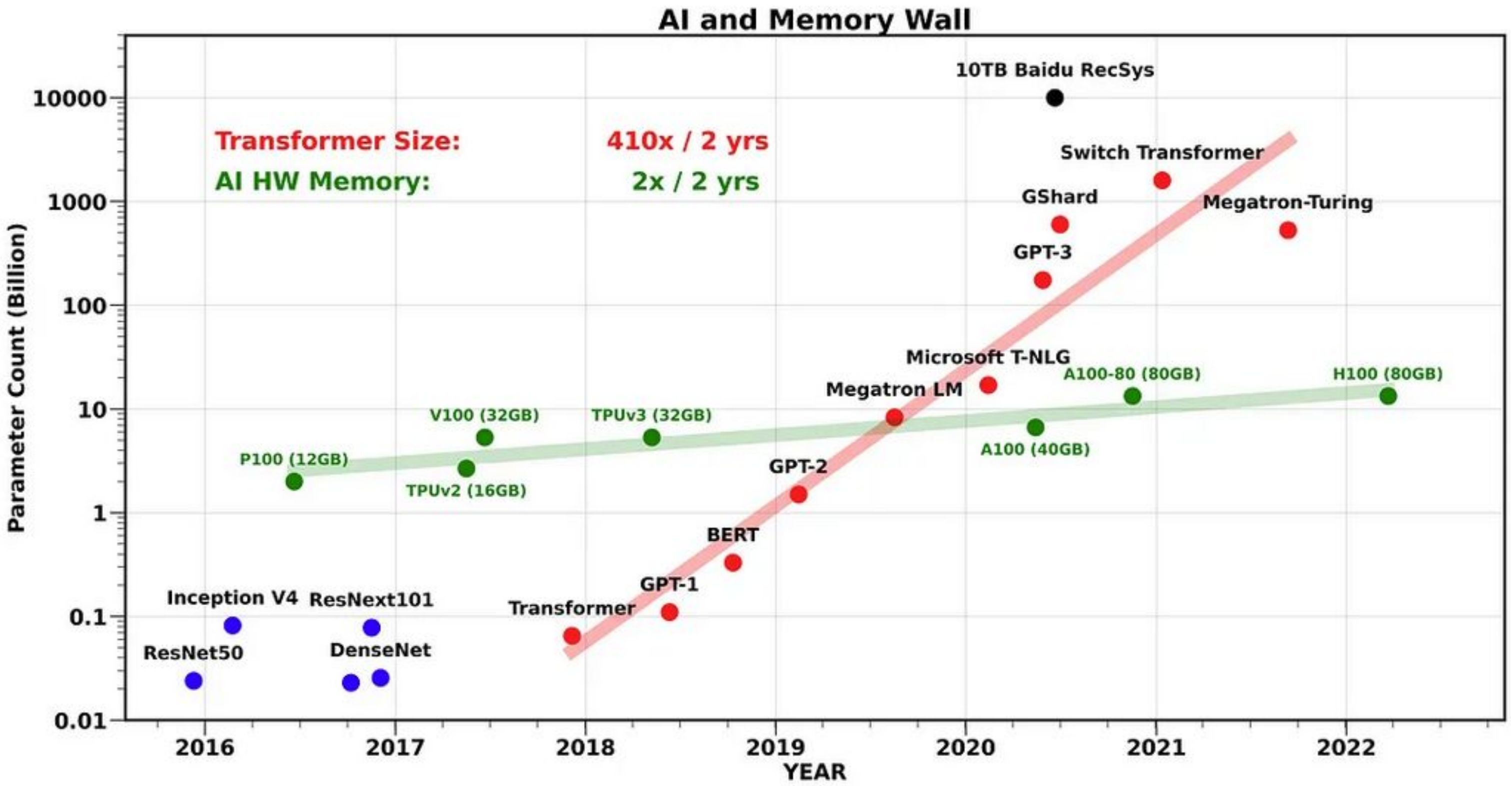
Coop: Memory is not a Commodity

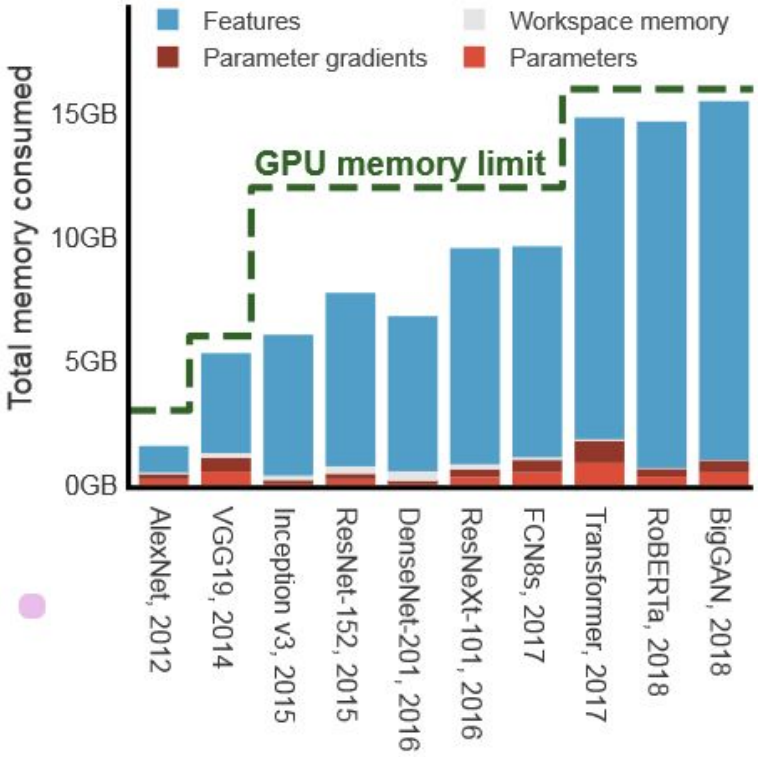
Jianhao Zhang, Shihan Ma, Peihong Liu, and Jinhui Yuan

Memory Wall



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Memory Bottleneck



P Jain et al. Checkmate: Breaking the memory wall with optimal tensor rematerialization. 2020

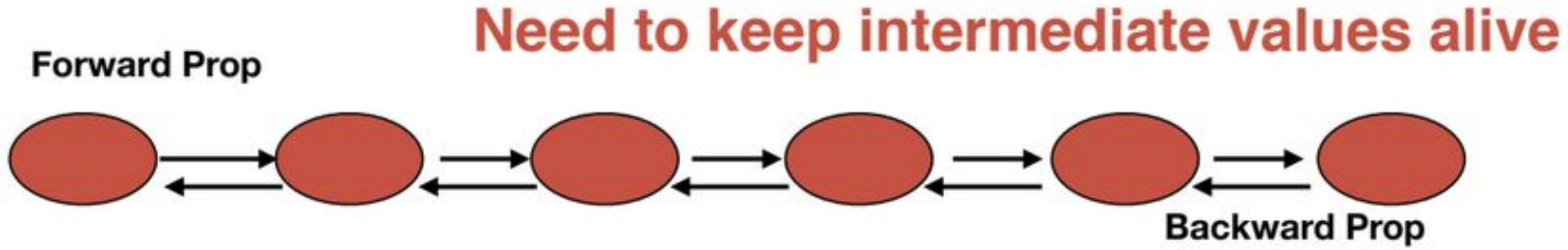
Gradient Checkpointing/Tensor Rematerialization

Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016

Gradient Checkpointing/Tensor Rematerialization

Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016

Traditional Backpropagation



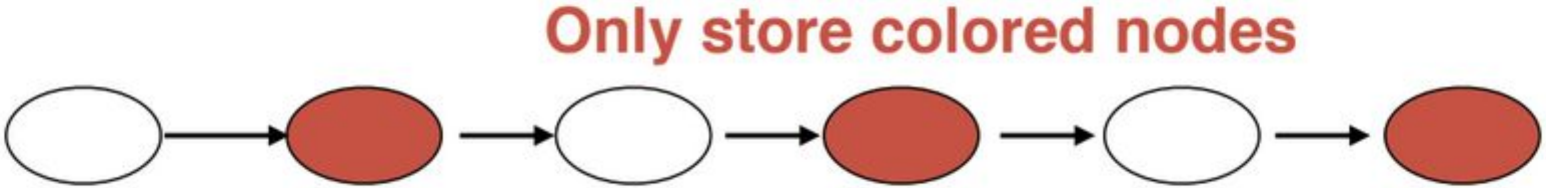
Memory Cost = O(N)

Based on slides by Tianqi Chen: https://drive.google.com/file/d/1YOWKaCfilzSjukavNmpTEsuyMBeoahKj/view

Gradient Checkpointing/Tensor Rematerialization

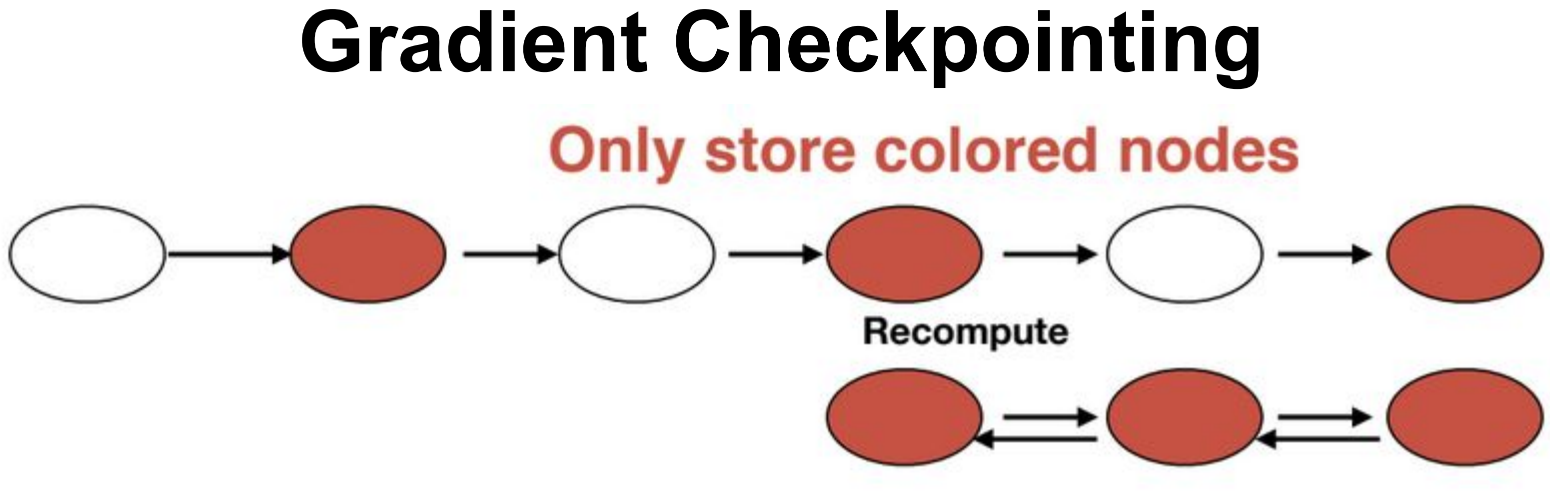
Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016

Gradient Checkpointing



Gradient Checkpointing/Tensor Rematerialization

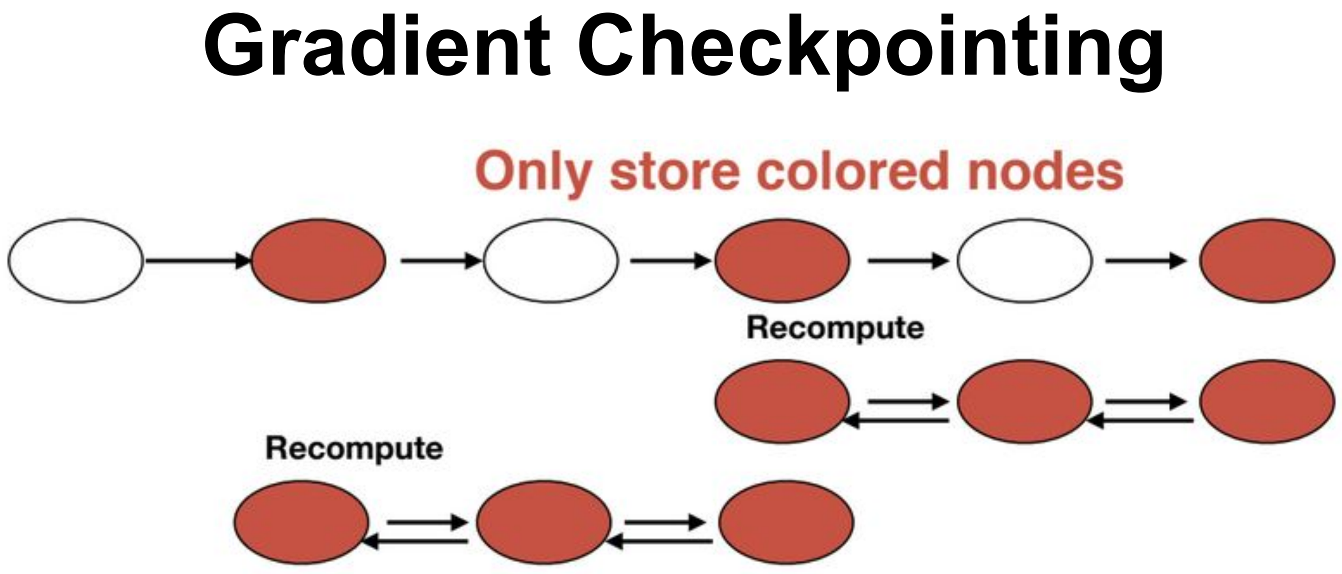
Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016



Based on slides by Tianqi Chen: https://drive.google.com/file/d/1YOWKaCfilzSjukavNmpTEsuyMBeoahKj/view

Gradient Checkpointing/Tensor Rematerialization

Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016



Based on slides by Tianqi Chen: https://drive.google.com/file/d/1YOWKaCfilzSjukavNmpTEsuyMBeoahKj/view

Gradient Checkpointing/Tensor Rematerialization

Chen et al. Training Deep Nets with Sublinear Memory Cost, 2016

Gradient Checkpointing

Memory Cost = O(K) + O(N/K)

$\Downarrow$

O(√N)

Based on slides by Tianqi Chen: https://drive.google.com/file/d/1YOWKaCfilzSjukavNmpTEsuyMBeoahKj/view

Checkmate

Jain et al. Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization, 2020

Rematerialization as integer linear program (ITP):

$$\underset{R,S,U,\textsc{Free}}{\arg\min} \quad \sum_{t=1}^{n}\sum_{i=1}^{t} C_i R_{t,i} \tag{1a}$$

subject to

$$R_{t,j} \le R_{t,i} + S_{t,i} \qquad \forall t\ \forall (v_i, v_j) \in E, \tag{1b}$$

$$S_{t,i} \le R_{t-1,i} + S_{t-1,i} \qquad \forall t \ge 2\ \forall i, \tag{1c}$$

$$\textstyle\sum_i S_{1,i} = 0, \tag{1d}$$

$$\textstyle\sum_t R_{t,n} \ge 1, \tag{1e}$$

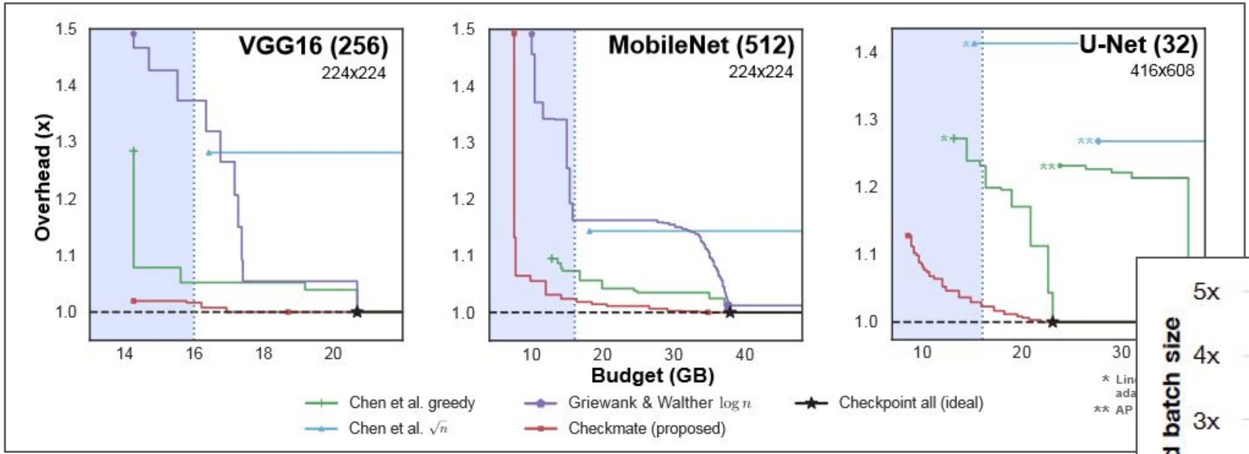
$$R_{t,i}, S_{t,i} \in \{0,1\} \qquad \forall t\ \forall i \tag{1f}$$

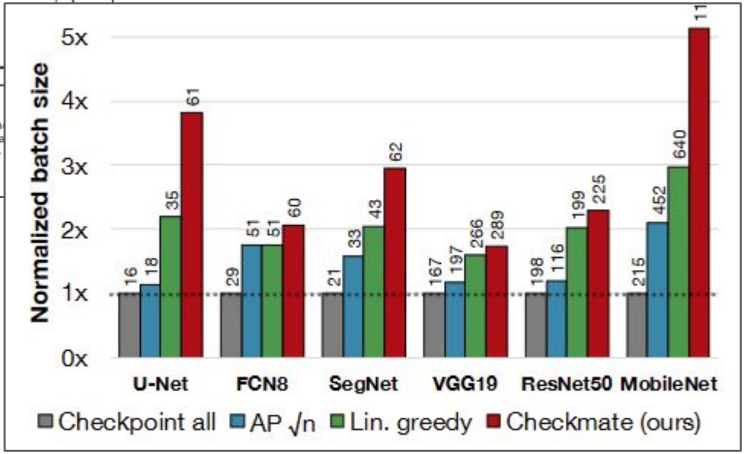
$$U_{t,k} \le M_{\text{budget}}$$

Jain et al. Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization, 2020

Checkmate Results

Jain et al. Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization, 2020





Jain et al. Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization, 2020

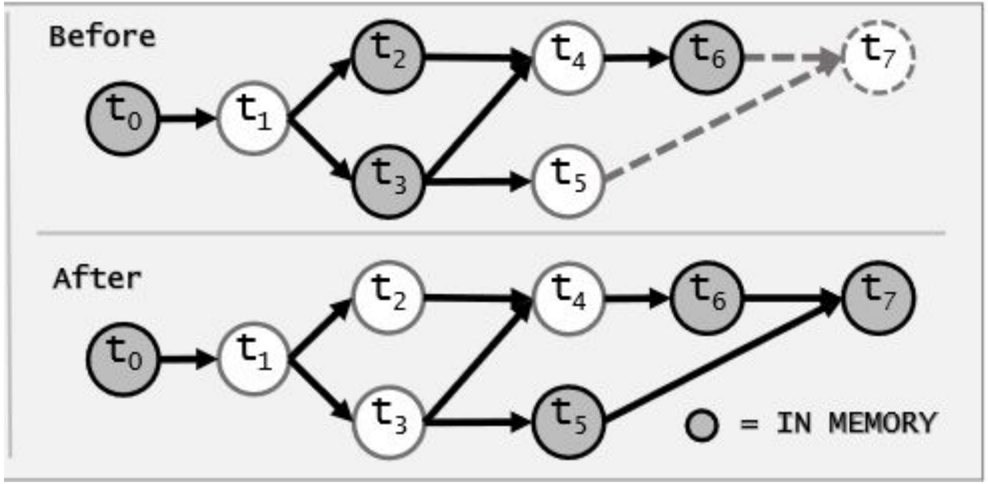
Static Assumption

Two main problems with static planning:

1. Dynamic models
2. Expensive for large models

Dynamic Tensor Rematerialization (DTR)

Kirisame et al. Dynamic Tensor Rematerialization, 2021



Kirisame et al. Dynamic Tensor Rematerialization, 2021

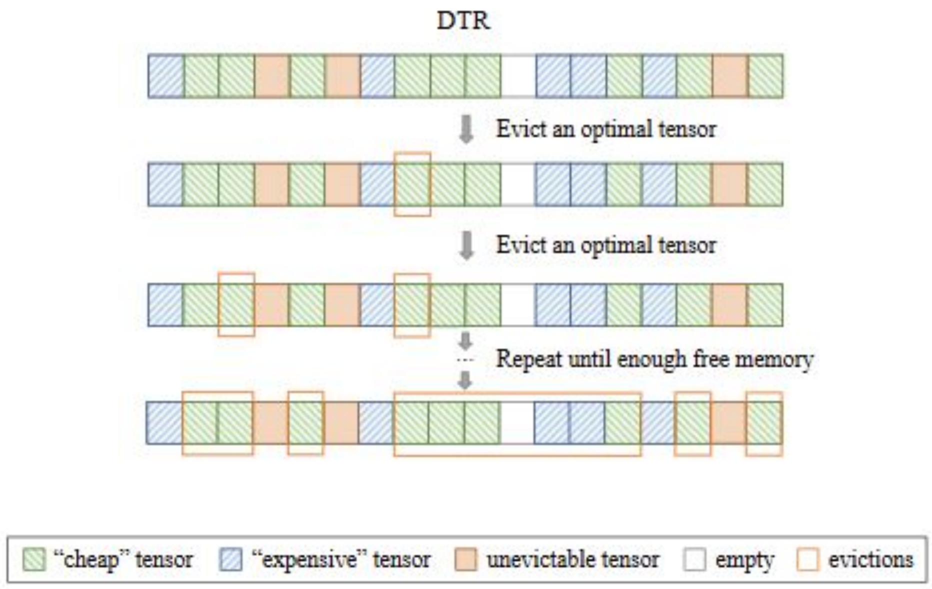
Dynamic Tensor Rematerialization (DTR)

Kirisame et al. Dynamic Tensor Rematerialization, 2021

- Cost c(t): the computation cost of tensor t
- Staleness s(t): the time since tensor t was last accessed
- Memory m(t): the size of tensor t

Heuristic Policy: minimize $h(t) = c(t) / (m(t) * s(t))$

Memory Fragmentation



Zhang et al. Coop: Memory is not a commodity, 2024

MegTaiChi

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

Goal: Combine tensor partitioning and rematerialization

Challenges:

1. Tensor partitioning is static; rematerialization is dynamic
2. Which tensors to evict?
3. How to optimize memory space?

MegTaiChi

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

Goal: Combine tensor partitioning and rematerialization

Challenges:

1. Tensor partitioning is static; rematerialization is dynamic
2. Which tensors to evict?
3. How to optimize memory space?

MegTaiChi: Key Parts

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

1. Dynamic Tensor Partition (DTP)
2. Dynamic Tensor Evicting (DTE)
3. Tensor Memory Allocation (TMA)

MegTaiChi: Dynamic Tensor Evicting (DTE)

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

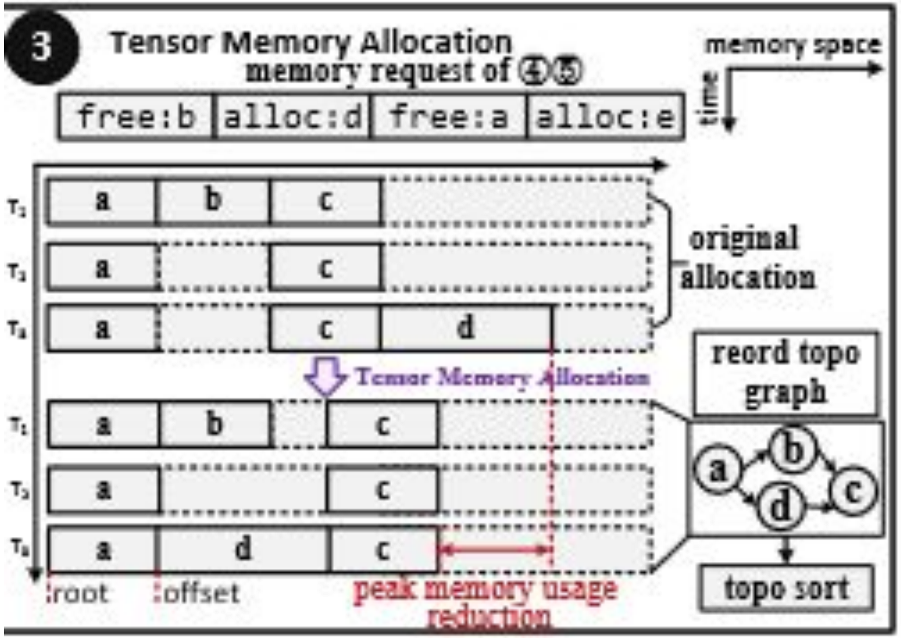
Tensor Evicting Mechanism:

Heuristic Policy: $$\min_{t \in M} \frac{\min(\mathbb{C}_r(t), \mathbb{C}_s(t)) \cdot \beta^{ret(t)}}{\left(m(t) + M_{left}(t) + M_{right}(t)\right) \cdot s(t)}$$

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

MegTaiChi: Tensor Memory Allocation (TMA)

Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021



Hu et al. MegTaiChi: dynamic tensor-based memory management optimization for DNN training, 2021

Coop: Problem Formulation

Goal: Decrease memory fragmentation!!!

Cost Function: $$\underset{S,L}{\arg\min} \sum_{t \in S} h(t), \ \text{subject to } M(S,L) \geq M_R$$

Zhang et al. Coop: Memory is not a commodity, 2024

Coop: Key Strategies

1. Sliding Window Algorithm
2. Cheap Tensor Partitioning
3. Recomputable In-place

Coop: Sliding Window Algorithm

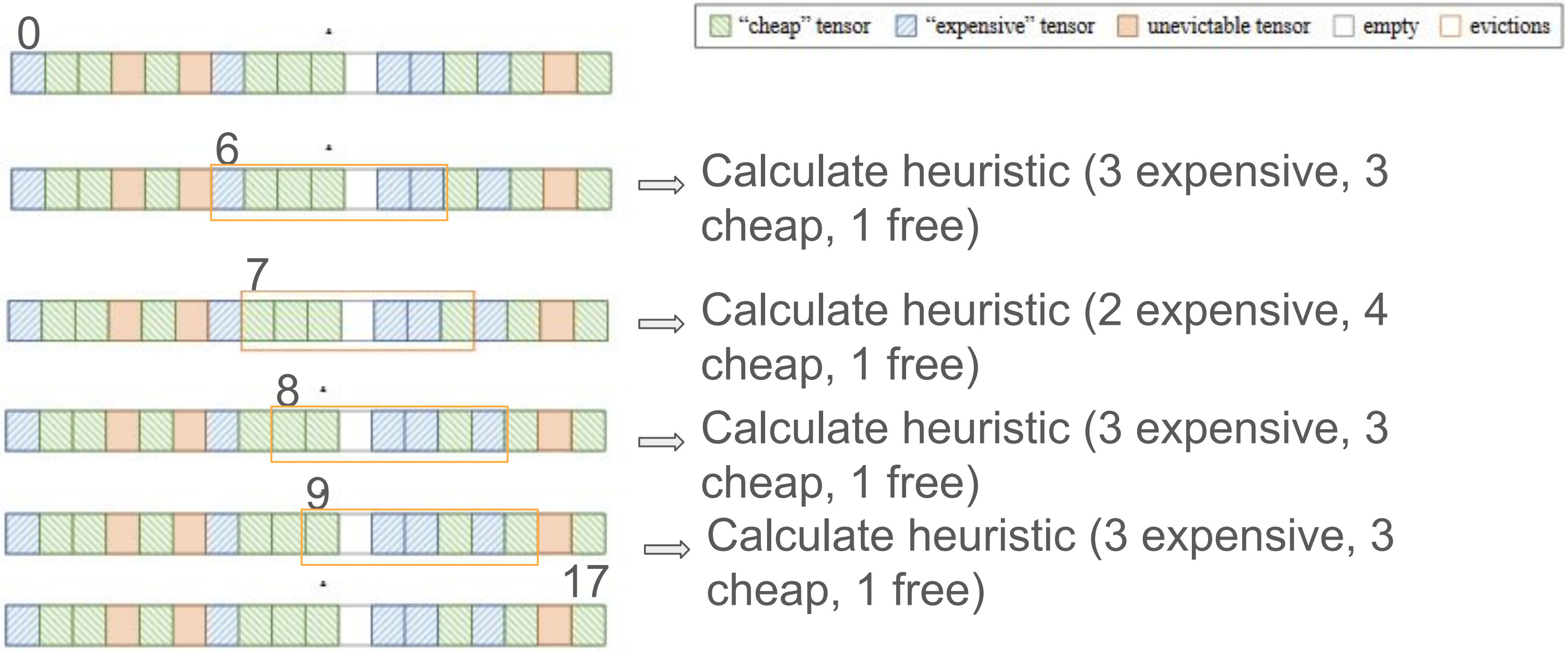
- Brute Force Approach: $O(2^N)$
- Sliding Window Algorithm: $O(N)$

Cost Function: $$\arg\min_{S,L} \sum_{t \in S} h(t), \text{ subject to } M(S,L) \geq M_R$$

Heuristic Policy: $h(t) = c(t) / s(t)$

Zhang et al. Coop: Memory is not a commodity, 2024

Coop: Sliding Window Algorithm



Calculate heuristic (3 expensive, 3 cheap, 1 free)

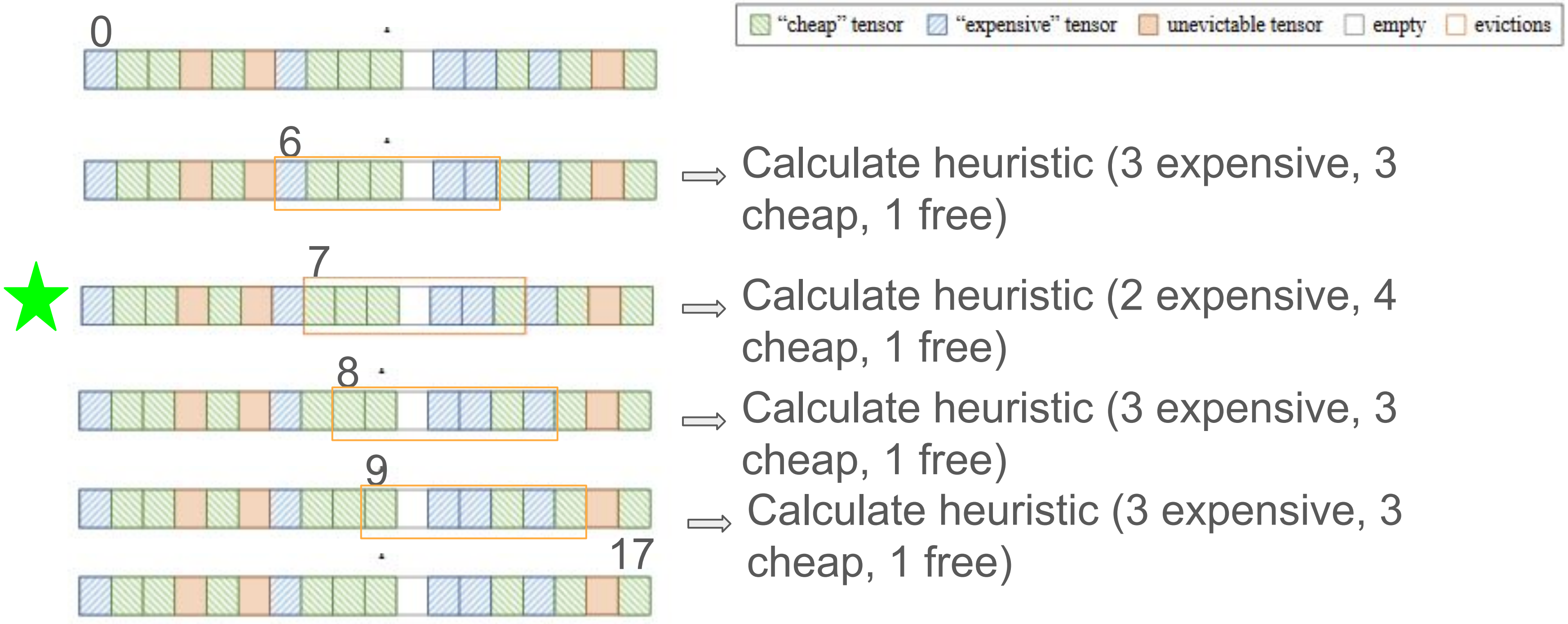
Calculate heuristic (2 expensive, 4 cheap, 1 free)

Calculate heuristic (3 expensive, 3 cheap, 1 free)

Calculate heuristic (3 expensive, 3 cheap, 1 free)

Based on Zhang et al. Coop: Memory is not a commodity, 2024

Coop: Sliding Window Algorithm



⇨ Calculate heuristic (3 expensive, 3 cheap, 1 free)

⇨ Calculate heuristic (2 expensive, 4 cheap, 1 free)

⇨ Calculate heuristic (3 expensive, 3 cheap, 1 free)

⇨ Calculate heuristic (3 expensive, 3 cheap, 1 free)

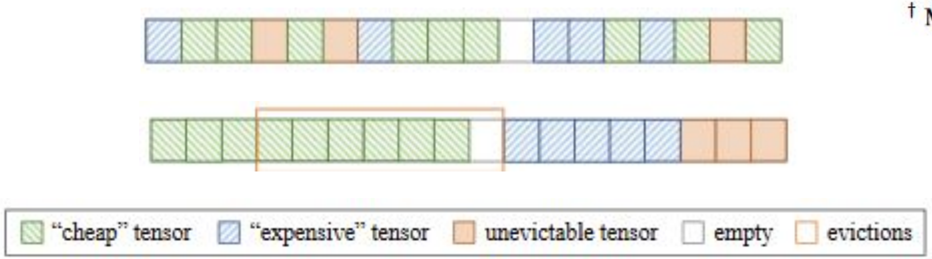
Based on Zhang et al. Coop: Memory is not a commodity, 2024

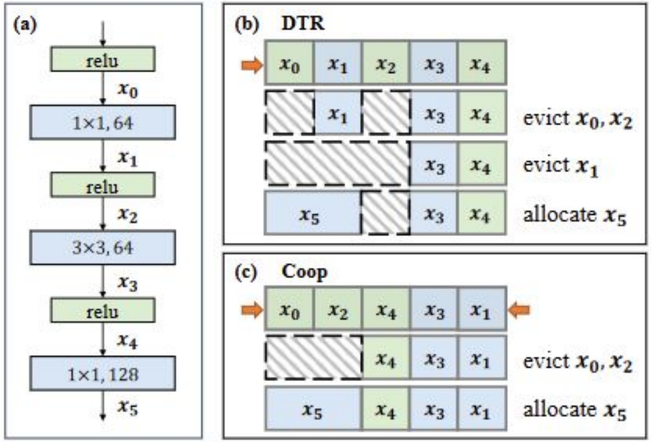
Coop: Cheap Tensor Partitioning

Table 1: Cost density of major operations in DNNs (μs/MB).

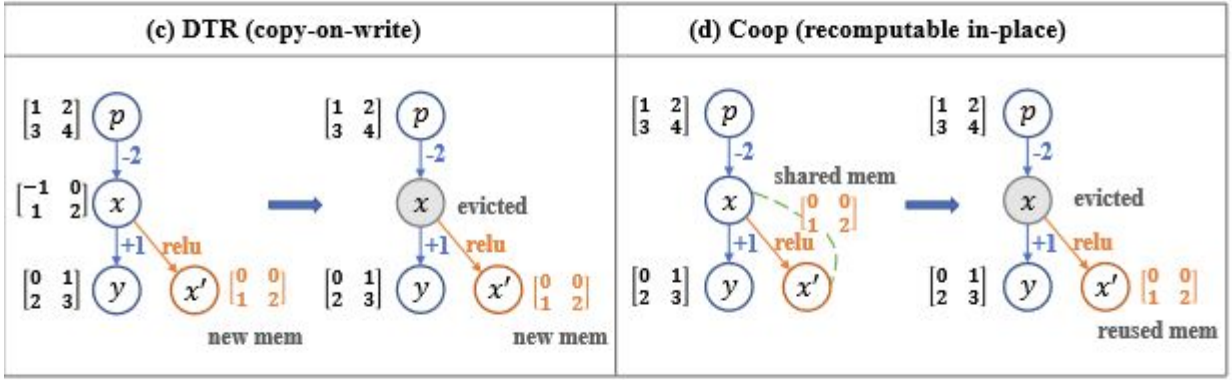
Operation	ResNet-50*	GPT-2†	U-Net*	Swin-T†
$\mathcal{C}_1$ Conv/MatMul	35.6	33.5	89.3	32.7
$\mathcal{C}_2$ Batch/LayerNorm	5.0	4.2	5.3	4.1
$\mathcal{C}_2$ ReLU/GELU	3.9	3.8	3.9	3.9

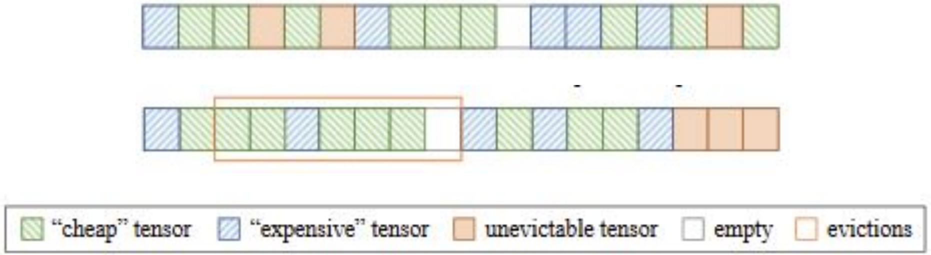
* Model with Conv, BatchNorm and ReLU
† Model with MatMul, LayerNorm and GELU.





Coop: Recomputable In-place





Evaluation

- Comparison methods

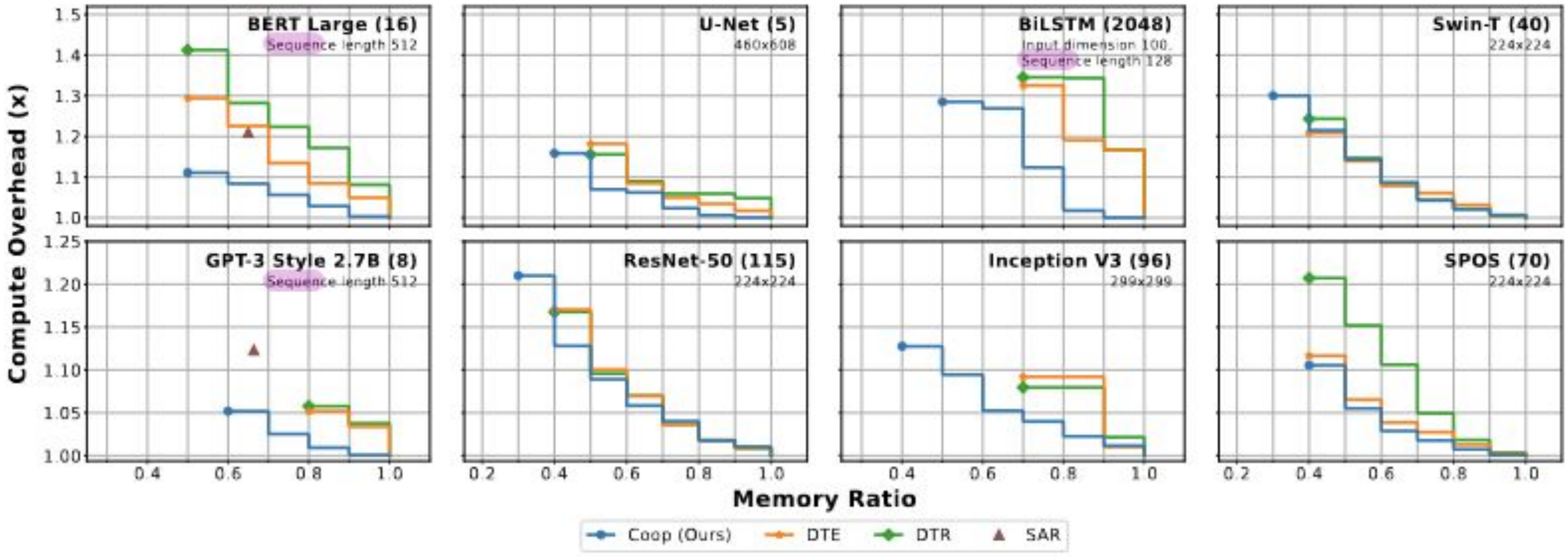
Method	Description	Automatic	MS*-aware	MS-optimized	Traversal Count
Coop	The proposed tensor rematerialization method.	✓	✓	✓	Single
DTE	Our impl. of Dynamic Tensor Evicting [23] in OneFlow.	✓	∼⋆	X	Multiple
DTR	Our impl. of Dynamic Tensor Rematerialization [16] in OneFlow.	✓	X	X	Multiple
SAR	Our impl. of Selective Activation Recomputation [38] in OneFlow.	X	X	X	-

* MS is the abbreviation of the memory system.

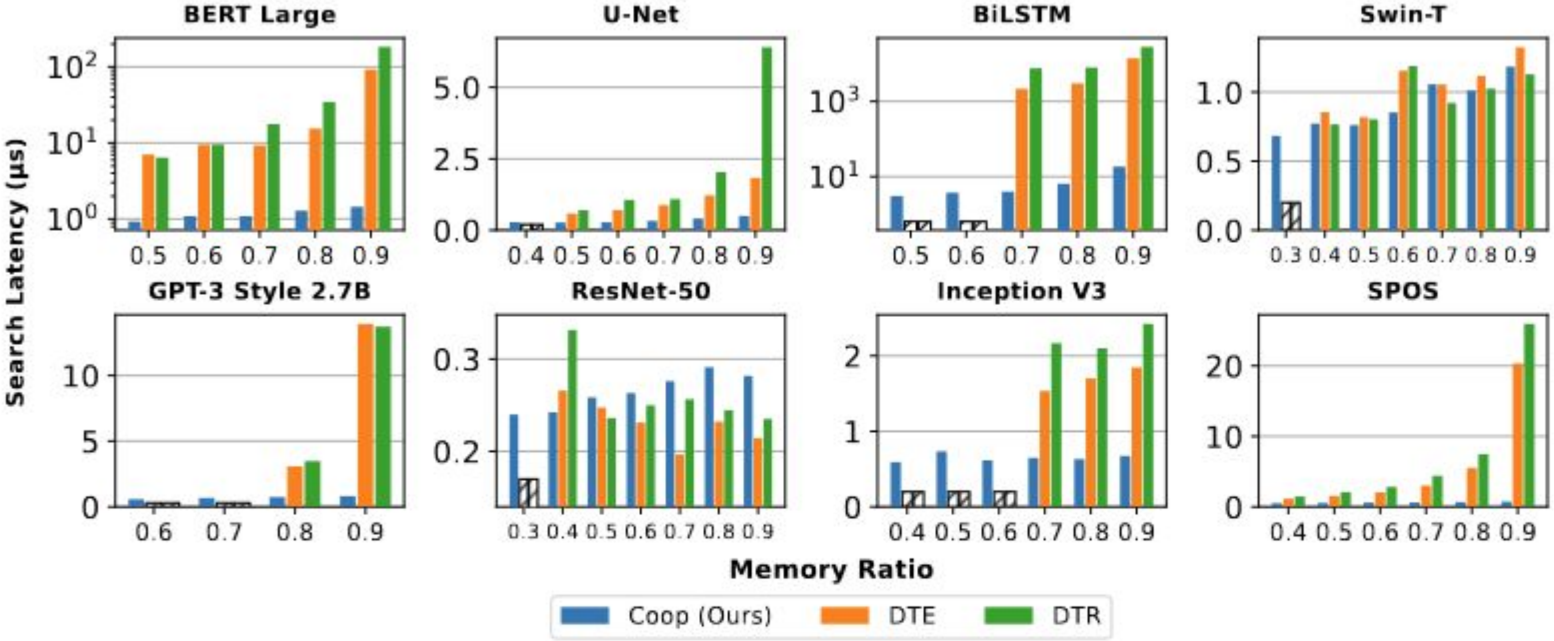
⋆ DTE's heuristic considers adjacent free memory but cannot promise a contiguous memory block is obtained.

- Eight models:
  - Two transformers (BERT Large and GPT-3 w/ 2.7B parameters)
  - Two dynamic models (BiLSTM and SPOS)
  - Other models (U-Net, ResNet, Inception v3, and Swin-T)
- Machine
  - 4 NVIDIA A100 GPU (80 GB, CUDA 11.7, CuDNN 8.5.0) and 56 Intel(R) Xeon(R) Platinum 8336C CPU cores running Ubuntu 20.04.

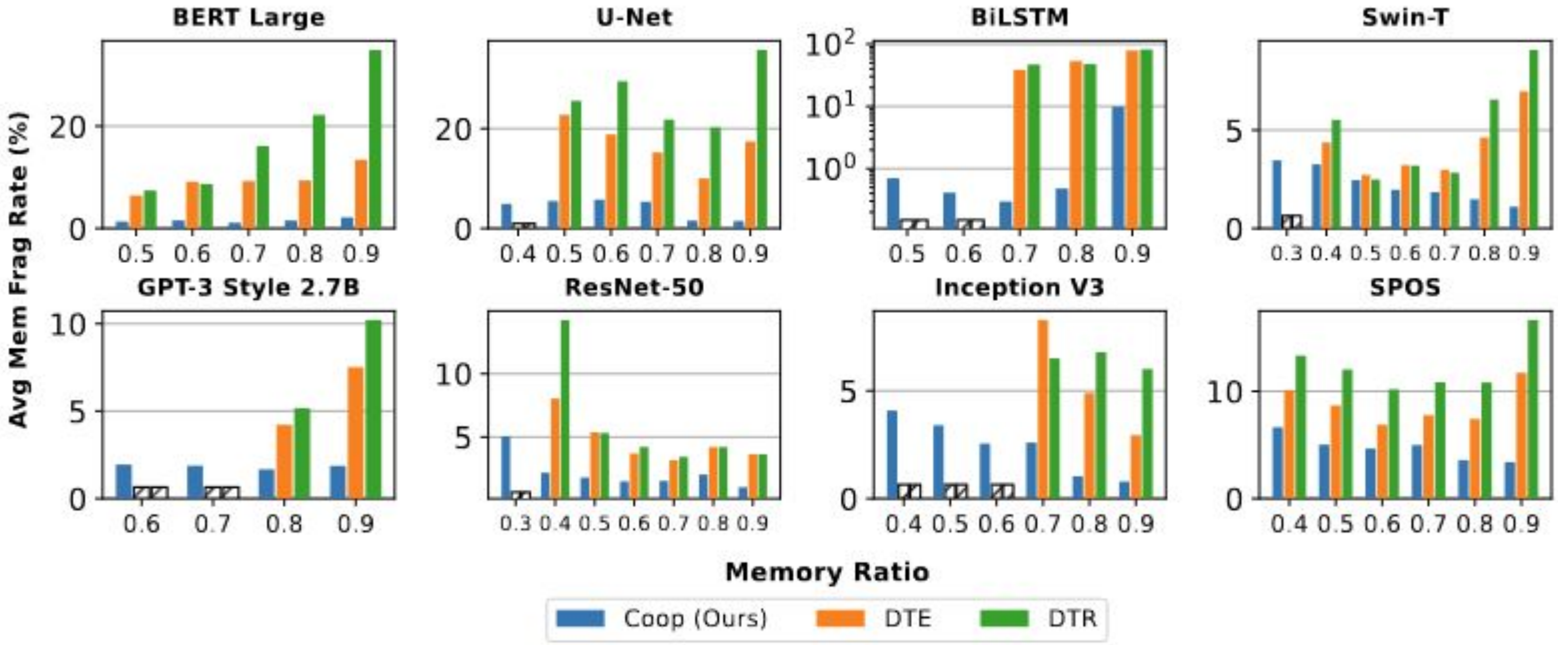
Results - Compute Overhead and Memory Budget



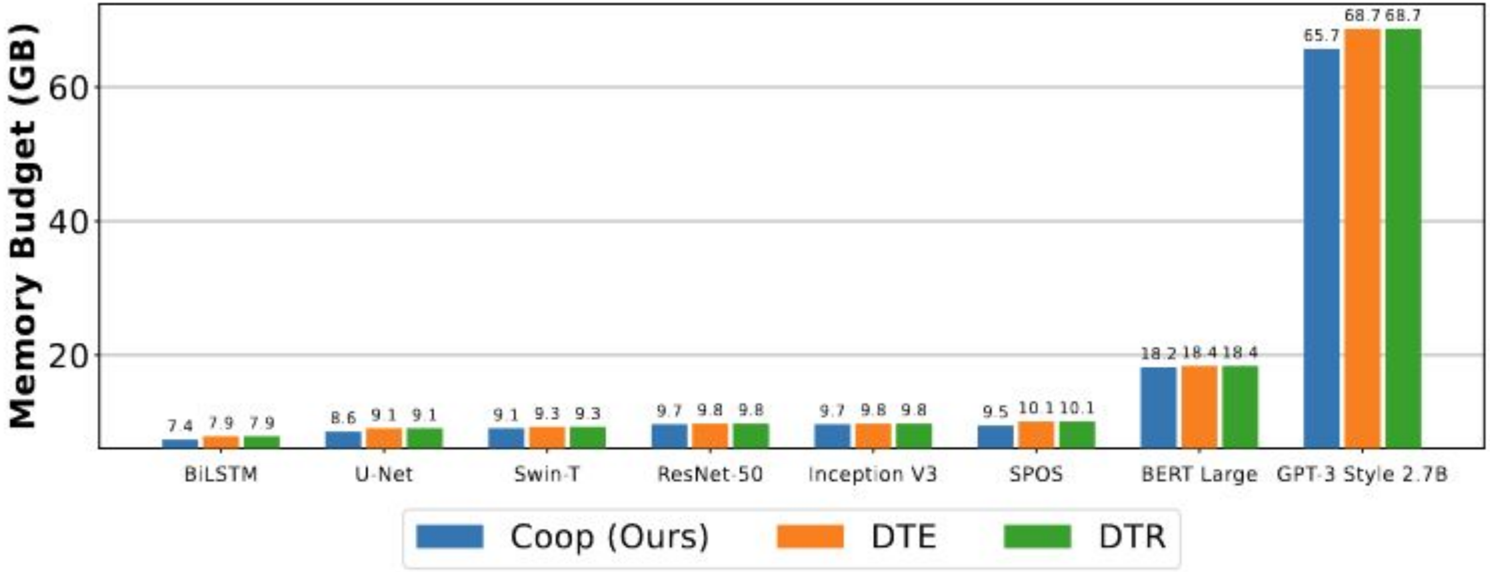
Results - Search Latency



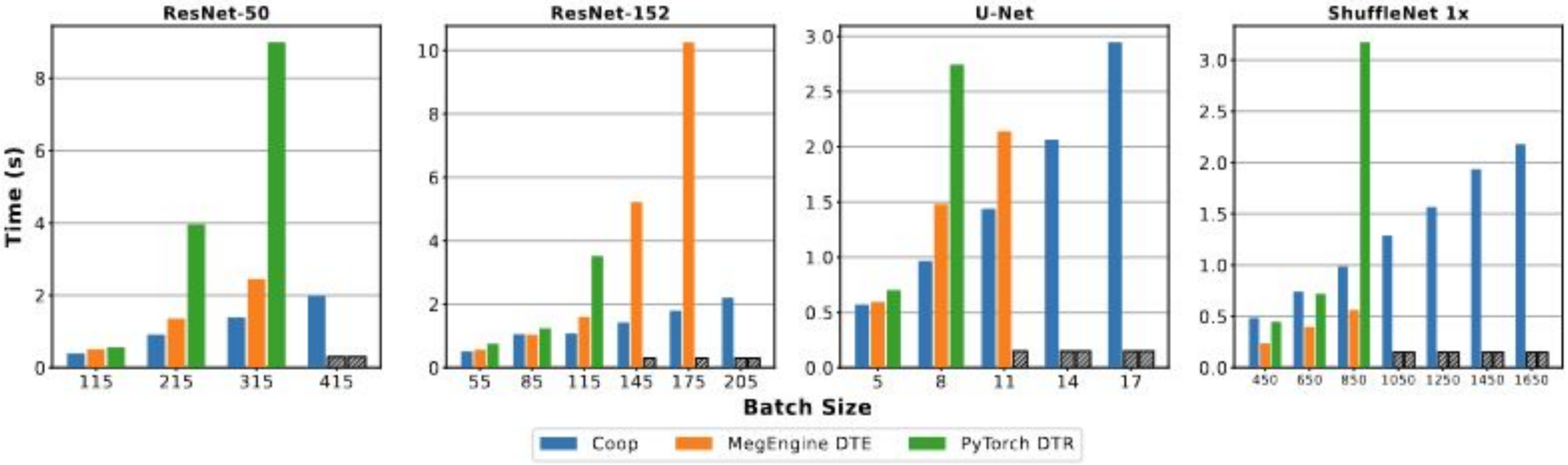
Results - Memory Fragmentation



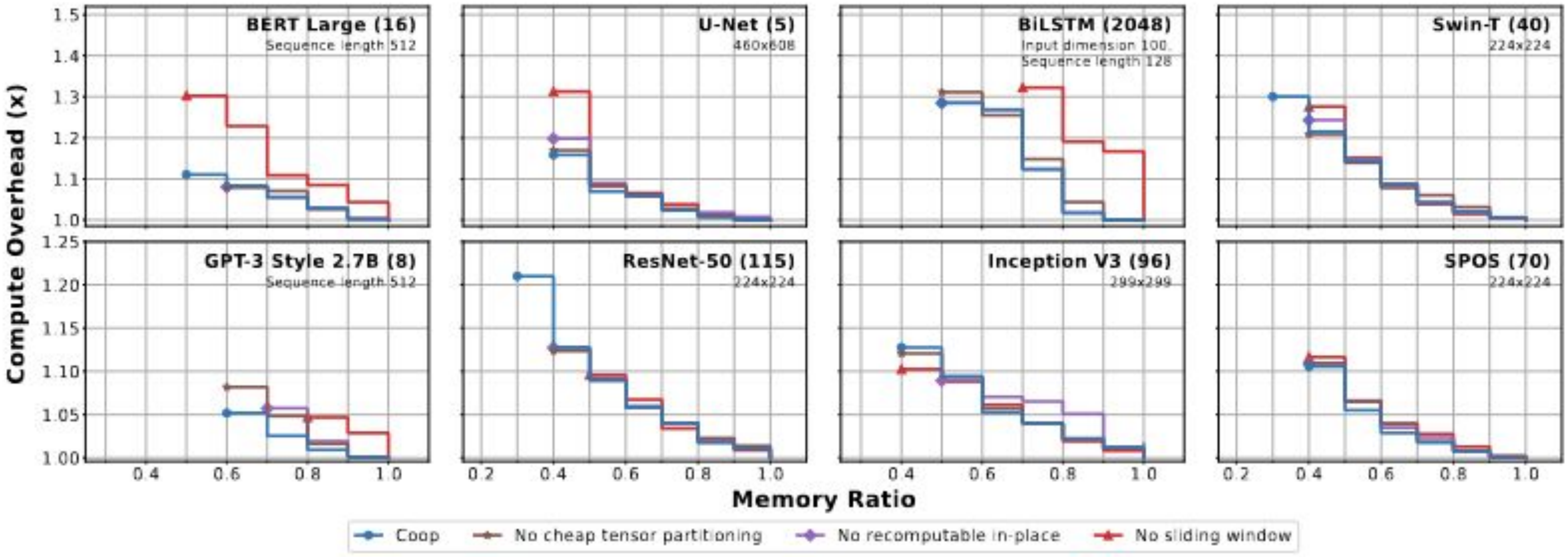
Results - Cutoff Memory



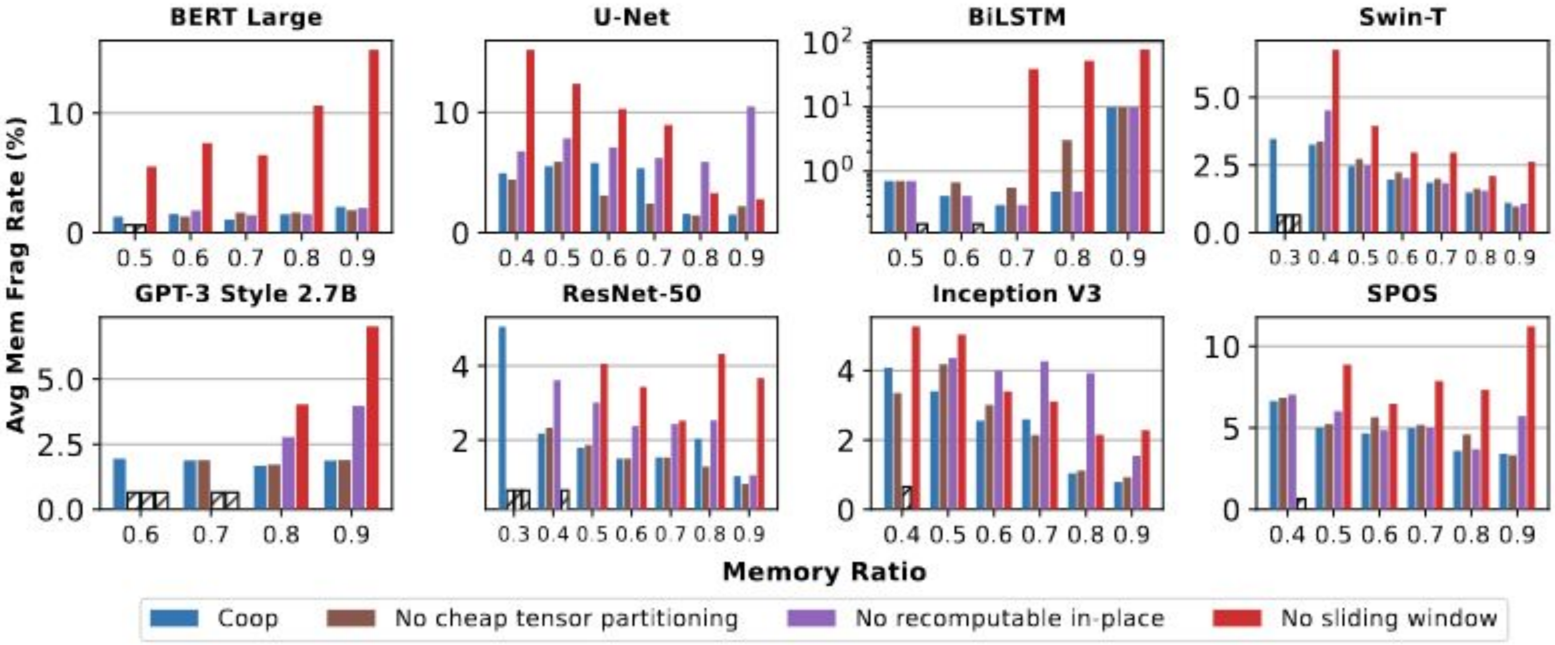
Comparison w/ official DTR and DTE implementations



Ablations - Compute Overhead



Ablations - Memory Fragmentation



Thoughts

- Strengths
  - Provides a solution to a major issue with DTR: Memory Fragmentation
- Weaknesses
  - Comparison against MegTaiChi paper
    - Authors compare against DTE but not DTE + TMA
      - TMA was important for reducing memory fragments due to fine-grained memory allocation
    - No mention of single-GPU vs multiple GPU
  - Only compared against two dynamic models
  - Not very unique
- Future work
  - Explore how their method would interact with tensor partitioning

Questions?

References

Gradient Checkpointing original paper: https://arxiv.org/abs/1604.06174

Gradient Checkpointing helpful explanation: https://medium.com/tensorflow/fitting-larger-networks-into-memory-583e3c758ff9

Gradient Checkpointing explanation video by Tianqi Chen: https://www.youtube.com/watch?v=t6NbNp9dfJQ

Associated slides: https://drive.google.com/file/d/1YOWKaCfilzSjukavNmpTEsuyMBeoahKj/view

Checkmate original paper: https://arxiv.org/abs/1910.02653

Dynamic Tensor Rematerialization original paper: https://arxiv.org/abs/2006.09616

Dynamic Tensor Rematerialization explanation video by one of the authors, Steven Lyubomirsky:: https://www.youtube.com/watch?v=S9KJ37Sx2XY

MegTaiChi original paper: https://dl.acm.org/doi/10.1145/3524059.3532394

Coop original paper: https://arxiv.org/abs/2311.00591

Coop poster presentation: https://neurips.cc/virtual/2023/poster/70826