ENERGY MINIMIZING-BASED TOKEN MERGING FOR ACCELERATING TRANSFORMERS

Hoai-Chau Tran *
University of Science - VNUHCM
thchau18@apcs.fitus.edu.vn

Duy M. H. Nguyen
University of Stuttgart
IMPRS for Intelligent Systems
German Research Center for Artificial Intelligence
Ho_Minh_Duy.Nguyen@dfki.de

Manh-Duy Nguyen
School of Computing, Dublin City University, Ireland
manh.nguyen5@mail.dcu.ie

Ngan Le
University of Arkansas
thile@uark.edu

Binh T. Nguyen
University of Science - VNUHCM
ngtbinh@hcmus.edu.vn

ABSTRACT

Model compression has been an active research field to reduce the size and complexity of the model. In a recent noteworthy study, ToMe and its variants utilize the Bipartite Soft Matching (BSM) algorithm in which tokens representing patches in an image are split into two sets, and top k similar tokens from one set are merged. This approach not only utilizes pretrained weights but also enhances speed and reduces memory usage. However, this algorithm has some drawbacks. The choice of a token-splitting strategy significantly influences the algorithm's performance since tokens in one set can only perceive tokens in the other set, leading to mismerging issues. Furthermore, although ToMe is effective in the initial layers, it becomes increasingly problematic in deeper layers as the number of tokens diminishes because of damaged informative tokens. To address these limitations, rather than relying on specific splitting strategies like BSM, we propose a new algorithm called PiToMe. Specifically, we prioritize the protection of informative tokens using an additional factor called the *energy score*. In experiments, PiToMe achieved up to a 50% memory reduction while exhibiting superior off-the-shelf performance on image classification (keeping 1.71% average performance drop compared to 2.6% for ToMe) and image-text retrieval (1.35% average performance drop compared to 6.89% for ToMe) compared to ToMe and ToMe-based approaches dependent solely on token similarity.

1 INTRODUCTION

Vision Transformers (ViTs) Dosovitskiy et al. (2020) have contributed to recent advancements in computer vision, enhancing the way we use deep learning models to represent images and videos. However, these transformer-based architectures often come with substantial memory requirements and high time complexity, especially as models grow larger. While there have been attempts to design new architectures (Dong et al., 2021; Yin et al., 2022; Rao et al., 2021; Liu et al., 2021; Zhou et al., 2022) to address these issues, the primary drawback of these new architectures is the need to retrain the model from scratch. Thus, it is imperative to find a solution that makes models faster and lighter without compromising the performance of pre-trained models.

In recent years, researchers have explored a novel research direction aimed at directly pruning or merging tokens (i.e., patches) passed into each layer of transformer encoders. This approach can

*Main author, Corresponding email: thchau18@apcs.fitus.edu.vn

leverage the pre-trained weights of the model, effectively reducing both the speed and memory footprint. A notable recent work is ToMe (Bolya et al., 2023) which introduced the Bipartite Soft Matching (BSM) algorithm, which is simple yet effective in merging tokens with high similarity. Since ToMe, several works have emerged, relying on the BSM algorithm to develop their variants (Cao et al., 2023; Bolya & Hoffman, 2023; Bonnaerens & Dambre, 2023; Chen et al., 2023); however, they only tested on small models, and the performance improvement is marginal. In this paper, we refer to them as BSM-based approaches.

ToMe and BSM-based approaches rely on the BSM algorithm, wherein all tokens representing patches in an image are divided into two sets, $\mathcal{A}$ and $\mathcal{B}$. Tokens in set $\mathcal{A}$ are then compared to those in set $\mathcal{B}$ using cosine similarity, and k tokens in set $\mathcal{A}$ with the highest similarity are selected for merging. However, this approach has some drawbacks:

- Firstly, the choice of a tokens-splitting strategy highly affects the performance of the algorithm. In the ToMe paper, the author chose to split based on odd and even indices. However, cases of mis-merging are still inevitable since tokens in set $\mathcal{A}$ can only perceive tokens in $\mathcal{B}$ but not themselves.
- Secondly, while the BSM algorithm works effectively in the initial layers where redundant tokens for backgrounds and noise are abundant, as tokens go deeper into the network, there is a risk of compromising informative tokens that represent the main object because of their high similarity

To address these limitations, we propose a new algorithm, named *PiToMe* to Protect Informative Tokens before Merging. Our approach prioritizes the protection of informative tokens using an additional criterion called *energy score*, in contrast to relying on specific splitting strategies as in BSM. In all experiments on two tasks, image classification, and image-text retrieval, using both large and small backbones models, our method demonstrates superior off-the-shelf performance compared to previous BSM-based approaches that depend solely on token similarity. Additional algorithm details are provided in Section 2.

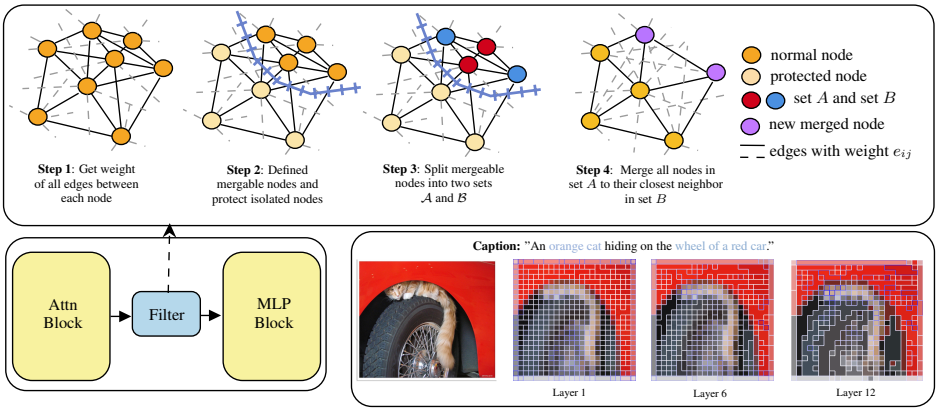


Figure 1: The core idea of PiToMe revolves around graph cutting, ensuring the preservation of important and informative nodes (i.e., patches) while selectively merging unimportant nodes, such as backgrounds and noises. For instance, consider an image showing a cat hiding behind the wheel of a red car. Our method can preserve critical nodes that characterize the cat and merge background and redundant tokens. Patches highlighted with a bolder blue border signify a higher attention score from the classification token.

2 METHODOLOGY

Inspired by the Graph Cut algorithm for image segmentation tasks, our methodology treats individual patches as nodes in a fully connected graph. Our goal is to efficiently identify and separate nodes using their *energy scores* and protect them from being merged. This helps protect informative/isolated tokens (low energy scores) while looking for tokens that are clustered together (high

energy scores) and considers merging them. As illustrated in Figure. 1, our approach include 4 main steps.

Step 1: We first obtain the nodes and compute edges for the input graphs. In our implementation, we opted for using the *key* vectors within the attention block to represent nodes $v_i \in \mathcal{V}$. The weight assigned to each edge is subsequently determined through cosine similarity:

$$\mathcal{E} = \frac{(\mathcal{V} \cdot \mathcal{V}^T)}{\|\mathcal{V}\|^2} \tag{1}$$

Let t be the number of nodes and h be the hidden size of $v_i \in \mathcal{V}$. The final output is the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \in \mathbb{R}^{t \times h}$ represent nodes and $\mathcal{E}$ is an adjacency matrix containing weights for all edges. The reason we chose *key* vectors as node representations is to align with the methodology in previous papers (Bolya et al., 2023; Bolya & Hoffman, 2023; Bonnaerens & Dambre, 2023; Chen et al., 2023), allowing for a direct comparison with the bipartite soft matching algorithm used by BSM-based approaches.

Step 2: In this steps the *energy score* for each node is calculated. Let i be the index of the current node and $\mathcal{N}(i)$ represent the set of neighbor nodes. The energy score s_i of node v_i is calculated using the following equation:

$$s_i = E(\mathbf{v}_i, \mathbf{e}_i) = -\frac{1}{t} \sum_{j \in \mathcal{N}(i)} f(e_{ij}) \text{ with } f(x) = \begin{cases} x & \text{if} x \geq m \\ \alpha(\exp(x-m)-1) & \text{otherwise} \end{cases} \tag{2}$$

In equation 2, and e_{ij} is the edge weight connecting node i to node j. Here, instead of summing all e_{ij}, the function $f(.)$ serves as a normalization tool, selectively considering nearby neighbors and discarding the influence of distant clusters on merging decisions. Here, m is a fixed constant representing the margin for each node. Nodes within this margin with high edge weight e_{ij} are considered true neighbors, while nodes outside this margin have e_{ij} replaced by a constant α, ensuring a lower bound for edges with minimal weights. The term $\exp(x-m)-1$ smooths the function $f(x)$ for neighboring nodes with e_{ij} proximity to the margin m. In experiments, we set $\alpha = 1.0$ and $m = 0.9 - 0.9 \times l_i/l$, where l_i is the current layer index and l is the number of encoder layers, indicating a growing margin as tokens move to deeper layers. Energy scores are estimated and sorted, and the top $2k$ nodes with the highest energy scores are selected for merging.

Step 3 & 4: Having identified mergeable tokens, we partition them into two sets, denoted as $\mathcal{A}$ and $\mathcal{B}$, each containing k nodes. All nodes in set $\mathcal{A}$ are merged with their nearest neighbors in set $\mathcal{B}$ through a weighted average procedure based on their energy scores.

Algorithm 1 PiToMe Algorithm

function PITOME(reduce ratio: r, input graph: $\mathcal{G}(\mathcal{V}, \mathcal{E})$) ▷ Function to prepare for merging
  $k \leftarrow t - t \cdot r$ ▷ Compute number of nodes to merge
  $\mathbf{s} \leftarrow \text{argsort}(E(\mathbf{v}_i, \mathbf{e}_i), \text{descending=True})$ ▷ Compute energy scores
  $\mathbf{n}_{protected} \leftarrow \mathbf{s}[:2 \cdot k], \mathbf{s}[2 \cdot k:]$ ▷ Identify mergeable and protected nodes
  $\mathbf{n}_a, \mathbf{n}_b \leftarrow \mathbf{n}_{merge}[:k], \mathbf{n}_{merge}[k:]$ ▷ Split mergeable nodes
  $\mathcal{E}_{merge} \leftarrow \mathcal{E}[\mathbf{n}_a][\mathbf{n}_b]$ ▷ get edge weights of mergeable nodes
  $\mathbf{n}_{dst} \leftarrow \text{argmax}(\mathcal{E}_{merge})$ ▷ Find closest neighbors
  function MERGE($\mathbf{X}$) ▷ Function to perform merging
    $\mathbf{X}_{protected} \leftarrow \mathbf{X}[\mathbf{n}_{protected}, :]$ ▷ Extract protected tokens
    $\mathbf{X}_{\mathcal{A}}, \mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}[\mathbf{n}_a, :], \mathbf{X}[\mathbf{n}_b, :]$ ▷ Extract tokens in set $\mathcal{A}$ and $\mathcal{B}$
    $\mathbf{X}_{\mathcal{A}}, \mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{A}} \times (1 - \mathbf{s}[\mathbf{n}_a]), \mathbf{X}_{\mathcal{B}} \times (1 - \mathbf{s}[\mathbf{n}_b])$ ▷ Weighted average
    $\mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{B}}.\text{scatter\_reduce}(\mathbf{n}_{dst}, \mathbf{X}_{\mathcal{A}}, \text{mode} = \text{"sum"})$ ▷ Merge tokens
    $\mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{B}} / \mathbf{s}_{\mathcal{B}}.\text{scatter\_reduce}(\mathbf{n}_{dst}, \mathbf{s}_{\mathcal{A}}, \text{mode} = \text{"sum"})$ ▷ Normalize merged tokens
    return $\text{cat}(\mathbf{X}_{protected}, \mathbf{X}_{\mathcal{B}})$ ▷ Concatenate and return merged tokens
  end function
  return MERGE ▷ Return merging lambda function
end function

The pseudo-code for our method is provided in algorithm 1. The final output is a $MERGE$ function which serves as a lambda function that can be applied to any matrix $\mathbf{X} \in \mathbb{R}^{t \times d}$ where t is the number of tokens in the current layer and d is the hidden size.

3 EXPERIMENTS

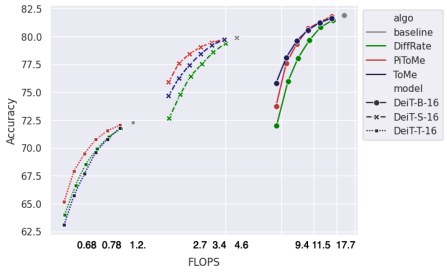
In our experiments, we focus on evaluating the off-the-shell performance (i.e. compressing the model and directly reusing weight without retraining) of our method across two different tasks: image classification and zero-shot image & text retrieval. Here the number of floating-point operations (FLOPS) that the model needed to perform inference for one sample is used as the main metric to benchmark the memory footprint as well as the speed of the model. Larger FLOPS mean the model requires higher memory and a longer time for training and inference.

3.1 IMAGE CLASSIFICATION

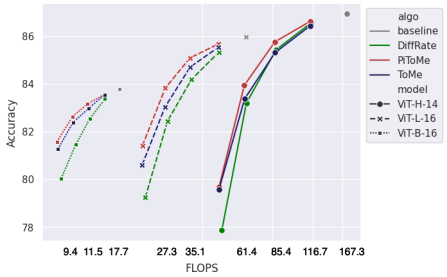
In this image classification experiment, we employed 5 ViT backbones of varying sizes—tiny (ViT-T), small (ViT-S), base (ViT-B), large (ViT-L), and huge (ViT-H) - which are pre-trained using either MAE (He et al., 2021) or DEIT (Touvron et al., 2021) styles. These backbones were utilized to assess both off-the-shelf and trained performance. All experiments were conducted on the ImageNet-1k dataset, which is a subset of ImageNet (Russakovsky et al., 2015) containing labeled images spanning 1000 categories.

Table 1: Comparison to recent SOTA models and other model compressing methods.

type	model	acc	FLOPS
	$\text{ViT}^{\text{MAE-B}}$	83.6	17.6
	$\text{ViT}^{\text{CLIP-B}}$	83.6	17.6
	Swin-B	84.0	15.4
	CSWin-B	84.2	15.0
	MViTv2-B	84.4	10.2
	MViTv2-L	85.3	42.1
merge	$\text{ToMe}^{\text{DEIT-T}}$	67.7	0.68
merge	$\text{PiToMe}^{\text{DEIT-T}}$	**69.5**	**0.68**
merge	$\text{ToMe}^{\text{DEIT-T}}$	68.9	0.79
merge	$\text{PiToMe}^{\text{DEIT-T}}$	**70.8**	**0.79**
	$\text{ViT}^{\text{DEIT-T}}$	72.3	1.2
prune	$\text{A-ViT}^{\text{DEIT-S}}$	78.6	2.9
prune	$\text{Dynamic-ViT}^{\text{DEIT-S}}$	79.3	2.9
prune	$\text{SP-ViT}^{\text{DEIT-S}}$	79.3	2.6
merge	$\text{ToMe}^{\text{DEIT-S}}$	78.1	2.7
merge	$\text{PiToMe}^{\text{DEIT-S}}$	**78.9**	**2.7**
merge	$\text{E-ViT}^{\text{DEIT-S}}$	79.5	2.9
merge	$\text{ToMe}^{\text{DEIT-S}}$	79.4	2.7
merge	$\text{PiToMe}^{\text{DEIT-S}}$	**79.7**	**2.7**
	$\text{ViT}^{\text{DEIT-S}}$	79.8	4.6
merge	$\text{ToMe}^{\text{MAE-L}}$	83.9	31.0
merge	$\text{PiToMe}^{\text{MAE-L}}$	**84.6**	**31.0**
merge	$\text{ToMe}^{\text{MAE-L}}$	85.0	31.0
merge	$\text{PiToMe}^{\text{MAE-L}}$	**85.2**	**31.0**
	$\text{ViT}^{\text{MAE-L}}$	85.7	61.6
merge	$\text{ToMe}^{\text{MAE-H}}$	85.9	92.11
merge	$\text{PiToMe}^{\text{MAE-H}}$	**86.3**	**92.11**
merge	$\text{ToMe}^{\text{MAE-H}}$	86.3	108.3
merge	$\text{PiToMe}^{\text{MAE-H}}$	**86.6**	**108.3**
	$\text{ViT}^{\text{MAE-H}}$	86.9	167.4



(a) DEIT models



(b) MAE models

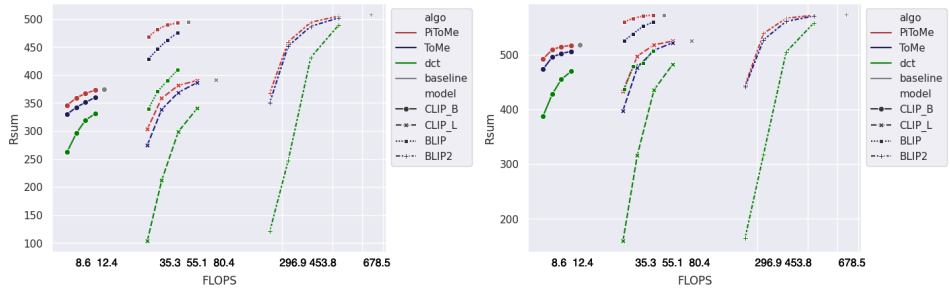
Figure 2: Off-the-shell performance of ViT backbones on the ImageNet dataset. Here, the FLOPS axis has been divided by 10^9 and rescaled using a logarithmic scale for better visualization. Apart from ToMe (Bolya et al., 2023) we also included an additional BSM-based approach which is DiffRate (Chen et al., 2023).

Table 1 showcases our experimental results, comparing our approaches with previous works, including recent SOTA models (Dong et al., 2021; Liu et al., 2021; He et al., 2021; Li et al., 2022b), and other token merging/pruning methods (Bolya et al., 2023; Rao et al., 2021; Yin et al., 2022; Zhou et al., 2022). Models with blue background are used off-the-shelf without training, while gray indicates models retrained from scratch. Off-the-shelf results, illustrated in figure 2, demonstrate that our method maintains high accuracy (1.35% average performance drop) after reducing up to 50% of FLOPS, showcasing superior performance with comparable throughput. In table **??**, after

retraining, models that are compressed by PiToMe outperformed previous merging/pruning methods by a large margin and reached close to the performance of original baseline models.

3.2 IMAGE-TEXT RETRIEVAL

In this image-text retrieval experiment, we evaluate our algorithm using three distinct backbones: CLIP (Radford et al., 2021), BLIP (Li et al., 2022a), and BLIP2 (Li et al., 2023). CLIP includes two ViT sizes, base (CLIP-B) and large (CLIP-L), with 12 and 24 layers, respectively. The BLIP model utilizes the ViT-B with 12 layers, while BLIP2 employs the ViT-G backbone with 48 layers. Flickr30k (Plummer et al., 2015) and MSCOCO (Lin et al., 2014) are used in this experiment, which is also frequently used in previous works for this task. The evaluation metric is based on recall@k, a widely employed metric in information retrieval and recommendation systems. Higher recall@k values indicate better performance, reflecting the model's effectiveness in retrieving relevant items. For further details on the training strategy, we refer readers to Li et al. (2022a).



(a) MSCOCO (b) Flickr30k

Figure 3: Off-the-shell performance for image-text retrieval task. Here, we evaluate the performance of compressed vison-languages models using three different algorithms, the first is our algorithm PiToMe, two others are the BSM algorithm used in ToMe (Bolya et al., 2023), and the DCT algorithm (He et al., 2023). The FLOPS axis is divided by 10^9 and rescaled using a logarithmic scale to enhance visualization. The Rsum metric represents the sum of $R@1 + R@5 + R@10$ for both image-to-text and text-to-image retrieval. The maximum score for Rsum is 600, indicating a perfect 100% recall score for all $R@k$.

From figure 3a, 3b, we can see that the results for off-the-shell models are consistent with the image classification task. Our algorithm proved a clear advantage in which we outperform other model compressing methods and recent SOTA models (Li et al., 2021; Chen et al., 2020; Varma et al., 2023; Sun et al., 2021) by a large margin for both image-to-text and text-to-image retrieval. This result is consistent for all backbones. Although retraining models is not necessary, we also included extra experiments in which we retrained all models from scratch in section A.1. To illustrate the effectiveness of our approach, more visualizations are also presented in Appendix A.3.

4 CONCLUSION

This paper introduces PiToMe, a new algorithm that utilizes graph cutting to protect informative tokens throughout the token merging process. Through experiments on image classification and image-text retrieval tasks, our algorithm consistently outperforms previous methods using token merging and pruning, given the same running time and memory usage. While our focus has been on tasks involving encoder ViT models, specifically using ViT encoders for image understanding in classification and retrieval, we believe the applicability of our approach extends beyond these scenarios. In the future, we will broaden the scope of our work to include decoder models to adapt to more tasks like stable diffusion, image captioning, text classification, and summarization.

REFERENCES

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4599–4603, 2023.

URL https://api.semanticscholar.org/CorpusID:257833518.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. 2023. doi: 10.48550/ARXIV.2307.10780. URL https://arxiv.org/abs/2307.10780.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12890–12903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.721. URL https://aclanthology.org/2023.acl-long.721.

Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate : Differentiable compression rate for efficient vision transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17118–17128, 2023. doi: 10.1109/ICCV51070.2023.01574.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pp. 104–120, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58576-1. doi: 10.1007/978-3-030-58577-8_7. URL https://doi.org/10.1007/978-3-030-58577-8_7.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021. URL https://arxiv.org/abs/2107.00652.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. Fourier transformer: Fast long range modeling by removing sequence redundancy with fft operator. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.570. URL http://dx.doi.org/10.18653/v1/2023.findings-acl.570.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/li22n.html.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23q.html.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. Lightningdot: Pre-training visual-semantic embeddings for real-time image-text retrieval. In *NAACL-HLT*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357, July 2021.

Maya Varma, Jean-Benoit Delbrouck, Sarah Hooper, Akshay Chaudhari, and Curtis Langlotz. Villa: Fine-grained vision-language representation learning from real-world data. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22168–22178, 2023. doi: 10.1109/ICCV51070.2023.02031.

Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yuxuan Zhou, Wangmeng Xiang, Chao Li, Biao Wang, Xihan Wei, Lei Zhang, Margret Keuper, and Xiansheng Hua. Sp-vit: Learning 2d spatial priors for vision transformers. In *The 33rd British Machine Vision Conference*, 2022. URL https://bmvc2022.mpi-inf.mpg.de/0564.pdf.

Table 2: Performance of token compressing methods when retraining

dataset	model	method	Ri@1	Ri@5	Ri@10	Rt@1	Rt@5	Rt@10	Rsum
Flickr30k	CLIP-B	baseline	81.14	95.29	98.08	90.04	99.04	99.87	564.36
		DCT$_{r=0.925}$	76.92	92.42	96.06	84.33	96.72	99.24	540.64
		ToMe$_{r=0.925}$	79.54	**94.86**	97.28	**90.60**	98.36	**99.42**	**561.08**
		PiToMe$_{r=0.925}$	**80.01**	94.84	**97.36**	89.92	**98.77**	99.35	560.32
	CLIP-L	baseline	81.17	95.64	98.56	92.97	99.4	100.00	568.23
		DCT$_{r=0.95}$	67.50	89.2	94.24	82.12	96.39	96.60	527.73
		ToMe$_{r=0.95}$	79.72	95.21	97.48	91.64	99.31	99.72	561.28
		PiToMe$_{r=0.95}$	**79.84**	**95.44**	**97.70**	**92.48**	**99.40**	**99.99**	**565.44**
	BLIP	baseline	83.5	96.64	98.30	94.43	99.60	100.00	572.44
		DCT$_{r=0.925}$	76.74	93.74	96.05	89.82	98.9	99.29	556.22
		ToMe$_{r=0.925}$	82.04	96.02	97.94	92.22	99.4	99.81	567.50
		PiToMe$_{r=0.925}$	**82.23**	**95.80**	**98.08**	**94.54**	**99.6**	**99.99**	**569.98**
MSCOCO	CLIP-B	baseline	54.67	78.68	86.50	66.06	86.99	93.89	466.46
		DCT$_{r=0.925}$	43.98	74.12	80.75	54.58	79.99	85.54	418.96
		ToMe$_{r=0.925}$	48.02	74.38	83.27	55.70	82.04	89.36	433.66
		PiToMe$_{r=0.925}$	**52.09**	**78.10**	**86.15**	**65.26**	**86.88**	**92.78**	**462.56**
	CLIP-L	baseline	55.45	82.69	88.40	69.26	90.44	94.89	483.68
		DCT$_{r=0.95}$	48.49	74.46	83.26	62.18	84.91	91.58	444.99
		ToMe$_{r=0.95}$	52.99	77.47	85.47	65.34	87.42	93.04	467.67
		PiToMe$_{r=0.95}$	**53.30**	**77.75**	**85.69**	**68.66**	**89.46**	**94.16**	**469.03**
	BLIP	baseline	57.31	81.83	88.91	75.78	93.8	96.62	494.22
		DCT$_{r=0.925}$	53.52	79.28	87.09	70.04	90.40	94.9	476.48
		ToMe$_{r=0.925}$	56.46	81.30	**88.66**	68.98	90.16	95.2	482.35
		PiToMe$_{r=0.925}$	**56.93**	**81.68**	88.63	**73.40**	**91.92**	**95.94**	**490.30**

Table 3: When provided with an equal percentage of remaining memory, our algorithm can achieve comparable speeds while significantly enhancing the performance across all models, whether in off-the-shelf or retrained settings.

model	method	img/s	% memory
CLIP-B $r = 0.925$	baseline	91	100%
	DCT	99	69.0%
	ToMe	102	69.0%
	PiToMe	102	69.0%
CLIP-L $r = 0.95$	baseline	47	100%
	DCT	57	60.5%
	ToMe	60	60.5%
	PiToMe	60	60.5%
BLIP $r = 0.925$	baseline	48	100%
	DCT	60	64.9%
	ToMe	66	64.9%
	PiToMe	65	64.9%
BLIP2 $r = 0.95$	baseline	10	100%
	DCT	18	45.5%
	ToMe	23	45.5%
	PiToMe	21	45.5%

Table 4: Compare to SOTA models

dataset	model	$Rsum$	FLOPS
FLickr30k	UNITER	550.90	-
	VILLA	551.24	-
	LightingDOT	532.26	-
	ALBEF	564.58	55.14
	CLIP-L	568.23	80.85
	BLIP	572.24	55.14
	PiToMe$^{\text{BLIP}}_{r=0.925}$	569.98	35.28
	PiToMe$^{\text{BLIP}}_{r=0.925}$	565.58	35.28
	BLIP2	572.72	678.45
	PiToMe$^{\text{BLIP2}}_{r=0.95}$	566.25	296.93
	PiToMe$^{\text{BLIP2}}_{r=0.975}$	572.81	434.50
MSCOCO	ALBEF	478.39	55.14
	CLIP-L	483.68	80.85
	BLIP	494.34	55.14
	PiToMe$^{\text{BLIP}}_{r=0.925}$	490.30	35.28
	PiToMe$^{\text{BLIP}}_{r=0.925}$	481.78	35.28
	BLIP2	507.46	678.45
	PiToMe$^{\text{BLIP2}}_{r=0.95}$	494.92	296.93
	PiToMe$^{\text{BLIP2}}_{r=0.975}$	504.95	434.50

A Appendix

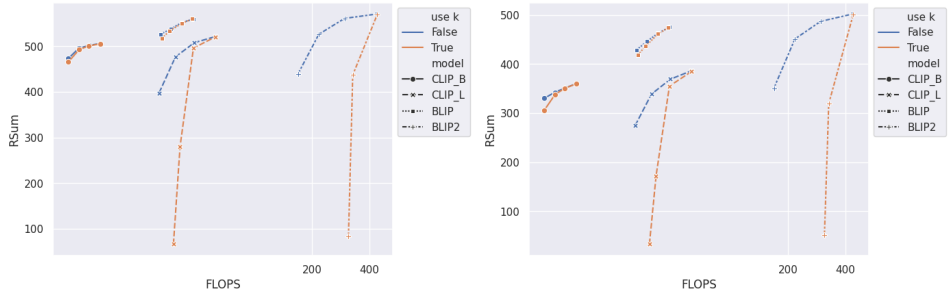
A.1 Detailed experiment results for retrained backbones on image-text retrieval tasks

We also conducted an experiment in which we utilized pre-trained backbones and finetuned them from scratch. All results regarding the retrieval performance, speed, and memory are presented in

tables 2, 3. We also included table 4 to compare with recent SOTA architectures where records with blue background denote models used off-the-shelf without training, while gray indicates models retrained from scratch. Here, it is evident that our algorithm consistently achieves the best performance for almost every backbone for image-text retrieval.

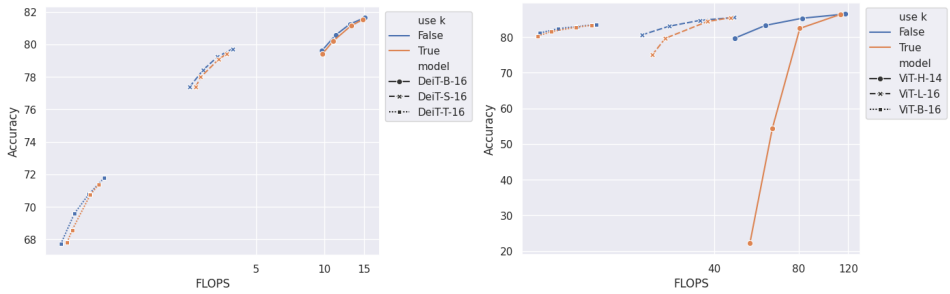
A.2 Performance of ToMe with different token merging schedules

Unlike in ToMe, where the author employed a fixed reduction schedule with an integer parameter k for each layer, we discovered that this merging strategy is suboptimal for off-the-shell performance. The reason is that there tends to be more redundancy in tokens within the initial layers, while the remaining tokens in later layers become progressively more informative. In this paper, we opt to reduce the model by a fixed percentage of r. This allows us to eliminate redundant tokens in the early layers while preserving informative tokens in the later layers.



(a) MSCOCO (b) Flickr30k

Figure 4: Off-the-shell performance of all backbones for image-text retrieval task using different token merging schedules.

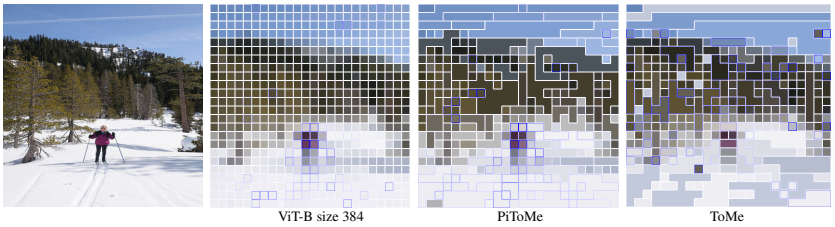


(a) DEIT backbones models (b) MAE backbone models

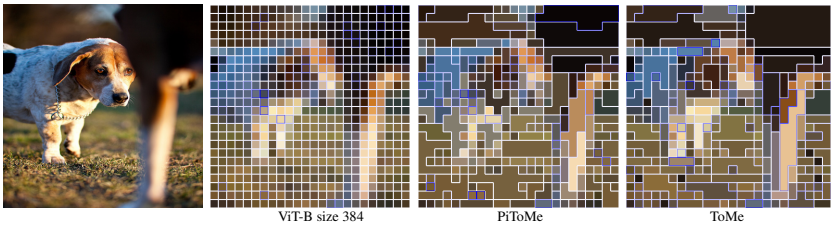
Figure 5: Off-the-shell performance of all backbones for image classification task using different token merging schedules.

A.3 Analysis

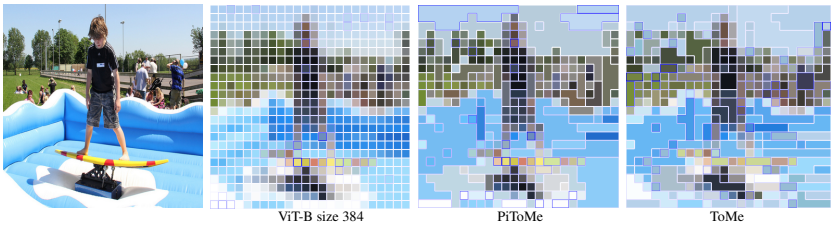
To better explain the effectiveness of PiToMe, we have included many visualizations featuring examples with images and captions sampled from the MSCOCO dataset, as shown in Figures 6a, 6b, 6c, 6d, 6e, 6f, we employed the BLIP backbone and set the reduction percentage to $r = 0.9$, visualizing the final representation of all remain tokens in the last layers. Tokens with bolder blue borders indicate higher attention scores from the classification (CLS) token. Here, it is evident that unlike Tome, which directly merges neighboring nodes with high similarity and potentially damages informative tokens, PiToMe can safeguard important tokens based on their energy scores. Consequently, PiToMe preserves attention maps for important information.



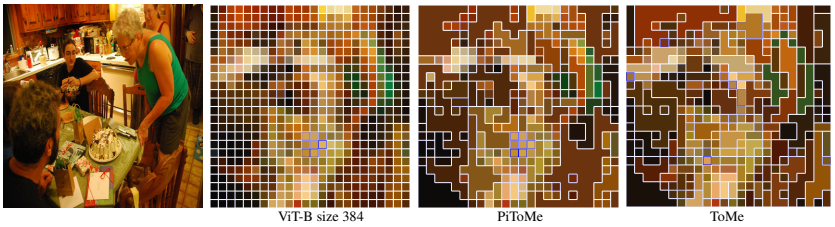
(a) This photo depicts someone cross-country skiing in the mountains.



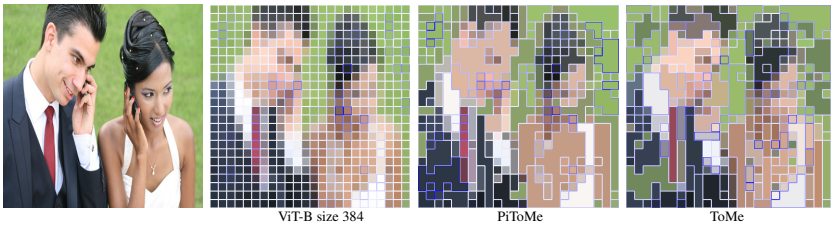
(b) A brown and white dog standing next to another dog.



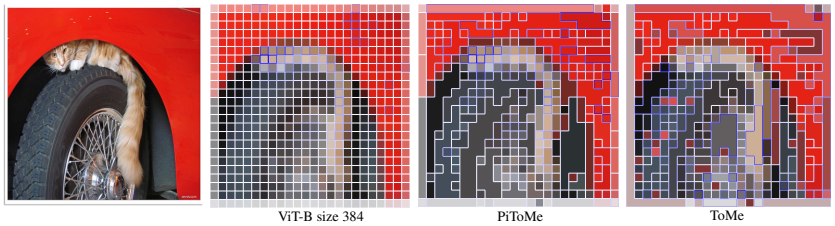
(c) A boy is standing in an inflatable pool on a surfboard.



(d) A woman blowing out the candles on a cake on a table.



(e) A man sitting next to a woman while they both talk on cell phones.



(f) An orange cat hiding on the wheel of a red car.