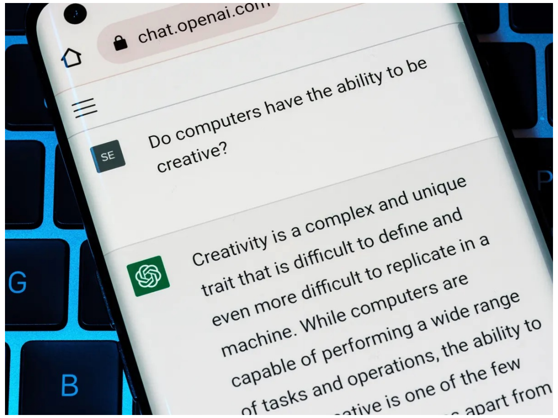
Zero-TPrune: Zero-Shot Token Pruning through Leveraging of the Attention Graph in Pre-trained Transformers

Presenter: Hongjie Wang
Advisor: Prof. Niraj K. Jha
Department of Electrical & Computer Engineering
Princeton University

PRINCETON UNIVERSITY

The Era of Artificial Intelligence

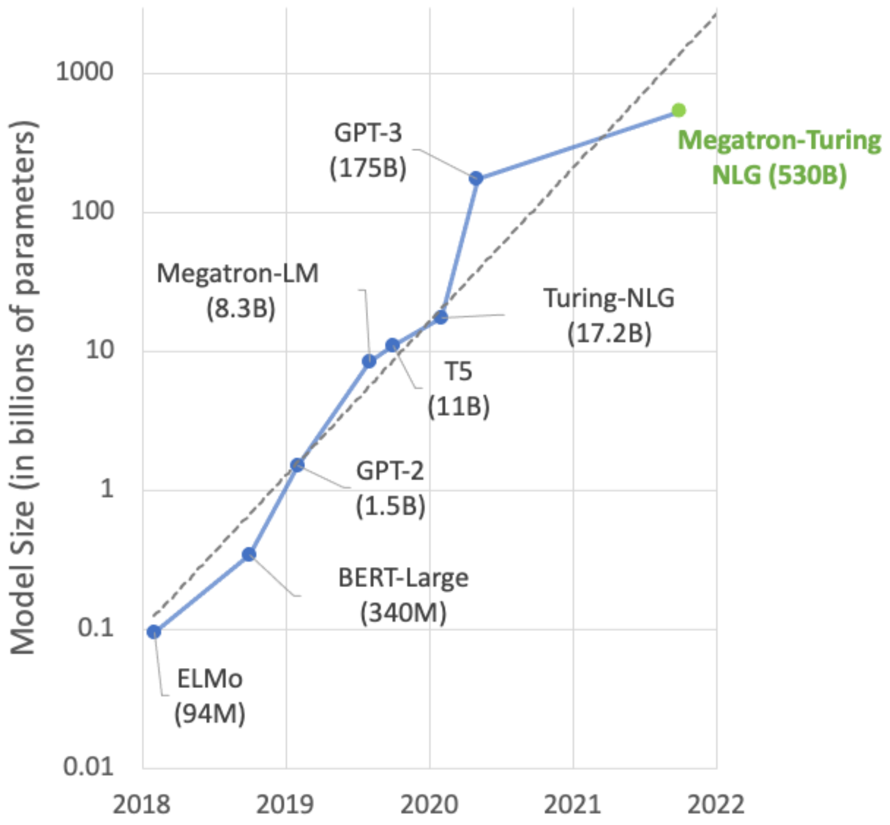


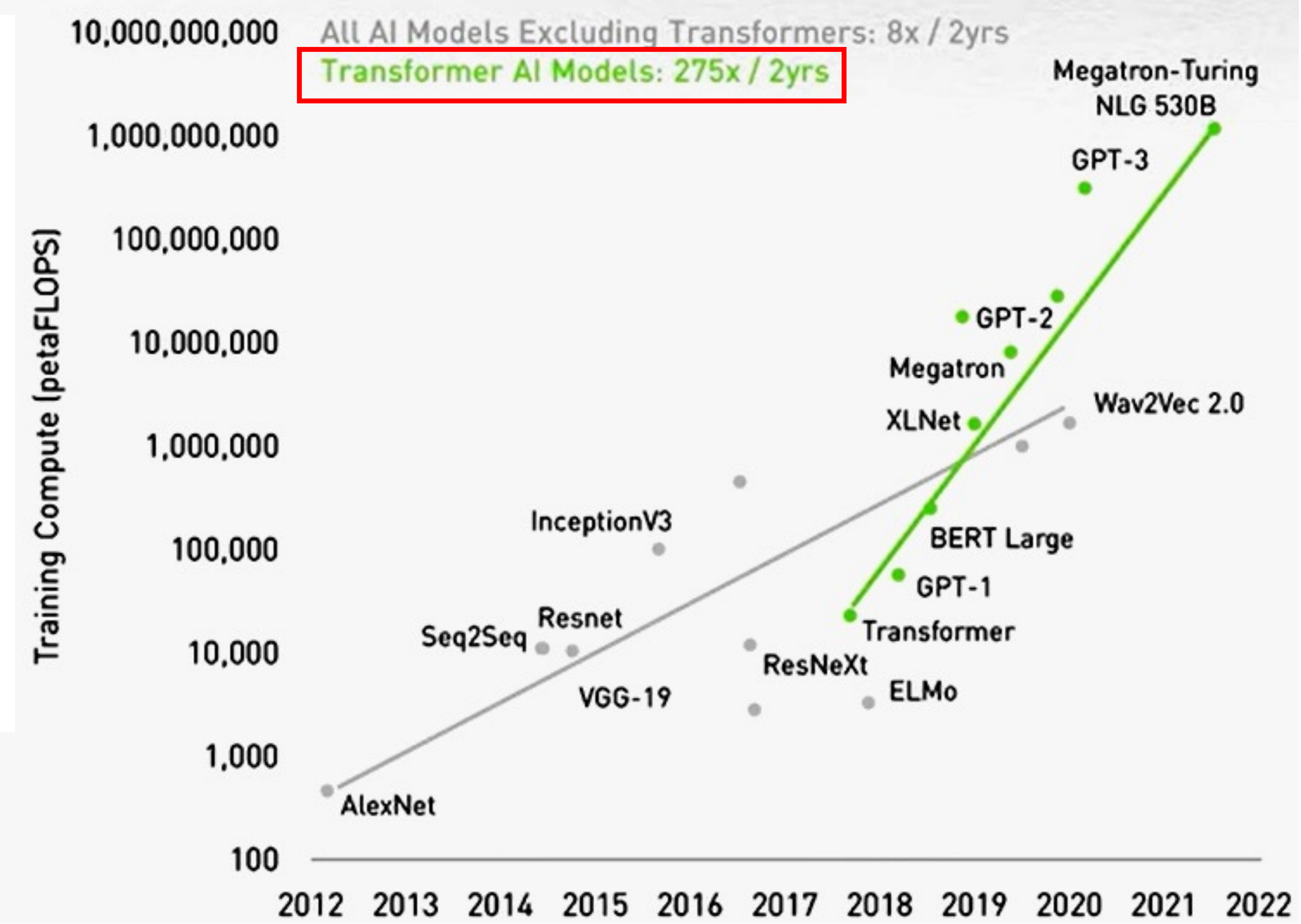
Text to text:
ChatGPT, Bard, Jasper …

Text/Image to image:
DALL-E, DeepAI, MidJourney …

Text to speech:
VITS, Genny, Diffsinger…

Increasing Size of Transformers





https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/

Challenge: Expensive Inference with Transformers

- Example: ChatGPT
- Inference: At least 8 Nvidia Tesla A100 GPUs needed (~$20,000/GPU)
- Electricity usage: $0.01-0.1 per query, $1-3 million in its first five days when opened to public

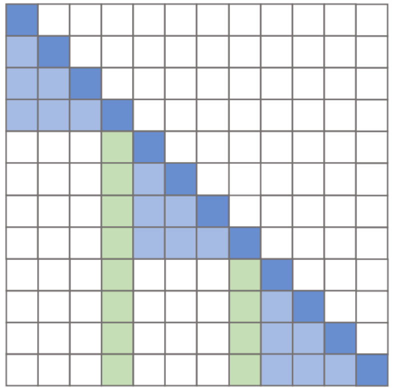
We want to prune Transformers!

Estimated by Nathan Baschez @ Twitter

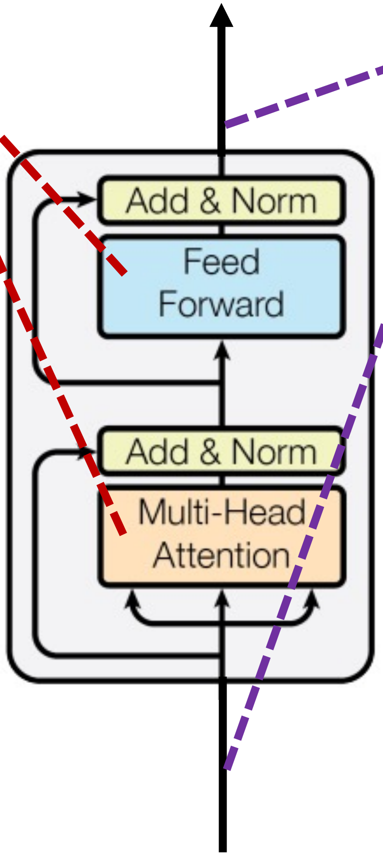
Transformer Pruning Methods

Exploit Structure Sparsity

E.g., Sparse Attention



- High pruning rate
- Hard to be fully utilized by hardware
- Dedicated design



Child et al., *arXiv*, 2019
Yin et al., *CVPR*, 2022

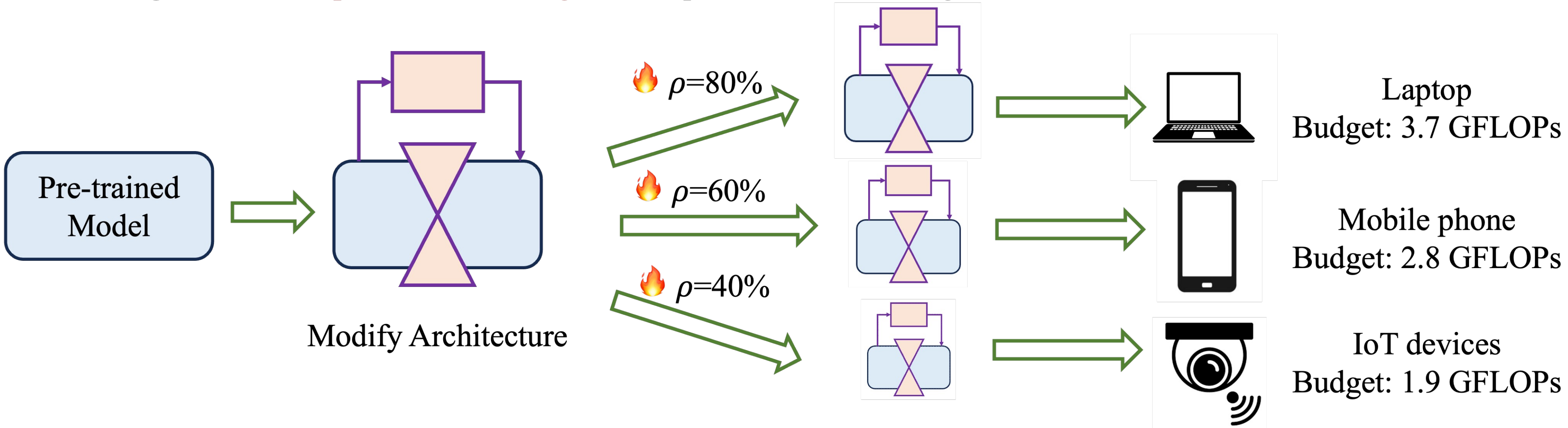
Exploit Feature Sparsity

E.g., Token Pruning

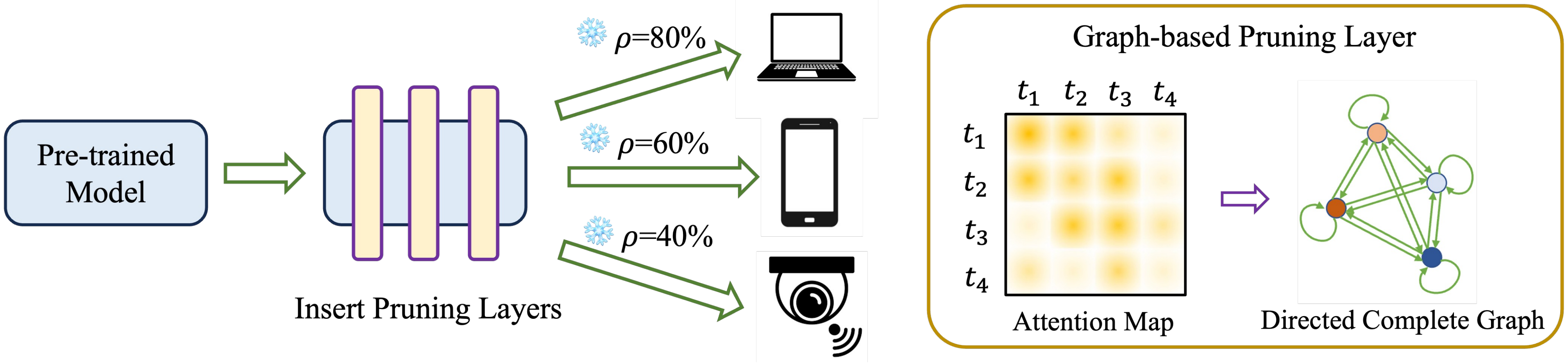
- Medium pruning rate
- Easy to be fully utilized by hardware
- Universal for various backbones

Pruning Requires Re-Training

Most existing methods: expensive re-training 🔥 is required for each configuration

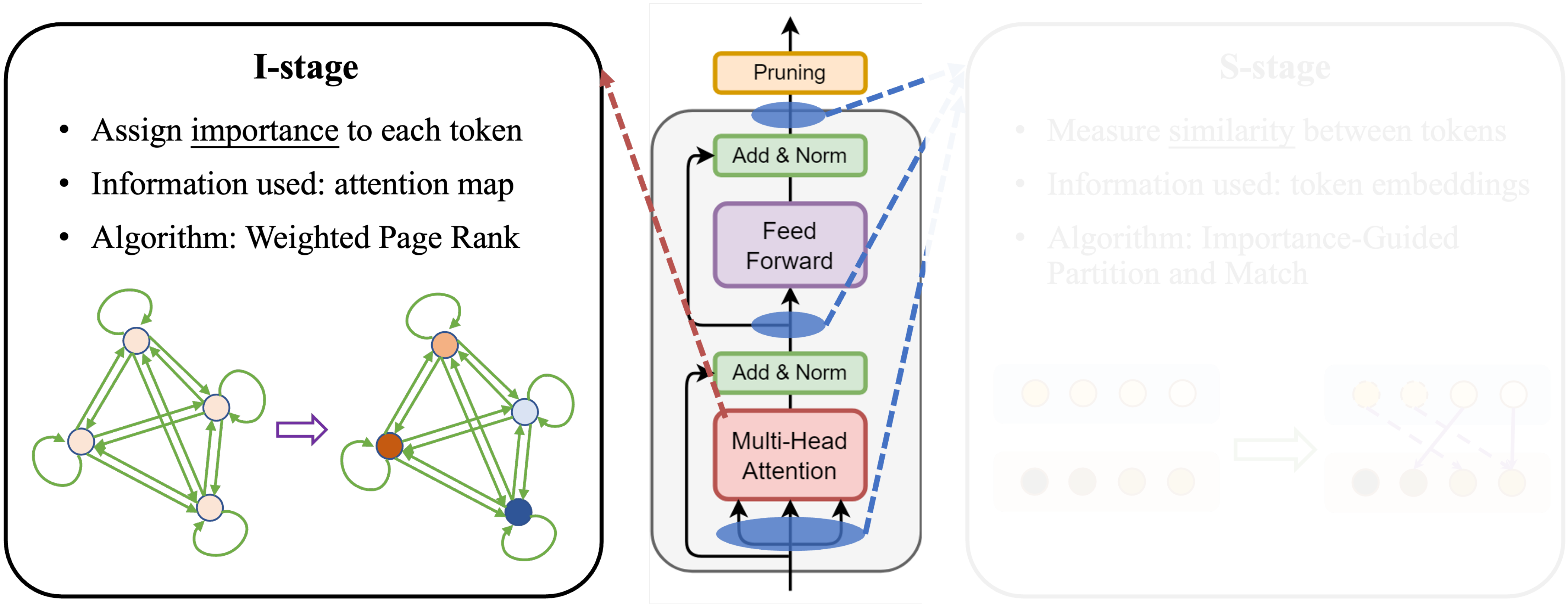


Our method: pruning deployment is training-free and can switch between different configurations at no computational cost ❄️



Overview

- Our methodology the first to consider both importance and similarity of tokens in performing token pruning

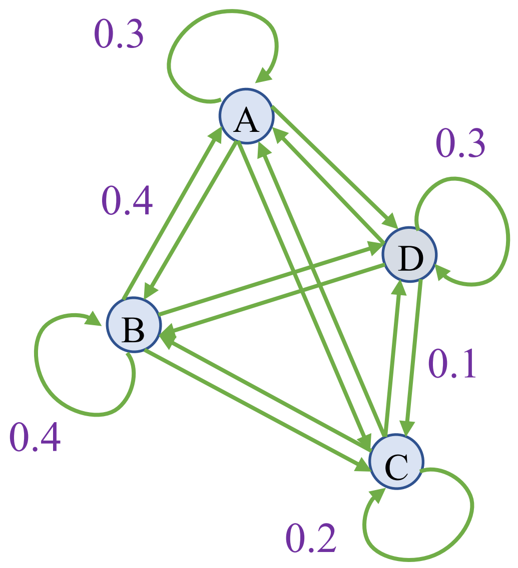


Closer Look at the Transformer Block

- For each head, the attention probability between tokens: elements in matrix $A^{(h,l)}$
- $A^{(h,l)}$: adjacency matrix of a complete, weighted, directed graph with hundreds of nodes

$$\text{Query} \longrightarrow$$

$$\text{Key} \Big\downarrow \quad \begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{pmatrix} 0.3 & 0.4 & 0.2 & 0.3 \\ 0.4 & 0.4 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.3 \end{pmatrix} \end{array} =$$

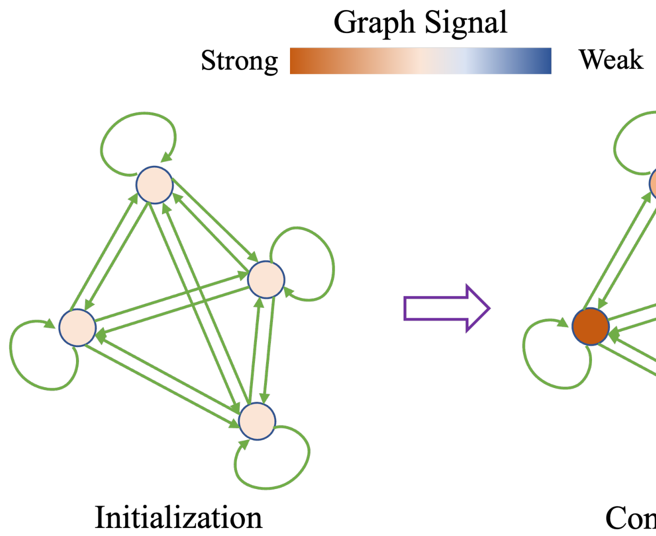


Utilize the information in this graph to select unimportant tokens (nodes)!

Weighted Page Rank (WPR) Algorithm

- Vanilla Page Rank algorithm: links between web pages unweighted
- Consider adjacency matrix $A^{(h,l)}$ as a graph operator, and apply it to the uniformly initialized graph signal iteratively until convergence

$$s^{(l)}(\mathrm{x}_i) = \frac{1}{N_h}\frac{1}{n}\sum_{h=1}^{N_h}\sum_{j=1}^{n}\mathbf{A}^{(h,l)}(\mathrm{x}_i,\mathrm{x}_j)\cdot \boldsymbol{s}^{(l)}(\boldsymbol{x}_j)$$



Require: $N > 0$ is the number of nodes in the graph; $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of this graph; $s \in \mathbb{R}^N$ represents the graph signal

Ensure: $s \in \mathbb{R}^N$ represents the importance score of nodes in the graph

$s^0 \leftarrow \frac{1}{N} \times e_N$ ▷ Initialize the graph signal uniformly

$t \leftarrow 0$

while $(|s^t - s^{t-1}| > \epsilon)$ **or** $(t = 0)$ **do** ▷ Continue iterating if not converged

  $t \leftarrow t + 1$

  $s^t \leftarrow A^T \times s^{t-1}$ ▷ Use the adjacency matrix as a graph shift operator

end while

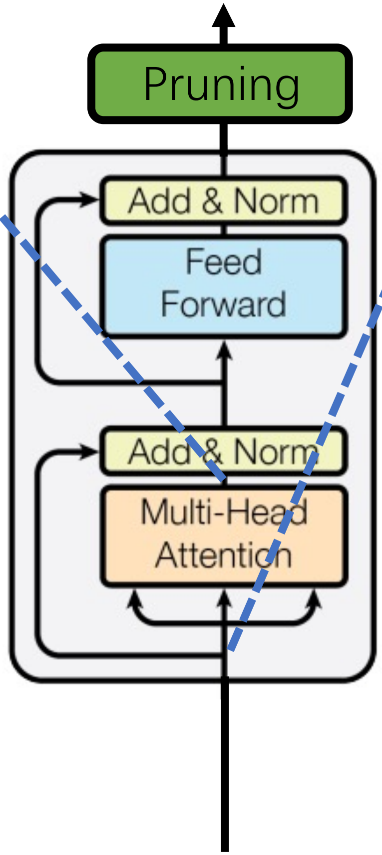
$s \leftarrow s^t$

Overview

- Our methodology the first to consider both importance and similarity of tokens in performing token pruning

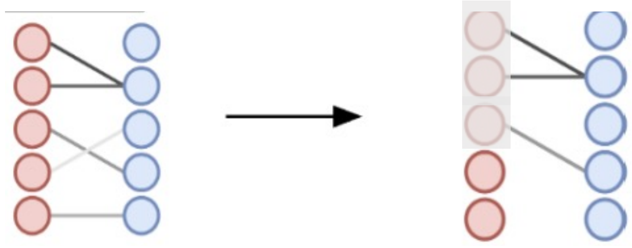
I-stage

- Assign importance to each token
- Information source: attention matrix
- Algorithm: Weighted Page Rank

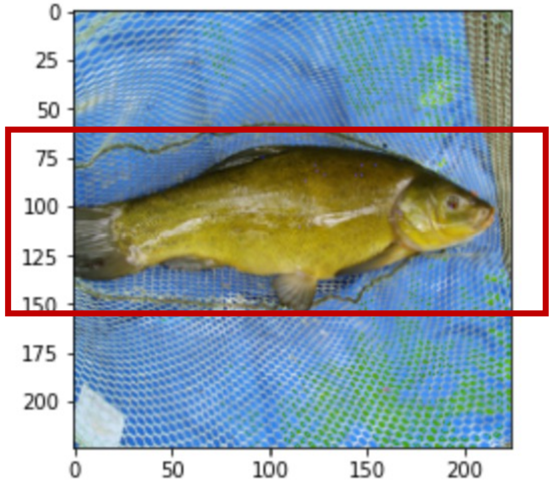


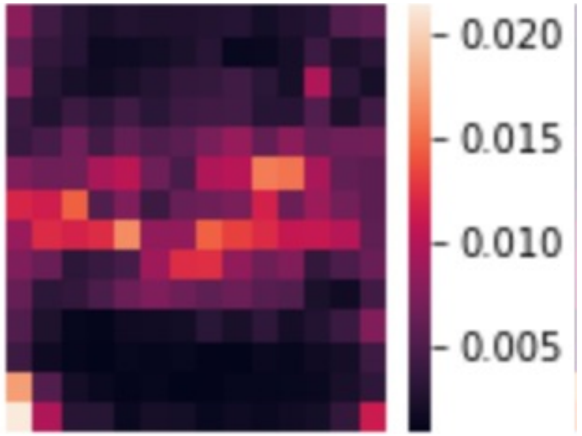
S-stage

- Measure similarity between tokens
- Information source: token embedding vectors
- Algorithm: Importance-guided group matching

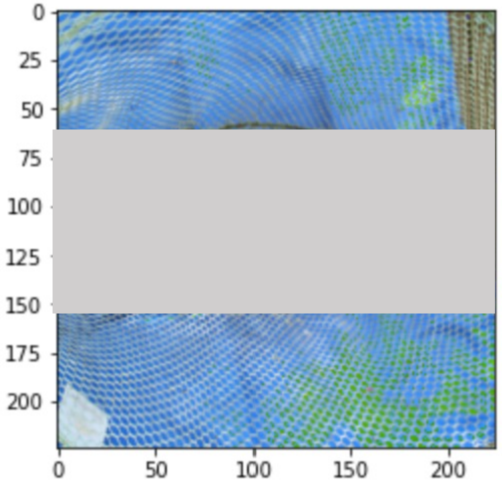


Are Important Tokens Really Necessary?

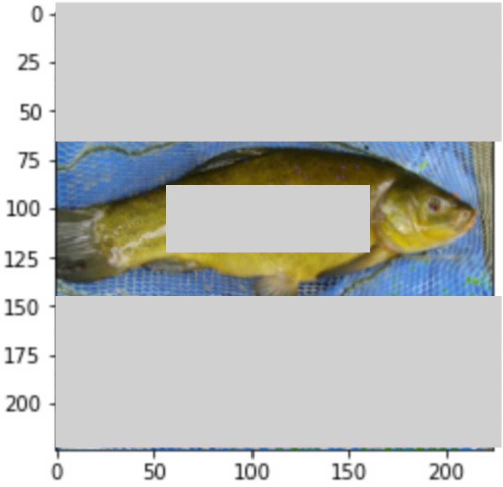




A fish!



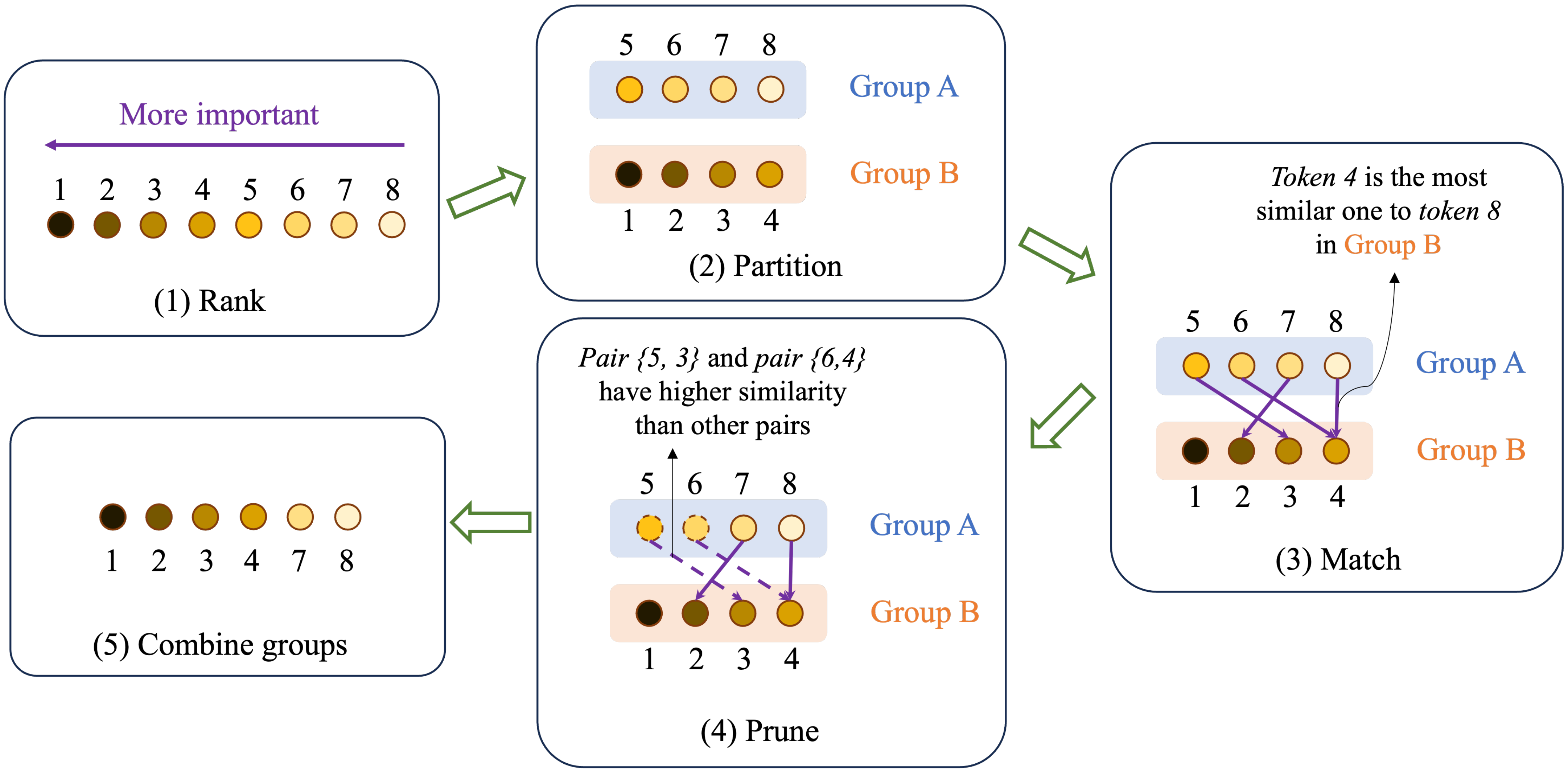
???



? A fish!

Once some important tokens are selected,
some other important tokens are no longer necessary!

Importance-Guided Group Matching



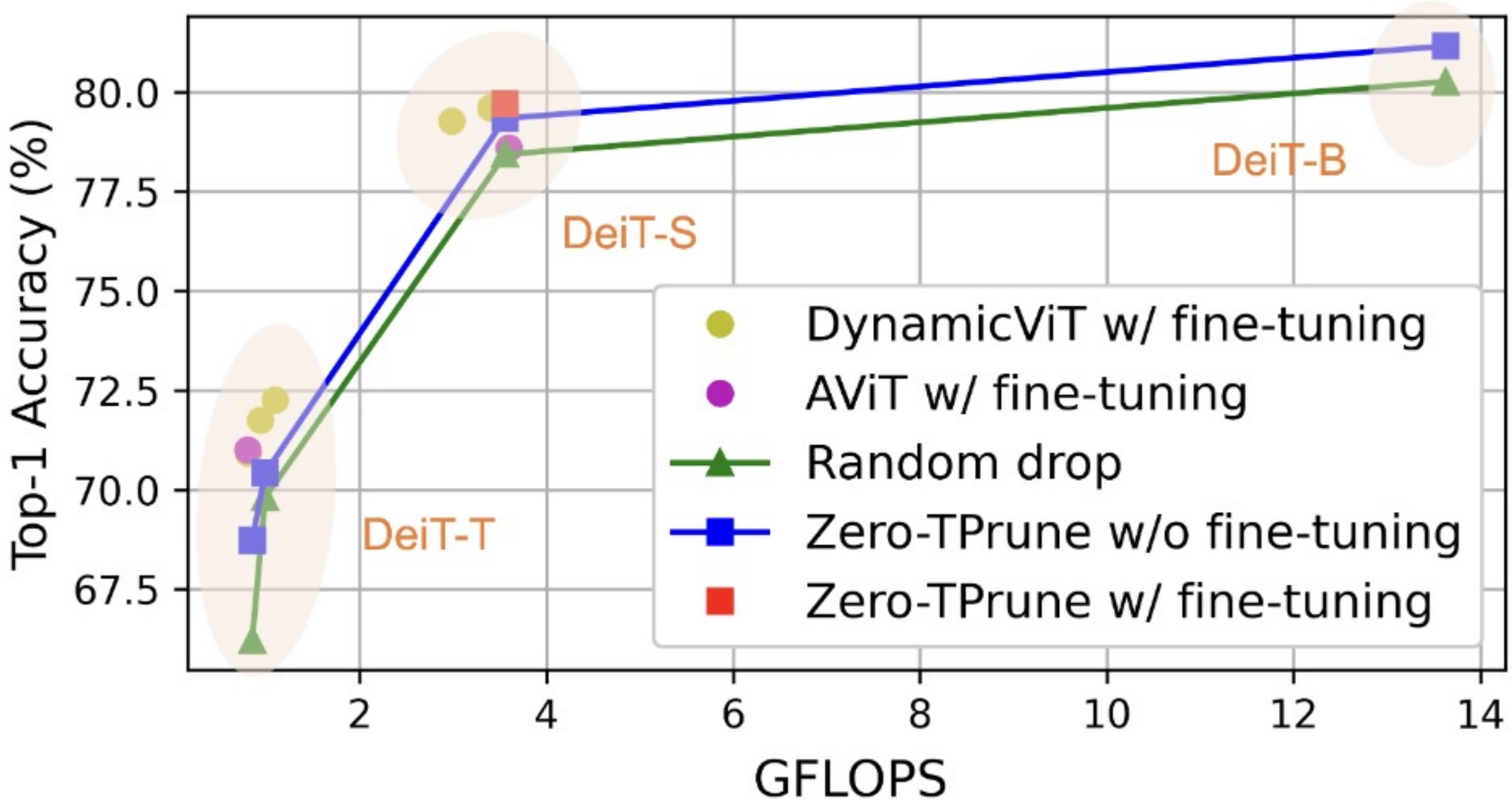
Visual Examples

Comparison Experiment Setup

- Pre-trained Transformer backbones:
  - DeiT [1], MAE [2], AugReg [3], SWAG [4], LV-ViT[5], T2T-ViT[6]
- Task: Image classification
- Dataset: ImageNet, 224px images (if not specified)
- Baselines:
  - Fine-tuning required methods: DynamicViT [7], A-ViT [8]
  - Off-the-shelf methods: ATS [9], ToMe [10]

[1] Touvron et al., *ICML*, 2021
[2] He et al., *CVPR,* 2022
[3] Steiner et al., *TMLR,* 2022
[4] Singh et al., *CVPR*, 2022
[5] Jiang et al., *NeurIPS*, 2021
[6] Yuan et al., *ICCV,* 2021
[7]Rao et al., *NeurIPS,* 2021
[8] Yin et al., *CVPR*, 2022
[9] Fayyaz et al., *ECCV,* 2022
[10] Bolya et al., *ICML,* 2023

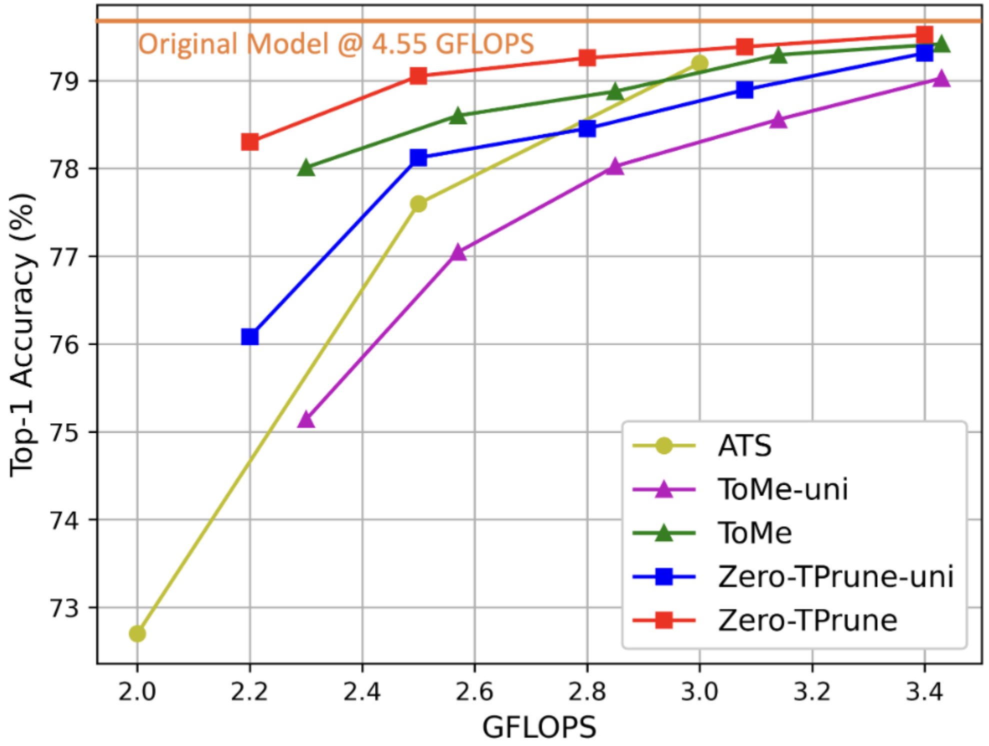
Comparisons with Methods that Require Fine-tuning



Comparisons with Off-the-shelf Methods: DeiT-S

- Compared to state-of-the-art, Zero-TPrune reduces accuracy loss by 33%

Method	Acc@top1	GFLOPS	Throughput(img/s)
DeiT-S	79.8%	4.55	1505.9
+ ATS	79.2% (-0.6%)	3.00 (-33.4%)	2062.3 (+36.9%)
+ ToMe	78.9% (-0.9%)	2.95 (-35.2%)	2263.9 (+50.3%)
+ Zero-TP-a	**79.4% (-0.4%)**	2.97 (-34.7%)	2188.4 (+45.3%)
+ Zero-TP-b	**79.1% (-0.7%)**	2.50 (-45.1%)	2458.4 (+63.2%)
+ Zero-TP-c	**79.8% (-0.0%)**	3.97 (-12.7%)	1673.2 (+11.1%)



Comparisons with Off-the-shelf Methods: Medium Models

Method	Acc@top1	GFLOPS	Method	Acc@top1	GFLOPS
AugReg	81.41%	4.55	MAE	83.62%	55.4
+ ATS	79.21%	2.80	+ATS	82.07%	42.3
+ ToMe	79.30%	2.78	+ToMe	82.69%	42.2
+ Zero-TP	**80.22%**	**2.79**	+Zero-TP	**82.93%**	42.3
LV-ViT-S	83.3%	6.6	SWAG	85.30%	55.6
+ ATS	80.4%	3.5	+ATS	84.21%	43.8
+ ToMe	79.8%	3.6	+ToMe	85.09%	43.8
+ Zero-TP	**81.5%**	**3.5**	+Zero-TP	**85.17%**	43.8

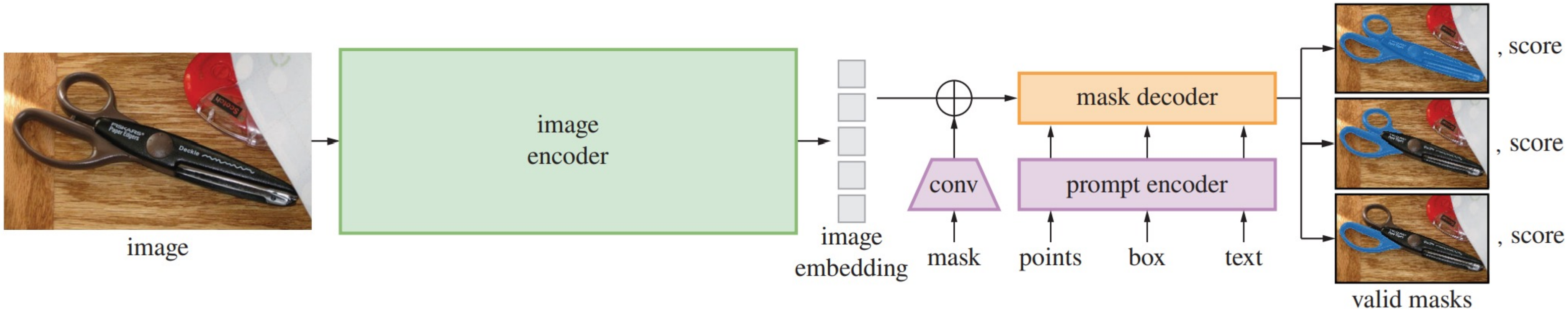
* WAG models perform inference on 384px images

Conclusions

- Zero-TPrune: the first zero-shot token pruning method that exploits both the importance and similarity of tokens
- Attention matrix → attention graph: Weighted Page Rank reduces noise from unimportant tokens during importance assignment
- Guided by importance: similarity-based matching and pruning are more precise
- Zero-TPrune can increase the throughput of off-the-shelf pre-trained Transformers by 45% with only 0.4% accuracy loss
- Compared with state-of-the-art methods, Zero-TPrune reduces accuracy loss by more than 30%

Future Work

- The prevailing "pretraining → downstream tasks" pattern naturally offers the potential to perform zero-shot pruning
- More tasks: segmentation, reconstruction, detection



Segment Anything (SAM)

Kirillov et al., *arXiv,* 2023

Future Work

- The prevailing "pretraining → downstream tasks" pattern naturally offers the potential to perform zero-shot pruning
- More tasks: segmentation, reconstruction, generation
- More architectures: diffusion models