UNIVERSITY

UC San Diego

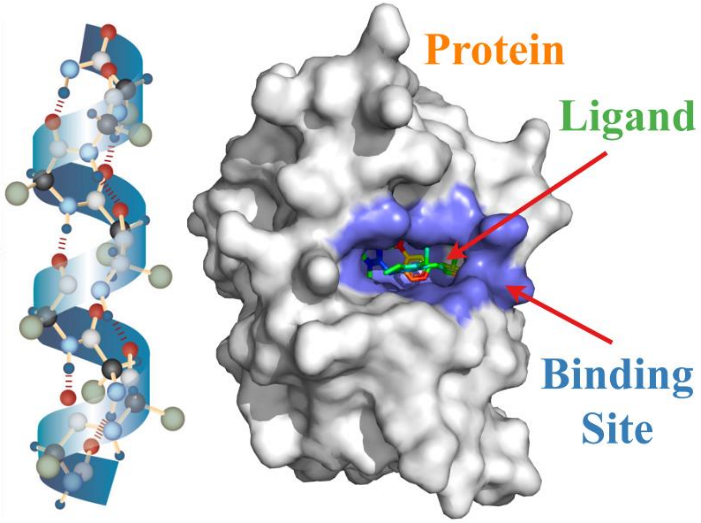
Point Cloud Data



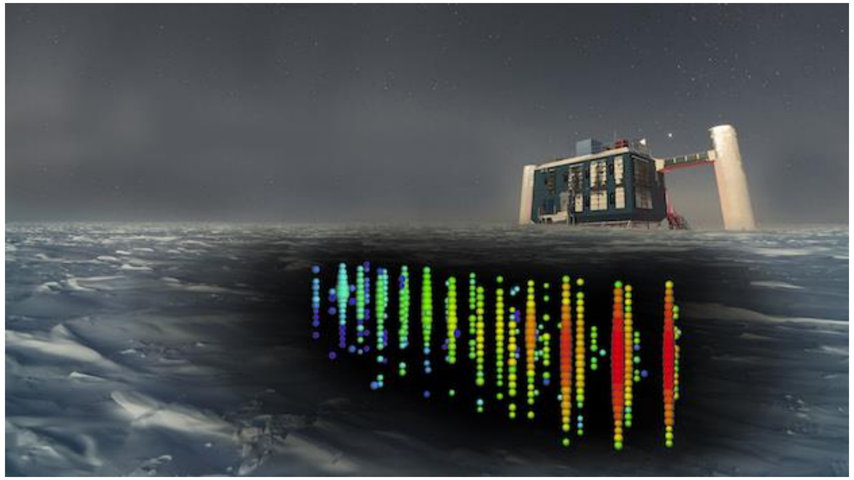
- Autonomous Driving



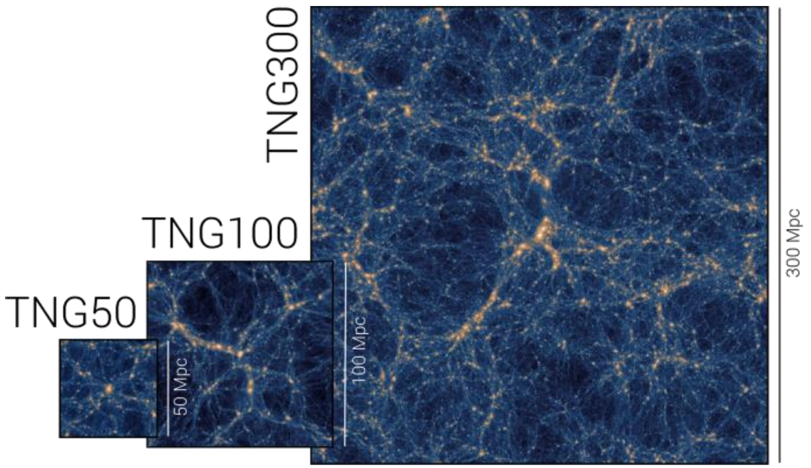
- Drug Discovery



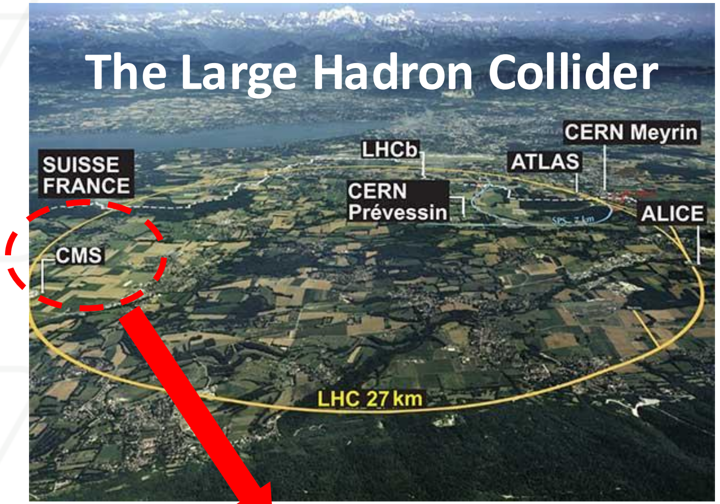
- Neutrino Detection

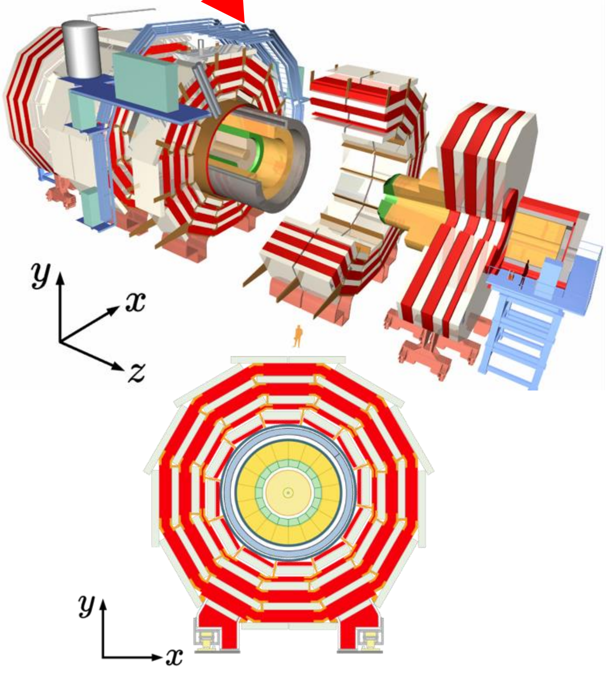


- Galaxy Evolution

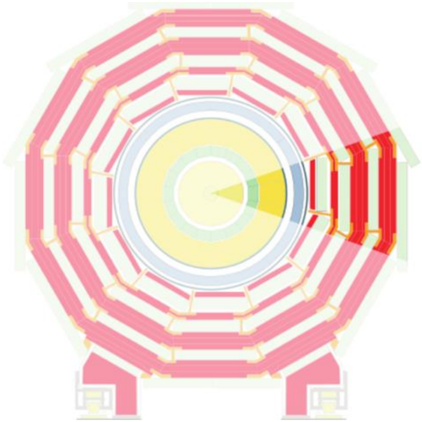


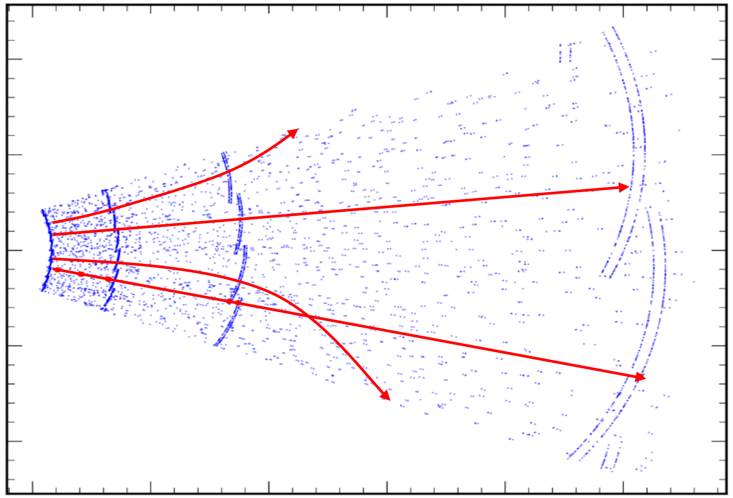
Point Clouds in High-Energy Physics

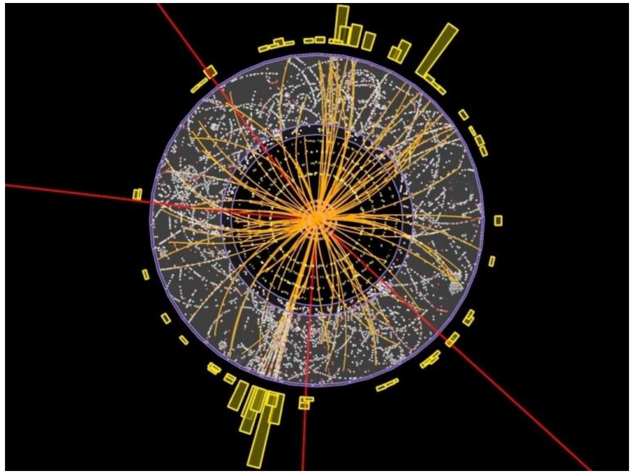


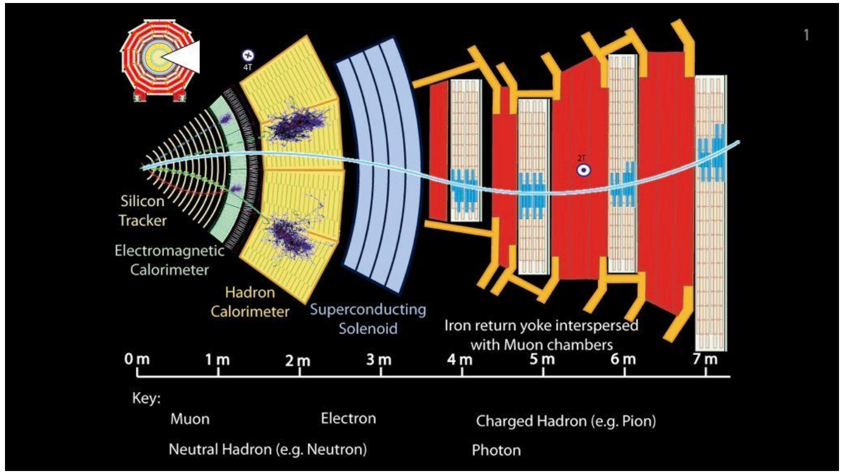


Detector Illustration



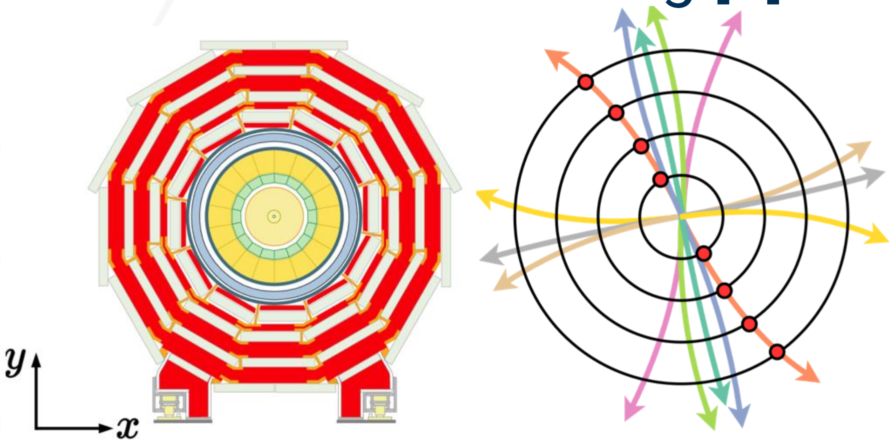




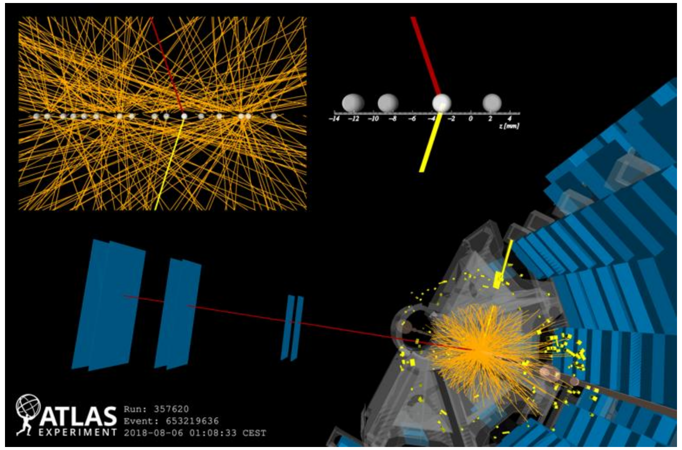


Point Clouds in High-Energy Physics

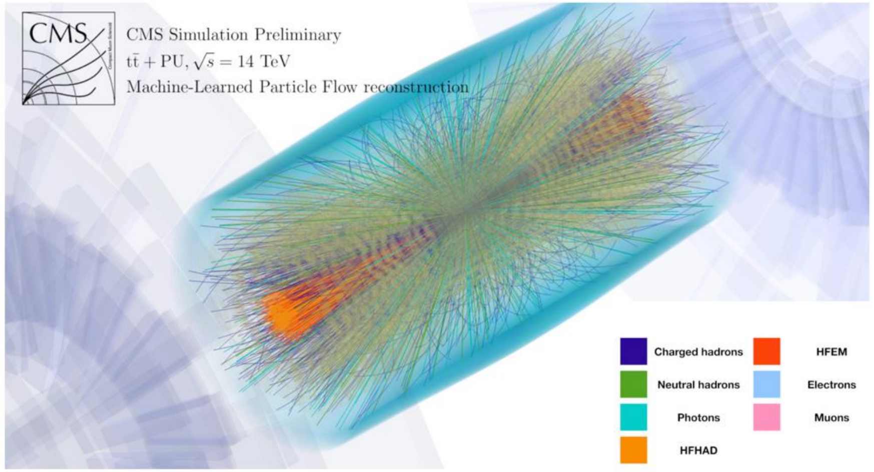
- Particle Tracking [1]



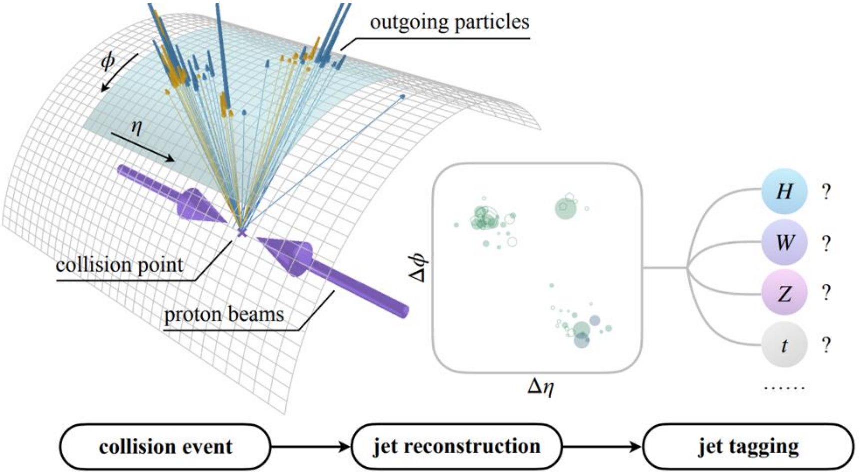
- Pileup Mitigation [2]



- Particle-flow Reconstruction [3]



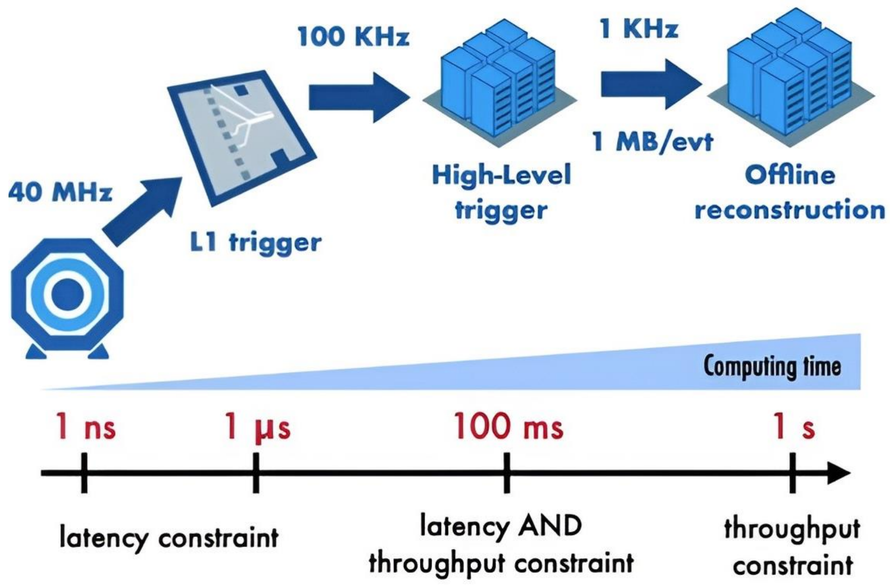
- Jet Tagging [4]



• • •

Current Computational Challenges

- Large-sized point clouds
  - Over ***60k points/cloud*** for the tracking task
- Large amount of data
  - LHC can produce 1 billion particle collisions per second (***1PB data/sec!***) [5]
- Online compute & low latency requirement
  - Data preprocessing ***can't be done offline!***

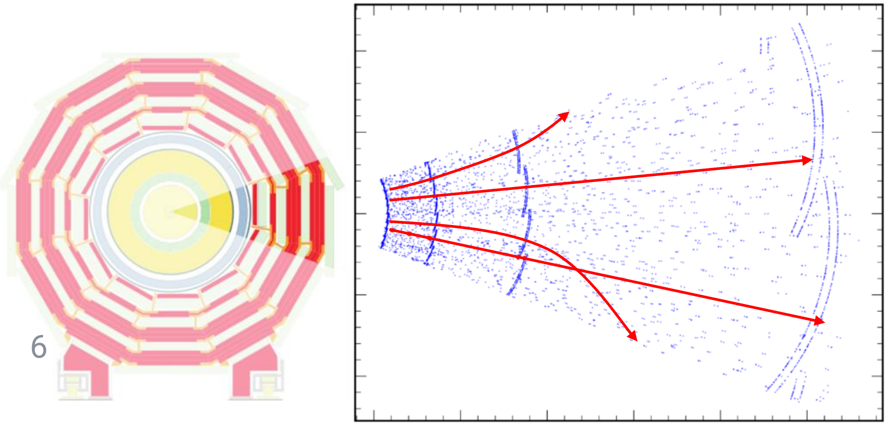


Current Computational Challenges

- Large-sized point clouds
  - Over ***60k points/cloud*** for the tracking task
- Large amount of data
  - LHC can produce 1 billion particle collisions per second (***1PB data/sec!***) [5]
- Online compute & low latency requirement
  - Data preprocessing ***can't be done offline!***

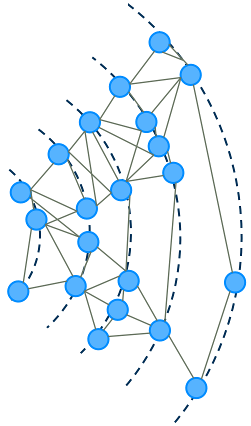
Popular solutions

- Graph Neural Networks [2, 3, 4, 5, 6, 7]
- By converting point clouds to graphs



via, e.g., geometric relations

or kNN graphs



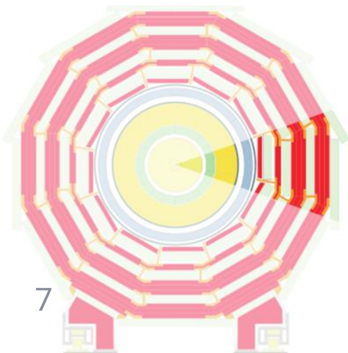
- *Leveraging the sparsity in the data, GNNs can be fast once graph is built*

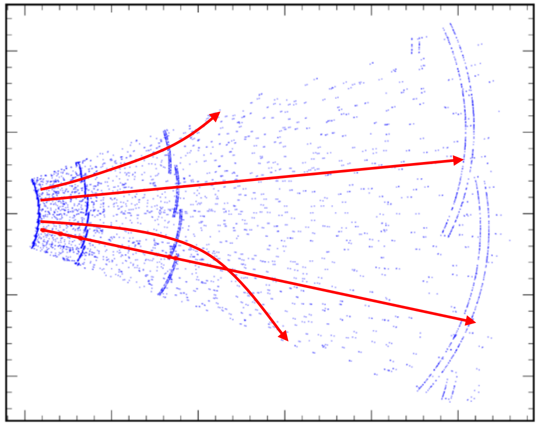
Current Computational Challenges

- Large-sized point clouds
  - Over ***60k points/cloud*** for the tracking task
- Large amount of data
  - LHC can produce 1 billion particle collisions per second (***1PB data/sec!***) [5]
- Online compute & low latency requirement
  - Data preprocessing ***can't be done offline!***

Popular solutions

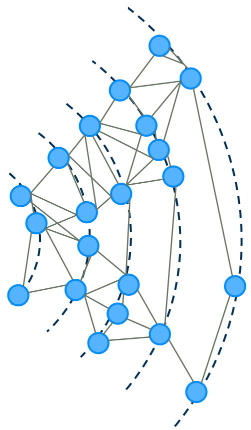
- Graph Neural Networks [2, 3, 4, 5, 6, 7]
- By converting point clouds to graphs





via, e.g., geometric relations

or kNN graphs



GNNs are not fast enough!

1. *Building graphs* ***can be expensive***
   - *kNN may have $\mathcal{O}(n^2)$ complexity!*
2. ***Irregular computation*** & random *memory access*
   - *Not hardware friendly!*

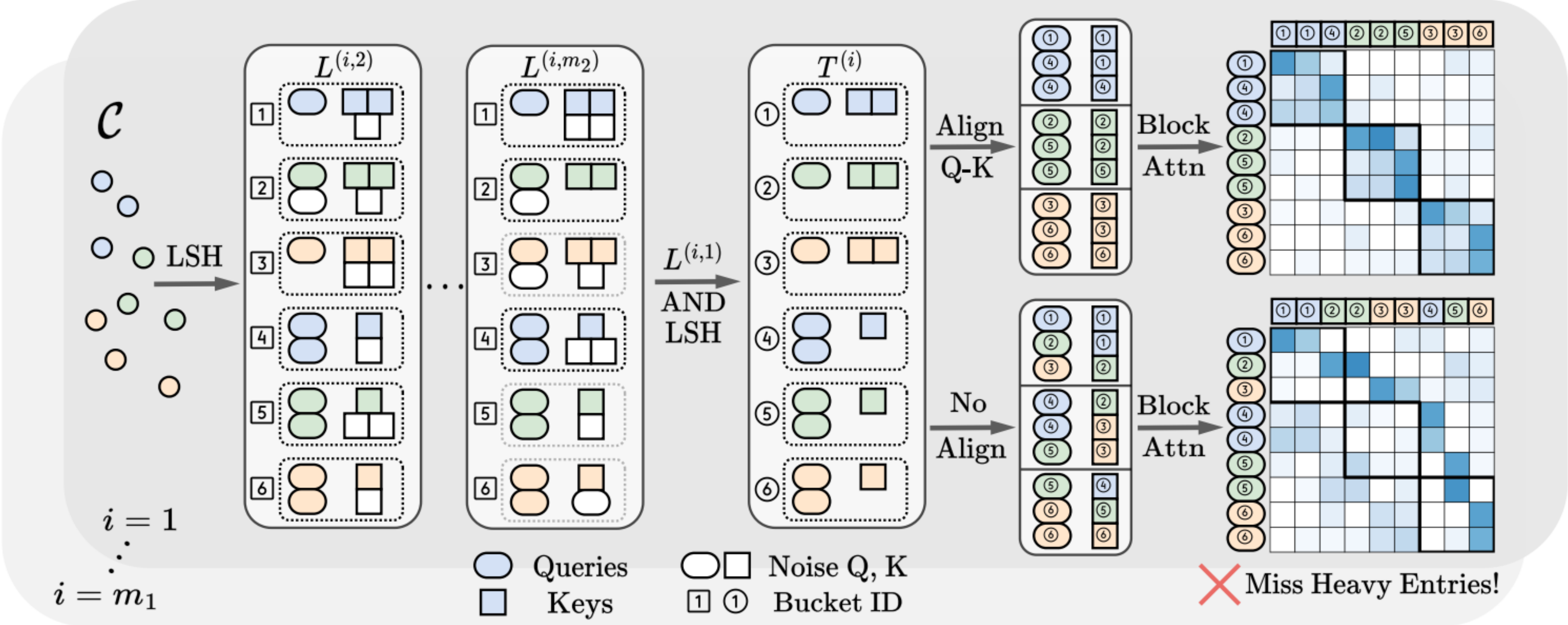
Our Solution: LSH-based Efficient Point Transformer (HEPT)

- We present HEPT, an efficient point transformer based on OR & AND LSH
  - No graph construction
  - Only regular computations
  - Linear complexity

100x+ faster than GNNs! (on GPUs)

On the tracking task, ~60k points/cloud

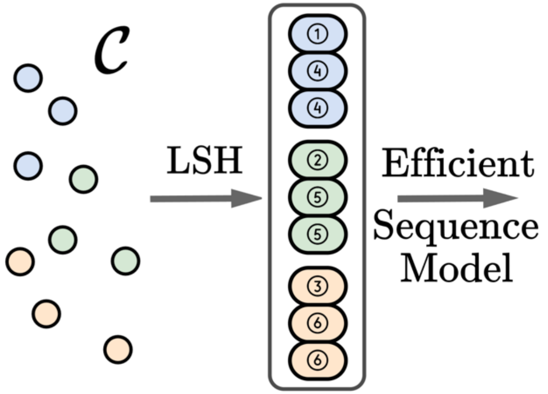
- Architecture
  - Assign hash codes using OR & AND E2LSH. Similar items share close 1D hash codes
  - Sort items based on hash codes. Then compute block-diagonal attention



Our Solution: LSH-based Efficient Point Transformer (HEPT)

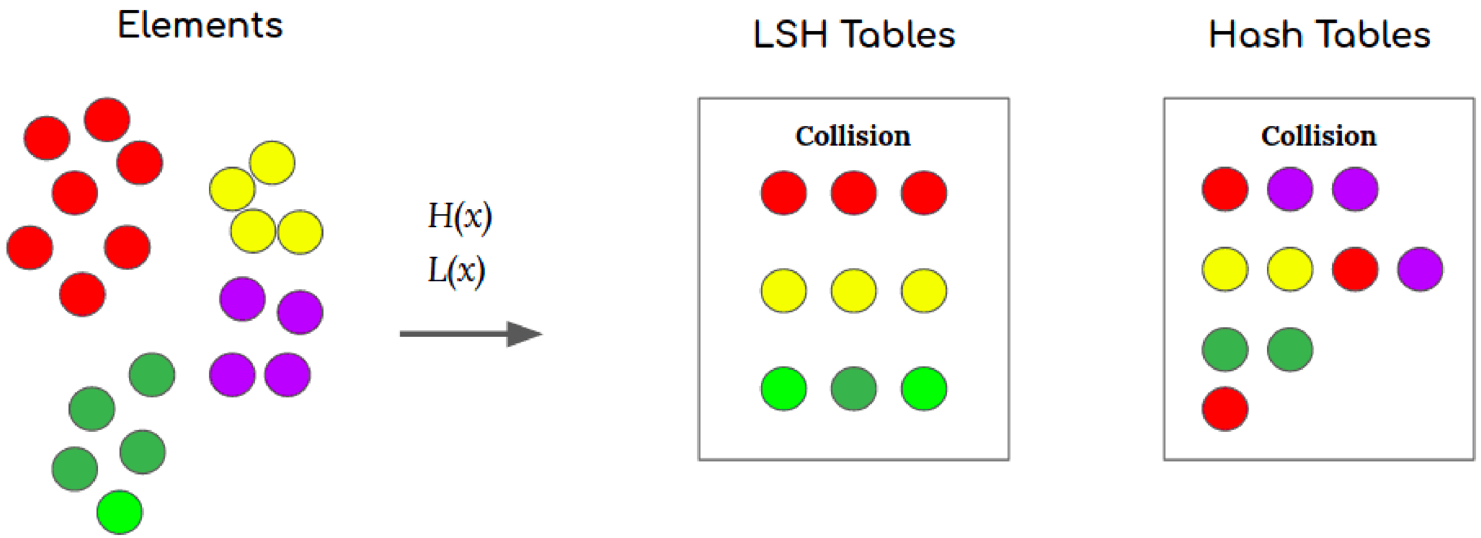
Summary

- HEPT is a point cloud serialization model
  - point clouds --> ***1D seq*** via ***LSH***
  - enable the use of ***efficient sequence models***
  - ***randomized*** serialization patterns



e.g.,
- *Local window attn*
- *State-space models*

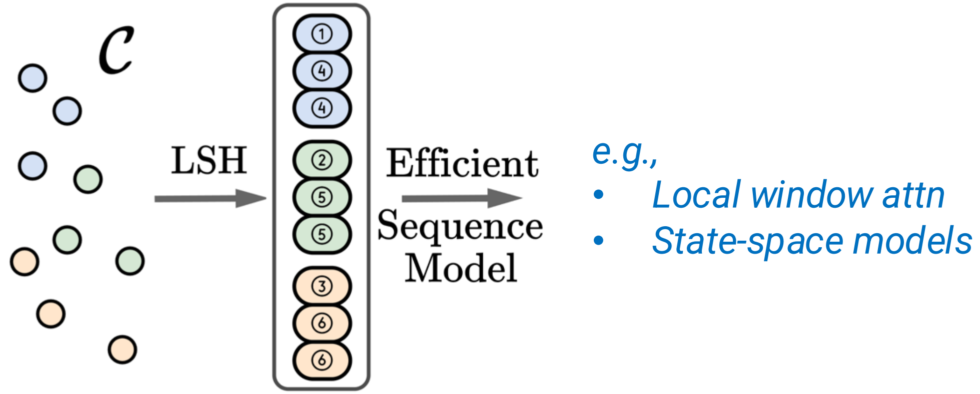
- Prelim: locality-sensitive hashing (LSH)
  - LSH hashes ***similar items*** to the same or ***similar buckets*** w/ high prob.



Our Solution: LSH-based Efficient Point Transformer (HEPT)

Summary

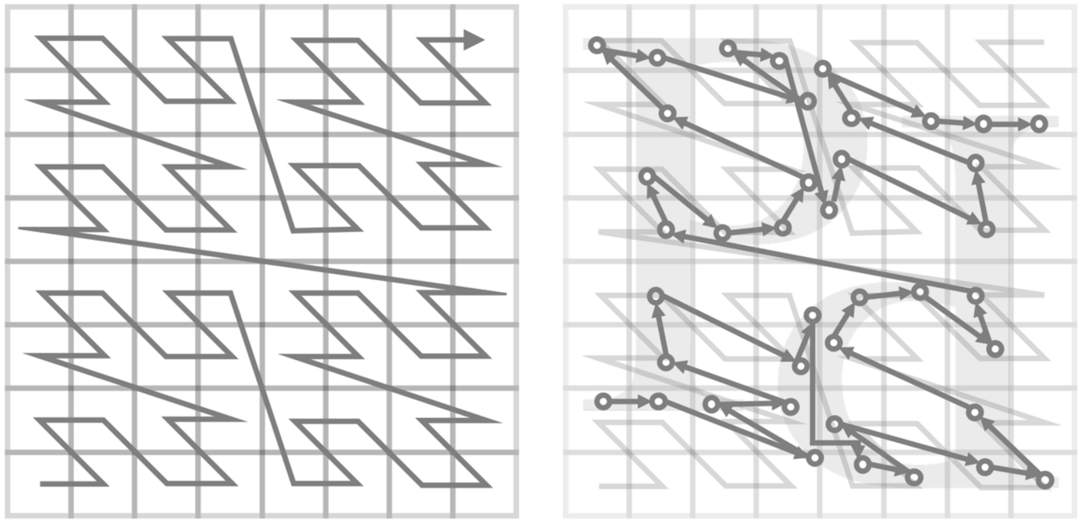
- HEPT is a point cloud serialization model
  - point clouds --> ***1D seq*** via ***LSH***
  - enable the use of ***efficient sequence models***
  - ***randomized*** serialization patterns



and point/vision mamba…

- Different from concurrent work [8], which
  - use ***fixed*** serialization patterns
  - ***hard to analyze*** to provide theoretical guarantees
  - ***cannot preserve*** certain locality patterns

(a) Z-order

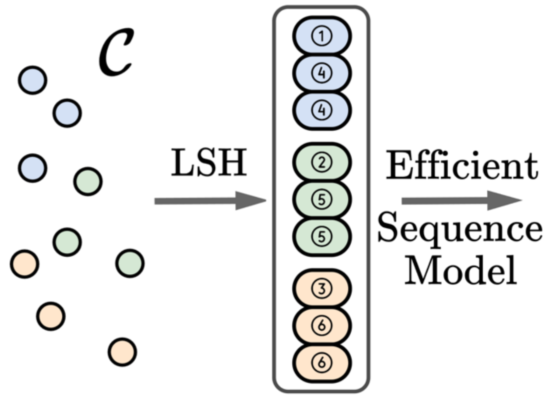


[8] Wu, Xiaoyang, et al. Point Transformer V3: Simpler Faster Stronger. CVPR, 2024.

Our Solution: LSH-based Efficient Point Transformer (HEPT)

Summary

- HEPT is a point cloud serialization model
  - point clouds --> ***1D seq*** via ***LSH***
  - enable the use of ***efficient sequence models***
  - ***randomized*** serialization patterns



e.g.,
- *Local window attn*
- *State-space models*

and point/vision mamba…

- Different from concurrent work [8], which
  - use ***fixed*** serialization patterns
  - ***hard to analyze*** to provide theoretical guarantees
  - ***cannot preserve*** certain locality patterns

(a) Z-order

- HEPT:
  - ***No hand-crafted*** serialization patterns!
  - ***Provable capability*** to preserve locality!
  - Can work even for ***high-dim data***!

[8] Wu, Xiaoyang, et al. Point Transformer V3: Simpler Faster Stronger. CVPR, 2024.

Theoretical Results

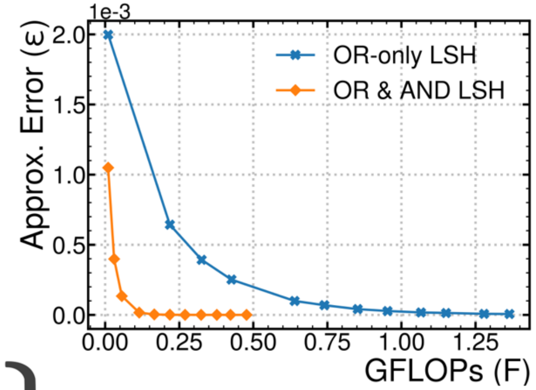
- The sparsity in the data lays the foundation of building an efficient model
  - i.e., a point primarily interacts with its local neighbors
- Our efficient transformer is built based on this property, with theoretical guarantees

Theoretical Results

- The sparsity in the data lays the foundation of building an efficient model
  - i.e., a point primarily interacts with its local neighbors
- Our efficient transformer is built based on this property, with theoretical guarantees
- Analyzed & compared two popular techniques
  - ***random Fourier features*** (RFFs) & ***locality-sensitive hashing*** (LSH)
    - Used by RFA [9], Performer [10], Reformer[11], SMYRF [12], HyperAttn [13], etc.
  - by examining the trade-off between
    - ***approximation error*** (ϵ) & ***computational complexity*** (Flops, F)

Theoretical Results

- The sparsity in the data lays the foundation of building an efficient model
  - i.e., a point primarily interacts with its local neighbors
- Our efficient transformer is built based on this property, with theoretical guarantees
- Analyzed & compared two popular techniques
  - ***random Fourier features*** (RFFs) & ***locality-sensitive hashing*** (LSH)
    - Used by RFA [9], Performer [10], Reformer[11], SMYRF [12], HyperAttn [13], etc.
  - by examining the trade-off between
    - ***approximation error*** (ϵ) & ***computational complexity*** (Flops, F)



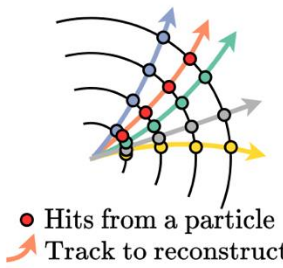
Theoretical Results

1. ***RFFs*** *are consistently* ***worse than LSH*** *under subquadratic complexity*
2. LSH is better. *However,* ***OR-only LSH can't sufficiently reduce the error*** *if F is near-linear*
3. Utilizing ***OR & AND LSH significantly improves performance***, exponentially reducing the error w/ near-linear complexity

- *widely used by prior works*
- *but suboptimal!*

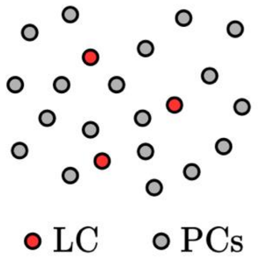
Empirical Results

- Tasks



(a) Charged Particle Tracking

- A **representation learning** task
  - learn close embeddings for points originating from the same particle



(b) Pileup Mitigation

- A **binary point classification** task
  - predict if a neutral particle is from pileup collisions or not

HEPT achieves ***SOTA accuracy!***

Table 1: Predictive performance on the three datasets. The **Bold**†, **Bold**‡, and **Bold** highlight the first, second, and third best results, respectively. Underline indicates the best transformer baselines.

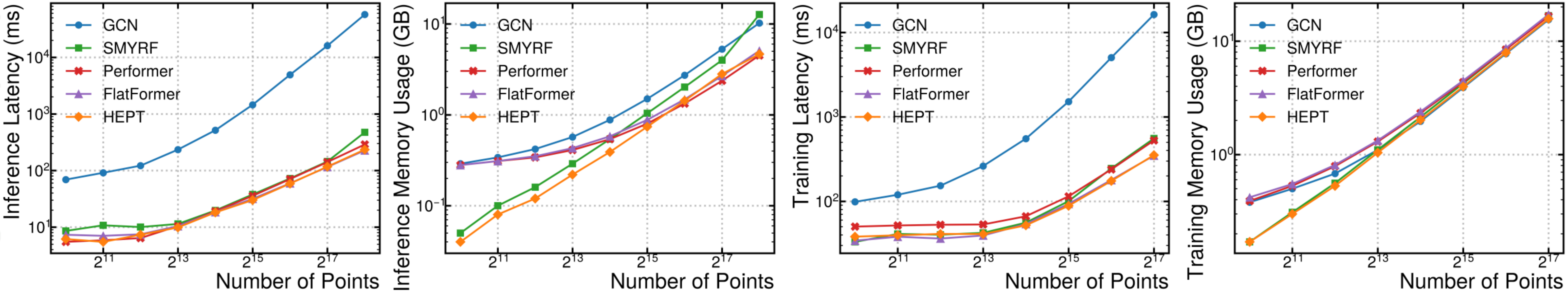
	Tracking-6k (AP@k)	Tracking-60k (AP@k)	Pileup-10k (AUC)
Random	5.88	5.71	4.22
SOTA GNNs	**91.00**‡	**90.89**‡	**40.26**
Reformer	72.37	<u>72.47</u>	36.70
SMYRF	72.98	71.18	25.20
HyperAttn	71.49	70.22	25.31
Performer	73.17	72.07	28.36
FLT	72.55	71.45	25.26
ScatterBrain	73.35	72.06	30.95
PointTrans	72.33	70.81	<u>40.26</u>
FlatFormer	<u>74.22</u>	70.23	38.61
GCN	79.61	75.38	40.10
DGCNN	**90.74**	**88.66**	33.75
GravNet	90.11	87.99	40.10
GatedGNN	80.98	78.42	40.26
Performer-k_{HEPT}	71.97	69.20	32.81
SMYRF-k_{HEPT}	83.19	71.04	**40.31**‡
FlatFormer-k_{HEPT}	88.18	85.06	39.99
HEPT	**92.66**†	**91.93**†	**40.39**†

Table 3: Ablation studies of HEPT.

	Tracking-60k
HEPT w/o k_{HEPT}	72.28
OR-only LSH	71.42
OR-only LSH*	78.22
OR & AND LSH	70.98
OR & AND LSH*	88.54

Empirical Results

- Scalability Analysis



HEPT is one of the ***most efficient*** transformers!

Achieve over ***100x speedup*** on GPUs compared to GNNs on Tracking-60k (60k points/cloud)

Georgia Tech

Ongoing Work

- Further improving HEPT for the tracking task (potentially include other tasks)
- Integrating HEPT with FlashAttention 2

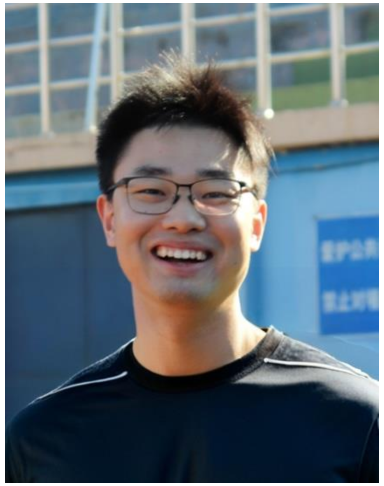
Table 2: Comparison of Attention Computation Method, Inference Speeds, and Block Size on A100

Attention Method	Speed at BS 100 (ms)	Speed at BS 500 (ms)	Speed at BS 1000 (ms)
Original HEPT	25.34	56.68	91.41
FlashAttention 2	17.05	20.49	23.20
FlexAttention	34.06	35.26	35.84

Conclusion

- **Paper:** https://arxiv.org/abs/2402.12535
- **GitHub:** https://github.com/Graph-COM/HEPT


Siqi Miao[1]


Zhiyuan Lu[2]


Mia Liu[3]


Javier Duarte[4]


Pan Li[1]

Georgia Tech
北京郵電大學 Beijing University of Posts and Telecommunications
PURDUE UNIVERSITY
UC San Diego

References

1. Siqi Miao et al. Locality-sensitive hashing-based efficient point transformer with applications in high-energy physics. ICML, 2024.
2. J Arjona Martínez et al. Pileup mitigation at the large hadron collider with graph neural networks. The European Physical Journal Plus, 2019.
3. Pata, Joosep, et al. MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks. The European Physical Journal C, 2021.
4. Qu, Huilin, and Loukas Gouskos. Jet tagging via particle clouds. Physical Review D, 2020.
5. Gaillard, M. Cern data centre passes the 200-petabyte milestone. 2017.
6. DeZoort, Gage, et al. Charged particle tracking via edge-classifying interaction networks. Computing and Software for Big Science, 2021.
7. Thais, Savannah, et al. Graph neural networks in particle physics: Implementations, innovations, and challenges. arXiv preprint, 2022.
8. Wu, Xiaoyang, et al. Point Transformer V3: Simpler Faster Stronger. CVPR, 2024.
9. Peng, Hao, et al. Random feature attention. ICLR, 2021.
10. Choromanski, Krzysztof, et al. Rethinking attention with performers. ICLR, 2021.
11. Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. ICLR, 2020.
12. Daras, Giannis, et al. Smyrf-efficient attention using asymmetric clustering. NeurIPS, 2020.
13. Han, Insu, et al. Hyperattention: Long-context attention in near-linear time. ICLR, 2024.

Georgia Tech