Accelerating Transformers with Spectrum-Preserving Token Merging

Hoai-Chau Tran[*1,2], **Duy M. H. Nguyen**[*1,3,4], **Duy M. Nguyen**[5], **TrungTin Nguyen**[6], **Ngan Le**[7],
Pengtao Xie[8,9], **Daniel Sonntag**[1,10], **James Zou**[11], **Binh T. Nguyen**[†2], **Mathias Niepert**[†3,4]

[1]German Research Center for Artificial Intelligence (DFKI), [2]University of Science - VNUHCM
[3]Max Planck Research School for Intelligent Systems (IMPRS-IS), [4]University of Stuttgart,
[5]Dublin City University, [6]University of Queensland, [7]University of Arkansas, [8]MBZUAI,
[9]UC San Diego, [10]Oldenburg University, [11]Stanford University.

Abstract

Increasing the throughput of the Transformer architecture, a foundational component used in numerous state-of-the-art models for vision and language tasks (e.g., GPT, LLaVa), is an important problem in machine learning. One recent and effective strategy is to merge token representations within Transformer models, aiming to reduce computational and memory requirements while maintaining accuracy. Prior works have proposed algorithms based on Bipartite Soft Matching (BSM), which divides tokens into distinct sets and merges the top k similar tokens. However, these methods have significant drawbacks, such as sensitivity to token-splitting strategies and damage to informative tokens in later layers. This paper presents a novel paradigm called PITOME, which prioritizes the preservation of informative tokens using an additional metric termed the *energy score*. This score identifies large clusters of similar tokens as high-energy, indicating potential candidates for merging, while smaller (unique and isolated) clusters are considered as low-energy and preserved. Experimental findings demonstrate that PITOME saved from 40-60% FLOPs of the base models while exhibiting superior off-the-shelf performance on image classification (0.5% average performance drop of ViT-MAEH compared to 2.6% as baselines), image-text retrieval (0.3% average performance drop of CLIP on Flickr30k compared to 4.5% as others), and analogously in visual questions answering with LLaVa-7B. Furthermore, PITOME is theoretically shown to preserve intrinsic spectral properties to the original token space under mild conditions. Our implementation is available at this link.

1 Introduction

Vision Transformers (ViTs) [1] have been integral to recent advancements in computer vision, leading to state-of-the-art deep learning architectures for representing images and videos [2–5]. However, these transformer-based architectures incur substantial memory costs and have a quadratic time complexity in the number of tokens due to the self-attention layers. This challenge becomes particularly severe as model sizes increase, as observed in Large Language Models (LLMs) [6].

To address such limitations, several efforts focus on designing a more *efficient attention* mechanism by making it linearly scale with input tokens [7, 8], integrating vision or language domain-specific modules [9, 10], or pruning the head numbers in ViT [11, 12]. Others propose *dynamically pruning* less important tokens *w.r.t.* pre-defined metrics using learnable masks [13, 14]. However, a primary downside of these novel methodologies lies in the necessity to retrain the model from scratch, therefore hindering the leveraging of well-trained models such as LLMs. Moreover, most pruning-based techniques may not accelerate the training process, which arises from the dynamic removal

---

[*]Co-first author, [†]Corresponding Authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

<center>ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate</center>

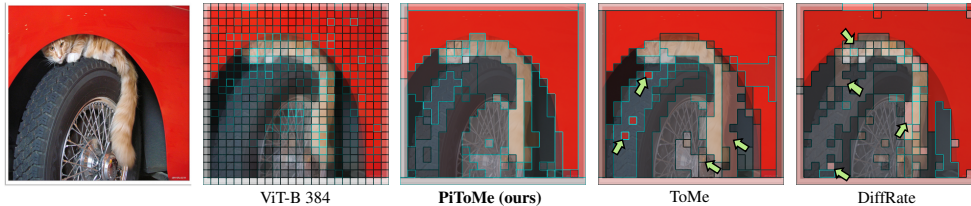



Figure 1: A comparison of token merging methods. Patches of the same color are merged. Green arrows highlight incorrect merges, avoided by PITOME. Position of tokens with high attention scores (cyan borders, zoom for clarity) in PITOME are maintained proportionality akin to ViT-base 384.

of tokens in each sample, resulting in a mismatch of dimensions and consequently preventing the batching of samples with consistent dimensions.

Recent research has introduced a novel *token merging* technique. Instead of pruning, this method combines tokens with high semantic similarity, removing background tokens and merging less informative foreground ones. Its versatility extends to training and non-training scenarios, drastically reducing compute and memory usage. A notable example is ToMe [15], which introduced the Bipartite Soft Matching (BSM) algorithm, prominent for its simplicity and effectiveness in merging highly similar tokens. Since ToMe, several works, including ToFu [16], Pumer [17], LTPM [18], and DiffRate [19], have built upon BSM with various adaptations in vision and language domains. In BSM, tokens representing image patches are separated into sets $\mathcal{A}$ and $\mathcal{B}$, and their pairwise cosine similarity is computed. The top k similar pairs of tokens between the sets $\mathcal{A}$ and $\mathcal{B}$ are merged. However, the performance of this algorithm is sensitive to the token-splitting strategy. For instance, ToMe's approach, which first splits tokens based on index parity, can lead to incorrect merging since tokens in $\mathcal{A}$ can subsequently only be merged with those in $\mathcal{B}$ (Figure 1). Moreover, while BSM excels in initial layers with many redundant tokens, deeper layers risk merging informative tokens due to latent object correlations. Though current enhancements [19] mitigated this by considering token attention scores in BSM [20], their adaptability to different ViT architectures, each with potentially distinct attention score distributions [21], remains a challenge.

In this work, we propose PITOME (Protect Informative Tokens before Merging), a method designed to safeguard crucial information-bearing tokens prior to the merging step. Our method prioritizes preserving informative tokens by utilizing an additional metric termed the *energy score* inspired by connections to *graph energy* in spectral graph theory [22, 23] (Theorem 1). Specifically, our energy score assesses large clusters of similar tokens as possessing high energy (like background and repeated textures), thereby marking them as suitable candidates for merging, while smaller, distinct regions (foreground) are deemed low-energy and thus treated as protected informative tokens. The proposed energy term operates on the graph built for input tokens, taking into account their relationships and aggregating information from nearby neighbors when their similarities exceed certain thresholds. This approach facilitates a deeper contextual comprehension compared to previous works [15–17, 19] that rely solely on attention scores or feature embedding per token. Subsequently, we only select the highest-scoring tokens and pass them on for merging in the next steps, ensuring the preservation of important tokens, particularly in the latter stages when only a few remaining ones. During the merging process, we continue leveraging sorted energy vectors from earlier stages by distributing tokens with similar energy into two sets, $\mathcal{A}$ and $\mathcal{B}$, resulting in candidates in $\mathcal{A}$ having a high probability of finding compatible matches in $\mathcal{B}$. Matched tokens are then merged using a weighted average feature embedding to create a new token representation.

The empirical results demonstrate that despite the increased computational cost associated with energy score calculations, PITOME exhibits comparable speed to other BSM-based approaches since the matching is performed on a smaller, high-energy token set. At the same time, it consistently shows superior accuracy across various experimental scenarios. Additionally, we present theoretical insights into PITOME, showing that, under moderate assumptions — such as the discriminative nature of feature embeddings generated by ViT for node pairs within and across distinct objects — our algorithm efficiently preserves the spectral properties of the initial input tokens, maintaining the eigenvalues derived from normalized Laplacian matrices of the original tokens [24–26]. To summarize, our contributions encompass:

- A new token merging procedure for accelerating ViT architectures is designed to protect crucial yet small-region tokens while identifying redundant ones for merging based on contextual token correlations captured by our energy score functions.

<center>2</center>

- Our PITOME runs as fast as other BSM-based approaches while achieving SOTA performance on diverse tasks, ranging from image-text retrieval (Sec. 4.1), visual question answering with LLMs (Sec. 4.2), image classification (Sec. 4.3), and text classification (Sec. 4.4). In several cases, PITOME is shown to reduce up to $40 - 60\%$ FLOPs of base models while only dropping performance around $0.3 - 0.5\%$ (CLIP model on Flick30k).

- We also present theoretical findings indicating that, given reasonable assumptions, PITOME can effectively approximate the spectral distance between the initial token spaces and the merged token set. This sheds light on why PITOME tends to outperform baselines in practical applications and contributes to a better understanding of the potential limitations inherent in BSM-based methods, such as those in [15, 16, 19, 17, 27].

2 Related Work

Efficient Attention Mechanisms. Various efforts have sought to enhance the efficiency of transformers in both NLP and Vision domains. Some concentrate on accelerating attention computation [28, 29, 8] through approximation techniques involving hashing [30], low-rank [31], or sparse approximations [32]. Others explore strategies such as head or feature pruning [11, 33] or the integration of domain-specific modules [9, 5, 34, 10]. However, many of them necessitate joint training with the backbone model from scratch. For instance, DynamicViT [35] runs approximately 150 hours of fine-tuning on an NVIDIA A100 GPU to prune the DeiT-S model [36]. In contrast, we focus on accelerating existing ViT models by token merging, which applies to training and non-training scenarios.

Dynamic Token Pruning. Several studies have explored token pruning in transformer models across NLP [37–39] and vision domains [40–42, 27]. However, like efficient transformers, these methods typically require training. Additionally, most pruning techniques are dynamic, meaning the number of tokens varies across different inputs, which improves accuracy but complicates batching for practical deployment. To address this, numerous pruning methods employ masks during the training phase rather than directly eliminating tokens; however, it yields to cancel out the speed advantages associated with pruning.

Token Merging. Leading techniques such as ToMe [15] and its improvements [17, 43, 18, 19, 16, 44], build upon lightweight Bipartite Soft Matching (BSM). These methods exhibit speeds comparable to pruning while achieving superior performance. They have demonstrated the ability to double the throughput of state-of-the-art Vision Transformers (ViT) on both images and videos with minimal accuracy degradation in various scenarios. However, BSM-based approaches are sensitive to the selection of sets in the matching process, potentially resulting in the loss of informative tokens due to heuristic merging procedures. To address these issues, methods like DiffRate [19] and Crossget [44] leverage attention scores in ViT or cross-modal guidance to identify important tokens during the matching process, though they remain sensitive to the distribution of the token space, especially with imbalanced clusters. Another direction involves adapting more intricate algorithms, such as k-means [45], spectral clustering [46], graph pooling [47], or graph coarsening [24, 48], to merge similar tokens. While these strategies offer some guarantees and well-controlled outputs, their iteration schemes are highly complex and may not align with the goal of reducing model complexity in ViT layers. Our PITOME, on the other hand, enables the advantages of both approaches. It maintains efficiency comparable to BSM, remains robust to token partitioning strategies, and offers a reasonable trade-off between speed and accuracy. Moreover, PITOME is theoretically proved to approximate the spectral spectrum of the original token space under reasonable assumptions, resembling the behavior of other spectral clustering methods.

3 Methodology

3.1 Token Merging Formulation

We apply token merging to each transformer block of the ViT architecture (Figure 2-a). Given the input token of the l-th block $\mathbf{X}^l \in \mathbb{R}^{N \times h}$ where N and h are the token length and token hidden embeddings, a forward step in one Transformer block can be formulated as follows:

$$\hat{\mathbf{X}}^l = \mathbf{X}^l + \text{Attention}(\mathbf{X}^l \mathbf{W}_Q, \mathbf{X}^l \mathbf{W}_K, \mathbf{X}^l \mathbf{W}_V), \ \mathbf{X}^{l+1} = \hat{\mathbf{X}}^l + \text{MLP}(\hat{\mathbf{X}}^l) \quad (1)$$

where Attention and MLP are the self-attention and multiple layer perceptron components. We then apply merge operations on $\hat{\mathbf{X}}^l$ and compute the output of the reduced MLP block as:

$$\mathbf{X}^{l+1} = \hat{\mathbf{X}}_m^l + \text{MLP}(\hat{\mathbf{X}}_m^l), \ \text{where} \ \hat{\mathbf{X}}_m^l = \mathbf{F}_{\text{mer}}(\hat{\mathbf{X}}^l, \mathbf{X}^l \mathbf{W}_K, r). \quad (2)$$

3

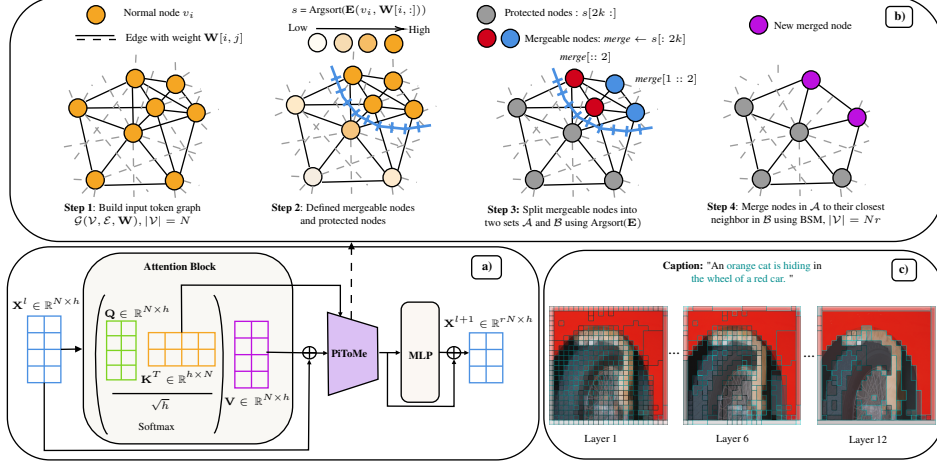



Figure 2: **a)** PITOME can be inserted inside transformer block; **b)** Energy scores are computed to identify mergeable and protective tokens; **c)** Our algorithm gradually merges tokens in each block.

Here $\mathbf{F}_{\text{mer}}(.)$ is the merging operation that receives $\hat{\mathbf{X}}^l$ as input for compressing, $\mathbf{X}^l \mathbf{W}_K$ (key matrices) as the token features of $\hat{\mathbf{X}}^l$ following prior work [15, 43, 18, 19], and r is the fraction of remaining tokens. The output $\hat{\mathbf{X}}_m^l \in \mathbb{R}^{rN \times h}$ serves as input for the MLP layer to produce $\mathbf{X}^{l+1} \in \mathbb{R}^{rN \times h}$. We present the PITOME $\mathbf{F}_{\text{mer}}(.)$ function in the next section.

3.2 Energy-based Merging

We propose to use a new term called *energy score* to evaluate the redundancy of each token, which is then used to protect informative or isolated tokens (low energy scores) while considering tokens that are in the large cluster as high energy scores and characterizing them as merging candidates. Figure 2-b illustrates the main steps in PITOME.

Token Graph Construction: Given a set of N token inputs in $\hat{\mathbf{X}}^l$, we build a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ with $\mathcal{V}$ a set of $N = |\mathcal{V}|$ nodes, $\mathcal{E}$ a set of $M = |\mathcal{E}|$ edges defined by connecting one token to the remaining ones in $\mathcal{G}$, $\mathbf{W} \in \mathbb{R}^{N \times N}$ be a weighted adjacency matrix. We opt for using the *key* vectors $\mathbf{K} = \mathbf{X}^l \mathbf{W}_K \in \mathbb{R}^{N \times h}$ as node features of $\mathcal{V}$, i.e., $v_i \in \mathcal{V}$ has h feature dimensions. The weight $\mathbf{W}[i, j]$ assigned to an edge $e_{ij} \in \mathcal{E}$ connects v_i and v_j is computed by cosine distance:

$$\mathbf{W}[i, j] = 1 - \cos(v_i, v_j), \text{ where } \cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}, \quad \forall v_i \in \mathcal{V}, v_j \in \mathcal{V}. \quad (3)$$

For simplicity, $\mathbf{W}[i, :]$ and $\mathbf{W}[:, i]$ denote the i-th row and column, *resp.*; $[N]$ stands for $\{1, \dots, N\}$.

Token Energy Scores: In this step, the *energy score*, denoted as $\mathbf{E} = (E_i)_{i \in [N]}$, is computed for each node (Figure 2-a, Step 2). The term is inspired by the concept of *graph energy* in spectral graph theory [22, 23], defined as the sum of the absolute eigenvalues of the adjacency matrix $\mathbf{W}$. We also leverage such structures of $\mathbf{W}$ to find correlations among tokens and to estimate token redundancy. Instead of using independent token values such as attention scores [19], our energy leads to better performance (Figure 6, Appendix) and provides theoretical connections to the spectral properties of the original token graphs (Theorem 1).

Let i be the index of the current node and $\mathcal{N}(i)$ represent the set of neighbor nodes. The energy score $E_i \equiv E_i(v_i, \mathbf{W}[i, :])$ of node v_i is calculated using the following equation:

$$E_i(v_i, \mathbf{W}[i, :]) = \frac{1}{N} \sum_{j \in \mathcal{N}(i)} f_m(\cos(v_i, v_j)), \; f_m(x) = \begin{cases} x & \text{if } x \geq m \\ \alpha(\exp(x - m) - 1) & \text{otherwise} \end{cases}. \quad (4)$$

Rather than accumulating all $\cos(v_i, v_j)$ values, the function $f_m(.)$ in Eq.(4) mimics the exponential linear unit activation function [49], focusing on similar tokens even if they are far apart, while ignoring dissimilar ones. Here, m is a dynamic margin value varying at each layer in the ViT model. Nodes within this margin, i.e., $(x > m)$ with high cosine similarity $\cos(v_i, v_j)$ are considered true neighbors, potentially representing tokens belonging to the same object. Nodes outside this margin have $\cos(v_i, v_j)$ replaced by a constant α, providing a lower bound for minimal edge weights. The

4

term $\exp(x - m) - 1 < 0$ smooths the function $f(x)$ for neighboring nodes near 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 total number of encoder layers, indicating an increasing margin as tokens move to deeper layers. The ablation studies for the α and m values are presented in Section 4.5.

Intuitively, Eq.(4) reflects the number of tokens potentially representing the same object. Tokens belonging to large objects (e.g., background) will have high energy scores, indicating potential candidates for merging, while smaller ones (e.g., foreground) will have low energy scores and are considered to be protected. This guides us to sort the energy vectors $\mathbf{E}$ in descending order and choose only the top $2k$ nodes with the highest scores as mergeable candidates and the remaining ones as protective tokens, i.e, $\mathbf{s} = \mathrm{argsort}(\mathbf{E})$, *merge* $\leftarrow \mathbf{s}[: 2k]$, *protect* $\leftarrow \mathbf{s}[2k :], k = N - Nr$.

Ordered Energy-based Bipartite Soft Matching: Having identified mergeable tokens in the *merge* set, we continue exploit the sorted order in $\mathbf{E}$ to form two sets $\mathcal{A}$ and $\mathcal{B}$ in BSM, each containing k nodes. Specifically, tokens with odd and even indices in *merge* are selected for $\mathcal{A}$ and $\mathcal{B}$, *resp.* given the fact that those in the same object should have similar energy scores, resulting in likely distributing in consecutive positions in $\mathrm{argsort}(\mathbf{E})$. In other words, our choosing has a high probability that one token in $\mathcal{A}$ always finds its best match in the same object in $\mathcal{B}$. This sets us apart with random partitions based on spatial indices in images like [15, 16].

Tracking Token Sizes All nodes in set $\mathcal{A}$ are then merged with their nearest neighbors in set $\mathcal{B}$ through the *fast* BSM algorithm. Following prior works [15, 16], we also add proportional attention to balance the effect of the merged token on the output of the softmax function: $\mathbf{A} = \mathrm{Softmax}\left(\mathbf{X}^l\mathbf{W_Q} \cdot (\mathbf{X}^l\mathbf{W_K})^T/\sqrt{h} + \log \mathbf{m}\right)$ where $\mathbf{m}$ is a row vector containing the size of each token, i.e., the number of data patches the token represents. The pseudo-code for our method is provided in Algorithm 1 (Appendix) with complexity analysis.

3.3 Connection to Graph Coarsening with Spectral Preservation

In this section, we employ tools from spectral graph theory to show a spectral distance preservation of PITOME. We note that similar properties can be obtained by using more complicated clustering algorithms such as K-mean [45] or spectral clustering [46, 47, 24]; however, these methods are typically loop-based algorithms, which are computationally expensive and not suitable for batch-type data. Our PITOME, in contrast, is as fast as BSM methods but theoretically preserves spectral properties of input token graphs.

We begin by introducing Definitions 1 and 2 of graph coarsening and lifting, *resp.*, to justify the spectral distance constructed in equation (5), measuring the similarity between the original and coarse graphs. For more thorough coverage of the mathematics of graph coarsening and graph lifting, we refer the reader to [50–53]. In short, *we treat the result of token merging as a graph coarsening process* (Figure 8, Appendix). We then create the *lifted graph* as a reconstruction from this coarsened version to assess the spectral distance to the original token graph.

Definition 1 (Graph Coarsening). *Given a weighted graph* $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$, *we denote* $\mathcal{P} = \{\mathcal{V}_i\}_{i \in [n]}$ *where* $\mathcal{V} = \cup_{i \in [n]} \mathcal{V}_i$, *be a partition of its node into* n *disjoint sets. The coarsened graph of* $\mathcal{G}$ *w.r.t.* $\mathcal{P}$ *is the weighted graph* $\mathcal{G}_c$, *where each partition in* $\mathcal{P}$ *is aggregated into a single node, denoted* $\{\nu_i\}_{i \in [n]}$, *by averaging the elements within each partition. The elements of the adjacency matrix are given by* $\mathbf{W}_c[i, j] = \sum_{v_i \in \mathcal{V}_i} \sum_{v_j \in \mathcal{V}_j} \mathbf{W}[i, j]/(|\mathcal{V}_i||\mathcal{V}_j|)$. *We denote the combinatorial and normalized Laplacians of* $\mathcal{G}$ *by* $\mathbf{L} = \mathbf{D} - \mathbf{W}$ *and* $\mathcal{L} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, *resp., where* $\mathbf{D}$ *is the diagonal degree matrix with* $\mathbf{D}[i, i] = d_i := \sum_{j=1}^{N} \mathbf{W}[i, j]$. *Similarly, the definition of the coarsened Laplacian matrices follows directly:* $\mathbf{L}_c = \mathbf{D}_c - \mathbf{W}_c$ *and* $\mathcal{L}_c = \mathbf{I}_n - \mathbf{D}_c^{-1/2}\mathbf{W}_c\mathbf{D}_c^{-1/2}$. *Finally, the eigenvalues and eigenvectors of* $\mathcal{L}$ *(resp.* $\mathcal{L}_c$*) are denoted as* $\boldsymbol{\lambda}$ *and* $\mathbf{u}$ *(resp.* $\boldsymbol{\lambda}_c$ *and* $\mathbf{u}_c$*).*

Definition 2 (Graph Lifting). *We call* $\mathcal{G}_l(\mathcal{V}_l, \mathcal{E}_l, \mathbf{W}_l)$ *the lifted graph of* $\mathcal{G}$ *if the adjacency matrix elements are given by* $\mathbf{W}_l[i, j] = \mathbf{W}_c[i, j]$. *We denote the node degree of* $v_{li} \in \mathcal{V}_l$ *by* $d_{li} = \sum_{j=1}^{N} \mathbf{W}_l[i, j]$. *The combinatorial and normalized Laplacians of* $\mathcal{G}_l$ *is then defined as* $\mathbf{L}_l = \mathbf{D}_l - \mathbf{W}_l$ *and* $\mathcal{L}_l = \mathbf{I}_N - \mathbf{D}_l^{-1/2}\mathbf{W}_l\mathbf{D}_l^{-1/2}$, *resp., where* $\mathbf{D}_l$ *is the diagonal degree matrix with* $\mathbf{D}[i, i] = d_{li}$. *Then, we denote,* resp.*, the eigenvalues and eigenvectors of* $\mathcal{L}_l$ *by* $\boldsymbol{\lambda}_l$ *and* $\mathbf{u}_l$.

5

Lemma 1 (Eigenvalue Preservation, see *e.g.,* [50, 51, 54, 55]). *The normalized Laplacian eigenvalues of the lifted graph $\boldsymbol{\lambda}_l$ contain all the eigenvalues of the coarse graph $\boldsymbol{\lambda}_c$ and additional eigenvalues 1 with $(N - n)$ multiplicity.*

Through Lemma 1, we can use the lifted graph $\mathcal{G}_l$ as a proxy for the coarse graph $\mathcal{G}_c$, and define:

$$\mathrm{SD}(\mathcal{G}, \mathcal{G}_c) = \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_l\|_1 = \sum_{i=1}^{N} |\lambda_i - \lambda_{li}| \text{ as a spectral distance.} \tag{5}$$

Next, we present our main theoretical result demonstrating how spectral distance characterizes the superiority of our novel PITOME paradigm over the state-of-the-art approaches as ToMe [15, 16]. The Theorem 1 quantifies how similar the original $\mathcal{G}$ is to its coarsened counterpart $\mathcal{G}_c$, and is proved in Appendix E.

Theorem 1 (Spectrum Consistent of Token Merging). *Suppose the graphs $\mathcal{G}_0^{(s)}$, $\mathcal{G}_{\mathrm{PITOME}}^{(s)}$, and $\mathcal{G}_{ToMe}^{(s)}$ are coarsened from the original graph $\mathcal{G}$ by iteratively merging pairs of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0^{(s)} = \{\mathcal{V}_{0i}^{(s)}\}_{i \in [s]}$, the PITOME-partition $\mathcal{P}_{\mathrm{PITOME}}^{(s)} = \{\mathcal{V}_{\mathrm{PITOME}i}^{(s)}\}_{i \in [s]}$, defined by PITOME in Algorithm 1, and the ToMe-partition [15, 16], $\mathcal{P}_{ToMe}^{(s)} = \{\mathcal{V}_{ToMei}^{(s)}\}_{i \in [s]}$, for $s = N, \ldots, n+1$. We assume some standard mild assumptions: (A1) $\mathbb{E}[\cos(v_{a_s}, v_{b_s})] \to 1, \quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s]$; (A2) there exists a margin m s.t., $\cos(v_{a_s}, v_{b_s}) \geq m > \cos(v_{a_s}, v_{c_s}), \quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{c_s} \in \mathcal{V}_{0j}^{(s)}, \forall i \neq j \in [s]$; and (A3) there is an order of cardinality in the true partition, without loss of generality, we assume $N_1^{(s)} \geq N_2^{(s)} \geq \ldots \geq N_s^{(s)}$, where $N_i^{(s)} = |\mathcal{V}_{0i}^{(s)}|, \forall i \in [s]$. Then it holds that:*

1. *The spectral distance between the original $\mathcal{G} \equiv \mathcal{G}_0^{(N)}$ and the PITOME-coarse $\mathcal{G}_{\mathrm{PITOME}}^{(n)}$ graphs converges to 0, i.e., $SD(\mathcal{G}, \mathcal{G}_{\mathrm{PITOME}}^{(n)}) \to 0$,*

2. *The spectral distance between the original $\mathcal{G}$ and the ToMe-coarse $\mathcal{G}_{ToMe}^{(n)}$ graphs converges to a non-negative constant C, with a high probability that $C > 0$.*

Intuitively, Theorem 1 states that, given assumptions (i) tokens are closely embedded within classes and distinct between classes (*A1, A2*), and (ii) the number of tokens per class follows certain orders (*A3*), the spectral distance between PITOME and the original tokens in Eq.(5) will converge to 0. In contrast, with ToMe partitions, a non-eliminable constant likely remains.

4 Experiments

We focus on two settings: *Off-the-Shelf Performance* , where we evaluate the models' performance immediately after compression without training, and *Retrained* , where we treat the compression algorithms as pooling functions and retrain the models on downstream tasks. The experiments cover four tasks: (i) *image & text retrieval*, (ii) *visual question answering (VQA)*, (iii) *image classification*, and (iv) *text classification*. We use the number of floating-point operations (FLOPS) needed for inference on one sample as the main metric to benchmark memory footprint and speed. Higher FLOPS indicate greater memory requirements and longer training and inference times.

4.1 Image & Text Retrieval

We evaluate PITOME on the image-text retrieval task using three different backbone models CLIP [56], ALBEF [57], and BLIP [58] on two frequently used Flickr30k [59] and MSCOCO [60] datasets. Our experiment is benchmarked using recall@k [61], where a higher recall@k indicates the model's effectiveness in retrieval. In Figure 3, we benchmarked PITOME against other SOTA *merging* or *pruning*-based methods such as ToMe [15], ToFu [16], DiffRate [19], and DCT [62] on *off-the-shelf* setting when varying amount of merged tokens at each layer. Given the same FLOPS, it is clear that PITOME consistently outperforms previous compression algorithms across all backbones. The performance gap increases as we decrease the percentage r of tokens retained in each layer. The same behavior remains consistent in Table 2, where we set $r = 0.925$ and *retrain* pre-trained checkpoints of BLIP and CLIP. For more details about the training method, please refer to Li et al. [58].

In Table 1, we compare PITOME using compression ratios of $r \in \{0.95, 0.975\}$ on BLIP and BLIP-2 against other advanced architectures such as ViLT [63], LightningDOT [64], UNITER [65],

6

METER [66], CLIP-L [67], and ALBEF [68]. The results show that PITOME consistently surpasses those architectures by a significant margin. Moreover, the performance drop on the base BLIP/BLIP-2 is minimal while achieving substantial reductions in memory footprint and *rerank times—nearly halving for BLIP* and *tripling for BLIP-2*. Additionally, the speedup can further improve with increased batch and model sizes.

4.2 Visual Question Answering (VQA) with Large Vision-Language Models

This experiment focuses on assessing the off-the-shelf performance of large vision-language models like LLaVa [2]. We extensively conduct experiments across six VQA datasets: VQA-v2 [69], GQA [70] (academic questions), VizWiz [71] (visually impaired individuals), ScienceQA [72] (zero-shot scientific question answering), TextVQA [73] (text-rich VQA tasks), and MME-Perception [74] (visual perception with yes/no question). More details on the number of samples in each dataset are in the Appendix. All experiments are conducted using *LLAVA-1.5 7B* and *LLAVA-1.5 13B* with the lmms_eval library [75] provided by the LMMs-Lab team.

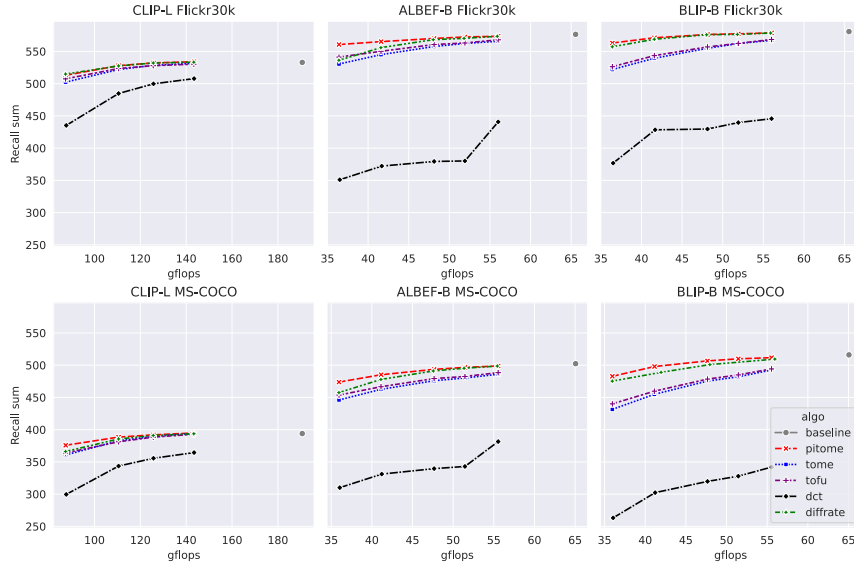



Figure 3: **Off-the-shell** **Image-Text Retrieval comparison** between PITOME v.s. merging/pruning methods on different backbones on tasks when varying the number of merged tokens. Here, Recall sum $=Rt@1 + Rt@5 + Rt@10 + Ri@1 + Ri@5 + Ri@10$ is close to 600, indicating recall scores at top 1,5, and 10 for retrieving image and text reached close to 100%. PITOME curves, in most cases, are above other baselines.

Table 1: **Image-Text Retrieval comparison.** PITOME without training are in blue, and with training in gray. PITOME achieves SOTA while saving $36\% - 56\%$ in FLOPS and speeding up by $\times 1.4$ to $\times 1.6$ compared to the base models.

Datasets	Methods	Rt@1 ↑	Ri@1 ↑	ZS Retrieval Rsum ↑	Reranked Rsum ↑	ViT FLOPS ↓	Total FLOPS ↓	ZS Retrieval Time ↓	Total Time ↓
Flickr30k	ViLT	83.50	64.40	490.60	525.70	-	55.90	-	-
	LightingDOT	83.90	69.90	532.26	-	-	-	-	-
	UNITER	92.87	83.73	521.90	542.80	-	949.9	-	-
	METER	94.30	82.22	560.54	570.72	-	-	-	-
	CLIP-L	92.90	81.34	568.23	-	80.85	-	25s	-
	ALBEF	94.91	85.32	564.58	575.00	55.14	65.54	16s	58s
	PITOME$_{r=0.95}^{BLIP}$	95.72	86.32	567.58	577.81	38.55	47.65	13s	56s
	PITOME$_{r=0.95}^{BLIP}$	96.61	87.18	569.98	579.35	38.55	47.65	13s	56s
	BLIP	96.86	87.48	572.24	580.76	55.14	65.54	16s	1m17s
	PITOME$_{r=0.95}^{BLIP2}$	96.83	87.84	566.25	580.77	296.93	390.77	45s	1m21s
	PITOME$_{r=0.975}^{BLIP2}$	97.55	89.04	572.81	583.72	434.50	564.78	1m5s	1m54s
	BLIP2	97.61	89.79	572.72	584.76	678.45	900.77	1m37s	3m15s
MS-COCO	ViLT	61.50	42.70	420.20	439.20	-	55.90	-	-
	CLIP-L	70.78	53.79	478.18	-	80.85	-	2m10s	-
	METER	76.16	57.08	-	495.95	-	-	-	-
	ALBEF	76.94	60.24	478.39	500.44	55.14	65.54	43s	5m29s
	PITOME$_{r=0.95}^{BLIP}$	79.46	62.50	485.99	506.65	38.85	47.65	51s	4m30s
	PITOME$_{r=0.95}^{BLIP}$	80.44	63.91	493.33	512.66	38.85	47.65	51s	4m30s
	BLIP	81.82	64.36	494.34	516.03	55.14	65.54	1m3s	7m10s
	PITOME$_{r=0.95}^{BLIP-2}$	82.29	65.54	494.92	518.44	296.93	390.77	3m33s	6m34s
	PITOME$_{r=0.975}^{BLIP-2}$	84.12	67.37	504.95	527.06	434.50	564.78	5m13s	9m24s
	BLIP-2	85.32	68.26	507.46	528.63	678.45	900.77	7m52s	16m12s

Table 2: **Retrained Image-Text Retrieval comparison** when retraining from scratch on CLIP and BLIP backbones. $Rk = Rk@1 + Rk@5 + Rk@10$, $k \in \{t, i\}$,

Models	Algo.	Rt ↑	Ri ↑	GFLOPS ↑	Eval Speed ↑	Train Speed ↑
CLIP$_{Flickr}$	Baseline	291.80	275.52	x1.00	x1.00	x1.00
	ToMe	287.30	270.52	x2.10	x1.39	x1.79
	ToFu	288.32	269.68	x2.10	x1.39	x1.76
	DCT	279.70	258.24	x2.10	x1.30	x1.37
	DiffRate	289.33	266.45	x2.10	x1.39	x1.78
	PITOME	291.50	270.94	x2.10	x1.39	x1.78
BLIP$_{FLickr}$	Baseline	296.70	284.06	x1.00	x1.00	x1.00
	ToMe	294.80	280.64	x1.57	x1.66	x1.60
	ToFu	296.46	281.04	x1.57	x1.65	x1.59
	DCT	291.79	275.22	x1.57	x1.61	x1.45
	DiffRate	292.77	279.46	x1.57	x1.65	x1.59
	PITOME	296.00	282.36	x1.57	x1.66	x1.59
CLIP$_{coco}$	Baseline	256.30	222.21	x1.00	x1.00	x1.00
	ToMe	248.64	215.03	x2.10	x1.38	x1.79
	ToFu	248.99	216.56	x2.10	x1.39	x1.79
	DCT	240.04	211.28	x2.10	x1.34	x1.37
	DiffRate	248.87	215.45	x2.10	x1.39	x1.79
	PITOME	250.70	217.01	x2.10	x1.39	x1.79
BLIP$_{coco}$	Baseline	273.72	241.30	x1.00	x1.00	x1.00
	ToMe	266.86	234.67	x1.57	x1.90	x1.85
	ToFu	266.18	233.87	x1.57	x1.90	x1.85
	DCT	264.38	230.19	x1.57	x1.86	x1.78
	DiffRate	265.45	235.11	x1.57	x1.84	x1.85
	PITOME	268.42	236.25	x1.57	x1.88	x1.85

7

Let L denote the number of layers in the CLIP encoder and N the number of visually encoded tokens. In our experiment, we apply PITOME to the ViT vision encoder of LLAVA, retaining only r percent of tokens in each layer. This results in $r^L N$ tokens being fed into the LLM, significantly enhancing inference speed. We used LLaVA-1.5-7B and LLaVA-1.5-13B checkpoints to run off-the-shelf settings. Tables 3 and 4, along with Figure 4, illustrate that the PITOME algorithm consistently achieves superior performance compared to other merging and pruning methods, as well as existing SOTA models such as BLIP-2 [76], InstructBLIP [77], IDEFICS-9B/80B [78], with *inference time nearly halved*. Remarkably, in some datasets like VisWiz and ScienceQA, the compressed model even surpasses the baseline model. We contend that this improvement stems from the merging of less significant tokens in PITOME, potentially enhancing the robustness of the language model (LLM).

4.3 Image Classification on Imagenet-1k

In this task, we employed five 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 [79] or DEIT [80] styles. These backbones were utilized to assess both off-the-shelf and retrained performance. All experiments were conducted on the ImageNet-1k dataset, which is a subset of ImageNet [81] containing labeled images spanning 1000 categories.

Table 3: **Off-the-shelf** *LLaVA-1.5 7B* (r=0.9) and *LLaVA-1.5 13B* (r=0.925) performance vs. PITOME and other token pruning/merging methods on six VQA datasets: VQA-v2 [69], GQA [70], VisWiz [71], TextVQA [73], MME [74] ScienceQA image (ScienceQA^I) [72].

Model	LLM	VQA^{v2}↑	GQA ↑	VisWiz↑	ScienceQA^I↑	TextVQA ↑	MME↑
BLIP-2	Vicuna-13B	41.0	41.0	19.6	61.0	42.5	1293.8
InstructBLIP	Vicuna-7B	-	49.2	34.5	60.5	50.1	-
InstructBLIP	Vicuna-13B	-	49.5	33.4	63.1	50.7	1212.8
IDEFICS-9B	LLaMA-7B	50.9	38.4	35.5	-	25.9	-
IDEFICS-80B	LLaMA-65B	60.0	45.2	36.0	-	30.9	-
LLaVA-1.5-7B		76.6	62.0	54.4	70.4	46.0	1514.7
ToMe		75.2	59.5	55.9	68.7	41.1	1412.4
ToFu	Vicuna-7B	75.1	59.4	55.8	68.5	41.2	1405.3
DCT		67.8	56.2	55.7	65.8	26.3	1193.9
DiffRate		72.0	57.9	55.4	66.4	30.6	1341.0
PITOME		**75.4**	**59.9**	**55.9**	**69.0**	**43.0**	**1448.1**
LLaVA-1.5-13B		78.3	63.2	56.7	72.8	48.7	1522.6
ToMe		76.0	59.9	55.9	73.8	43.1	1470.3
ToFu	Vicuna-13B	76.1	60.1	56.1	74.0	43.0	1471.0
DCT		70.8	57.3	56.1	70.3	23.9	1355.8
DiffRate		73.4	58.5	54.6	70.6	32.8	1395.4
PITOME		**76.8**	**60.2**	**56.1**	**74.0**	**45.6**	**1490.1**

Table 4: **Inference time** of LLaVA-1.5-7B and LLaVA-1.5-13B models when running on *five V100-GPUs* and *five A100-GPUs*.

Model	VQA^{v2}↓	GQA ↓	VisWiz ↓	ScienceQA^I↓	TextVQA↓	MME↓
LLava-1.5-7B	09h:05m	10m:25s	04m:36s	01m:50s	10m:12s	02m:32s
ToMe	05h:38m	06m:34s	03m:26s	01m:07s	07m:37s	01m:24s
ToFu	05h:35m	06m:32s	03m:29s	01m:06s	07m:40s	01m:24s
DCT	05h:59m	06m:41s	03m:28s	01m:08s	08m:16s	01m:27s
DiffRate	05h:39m	06m:39s	03m:26s	01m:06s	07m:36s	01m:21s
PITOME	05h:44m	06m:37s	03m:26s	01m:07s	07m:37s	01m:23s
LLava-1.5-13B	13h:11m	13m:05s	07m:36s	04m:54s	15m:04s	02m:59s
ToMe	09h:28m	09m:35s	05m:58s	03m:31s	11m:48s	02m:16s
ToFu	09h:26m	09m32s	05m:58s	03m:26s	11m:45s	02m:15s
DCT	10h:02m	10m:53s	06m:46s	03m:45s	12m:57s	02m:34s
DiffRate	09h:33m	09:m44s	06m:01s	03m:37s	11m:52s	02m:18s
PITOME	09h:32m	09m:39s	06m:03s	03m:35s	12m:08s	02m:17s

Figure 4: **Off-the-shelf** performance of PITOME on **LLaVA-1.5-7B** with different compressing ratio r.

Table 5 and Figure 5 present our experimental results, comparing PITOME with recent works, including SOTA efficient transformers such as Swin-B [82], CSWin-B [82], MViT-B/L [83], MAE [79], and other token merging/pruning methods [84, 85, 41]. We observe that PITOME maintains high accuracy with an average performance drop of only 0.5% after reducing up to 44% of FLOPS (MAE-H), showcasing superior performance with comparable throughput. It is important to note that *dynamic pruning-based methods* such as A-ViT [85], Dynamic ViT [84], and SP-ViT [13] do not accelerate training speed due to using additional masks for padding tokens into a same dimension. On the retraining settings, we note that models compressed by PITOME also surpass merging/pruning methods by a large margin and approach the performance of the original models.

8

4.4 Text Classification

While previous studies have focused on benchmarking BSM-based algorithms within the vision or vision-language domain, we also extend experiments to the text domain, where input sequence lengths vary by sample. Specifically, we apply compression algorithms to the first three layers of the BERT model [86], reducing the number of tokens by 20% in each layer. Our experiments utilize the SST-2 dataset [87] with an average sequence length of 23.2 tokens and the IMDb dataset [88] with an average sequence length of 292.2 tokens.

As demonstrated in Table 6 and Figure 11 (Appendix), our findings indicate that PITOME performs better than other BSM-based baselines. Additionally, after retraining, the compressed BERT models achieve competitive records while significantly accelerating training speed compared to previous pruning methods such as PowER-BERT [89], Fisher [90], and LTP [91], as well as BERT-based efficient models like DistilBERT [92] and ALBERT [93]. Notably, we observe only a 0.4% performance drop on the IMDb dataset and even surpass the original BERT model by 0.3% on the SST-2 dataset. For detailed empirical results on this task, please refer to Appendix D.

Table 5: **Image Classification**: Performance of PITOME on Imagenet-1k, both off-the-shelf (OTS acc) and after retraining (Trained acc), across ViT backbones. We benchmark with different architectures and merging/pruning methods.

Type	Model	OTS Acc.	Trained Acc.	Flops ↓	Train speed up
Other models	Swin-B	n/a	84.0	15.4	×
	CSWin-B	n/a	84.2	15.0	×
	MViTv2-B	n/a	84.4	10.2	×
	MViTv2-L	n/a	85.3	42.1	×
merge	ToMe^{DEIT-T}	68.9	70.0	0.79	✓
	ToFu^{DEIT-T}	69.6	70.5	0.79	✓
	DCT^{DEIT-T}	67.6	68.7	0.79	✓
	DiffRate^{DEIT-T}	69.9	70.7	0.79	✓
	PITOME^{DEIT-T}	**70.8**	**71.6**	**0.79**	✓
	ViT^{DEIT-T}	72.3	72.3	1.2	×
prune	A-ViT^{DEIT-S}	n/a	78.6	2.9	×
	Dynamic-ViT^{DEIT-S}	n/a	79.3	2.9	×
	SP-ViT^{DEIT-S}	n/a	79.3	2.6	×
	E-ViT^{DEIT-S}	-	79.5	2.9	×
merge	ToMe^{DEIT-S}	77.7	79.4	2.9	✓
	ToFu^{DEIT-S}	77.8	79.6	2.9	✓
	DCT^{DEIT-S}	74.8	78.6	2.9	✓
	DiffRate^{DEIT-S}	76.8	79.5	2.9	✓
	PITOME^{DEIT-S}	**79.1**	**79.8**	**2.9**	✓
	ViT^{DEIT-S}	79.8	79.8	4.6	×
merge	ToMe^{MAE-L}	82.9	85.0	31.0	✓
	ToFu^{MAE-L}	83.8	85.1	31.0	✓
	DCT^{MAE-L}	82.8	84.4	31.0	✓
	DiffRate^{MAE-L}	83.2	85.3	31.0	✓
	PITOME^{MAE-L}	**84.6**	**85.3**	**31.0**	✓
	ViT^{MAE-L}	85.7	85.7	61.6	×
merge	ToMe^{MAE-H}	85.6	86.4	92.8	✓
	ToFu^{MAE-H}	85.8	86.4	92.8	✓
	DCT^{MAE-H}	84.3	86.0	92.8	✓
	DiffRate^{MAE-H}	85.9	86.6	92.8	✓
	PITOME^{MAE-H}	**86.4**	**86.7**	**92.8**	✓
	ViT^{MAE-H}	86.9	86.9	167.4	×

Table 6: **Text Classification**: PITOME vs other BERT-style compressed models and token pruning ones.

Dataset	Type	Model	Acc	Eval Flops ↑	Train Speed ↑
SST-2	compressed models	ALBERT	91.3	x1.0	x1.1
		DistiledBERT	91.1	x2.0	x1.7
	pruning +mask	BERT	91.4	x1.0	x1.0
		PowER-BERT	91.1	x2.5	x1.0
		Fisher	91.3	x1.6	x1.0
		LTP	91.3	x2.9	x1.0
	merging	PITOME	91.0	x1.9	x1.4
		ToMe	91.2	x1.9	x1.4
		ToFu	89.8	x1.9	x1.4
		DCT	90.7	x1.9	x1.4
		DiffRate	89.7	x1.9	x1.4
		PITOME	**91.7**	**x1.9**	**x1.4**
IMDb	compressed models	ALBERT	89.2	x1.0	x1.2
		DistiledBERT	93.0	x2.0	x1.9
	pruning +mask	BERT	94.0	x1.0	x1.0
		PowER-BERT	92.5	x2.7	x1.0
		TR-BERT	93.6	x2.3	x1.0
	merging	PITOME	93.2	x1.9	x1.8
		ToMe	93.3	x1.9	x1.8
		ToFu	92.6	x1.9	x1.8
		DCT	92.4	x1.9	x1.8
		DiffRate	92.4	x1.9	x1.8
		PITOME	**93.6**	**x1.9**	**x1.8**

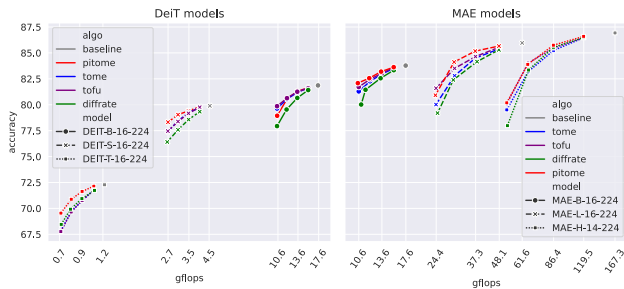



Figure 5: **Off-the-shelf results on Imagenet-1k. Zoom in for better view.**

4.5 PITOME Ablation Studies

Contributions of energy scores and related factors. To assess the performance of the components used in PITOME, we conduct the following settings: (i) PITOME without protecting important tokens by our energy in Step 2, i.e., using odd and even indices in sorted energy score array as two sets in BSM; (ii) PITOME where the merging process in Step 3 conducted on two randomly sets $\mathcal{A}$, $\mathcal{B}$ as baselines [15, 16] instead of leveraging ordered in sorted energy vectors $\mathbf{E}(.)$; (iii) PITOME without using our proposed energy score as in Eq(4) but utilizing other indicators like attention scores from the $[\text{CLS}]$ (PITOME w cls attn) token [19] or mean of attention scores; (iv) PITOME using a fixed of k removed token at each layer as ToMe [15] rather than a reducing ratio of r as our configuration.

We run experiments on image-text retrieval and text classification tasks, reporting the results in Table 7 for (i) and (ii), and in Figure 6 for (iii) and (iv). The results demonstrate that all factors contribute to

9

the performance of PITOME, with energy-based operations playing a particularly significant role. Additionally, reducing tokens with a ratio r effectively eliminates redundant tokens in early layers while preserving informative ones in later layers.

Margin m and α hyper-parameters. To validate the roles of these parameters in our energy score function in Eq.(4), we conduct ablation studies on image-text retrieval task with (v) adaptive margin m compared with a fixed value $m \in \{0.9, 0.45, 0.0, -1.0\}$ when varying the ratio r and (vi) given a fixed value of r, changing the smooth constant value α in $\alpha(\exp(x - m) - 1)$ with $x < m$. Results for these settings are summarized in Figure 7 and Table 8, respectively. We observe that while models with fixed tend to have the accuracy drop sharply when it is lower than some threshold, the adaptive margins achieve the best results across cases. We hypothesize that as the token space becomes sparser in deeper layers, PiToMe's fixed m approach likely assigns the same energy score to all tokens, making it difficult to isolate and protect tokens during merging. Table 8 also shows that $\alpha = 1.0$ is the best choice across margin values.

Further details, including additional ablation study results, visualizations (output merging, open-chat with LLaVa), and extra PITOME experiments, are provided in the Appendix.

Table 7: **Impact** of different settings in **Steps 2 and 3**.

Settings	Image-Text Retrieval		Text CLS.	
	ratio r	$Rsum$	ratio r	acc
PITOME w/o protecting tokens in step 2	0.925	568.9	0.6	86.99
	0.95	575.3	0.7	89.97
	0.975	578.2	0.8	91.67
PITOME using random split in step 3	0.925	567.9	0.6	87.17
	0.95	574.7	0.7	90.34
	0.975	578.4	0.8	91.73
PITOME	0.925	**573.4**	0.6	**89.20**
	0.95	**577.8**	0.7	**91.47**
	0.975	**580.1**	0.8	**93.26**

Table 8: **Impact** of the constant α on the image-text retrieval task. Results are in recall sum; higher is better.

ration r	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 0.$
0.85	**519.98**	518.66	515.90
0.875	**545.90**	544.22	542.54
0.90	**562.82**	562.42	561.92
0.925	**571.88**	571.10	570.62
0.95	**577.50**	577.43	577.40
0.975	**580.24**	579.82	579.76

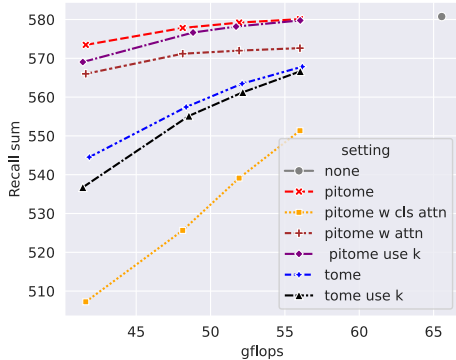



Figure 6: Ablation studies of PITOME.

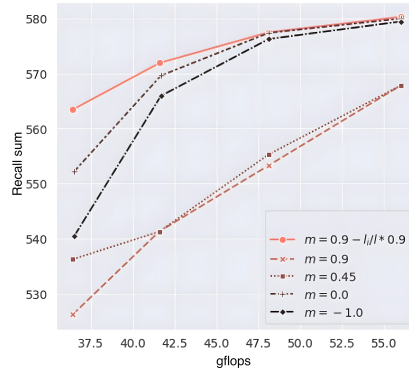



Figure 7: Ablation studies on adaptive margin m.

5   Conclusion

This paper introduces PITOME, a novel algorithm that employs energy concepts to protect informative tokens during the token merging process. Our algorithm matches the efficiency of heuristic merging methods while maintaining a theoretical connection to the spectral properties of the input token space. In experiments on image classification, image-text retrieval, and VQA with LLaVA-1.5 7B/13B, PITOME consistently outperforms recent token merging and pruning methods, given the equivalent runtime and memory usage.

Limitations and Future Works Although our focus has been on tasks using ViT encoders for a variety of applications, we believe it is important to extend PITOME to generative tasks such as image generation (e.g., stable diffusion) or segmentation. This extension, however, necessitates the development of an *unmerge mechanism* in the decoder, which remains an open question. Additionally, our energy score relies on a fully connected graph of input tokens, which can increase complexity as the input size grows. Constructing sparse graphs, therefore, might be beneficial for scaling in more challenging settings. Finally, designing a differentiable learning mechanism to optimize the reducing rate r for token merging could enhance robustness and versatility across different downstream tasks.

10

Acknowledgment

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Duy M. H. Nguyen. Duy M. H. Nguyen and Daniel Sonntag are also supported by the XAINES project (BMBF, 01IW20005), No-IDLE project (BMBF, 01IW23002), and the Endowed Chair of Applied Artificial Intelligence, Oldenburg University. Hoai-Chau Tran acknowledges the support from the AISIA Extensive Research Assistant Program 2023 (Batch 1) during this work and DFKI for supporting computing resources. TrungTin Nguyen acknowledges support from the Australian Research Council grant DP230100905. Ngan Le acknowledges funding support from the U.S. National Science Foundation (NSF) under Award No. OIA-1946391 and NSF EFRI BRAID 2223793. Binh T. Nguyen wants to thank the University of Science, Vietnam National University in Ho Chi Minh City for their support. Mathias Niepert acknowledges funding by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC and support by the Stuttgart Center for Simulation Science (SimTech).

References

[1] 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.

[2] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[3] Duy MH Nguyen, Hoang Nguyen, Nghiem Diep, Tan Ngoc Pham, Tri Cao, Binh Nguyen, Paul Swoboda, Nhat Ho, Shadi Albarqouni, Pengtao Xie, et al. Lvm-med: Learning large-scale self-supervised vision models for medical imaging via second-order graph matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[6] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[7] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[8] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *European Conference on Computer Vision*, pages 35–49. Springer, 2022.

[9] 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*, pages 10012–10022, 2021.

[10] Duy MH Nguyen, Hoang Nguyen, Truong TN Mai, Tri Cao, Binh T Nguyen, Nhat Ho, Paul Swoboda, Shadi Albarqouni, Pengtao Xie, and Daniel Sonntag. Joint self-supervised image-volume representation learning with intra-inter contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14426–14435, 2023.

11

[11] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022.

[12] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[13] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022.

[14] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2092–2101, 2023.

[15] 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.

[16] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[17] 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, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12890–12903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.721.

[18] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[19] 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)*, pages 17118–17128, 2023. doi: 10.1109/ICCV51070.2023.01574.

[20] Chaoya Jiang, Haiyang Xu, Chenliang Li, Ming Yan, Wei Ye, Shikun Zhang, Bin Bi, and Songfang Huang. Trips: Efficient vision-and-language pre-training with text-relevant image patch selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4084–4096, 2022.

[21] Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? *International Conference on Learning Representations*, 2023.

[22] R Balakrishnan. The energy of a graph. *Linear Algebra and its Applications*, 387:287–295, 2004.

[23] Ivan Gutman and Bo Zhou. Laplacian energy of a graph. *Linear Algebra and its applications*, 414(1):29–37, 2006.

[24] Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In *International conference on machine learning*, pages 3237–3246. PMLR, 2018.

[25] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph coarsening with preserved spectral properties. In *International Conference on Artificial Intelligence and Statistics*, pages 4452–4462. PMLR, 2020.

[26] Andreas Loukas. Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, 20(116):1–42, 2019.

12

[27] Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. *arXiv preprint arXiv:2305.17328*, 2023.

[28] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021.

[29] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[30] Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G Dimakis. Smyrf-efficient attention using asymmetric clustering. *Advances in Neural Information Processing Systems*, 33: 6476–6489, 2020.

[31] Valerii Likhosherstov, Krzysztof M Choromanski, Jared Quincy Davis, Xingyou Song, and Adrian Weller. Sub-linear memory: How to make performers slim. *Advances in Neural Information Processing Systems*, 34:6707–6719, 2021.

[32] Hongyu Ren, Hanjun Dai, Zihang Dai, Mengjiao Yang, Jure Leskovec, Dale Schuurmans, and Bo Dai. Combiner: Full attention transformer with sparse computation cost. *Advances in Neural Information Processing Systems*, 34:22470–22482, 2021.

[33] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, pages 396–414. Springer, 2022.

[34] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.

[35] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[37] Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR, 2020.

[38] Qihuang Zhong, Liang Ding, Juhua Liu, Xuebo Liu, Min Zhang, Bo Du, and Dacheng Tao. Revisiting token dropping strategy in efficient bert pretraining. *arXiv preprint arXiv:2305.15273*, 2023.

[39] Jungmin Yun, Mihyeon Kim, and Youngbin Kim. Focus on the core: Efficient attention via pruned token compression for document classification. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[40] Hongxu Yin, Arash Vahdat, Jose M 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*, pages 10809–10818, 2022.

[41] 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.

13

[42] Zhuoran Song, Yihong Xu, Zhezhi He, Li Jiang, Naifeng Jing, and Xiaoyao Liang. Cp-vit: Cascade vision transformer pruning via progressive sparsity prediction. *arXiv preprint arXiv:2203.04570*, 2022.

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

[44] Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *International Conference on Machine Learning*, 2024.

[45] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 12–21, 2023.

[46] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International conference on machine learning*, pages 874–883. PMLR, 2020.

[47] Junran Wu, Xueyuan Chen, Ke Xu, and Shangzhe Li. Structural entropy guided graph hierarchical pooling. In *International conference on machine learning*, pages 24017–24030. PMLR, 2022.

[48] Manoj Kumar, Anurag Sharma, Shashwat Saxena, and Sandeep Kumar. Featured graph coarsening with similarity guarantees. In *International Conference on Machine Learning*, pages 17953–17975. PMLR, 2023.

[49] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR 2016*, 2016.

[50] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph Coarsening with Preserved Spectral Properties. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4452–4462. PMLR, August 2020.

[51] Andreas Loukas. Graph Reduction with Spectral and Cut Guarantees. *Journal of Machine Learning Research*, 20(116):1–42, 2019.

[52] Christopher Brissette, Andy Huang, and George Slota. Spectrum Consistent Coarsening Approximates Edge Weights. *SIAM Journal on Matrix Analysis and Applications*, 44(3): 1032–1046, 2023. doi: 10.1137/21M1458119.

[53] Andreas Loukas and Pierre Vandergheynst. Spectrally Approximating Large Graphs with Smaller Graphs. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3237–3246. PMLR, July 2018.

[54] Hannu Toivonen, Fang Zhou, Aleksi Hartikainen, and Atte Hinkka. Compression of weighted graphs. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 965–973, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020566.

[55] Steve Butler. Interlacing for weighted graphs using the normalized Laplacian. *The Electronic Journal of Linear Algebra*, 16:90–98, 2007.

[56] 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, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

14

[57] 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.

[58] 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, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022.

[59] 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.

[60] 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, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[61] Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. *Thirty-First International Joint Conference on Artificial Intelligence (IJCAI))*, 2022.

[62] 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.

[63] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR, 2021.

[64] Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. Lightning-dot: Pre-training visual-semantic embeddings for real-time image-text retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 982–997, 2021.

[65] 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 *European conference on computer vision*, pages 104–120. Springer, 2020.

[66] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176, 2022.

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[68] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[69] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[70] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00686.

15

[71] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018. doi: 10.1109/CVPR.2018.00380.

[72] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[73] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8309–8318, 2019. doi: 10.1109/CVPR.2019.00851.

[74] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[75] Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Accelerating the development of large multimoal models, March 2024.

[76] 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, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023.

[77] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[78] IDEFICS. Introducing idefics: An open reproduction of state-of-the-art visual language model. https://huggingface.co/blog/idefics, 2023. Accessed: 2023-05-21.

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

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

[81] 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.

[82] 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.

[83] 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*, 2022.

[84] 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.

16

[85] 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.

[86] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of naacL-HLT*, 2019.

[87] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[88] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[89] Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR, 2020.

[90] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116, 2022.

[91] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 784–794, 2022.

[92] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[93] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[94] Henry Wolkowicz and George P. H. Styan. Bounds for eigenvalues using traces. *Linear Algebra and its Applications*, 29:471–506, 1980. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(80)90258-X.

[95] Hermann Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, December 1912. ISSN 1432-1807. doi: 10.1007/BF01456804.

17

Supplement to "Accelerating Transformers with Spectrum-Preserving Token Merging"

Contents

A Datasets Descriptions   **18**

B PITOME Algorithm   **19**
   B.1   Pseudo-Code Implementation   19
   B.2   PITOME Complexity Analysis   19
   B.3   Model complexity analysis   20

C Performance of ToMe with Different Token Merging Schedules   **20**

D Additional Experiments on Text Classification Task.   **21**

E Proof of Theorem 1   **22**
   E.1   Sketch of Proof   22
   E.2   Proof of Proposition 1   23
   E.3   Proof of Proposition 2   24
   E.4   Proof of Proposition 3   24
   E.5   Proofs of Technical Results   26

F Token Merging Outputs Visualization   **29**

G OpenChat with Compressed LLaVA-1.5 Model   **29**

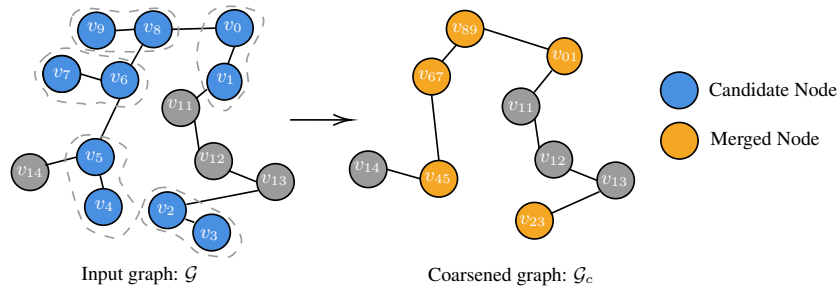



Figure 8: Token merging outputs can be seen as coarsened graph from an input graph.

A Datasets Descriptions

We present in Table 9 datasets used in our experiment. It is important to note that PITOME was run off-the-shelf on large datasets in VQA tasks such as VQA-v2 with $447.8k$ sample or GQQ with 12.5 questions, validating generalization and robustness of our merging mechanism.

18

Table 9: Brief statistic of all datasets used in this paper.

Task	Name	Modality	No. Train	No. Test
Image-Text retrieval	Flickr30k [59]	Vision, Text	29k images, each with 5 captions	1k images, each with 5 captions
	MS-COCO[60]	Vision, Text	approx 118k images, each with 5 captions	5k images, each with 5 captions
Visual Question Answering	VQA-v2[69]	Vision, Text	approx 443.7k questions	approx 447.8k questions
	GQA [70]	Vision, Text	approx 72k images with 943k questions	approx 10.2k images with 12.5k questions
	VisWiz [71]	Vision, Text	8k image and questions	approx 4.32k image and questions
	TextVQA[73]	Vision, Text	approx 28.4k images with 34.6k questions	approx 3.1k images with 4.2k questions
	MME [74]	Vision, Text	-	2375 images and questions
	ScienceQA[72]	Vision, Text	approx 4.11k image and questions	approx 2.11k image and questions
Text Classification	IMDb [88]	Text	25k movie reviews.	25k movie reviews.
	SST-2 [87]	Text	67,349 sentences	872 sentences
Image Classification	Imagenet-1k[81]	Vision	Approximately approx 1.28 million images	50k images (50 images per class)

B PITOME Algorithm

B.1 Pseudo-Code Implementation

The pseudo-code for our method is provided in Algorithm 1. Here $\mathbf{E}$ in line 3 is a vector that represents energy scores of all nodes calculated by the energy function 4. The final output is a MERGE(.) function, which serves as a lambda function that can be applied to any matrix $\hat{\mathbf{X}}^l$ at the l-th layer. The vector $\mathbf{m}$ is a vector that contains information about token sizes (i.e, the number of tokens being merged into each token).

Algorithm 1 PITOME Algorithm

1: **function** PITOME(remain token ratio: r, input graph: $\mathcal{G}(\mathcal{V},\mathcal{E})$) // Function to prepare for merging
2:    $k \leftarrow N - N \cdot r$ // Compute number of nodes to merge
3:    $\mathbf{s} \leftarrow$ argsort($\mathbf{E}$, descending=True)   // Compute energy scores
4:    $merge, protect \leftarrow \mathbf{s}[: 2 \cdot k], \mathbf{s}[2 \cdot k :]$ // Identify mergeable and protected nodes
5:    $\mathbf{n}_a, \mathbf{n}_b \leftarrow merge[:: 2], merge[1 :: 2]$ // Split mergeable nodes
6:    $\mathcal{E}_{merge} \leftarrow \mathcal{E}[\mathbf{n}_a][\mathbf{n}_b]$ // Get edge weights of mergeable nodes
7:    $\mathbf{n}_{dst} \leftarrow$ argmax($\mathcal{E}_{merge}$)   // Find closest neighbors
8:    **function** MERGE($\mathbf{X}$)   // Function to perform merging
9:        $\mathbf{X}_{protected} \leftarrow \mathbf{X}[protect, :]$ // Extract protected tokens
10:        $\mathbf{X}_{\mathcal{A}}, \mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}[\mathbf{n}_a, :], \mathbf{X}[\mathbf{n}_b, :]$ // Extract tokens in set $\mathcal{A}$ and $\mathcal{B}$
11:        $\mathbf{X}_{\mathcal{A}}, \mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{A}} \times \mathbf{m}[\mathbf{n}_a], \mathbf{X}_{\mathcal{B}} \times \mathbf{m}[\mathbf{n}_b]$ // Weighted average
12:        $\mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{B}}$.scatter_reduce($\mathbf{n}_{dst}, \mathbf{X}_{\mathcal{A}}$, mode = "sum")   // Merge tokens
13:        $\mathbf{X}_{\mathcal{B}} \leftarrow \mathbf{X}_{\mathcal{B}}/\mathbf{m}_{\mathcal{B}}$.scatter_reduce($\mathbf{n}_{dst}, \mathbf{m}_{\mathcal{A}}$, mode = "sum")   // Weighted average
14:        **return** cat($\mathbf{X}_{protected}, \mathbf{X}_{\mathcal{B}}$)   // Concatenate and return merged tokens
15:    **return** MERGE   // Return merging lambda function

B.2 PITOME Complexity Analysis

In algorithm 1, in line 3, the weighted graph is constructed through matrix multiplication, leading to a complexity of $\mathcal{O}\left(N^2h\right)$, where h is the dimension of input vectors. Next, the computed energy scores are sorted, which have the complexity of $\mathcal{O}\left(N\log(N)\right)$. Lastly, in line 7, the max operator for selecting the merge destination and the lambda $MERGE$ function, which performs tensor operations based on computed indices, lead to linear complexity. Combining these aspects, the overall time complexity of the PITOME function can be approximated as $\mathcal{O}\left(N^2h\right)$, considering the dominant factors contributing to computational cost. However, actual performance may vary depending on the specific PyTorch version and hardware utilization, with optimizations potentially altering these estimates.

19

(a) MSCOCO   (b) Flickr30k

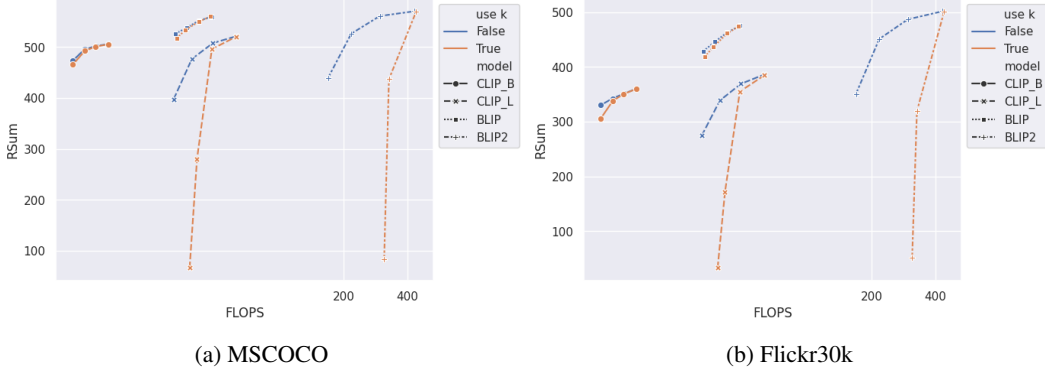



Figure 9: **Off-the-shelf** performance of all backbones for **image-text retrieval** task using different token merging schedules.

In the BSM algorithms used by ToMe, after the bipartition of tokens into two sets $\mathcal{A}$ and $\mathcal{B}$ using odd and even indices, each set has $N/2$ tokens, it also requires calculating a similarity matrix between tokens in these two sets. This operation also has the complexity of $\mathcal{O}\left(N^2 h\right)$. The similarity matrix is then sorted to get the top k tokens in set $\mathcal{A}$ with the highest similarity score used for merging, this operation also has the time complexity of $\mathcal{O}\left(N \log(N)\right)$, and the merging procedure is the same as our PITOME algorithm. So basically when compared to BSM, our algorithms have the same time complexity of $\mathcal{O}\left(N^2 h\right)$, although in practice the speed of ToMe is a little bit faster than PITOME (a few milliseconds), but our algorithms give a much better trade-off between speed and accuracy. For more details about the BSM algorithms please refer to the ToMe paper[15].

B.3 Model complexity analysis

In a standard Transformer layer, we have the time and space complexity to be $\mathcal{O}\left(N^2 h + Nh^2\right)$ and $\mathcal{O}\left(N^2 + Nh\right)$. In each layer ith of the model, we compress the number of tokens down to rN using PITOME algorithms with the complexity of $\mathcal{O}\left((r^{i-1}N)^2 h\right)$, so the layer i will enjoy the time and space complexity of $\mathcal{O}\left((r^{i-1}N)^2 h + r^i Nh^2 + (r^{i-1}N)^2 h\right)$ and $\mathcal{O}\left((r^{i-1}N)^2 + r^i Nh\right)$ *resp.* Vision language models like LLaVA directly use output tokens from ViT encoders. Let l be the number of layers in the ViT encoder since we utilize PITOME in each layer in the ViT encoder part; the LLM model will have the time and space complexity of $\mathcal{O}\left((r^l N_{\text{ViT}} + N_{\text{LLM}})^2 h + (r^l N_{\text{ViT}} + N_{\text{LLM}})h^2\right)$ and $\mathcal{O}\left((r^l N_{\text{ViT}} + N_{\text{LLM}})^2 + (r^l N_{\text{ViT}} + N_{\text{LLM}})h\right)$ which boosts the inference speed and saves a high amount of memory usage. The speedup could get even more impressive when used with higher batch size and image size. This also applies to models that use cross-attention modules for image text matching like BLIP and ALBEF, in which most of the computation expense comes from tokens encoded by the ViT model, the time and space complexity for the cross attention layers are $\mathcal{O}\left(r^l N_{\text{vision}} N_{\text{text}} h + N_{\text{text}} h^2\right)$ and $\mathcal{O}\left(r^l N_{\text{vision}} N_{\text{text}} + N_{\text{text}} h\right)$ *resp.*

C Performance of ToMe with Different Token Merging Schedules

In the original ToMe paper, the authors proposed a merging schedule that involves reducing tokens in each layer by a fixed k tokens per layer. However, as illustrated in Figure 6, we showed that this merging schedule is suboptimal for off-the-shell performance, and this section provides empirical results to confirm this claim.

In this experiment, we exclusively compare two versions of the BSM algorithms utilized in the ToMe paper: one that preserves a percentage r of tokens in each layer and another that reduces a fixed k tokens in each layer. However, for a more comprehensive comparison, we extensively apply these algorithms across 6 ViT backbones (DeiT-T, DeiT-S, DeiT-B, MAE-B, MAE-L, MAE-H) for image classification tasks using the Imagenet-1k dataset, as well as across 4 backbones for image-text retrieval tasks (CLIP-B, CLIP-L, BLIP, BLIP2) on FLickr30k and MSCOCO dataset.

20

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

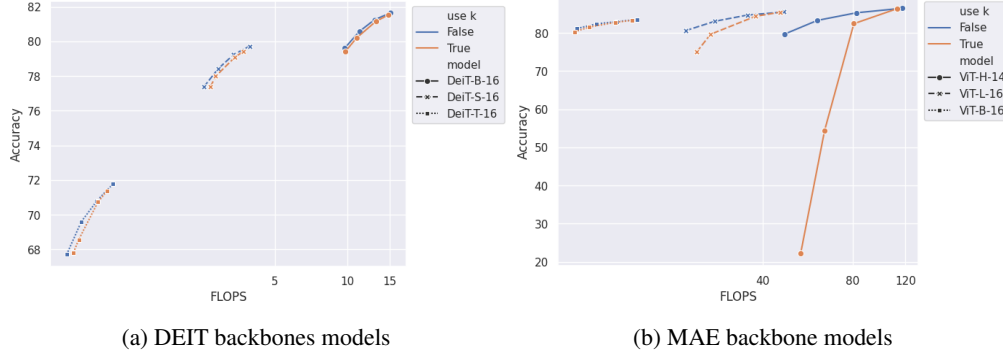



Figure 10: **Off-the-shell** performance of all backbones for **image classification** task using different token merging schedules.

From Figures 10b, 10a, 9b, and 9a, it is evident that, given the same FLOPS, the BSM version that uses the remaining percent r shows a clear advantage, outperforming the original schedule by a large margin. This gap becomes even more pronounced when benchmarked on large models like MAE-L, MAE-H, and BLIP-2. During the experiment, we also observed that models compressed using the ratio r tend to run a little faster since a large number of tokens are removed in earlier layers.

D   Additional Experiments on Text Classification Task.

This section comprehensively assesses the performance of PITOME against other baseline algorithms for sentiment text classification tasks. We conducted this experiment using the IMDb and SST-2, applying the compression algorithms to the first three layers of the model. To ensure consistency in our experimental results, we used two different backbones: BERT with 12 layers and DistilBERT with 6 layers. Since we only compress the first three layers, in addition to the baselines previously used throughout this paper, we introduced two additional baselines: ToFu-p and ToFu-m. ToFu-p uses BSM to prune tokens instead of merging them, while ToFu-m represents the original ToFu algorithm without pruning.

Table 10: **Performance** of PITOME versus baselines algorithms when training **BERT and Distiled-BERT when** retrained **from scratch**.

| Dataset | Model | Compress method | $r = 0.8$ | | | | $r = 0.75$ | | | | $r = 0.7$ | | | |
			accuracy	flops	eval speed	train speed	accuracy	flops	eval speed	train speed	accuracy	flops	eval speed	train speed
SST-2	BERT	ToMe	91.25 (-0.20)	x1.88	x1.14	x1.40	89.12 (-2.33)	x2.27	x1.22	x1.56	88.00 (-3.45)	x2.72	x1.30	x1.68
		ToFu	89.82 (-1.83)	x1.88	x1.15	x1.40	88.64 (-2.81)	x2.27	x1.22	x1.56	87.22 (-4.23)	x2.72	x1.29	x1.68
		DCT	90.66 (-0.79)	x1.81	x1.05	x1.12	89.31 (-2.14)	x2.13	x1.12	x1.43	87.76 (-3.69)	x2.72	x1.19	x1.61
		DiffRate	89.72 (-1.73)	x1.88	x1.14	x1.40	87.96 (-3.49)	x2.27	x1.21	x1.68	87.64 (-3.81)	x2.72	x1.22	x1.68
		PITOME	**91.72 (+0.27)**	**x1.88**	**x1.19**	x1.39	**90.28 (-1.17)**	**x2.27**	**x1.23**	x1.55	**88.67 (-2.78)**	**x2.72**	x1.25	x1.65
	DistiledBERT	ToMe	89.64 (-1.74)	x1.61	x1.01	x1.28	88.56 (-2.82)	x1.88	x1.03	x1.35	88.64 (-2.74)	x2.26	x1.07	x1.46
		ToFu	89.92 (-1.46)	x1.61	x1.02	x1.27	88.85 (-2.53)	x1.88	x1.04	x1.35	88.76 (-2.62)	x2.26	x1.07	x1.46
		DCT	89.59 (-1.79)	x1.51	x0.95	x1.16	88.08 (-3.30)	x1.61	x1.00	x1.22	87.70 (-3.68)	x1.88	x1.02	x1.32
		DiffRate	89.65 (-1.73)	x1.61	x1.01	x1.26	89.05 (-2.33)	x1.88	x1.02	x1.33	87.88 (-3.50)	x2.26	x1.07	x1.42
		PITOME	**90.31 (-1.07)**	**x1.61**	x1.02	x1.25	**89.28 (-2.10)**	**x1.88**	**x1.05**	x1.34	**88.89 (-2.49)**	**x2.26**	**x1.10**	x1.41
IMDb	BERT	ToMe	93.25 (-0.75)	x1.91	x1.68	x1.86	93.01 (-0.99)	x2.30	x1.88	x2.06	92.33 (-1.67)	x2.77	x2.10	x2.36
		ToFu	93.36 (-0.64)	x1.92	x1.68	x1.86	92.99 (-1.01)	x2.30	x1.89	x2.06	92.34 (-1.66)	x2.77	x2.09	x2.34
		DCT	92.39 (-1.61)	x1.90	x1.60	x1.79	92.22 (-1.78)	x2.30	x1.75	x1.94	91.31 (-2.69)	x2.77	x1.97	x2.25
		DiffRate	92.96 (-1.04)	x1.91	x1.68	x1.86	92.53 (-1.47)	x2.30	x1.86	x2.03	92.10 (-1.90)	x2.77	x2.08	x2.33
		PITOME	**93.52 (-0.48)**	**x1.91**	x1.66	x1.84	**93.27 (-0.73)**	**x2.30**	x1.84	x2.01	**92.74 (-1.26)**	**x2.77**	x2.08	x2.31
	DistiledBERT	ToMe	92.45 (-0.55)	x1.83	x1.47	x1.57	92.34 (-0.66)	x2.15	x1.57	x1.64	91.86 (-1.14)	x2.53	x1.69	x1.86
		ToFu	92.55 (-0.45)	x1.83	x1.47	x1.57	92.33 (-0.67)	x2.15	x1.57	x1.64	91.91 (-1.09)	x2.53	x1.69	x1.85
		DCT	92.38 (-0.62)	x1.61	x1.39	x1.50	91.87 (-1.13)	x1.79	x1.48	x1.62	91.69 (-1.31)	x2.19	x1.56	x1.77
		DiffRate	92.43 (-0.57)	x1.83	x1.44	x1.55	92.16 (-0.84)	x2.15	x1.57	x1.62	91.78 (-1.22)	x2.53	x1.63	x1.84
		PITOME	**92.71 (-0.29)**	**x1.83**	x1.43	x1.54	**92.55 (-0.45)**	**x2.15**	x1.54	x1.61	**92.06 (-0.94)**	**x2.53**	x1.62	x1.83

As demonstrated in Table 10 and Figure 11, our findings align with empirical results from previous tasks, indicating that PITOME consistently achieves superior performance compared to other baselines. Particularly noteworthy is the performance on the IMDb dataset with a large context length, where even after reducing FLOPs by 80%, models compressed by PITOME still maintain off-the-shelf accuracy above 85%, while other baseline algorithms see the off-the-shelf accuracy drop below 70%. Furthermore, Table 10 highlights that our algorithm can also facilitate better model

21

learning compared to other methods, achieving high accuracy that closely approaches that of the original model.

E   Proof of Theorem 1

E.1   Sketch of Proof

The proof sketch for Theorem 1 begins by defining coarsened and lifted versions of the original graph $\mathcal{G}$ (see Definitions 1 and 2) using the PITOME and ToMe algorithms. The goal is to demonstrate that the spectral distance between $\mathcal{G}$ and its PITOME-coarsened counterpart converges to zero, whereas the distance for ToMe remains bounded away from zero. The sketch proceeds by introducing Propositions 1, 2, and 3, which build toward the main result:

1. Proposition 1 establishes upper bounds on the edge weight differences between merged nodes under the PITOME and ToMe methods.

2. Proposition 2 employs standard mild assumptions to relate the cosine similarity among nodes within clusters, demonstrating that the upper bound error $\epsilon_{\text{PITOME}}^{(s)}$ between the edge weights of merged nodes in PITOME-coarsened graphs $\mathcal{G}_{\text{PITOME}}^{(n)}$ converges to 0, whereas this property does not hold for the ToMe-coarsened graphs $\mathcal{G}_{\text{ToMe}}^{(n)}$.

3. Proposition 3 completes the sketch by bounding the spectral distances of the coarsened graphs in terms of the edge weight differences from the earlier Propositions 1 and 2.

Each proposition is proven in detail within the appendix sections following this sketch. Note that Theorem 1 immediately follows the following Propositions 1, 2, and 3, which are proved *resp.* in Sections E.2, E.3, and E.4.

Proposition 1. *Suppose the graphs $\mathcal{G}_0^{(s)}$, $\mathcal{G}_{\text{PITOME}}^{(s)}$, and $\mathcal{G}_{\text{ToMe}}^{(s)}$ are coarsened from the original graph $\mathcal{G}$ by iteratively merging pairs of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0^{(s)} = \{\mathcal{V}_{0i}^{(s)}\}_{i \in [s]}$, the PITOME-partition $\mathcal{P}_{\text{PITOME}}^{(s)} = \{\mathcal{V}_{\text{PITOME}i}^{(s)}\}_{i \in [s]}$, defined by PITOME Algorithm 1, and the ToMe-partition [15, 16], $\mathcal{P}_{\text{ToMe}}^{(s)} = \{\mathcal{V}_{\text{ToMe}i}^{(s)}\}_{i \in [s]}$, for $s = N, \ldots, n+1$. We assume the following standard mild assumption:*

(A2). There exists a margin m s.t.

$$\cos(v_{a_s}, v_{b_s}) \geq m > \cos(v_{a_s}, v_{c_s}), \quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{c_s} \in \mathcal{V}_{0j}^{(s)}, \forall i \neq j \in [s]. \tag{6}$$

Then, the edge weights of merged nodes from PITOME Algorithm 1 and the ToMe-partition [15, 16] satisfy

$$\|\mathbf{W}[a_s,:] - \mathbf{W}[b_s,:]\|_1 \leq \epsilon^{(s)} \tag{7}$$

for some nonnegative upper bounds $\epsilon^{(s)}$, $s = N, N-1, \ldots, n+1$ defined as follows:

$$\epsilon^{(s)} = \begin{cases} 2(1 - \cos(v_{a_s}, v_{b_s})) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}, \forall i \in [s], \\ 3(1 - \beta) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i \neq j \in [s]. \end{cases} \tag{8}$$

Proposition 2. *We assume some standard mild assumptions:*

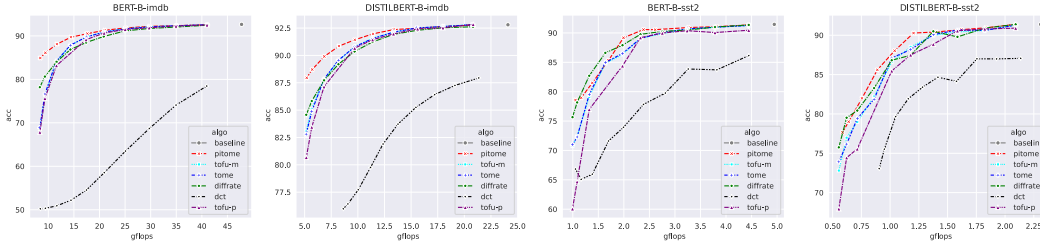



Figure 11: **Off-the-shelf** performance of various algorithms on the **text classification task**.

22

(A1). $\mathbb{E}[\cos(v_{a_s}, v_{b_s})] \to 1, \quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s].$

(A2). *There exists a margin m s.t.,* $\cos(v_{a_s}, v_{b_s}) \geq m > \cos(v_{a_s}, v_{c_s}), \quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{c_s} \in \mathcal{V}_{0j}^{(s)}, \forall i \neq j \in [s].$

(A3). *There is an order of cardinality in the true partition, without loss of generality, we assume* $N_1^{(s)} \geq N_2^{(s)} \geq \ldots \geq N_s^{(s)}$, *where* $N_i^{(s)} = |\mathcal{V}_{0i}^{(s)}|, \forall i \in [s].$

Here $\mathbb{E}(\cdot)$ stands for the expectation of the random variables that define the random events $M_{ab}^{(s)} := \{v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s]\}$, indicating that the two merged nodes belong to the same true partition $\mathcal{V}_{0i}^{(s)}$. Given the $\epsilon^{(s)}$ defined in equation (8), we obtain

1. *The upper bound error $\epsilon_{\mathrm{PITOME}}^{(s)}$ between the edge weights of merged nodes from PITOME-coarse $\mathcal{G}_{\mathrm{PITOME}}^{(n)}$ graphs converges to 0, i.e.,*

$$\epsilon_{\mathrm{PITOME}}^{(s)} \to 0 \text{ as } \mathbb{E}(\cos(v_{a_s}, v_{b_s})) \to 1, \text{ for any } s = N, \ldots, n+1.$$

2. *The upper bound error $\epsilon_{ToMe}^{(s)}$ between the edge weights of merged nodes from ToMe-coarse $\mathcal{G}_{ToMe}^{(n)}$ graphs converges to a non-negative constant $C^{(s)}$, with a high probability that $C^{(s)} > 0$, i.e.,*

$$\epsilon_{ToMe}^{(s)} \to C^{(s)} \text{ and } \mathbb{P}(C^{(s)} > 0) > 0 \text{ as } \mathbb{E}(\cos(v_{a_s}, v_{b_s})) \to 1, \text{ for any } s = N, \ldots, n+1.$$

Proposition 3. *Suppose the graphs $\mathcal{G}_0^{(s)}$, $\mathcal{G}_{\mathrm{PITOME}}^{(s)}$, and $\mathcal{G}_{ToMe}^{(s)}$ are coarsened from the original graph $\mathcal{G}$ by iteratively merging pairs of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0^{(s)} = \{\mathcal{V}_{0i}^{(s)}\}_{i \in [s]}$, the PITOME-partition $\mathcal{P}_{\mathrm{PITOME}}^{(s)} = \{\mathcal{V}_{\mathrm{PITOME}i}^{(s)}\}_{i \in [s]}$, defined by PITOME Algorithm 1, and the ToMe-partition [15, 16], $\mathcal{P}_{ToMe}^{(s)} = \{\mathcal{V}_{ToMei}^{(s)}\}_{i \in [s]}$, for $s = N, \ldots, n+1$. If the edge weights of merged nodes satisfy*

$$\|\mathbf{W}[a_s, :] - \mathbf{W}[b_s, :]\|_1 \leq \epsilon^{(s)}$$

for some nonnegative upper bounds $\epsilon^{(s)}$, $s = N, N-1, \ldots, n+1$. The spectral distances between the original $\mathcal{G} \equiv \mathcal{G}_0^{(N)}$ and the PITOME-coarse and ToMe-coarse $\mathcal{G}_{\mathrm{PITOME}}^{(n)}$ graphs are bounded as follows:

$$\max\{SD(\mathcal{G}, \mathcal{G}_{\mathrm{PITOME}}^{(n)}), SD(\mathcal{G}, \mathcal{G}_{ToMe}^{(n)})\} \leq \frac{3N}{2} \sum_{s=N}^{n+1} \epsilon^{(s)}.$$

E.2 Proof of Proposition 1

We want to prove that the edge weights of merged nodes from PITOME Algorithm 1 and the ToMe-partition [15, 16] satisfy

$$\|\mathbf{W}[a_s, :] - \mathbf{W}[b_s, :]\|_1 \leq \epsilon^{(s)}$$

for some nonnegative upper bounds $\epsilon^{(s)}$, $s = N, N-1, \ldots, n+1$.

Let us start with the simplest case, where we suppose the graph $\mathcal{G}_c$ is coarsened from $\mathcal{G}$ by merging one pair of nodes v_a and v_b using PITOME Algorithm 1 and the ToMe-partition. We can then demonstrate that the edge weights of the merged nodes satisfy certain conditions (for more details, refer to Appendix E.5.1):

$$\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1 \leq \epsilon \text{ for a nonnegative upper bound } \epsilon. \tag{9}$$

Recall that the graphs $\mathcal{G}_0^{(s)}$, $\mathcal{G}_{\mathrm{PITOME}}^{(s)}$, and $\mathcal{G}_{ToMe}^{(s)}$ are coarsened from the original graph $\mathcal{G}$ by iteratively merging pairs of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0^{(s)} = \{\mathcal{V}_{0i}^{(s)}\}_{i \in [s]}$, the PITOME-partition $\mathcal{P}_{\mathrm{PITOME}}^{(s)} = \{\mathcal{V}_{\mathrm{PITOME}i}^{(s)}\}_{i \in [s]}$, defined by PITOME Algorithm 1, and the ToMe-partition [15, 16], $\mathcal{P}_{ToMe}^{(s)} = \{\mathcal{V}_{ToMei}^{(s)}\}_{i \in [s]}$, for $s = N, \ldots, n+1$. By iteratively applying the merged 2-nodes inequalities (9), we obtain the desired upper bound as follows:

$$\|\mathbf{W}[a_s, :] - \mathbf{W}[b_s, :]\|_1 \leq \epsilon^{(s)}, \text{ for all } s = N, N-1, \ldots, n+1,$$

23

where

$$\epsilon^{(s)} = \begin{cases} 2(1 - \cos(v_{a_s}, v_{b_s})) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s], \\ 3(1 - \beta) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i \neq j \in [s]. \end{cases}$$

E.3 Proof of Proposition 2

Recall that in equation (8) of Proposition 1, we defined $\epsilon^{(s)}$, for $s = N, N-1, \ldots, n+1$, as follows:

$$\epsilon^{(s)} = \begin{cases} 2(1 - \cos(v_{a_s}, v_{b_s})) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s], \\ 3(1 - \beta) & \text{if } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i \neq j \in [s]. \end{cases}$$

Recall that we defined $\mathbb{E}(\cdot)$ as the expectation of the random variables that define the random events $M_{ab}^{(s)} := \{v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}, i \in [s]\}$, indicating that the two merged nodes belong to the same true partition $\mathcal{V}_{0i}^{(s)}$. Using the definition and the linearity of expectation, we have

$$\mathbb{E}\left(\epsilon^{(s)}\right) = 2[1 - \mathbb{E}\left(\cos(v_{a_s}, v_{b_s})\right)] + 3(1 - \beta)(1 - \mathbb{P}(M_{ab}^{(s)})). \tag{10}$$

Under Assumptions (A2) and (A3) and the energy-based merging mechanism of the PITOME algorithm, we can verify that $\mathbb{P}(M_{ab}^{(s)}) = 1$ via Lemma 2, which is proved in Appendix E.5.2. Refer to Lemma 3 for the ToMe approach (proof in Appendix E.5.2), noting that there exists a scenario where the random event $M_{ab}^{(s)}$ does not occur, leading to $\mathbb{P}(M_{ab}^{(s)}) < 1$. This leads to the desired results.

Lemma 2. *For the* PITOME *approach, it holds that* $\mathbb{P}(M_{ab}^{(s)}) = 1$, *i.e.,* *there exists a true partition* $\mathcal{V}_{0i}^{(s)}$, $i \in [s]$, *such that* $v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0i}^{(s)}$.

Lemma 3. *For the ToMe approach, there is a case where the random event* $M_{ab}^{(s)}$ *does not occur and therefore* $\mathbb{P}(M_{ab}^{(s)}) < 1$.

E.4 Proof of Proposition 3

Let us start with the simplest case, where we suppose the graph $\mathcal{G}_c$ is coarsened from $\mathcal{G}$ by merging one pair of nodes v_a and v_b. We can then prove the following property: If the edge weights of merged nodes satisfy

$$\|\mathbf{W}[a,:] - \mathbf{W}[b,:]\|_1 \leq \epsilon$$

then the spectral distance between the original and lifted graphs is bounded by

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_l\|_1 \leq \frac{3}{2}\epsilon. \tag{11}$$

Proof of the 2-node triangle inequality 11. Indeed, since the coarse graph $\mathcal{G}_c$ is coarsened by merging a pair of nodes v_a and v_b, the edge weights of the lifted graph $\mathcal{G}_l$ can be formulated as:

$$\mathbf{W}_l[i,j] = \begin{cases} \frac{\mathbf{W}[a,a] + 2\mathbf{W}[a,b] + \mathbf{W}[b,b]}{4} & \text{if } i,j \in \{a,b\}, \\ \frac{\mathbf{W}[a,j] + \mathbf{W}[b,j]}{2} & \text{if } i \in \{a,b\} \text{ and } j \notin \{a,b\}, \\ \frac{\mathbf{W}[i,a] + \mathbf{W}[i,b]}{2} & \text{if } i \notin \{a,b\} \text{ and } j \in \{a,b\}, \\ \mathbf{W}[i,j] & \text{otherwise.} \end{cases} \tag{12}$$

Here we use the fact that the adjacency matrix elements of $\mathcal{G}_l$ are given by

$$\mathbf{W}_l[i,j] = \frac{\sum_{v_i \in \mathcal{V}_i} \sum_{v_j \in \mathcal{V}_j} \mathbf{W}[i,j]}{|\mathcal{V}_i||\mathcal{V}_j|} \text{ where } v_i \in \mathcal{V}_i, \text{ and } v_j \in \mathcal{V}_j. \tag{13}$$

The corresponding node degree of $\mathcal{G}_l$ is

$$d_{li} = \begin{cases} \frac{d_a + d_b}{2} & \text{if } i \in \{a,b\}, \\ d_i & \text{otherwise.} \end{cases} \tag{14}$$

Using the result from Lemma 4, we can bound on the eigenvalue gap between $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_l$ via the perturbation matrix $\mathbf{E} = \mathbf{L} - \mathbf{L}_l = \mathbf{D} - \mathbf{D}_l + \mathbf{W}_l - \mathbf{W}$ as follows:

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_l\|_1 = \sum_{i=1}^{N} |\lambda_i - \lambda_{li}| \leq N\|\mathbf{E}\|_2 \leq N\sqrt{\|\mathbf{E}\|_\infty \|\mathbf{E}\|_1}. \tag{15}$$

24

Here the last inequality follows from the upper bound of the spectral norm $\|\mathbf{E}\|_2$ of [94], and we defined $\|\mathbf{E}\|_\infty = \max_i \sum_j |\mathbf{E}[i,j]|$, which is simply the maximum absolute row sum of the matrix, and $\|\mathbf{E}\|_1 = \max_j \sum_i |\mathbf{E}[i,j]|$, which is simply the maximum absolute column sum of the matrix.

Lemma 4 (Perturbations of eigenvalues: Weyl's inequality from [95]). *Let $\mathbf{L} \in \mathbb{R}^{N \times N}$ and $\mathbf{L}_l \in \mathbb{R}^{N \times N}$ be symmetric matrices. Then for all $i = 1, \ldots, N$,*

$$\max_{i=1,\ldots,N} |\boldsymbol{\lambda}_{li}(\mathbf{L}_l) - \boldsymbol{\lambda}_i(\mathbf{L})| \le \|\mathbf{L}_l - \mathbf{L}\|_2, \tag{16}$$

where $\|\mathbf{L}_l - \mathbf{L}\|_2$ is the induced 2-norm of $\mathbf{L}_l - \mathbf{L}$.

The equations (12), (13), and (14) yield the following identities:

$$\mathbf{W}_l[i,j] - \mathbf{W}[i,j] = \begin{cases} \frac{\mathbf{W}[a,a] + 2\mathbf{W}[a,b] + \mathbf{W}[b,b]}{4} - \mathbf{W}[i,j] & \text{if } i,j \in \{a,b\}, \\ \frac{\mathbf{W}[a,j] + \mathbf{W}[b,j]}{2} - \mathbf{W}[i,j] & \text{if } i \in \{a,b\} \text{ and } j \notin \{a,b\}, \\ \frac{\mathbf{W}[i,a] + \mathbf{W}[i,b]}{2} - \mathbf{W}[i,j] & \text{if } i \notin \{a,b\} \text{ and } j \in \{a,b\}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

and

$$\mathbf{D}[i,i] - \mathbf{D}_l[i,i] = \begin{cases} d_i - \frac{d_a + d_b}{2} & \text{if } i \in \{a,b\}, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Now we want to prove that $\|\mathbf{E}\|_\infty \le \epsilon$ and $\|\mathbf{E}\|_1 \le \epsilon$. Let us first focus on the first term $\|\mathbf{E}\|_\infty = \max_i \sum_j |\mathbf{E}[i,j]|$. Via the triangle inequality, we have

$$\|\mathbf{E}\|_\infty \le \|\mathbf{D} - \mathbf{D}_l\|_\infty + \|\mathbf{W}_l - \mathbf{W}\|_\infty. \tag{19}$$

Then, again using the triangle inequality and the assumption of Proposition 3, we obtain the first upper bound on the first term of the equation (19) as follows:

$$\|\mathbf{D} - \mathbf{D}_l\|_\infty = \max_i \sum_j |\mathbf{D}[i,j] - \mathbf{D}_l[i,j]| = \max_{i \in \{a,b\}} \left| d_i - \frac{d_a + d_b}{2} \right|$$

$$= \left| d_a - \frac{d_a + d_b}{2} \right| = \left| \frac{d_a - d_b}{2} \right|$$

$$= \frac{1}{2} \left| \sum_{j=1}^N \mathbf{W}_{aj} - \sum_{j=1}^N \mathbf{W}_{bj} \right| \le \frac{1}{2} \sum_{j=1}^N |\mathbf{W}[a,j] - \mathbf{W}[b,j]|$$

$$= \frac{1}{2} \|\mathbf{W}[a,:] - \mathbf{W}[b,:]\|_1 \le \frac{\epsilon}{2}. \tag{20}$$

For the second upper bound term from the equation (19), we consider two cases for each index $i = 1, \ldots, N$: $i \in \{a,b\}$ (**Case 1**) and $i \notin \{a,b\}$ (**Case 2**).

Case 1. Assume that $i \in \{a,b\}$. Since a and b have the same role, we can take $i = a$ without loss of generality. Using the equation (17), it holds that

$$\|\mathbf{W}_l - \mathbf{W}\|_\infty = \max_{i \in \{a,b\}} \sum_j |\mathbf{W}_l[i,j] - \mathbf{W}[i,j]|$$

$$= \left| \frac{2\mathbf{W}[a,b] + \mathbf{W}[b,b] - 3\mathbf{W}[a,a]}{4} \right| + \left| \frac{\mathbf{W}[a,a] + \mathbf{W}[b,b] - 2\mathbf{W}[a,b]}{4} \right| + \sum_{j \notin \{a,b\}} \left| \frac{\mathbf{W}[a,j] - \mathbf{W}[b,j]}{2} \right|$$

$$= \left| \frac{(\mathbf{W}[b,b] - \mathbf{W}[a,b]) + 3(\mathbf{W}[a,b] - \mathbf{W}[a,a])}{4} \right|$$

$$+ \left| \frac{(\mathbf{W}[a,a] - \mathbf{W}[a,b]) + (\mathbf{W}[a,b] - \mathbf{W}[b,b]) + 2(\mathbf{W}[b,b] - \mathbf{W}[a,b])}{4} \right|$$

$$+ \sum_{j \notin \{a,b\}} \left| \frac{\mathbf{W}[a,j] - \mathbf{W}[b,j]}{2} \right|.$$

25

Therefore, it holds that

$$
\begin{aligned}
\|\mathbf{W}_l - \mathbf{W}\|_\infty \leq{} & \frac{1}{4}|\mathbf{W}[a,b] - \mathbf{W}[b,b]| + \frac{3}{4}|\mathbf{W}[a,a] - \mathbf{W}[a,b]| \\
& + \frac{1}{4}|\mathbf{W}[a,a] - \mathbf{W}[a,b]| + \frac{1}{4}|\mathbf{W}[a,b] - \mathbf{W}[b,b]| + \frac{2}{4}|\mathbf{W}[a,b] - \mathbf{W}[b,b]| \\
& + \frac{1}{2}\sum_{j\notin\{a,b\}}|\mathbf{W}[a,j] - \mathbf{W}[b,j]| \ \text{(using triangle inequalities)} \\
={} & |\mathbf{W}[a,b] - \mathbf{W}[b,b]| + |\mathbf{W}[a,a] - \mathbf{W}[a,b]| + \frac{1}{2}\sum_{j\notin\{a,b\}}|\mathbf{W}[a,j] - \mathbf{W}[b,j]| \\
\leq{} & |\mathbf{W}[a,b] - \mathbf{W}[b,b]| + |\mathbf{W}[a,a] - \mathbf{W}[a,b]| + \sum_{j\notin\{a,b\}}|\mathbf{W}[a,j] - \mathbf{W}[b,j]| \\
={} & \|\|\mathbf{W}[a,:] - \mathbf{W}[b,:]\|\|_1 \leq \epsilon.
\end{aligned}
\tag{21}
$$

Case 2. Assume that $i \notin \{a,b\}$. Using the equation (17), we obtain

$$
\begin{aligned}
\|\mathbf{W}_l - \mathbf{W}\|_\infty &= \max_{i\notin\{a,b\}}\sum_{j\in\mathcal{V}}|\mathbf{W}_l[i,j] - \mathbf{W}[i,j]| \\
&= \left|\frac{\mathbf{W}[i,a] + \mathbf{W}[i,b]}{2} - \mathbf{W}[i,a]\right| + \left|\frac{\mathbf{W}[i,a] + \mathbf{W}[i,b]}{2} - \mathbf{W}[i,b]\right| \\
&= \left|\frac{\mathbf{W}[i,a] - \mathbf{W}[i,b]}{2}\right| + \left|\frac{\mathbf{W}[i,a] - \mathbf{W}[i,b]}{2}\right| \\
&= |\mathbf{W}[i,a] - \mathbf{W}[i,b]| \leq \|\|\mathbf{W}[a,:] - \mathbf{W}[b,:]\|\|_1 \leq \epsilon.
\end{aligned}
\tag{22}
$$

Combining (21) and (22), we obtain

$$
\|\mathbf{W}_l - \mathbf{W}\|_\infty \leq \epsilon.
$$

This leads to

$$
\|\mathbf{E}\|_\infty \leq \frac{3}{2}\epsilon
$$

when using the inequalities (19) and (20). Similarly, we can show that $\|\mathbf{E}\|_1 \leq \frac{3}{2}\epsilon$. Therefore, the equation (15) leads to the desired claim as follows:

$$
\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_l\|_1 \leq N\sqrt{\frac{3}{2}\epsilon\frac{3}{2}\epsilon} = \frac{3N}{2}\epsilon.
$$

$\square$

Recall that the graphs $\mathcal{G}_0^{(s)}$, $\mathcal{G}_{\text{PITOME}}^{(s)}$, and $\mathcal{G}_{\text{ToMe}}^{(s)}$ are coarsened from the original graph $\mathcal{G}$ by iteratively merging pairs of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0^{(s)} = \{\mathcal{V}_{0i}^{(s)}\}_{i\in[s]}$, the PITOME-partition $\mathcal{P}_{\text{PITOME}}^{(s)} = \{\mathcal{V}_{\text{PITOME}i}^{(s)}\}_{i\in[s]}$, defined by PITOME Algorithm 1, and the ToMe-partition [15, 16], $\mathcal{P}_{\text{ToMe}}^{(s)} = \{\mathcal{V}_{\text{ToMe}i}^{(s)}\}_{i\in[s]}$, for $s = N,\ldots,n+1$. By iteratively applying the triangle inequalities (11), the spectral distances between the original $\mathcal{G} \equiv \mathcal{G}_0^{(N)}$ and the PITOME-coarse $\mathcal{G}_{\text{PITOME}}^{(n)}$ and ToMe-coarse $\mathcal{G}_{\text{ToMe}}^{(n)}$ graphs are bounded as follows:

$$
\begin{aligned}
\max\{\text{SD}(\mathcal{G}, \mathcal{G}_{\text{PITOME}}^{(n)}), \text{SD}(\mathcal{G}, \mathcal{G}_{\text{ToMe}}^{(n)})\} &\leq \sum_{s=N}^{n+1}\max\{\text{SD}(\mathcal{G}_0^{(s)}, \mathcal{G}_{\text{PITOME}}^{(s-1)}), \text{SD}(\mathcal{G}_0^{(s)}, \mathcal{G}_{\text{ToMe}}^{(s-1)})\} \\
&\leq \frac{3N}{2}\sum_{s=N}^{n+1}\epsilon^{(s)}.
\end{aligned}
$$

E.5 Proofs of Technical Results

E.5.1 Proof of the merged 2-nodes inequality (9)

Recall that there exists the graph $\mathcal{G}_0$ coarsened from the original graph $\mathcal{G}$ by merging a pair of nodes v_{a_s} and v_{b_s} w.r.t. the true partition $\mathcal{P}_0 = \{\mathcal{V}_{0i}\}_{i\in[s]}$. Then, we have $\mathcal{V} = \mathcal{V}_{01} \cup \mathcal{V}_{02} \cup \ldots \cup \mathcal{V}_{0n}$, where $n = N - 1$. We also note that the energy score E_a of node v_a is calculated using the following

26

equation:

$$E_a = \frac{1}{N} \sum_{b \in \mathcal{N}(a)} f_m(\cos(v_a, v_b)), \quad f_m(x) = \begin{cases} x & \text{if } x \geq m \\ \alpha(\exp(x - m) - 1) & \text{otherwise} \end{cases}. \quad (23)$$

Inspired by the construction of the function f_m in equation (23) and according to Assumption (A2) in the inequalities (6), we can replace the smooth term $\alpha(\exp(x - m) - 1)$ by a constant β for simplicity. More precisely, if the nodes v_a and v_b are not considered true neighbours, *i.e.,* their cosine similarity is less than the margin m, then we can simplify the expression as follows:

$$\cos(v_a, v_b) = \beta := \sup_{v_a \in \mathcal{V}_{0i}, v_b \in \mathcal{V}_{0j}, i \neq j \in [N]} \alpha(\exp(\cos(v_i, v_j) - m) - 1) < 0. \quad (24)$$

To check the inequality (9), we examine the following term

$$\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1$$

in two cases:

Case 1. If two nodes v_a and v_b belong to the same true partition, say for example, $\mathcal{V}_{0i}, i \in [N]$, then since $n = N - 1$, we have $\mathcal{V}_{0i} = \{v_a, v_b\}$. Therefore, we can expand the previous 1-norm as follows:

$$\begin{aligned}
\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1 &= \sum_{k=1}^{N} |W[a, k] - W[b, k]| \\
&= \sum_{v_k \in \mathcal{V}_{0i}} |W[a, k] - W[b, k]| + \sum_{v_k \notin \mathcal{V}_{0i}} |W[a, k] - W[b, k]| \\
&= |W[a, a] - W[b, a]| + |W[a, b] - W[b, b]| \\
&= 2|1 - \cos(v_a, v_b)|. \quad (25)
\end{aligned}$$

Case 2. If $v_a \in \mathcal{V}_{0i}$ and $v_b \in \mathcal{V}_{0j}$ such that $i \neq j$, $i, j \in [N]$. Since $n = N - 1$, we have either $\mathcal{V}_{0i} = \{v_a, v_{0i}\}, \mathcal{V}_{0j} = \{v_b\}$ (Case 2.1) or $\mathcal{V}_{0i} = \{v_a\}, \mathcal{V}_{0j} = \{v_b, v_{0j}\}$ (Case 2.2). Let us first consider the Case 2.1, then it holds that

$$\begin{aligned}
\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1 &= \sum_{k=1}^{N} |W[a, k] - W[b, k]| \\
&= \sum_{v_k \in \mathcal{V}_{0i}} |W[a, k] - W[b, k]| + \sum_{v_k \in \mathcal{V}_{0j}} |W[a, k] - W[b, k]| + \sum_{v_k \notin \mathcal{V}_{0i}, v_k \notin \mathcal{V}_{0j}} |W[a, k] - W[b, k]| \\
&= |W[a, a] - W[b, a]| + |W[a, 0i] - W[b, 0i]| + |W[a, b] - W[b, b]| \\
&= (1 - \beta) + |1 - \cos(v_a, v_{0i}) - (1 - \beta)| + (1 - \beta) \\
&= 2(1 - \beta) + |\cos(v_a, v_{0i}) - \beta| \leq 3(1 - \beta). \quad (26)
\end{aligned}$$

Now, let us first consider the Case 2.2, then it holds that

$$\begin{aligned}
\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1 &= \sum_{k=1}^{N} |W[a, k] - W[b, k]| \\
&= \sum_{v_k \in \mathcal{V}_{0i}} |W[a, k] - W[b, k]| + \sum_{v_k \in \mathcal{V}_{0j}} |W[a, k] - W[b, k]| + \sum_{v_k \notin \mathcal{V}_{0i}, v_k \notin \mathcal{V}_{0j}} |W[a, k] - W[b, k]| \\
&= |W[a, a] - W[b, a]| + |W[a, 0j] - W[b, ji]| + |W[a, b] - W[b, b]| \\
&= (1 - \beta) + |1 - \cos(v_a, v_{0j}) - (1 - \beta)| + (1 - \beta) \\
&= 2(1 - \beta) + |\cos(v_a, v_{0j}) - \beta| \leq 3(1 - \beta). \quad (27)
\end{aligned}$$

Combining the previous equations (25), (26), and (27), we have

$$\|\mathbf{W}[a, :] - \mathbf{W}[b, :]\|_1 \leq \begin{cases} 2|1 - \cos(v_a, v_b)| & \text{if } v_a, v_b \in \mathcal{V}_{0i}, \forall i \in [N], \\ 3(1 - \beta) & \text{if } v_a \in \mathcal{V}_{0i}, v_b \in \mathcal{V}_{0j}, \forall i \neq j \in [N]. \end{cases}. \quad (28)$$

27

E.5.2 Proof of Lemma 2

Recall the following Assumptions (A2) and (A3):

(A2). There exists a margin m s.t., $\cos(v_{a_s}, v_{b_s}) \geq m > \cos(v_{a_s}, v_{c_s})$, $\quad \forall v_{a_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{b_s} \in \mathcal{V}_{0i}^{(s)}, \forall v_{c_s} \in \mathcal{V}_{0j}^{(s)}, \forall i \neq j \in [s]$.

(A3). There is an order of cardinality in the true partition, without loss of generality, we assume $N_1^{(s)} \geq N_2^{(s)} \geq \ldots \geq N_s^{(s)}$, where $N_i^{(s)} = |\mathcal{V}_{0i}^{(s)}|, \forall i \in [s]$.

In PITOME approach, using Assumption (A2), given any two nodes $v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i < j \in [s]$, the energy scores E_{a_s} and E_{b_s} of nodes v_{a_s} and v_{b_s}, respectively, are simplified as follows:

$$E_{a_s} = \frac{1}{N} \sum_{c_s \in \mathcal{N}(v_{a_s})} f_m(\cos(v_{a_s}, v_{c_s})), \text{ where } f_m(x) = \begin{cases} x & \text{if } x \geq m \\ \beta & \text{if } x < m \end{cases}$$

$$= \frac{1}{N} \sum_{v_{c_s} \in \mathcal{V}_{0i}^{(s)}} \cos(v_{a_s}, v_{c_s}) + \frac{N - N_i^{(s)}}{N} \beta \geq \frac{N_i^{(s)} m + (N - N_i^{(s)})\beta}{N}, \tag{29}$$

$$E_{b_s} = \frac{1}{N} \sum_{v_{c_s} \in \mathcal{V}_{0j}^{(s)}} \cos(v_{b_s}, v_{c_s}) + \frac{N - N_j^{(s)}}{N} \beta \leq \frac{N_j^{(s)} + (N - N_j^{(s)})\beta}{N}. \tag{30}$$

Given the choice of the universal margin m as follows: $m = \max\left\{ \frac{N_j^{(s)}}{N_i^{(s)}} : i < j \in [s] \right\}$, it holds that $E_{a_s} \geq E_{b_s}$. Indeed, this is guaranteed as long as we have

$$\frac{N_i^{(s)} m + (N - N_i^{(s)})\beta}{N} \geq \frac{N_j^{(s)} + (N - N_j^{(s)})\beta}{N}. \tag{31}$$

This is equivalent that

$$m \geq \frac{N_j^{(s)} + (N - N_j^{(s)})\beta - (N - N_i^{(s)})\beta}{N_i^{(s)}} = \frac{N_j^{(s)} + (N_i^{(s)} - N_j^{(s)})\beta}{N_i^{(s)}} \geq \frac{N_j^{(s)}}{N_i^{(s)}}. \tag{32}$$

Using Assumption (A3) and this choice of universal margin m, it holds that

$$E_{a_s} \geq E_{b_s}, \text{ for any } v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i < j \in [s]. \tag{33}$$

Recall that in PITOME approach, we use the ordered energy-based for the bipartite soft matching where we defined two set $\mathcal{A}$ and $\mathcal{B}$ with $|\mathcal{A}| = |\mathcal{B}| = k$ as follows:

$$\mathcal{A} = \left\{ v_1^e, v_3^e, \ldots, v_{2k-1}^e \right\}, \mathcal{B} = \left\{ v_2^e, v_4^e, \ldots, v_{2k}^e \right\}, \tag{34}$$

where the nodes $v_i^e, i \in [2k]$, are sorted in decreasing order based on their energy scores, i.e., $E_{v_i^e} > E_{v_{i+1}^e}, \forall i \in [2k-1]$.

We return to the proof of the Lemma 2 by contradiction. Assume that $v_{a_s} \in \mathcal{V}_{0i}^{(s)}, v_{b_s} \in \mathcal{V}_{0j}^{(s)}, i \neq j \in [s]$. Without loss of generality, we assume that $v_{a_s} = v_1^e \in \mathcal{V}_{01}^{(s)}$ and $N_1^{(s)} = |\mathcal{V}_{01}^{(s)}| > 1$. Note that PITOME algorithm selects $v_{b_s} \in \mathcal{B}$ such that

$$v_{b_s} = \arg\max v_{c_s \in \mathcal{B}} \cos(v_{a_s}, v_{c_s}).$$

This is equivalent that for any $v_{c_s} \in \mathcal{B}$, it holds that:

$$\cos(v_{a_s}, v_{c_s}) \leq \max v_{c_s \in \mathcal{B}} \cos(v_{a_s}, v_{c_s}) = \cos(v_{a_s}, v_{b_s}) < m.$$

Assumption (A2) implies that $v_{2i}^e \notin \mathcal{V}_{01}^{(s)}, \forall i \in [k]$. Since we have $N_1^{(s)} = |\mathcal{V}_{01}^{(s)}| > 1$, there exists at least one node in $\mathcal{A} \cap \mathcal{V}_{01}^{(s)}$, say for example v_3^e. Using (34), we have $E_{v_2^e} \geq E_{v_3^e}$, which contradicts (33) where we have $E_{v_3^e} > E_{v_2^e}$ since $v_3^e \in \mathcal{V}_{01}^{(s)}$ and $v_2^e \in \mathcal{V}_{0j}^{(s)}, j > 1$.

E.5.3 Proof of Lemma 3

On the contrary, in the Bipartite Soft Matching algorithm from the ToMe approach [15], the authors divide the tokens into two distinct sets $\mathcal{A}$ and $\mathcal{B}$ and merge the top k similar tokens using some partitioning style like sequential, alternating and random without considering the ordered energy-

28

based Bipartite Soft Matching like ours. This leads to a case where the random event $M_{ab}^{(s)}$ does not occur and therefore $\mathbb{P}(M_{ab}^{(s)}) < 1$. Indeed, this case happens when all the nodes from the true partition $\mathcal{V}_{01}^{(s)}$ are divided into the same set $\mathcal{A}$. Therefore, the Bipartite Soft Matching algorithm has to choose and select the node $v_{b_s} \in \mathcal{V}_{0j}^{(s)}, j > 1$, for merging. Actually, this case arises when all the nodes from the true partition $\mathcal{V}_{01}^{(s)}$ are distributed into the same set $\mathcal{A}$. Therefore, the Bipartite Soft Matching algorithm has to select the node $v_{b_s} \in \mathcal{V}_{0j}^{(s)}, j > 1$, for merging.

F   Token Merging Outputs Visualization

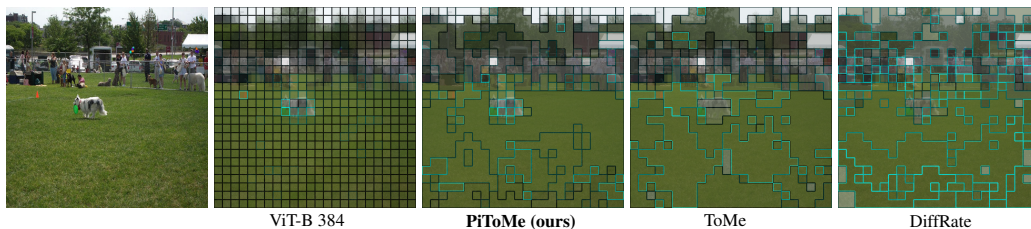
To better illustrate the effectiveness of PITOME, we have incorporated many visualizations show-casing examples randomly sampled from the MSCOCO dataset, as depicted in Figures 12a. In this analysis, we utilized the BLIP backbone and configured the reduction percentage to $r = 0.85$. These visualizations portray the final representation of all remaining tokens in the last layers. Notably, tokens outlined with bolder cyan borders signify higher attention scores from the classification (CLS) token. Apparently, unlike ToMe and DiffRate, PITOME adeptly approximates the spectral spectrum of the original token space, thereby preserving the distribution of all tokens. Consequently, PITOME demonstrates a lower level of information distortion due to reduced mis-merged tokens, thus retaining attention maps for critical information in the final layer. As illustrated in Figures 12a,12b,12c,12d,12e the attention map in the final layer of the model compressed by PITOME exhibits the closest resemblance to that of the baseline model.

G   OpenChat with Compressed LLaVA-1.5 Model

This section is dedicated to showcasing sample responses of LLaVA-1.5-7B when compressed using different algorithms. Here, we set the compression rate to $r = 0.9$ for each CLIP layer. All sample answers are given in tables 11, 13, 12. Texts that are highlighted by blue color indicate false information given by the model. From these sample answers, it is evident that, unlike other BSM-based approaches, our algorithm PITOME not only facilitates quicker model responses but also retains its capability to recognize objects, understand contextual information in the image, and provide correct answers closely aligned with those of the original LLaVA-1.5-7B model.

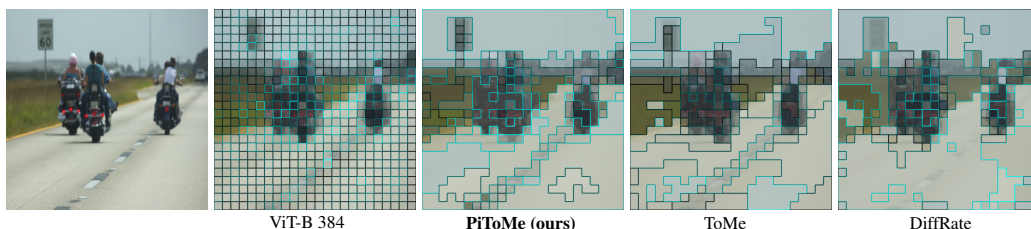
29

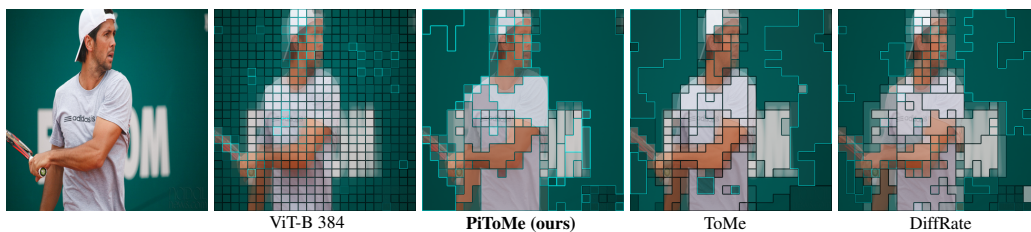
Figure 12: Visualizations using random images sampled from the MSCOCO dataset.

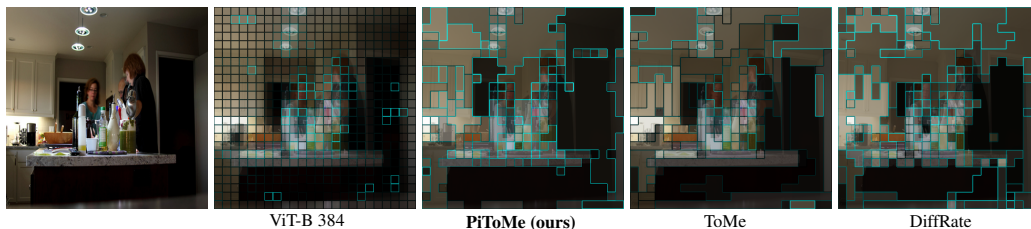
ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate

(a) A white dog catching a novelty flying disc in a competition.

ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate

(b) A woman riding a horse jumping it over obstacles.

ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate

(c) Three different motorcycle couples riding down a road.

ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate

(d) A man holding a tennis racquet on a tennis court.

ViT-B 384   **PiToMe (ours)**   ToMe   DiffRate

(e) A group of people who are around a kitchen counter.

30

Table 11

USER: Describe what you see

Model	Answer
LLaVA-1.5-7B	In the image, a baseball game is taking place with a batter standing at home plate, holding a baseball bat and preparing to swing. The catcher is positioned behind the batter, ready to catch the ball if the batter misses. There are several other players on the field , including teammates and opponents, all focused on the ongoing play. Some of the players are standing closer to the batter, while others are spread out across the field. The scene captures the excitement and anticipation of a crucial moment in the game.
LLaVA-1.5-7B + PITOME	In the image, a baseball player is in the middle of swinging a baseball bat, attempting to hit the ball during a game. There are several other people in the scene, some of whom are wearing baseball gloves , indicating that they are also participating in the game. The presence of multiple players and the action of the batter suggest that this is an exciting moment in the game, with the batter trying to make contact with the ball and potentially score a run for their team.
LLaVA-1.5-7B +ToMe	In the image, a group of people is gathered on a baseball field, with some of them holding baseball bats . It appears that they are either practicing or participating in a baseball game. There are several people standing in various positions on the field , and some of them are closer to the foreground while others are further away. The scene captures the excitement and camaraderie often associated with baseball, as people come together to enjoy the game and improve their skills.
LLaVA-1.5-7B + ToFu	In the image, there is a baseball game in progress with multiple players on the field . One player is holding a baseball bat and is in the process of swinging it, likely attempting to hit the ball. Another player is holding a baseball glove, ready to catch the ball if it is hit. There are several other players on the field, some of them wearing baseball gloves as well . A bench can be seen in the background, likely for players to sit on when they are not actively participating in the game. The scene captures the excitement and action of a baseball game.
LLaVA-1.5-7B + DiffRate	In the image, a group of people is playing a game of baseball on a field. There are several players on the field, with some of them holding baseball bats and wearing baseball gloves . The players are actively engaged in the game, with one player holding a baseball bat and another player wearing a baseball glove, ready to catch the ball. The other players are also positioned around the field, either waiting for their turn to bat or preparing to field the ball . The scene captures the excitement and energy of a live baseball game.

31

Table 12

USER: Describe what you see.

Model	Answer
LLaVA-1.5-7B	In the image, a man is standing on a folding chair in the back of a yellow taxi cab. He is holding a blue shirt, possibly drying it or preparing to put it on. The taxi is driving down a busy street with multiple cars around it, including another taxi in front of it. The man's position on the chair and the presence of the shirt suggest that he might be a street vendor or a worker who is using the taxi as a mobile workspace.
LLaVA-1.5-7B + PIToMe	In the image, a man is standing on the back of a moving car, holding a clothesline with clothes hanging from it. The car appears to be a taxi cab, and the man is wearing a yellow shirt. The scene seems to be taking place in a busy city, as there are multiple cars visible in the background. The man's unusual activity of hanging clothes from the back of a moving car likely attracts attention and curiosity from passersby.
LLaVA-1.5-7B +ToMe	The image shows a man standing on the back of a taxi cab, holding a large umbrella. The taxi is driving down a street, and the man appears to be using the umbrella to shield himself from the elements. In addition to the taxi, there are several other vehicles in the scene, including a truck and a car. The man on the taxi is the main focus of the image, and his actions suggest that he is trying to protect himself from rain or other weather conditions.
LLaVA-1.5-7B + ToFu	The image depicts a busy city street scene with a man standing on the back of a truck, possibly performing a stunt or participating in a parade. The truck is positioned in the middle of the street, and the man appears to be the main focus of the scene. There are several other people in the image, some of whom are walking or standing near the truck. Additionally, there are multiple cars parked or driving along the street, and a traffic light can be seen in the background. The overall atmosphere suggests an urban environment with a lively and bustling atmosphere.
LLaVA-1.5-7B + DiffRate	The image features a man wearing a yellow shirt, standing next to a yellow car. He appears to be holding onto the car, possibly for support or to steady himself. The car is parked on a street, and there are other cars visible in the background. In addition to the man and the car, there are two other people in the scene, one standing further back and another closer to the foreground. A handbag can also be seen placed on the ground near the center of the image.

32

Table 13

USER: Describe what is happening?

Model	Answer
LLaVA-1.5-7B	In the image, a woman is riding a horse and jumping over a hurdle, which is a part of an equestrian competition. The horse and rider are in the middle of the jump, showcasing their skills and coordination. There are several other people in the scene, some of whom may be fellow competitors or spectators. The presence of multiple people and the focus on the horse and rider suggests that this is a competitive event or practice session.
LLaVA-1.5-7B + PɪToMᴇ	In the image, a person is riding a horse while jumping over an obstacle, likely during an equestrian event or competition. The horse and rider are in the middle of the jump, showcasing their skill and coordination. There are several other people in the scene, some of whom might be fellow competitors, trainers, or spectators. One person is standing close to the rider on the horse , while others are scattered around the area. The overall atmosphere of the image suggests an exciting and challenging equestrian event.
LLaVA-1.5-7B +ToMe	In the image, a person is riding a horse and jumping over an obstacle, possibly a hurdle or a small hill . The horse and rider are in the middle of the action, showcasing their skill and coordination. The scene captures the excitement and challenge of equestrian sports, such as show jumping or eventing.
LLaVA-1.5-7B + ToFu	In the image, a person is riding a horse while standing in a corral . The horse and rider are positioned near a fence , which could be a part of a fenced-in area or a stable . The scene captures the bond between the rider and the horse as they interact and move together within the corral.
LLaVA-1.5-7B + DiffRate	In the image, there are two people riding horses , likely participating in a horseback riding lesson or practice session. The horses and riders are positioned next to each other, with one horse and rider on the left side and the other horse and rider on the right side of the scene . The riders are wearing helmets for safety, and the horses appear to be well-trained and cooperative. The scene captures the essence of a horseback riding activity, where the riders are learning to control and communicate with their horses effectively.

33

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provided experiments to support our claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included it at the end of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

34

Justification: All assumptions are already stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: we provided experiment descriptions in the main paper and appendix. Furthermore, we will release our GitHub implementation if the paper is accepted.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

35

Answer: [Yes]

Justification: We will publish the code when the paper is accepted or through an anonymous link if some reviewers ask for it.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provided details for experiments in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We conduct experiments on diverse datasets and follow the protocol used by previous works for fair comparisons.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

36

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: This information is included in our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our paper follows the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our work helps to reduce the carbon footprint when training large models using ViT. There is no negative societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

37

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not pose such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited papers and resources used in our experiment.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

38

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We don't have experiments involving crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

39