Leveraging local similarity for token merging in Vision Transformers

Karim Haroun$^{1,2[0009-0000-6972-6019]}$, Jean Martinet$^{2[0000-0001-8821-5556]}$, Karim Ben Chehida$^{1[0000-0002-5959-1832]}$, and Thibault Allenet$^{1[0009-0003-0810-3338]}$

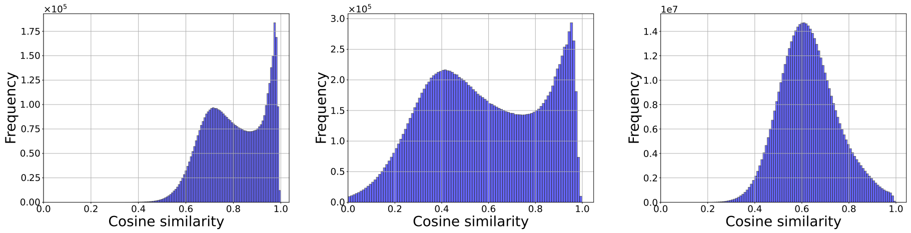
[1] French Alternative Energies and Atomic Energy Commission (CEA), France
[2] Université Côte d'Azur, CNRS, I3S, France
{karim.haroun, karim.benchehida, thibault.allenet}@cea.fr
{karim.haroun, jean.martinet}@univ-cotedazur.fr

Abstract. Vision Transformers (ViTs) have shown promising results in computer vision tasks, challenging CNN architectures on image classification, segmentation and object detection. However, their quadratic complexity $O(N^2)$, where N is the token sequence length, hinders their deployment on edge devices. To tackle this challenge, researchers have proposed various compressing schemes that exploit sparsity and redundancies. In this paper, we focus on one of these strategies, named token merging, which consists of dynamically and progressively combining similar tokens during inference, leading to computational savings. Most of the proposed methods compute similarities between all tokens before picking the highest score that leads to the merging decision. This contradicts the intuition that spatially close tokens are more similar than distant ones. In our paper, we show that the distribution of cosine similarity scores of adjacent token pairs is higher than the distribution of similarity scores of distant tokens. Based on this observation, we propose LoTM, a Local Token Merging approach where we constrain the merging window to a pair of adjacent tokens only. Our model is evaluated on a classification task using the ImageNet-1K dataset, as it outperforms most state-of-the-art approaches in accuracy given the same computational budget without requiring further training.

Keywords: Deep learning · Vision Transformers · Adaptive inference · Neural network compression.

1 Introduction

The remarkable success of Transformers in Natural Language Processing (NLP), as demonstrated by Vaswani et al. [26], has captured the attention of researchers in computer vision. As a result, numerous efforts have been made to adopt the Transformer as an alternative deep neural network architecture for computer vision tasks. A pioneering example is Vision Transformer (ViT) by Dosovitskiy et al. [7], which utilizes a fully Transformer-based architecture for image classification. ViT works by dividing an image into several local patches, thus creating



(a) (b) (c)

Fig. 1: Distribution of cosine similarity between pairs of tokens over 50k instances of ImageNet-1K validation set. (a) The similarity is computed between two adjacent tokens, i.e., spatially related tokens. (b) The similarity is computed between non-overlapping windows of four tokens. (c) The similarity is computed between all tokens of the sequence.

a visual sequence for the Transformer to process. Its self-attention mechanism evaluates the relationships between these patches, aggregating their information to form a high-level representation suitable for image recognition.

Following ViT, many variants have been developed [20, 25, 27, 28, 30]. For instance, DeiT [25] achieved state-of-the-art performance on the ImageNet-1K benchmark [5] without requiring pre-training on an extensive dataset like JFT-300M [24]. T2T-ViT [31], also trained from scratch on ImageNet-1K, enhances local and global information exchange via a T2T module before the transformer encoder. CrossViT [3] leverages multi-scale features within the vision transformer, while TNT [8] delves into the attention mechanisms within individual patches, breaking them down into smaller components. CrossFormer [33] introduces a novel approach using patches of varying sizes to establish cross-scale attention, showing significant improvements across key vision benchmarks.

These advancements have established vision transformers as strong alternatives to traditional Convolutional Neural Networks (CNNs) [15] in vision tasks, with notable examples including the Swin Transformer [18] and Twins [4]. However, compared to CNNs, vision transformers often do not significantly reduce computational costs and sometimes even require more resources.

Given the quadratic computational complexity of transformers $O(N^2)$, a logical approach is to reduce the number of tokens in the transformer to potentially speed up processing. Indeed, by selectively reducing the number of tokens propagated through the network, the goal is to achieve computational efficiency without compromising information integrity. Recent advancements have introduced two main strategies: The first involves pruning tokens based on their importance, typically identified through analysis of the attention scores of the CLS class embedding, where essential tokens are retained while less critical ones are discarded. The second strategy suggests merging tokens based on their similarity, where it fuses top-k most similar tokens at each Transformer layer. Moreover, recent works have proposed using both compression paradigms at once [2, 14].

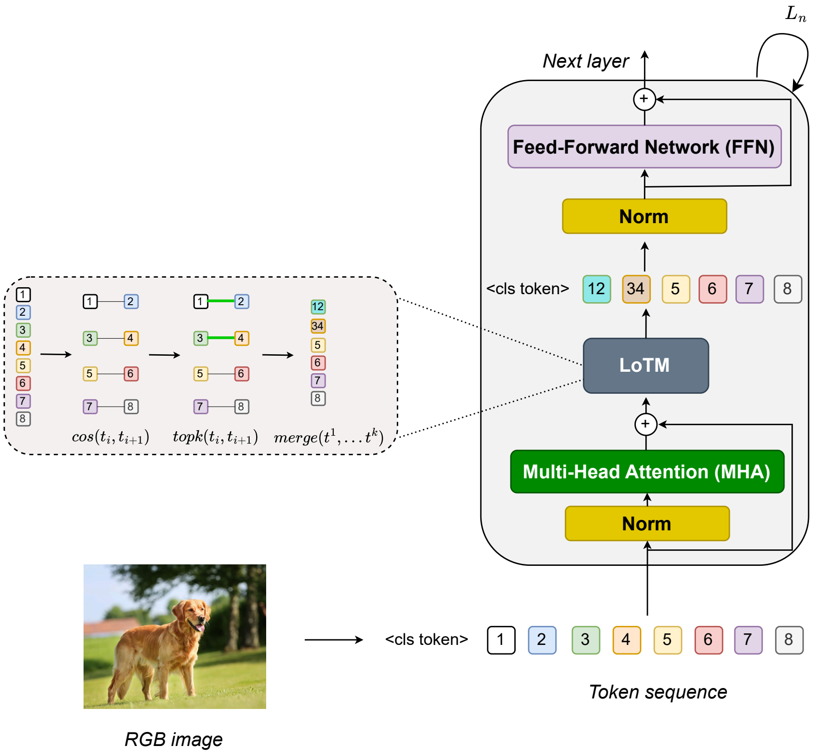


Fig. 2: Overview of LoTM architecture. By focusing on adjacent tokens, LoTM ensures that local structures within the image are preserved when applying token merging. This strategy effectively maintains the spatial integrity of the image information through the model. LoTM is applied on $\{4^{th}, 7^{th}, 11^{th}\}$ layers.

This paper delves into the second approach. In contrast to classical token merging approaches that fuse tokens based on a similarity measure between all token pairs in the input sequence, such as PatchMerger [23], or top-k closest tokens to a centroid like K-Medoids [19], our method restricts the merging strategy to pairs of spatially related tokens, i.e., adjacent ones. This is based on the intuition that tokens in proximity are likely to exhibit higher similarity.

To validate our intuition, we present three illustrations in Figure 1, which show the distribution of cosine similarity between tokens at the first layer of a DeiT-S [25] model, after the Multi-head Self-Attention (MSA) module, on the ImageNet-1K validation set. Figure 1(a) depicts the distribution of cosine similarity between adjacent tokens, which follows a bimodal shape, with most similarity scores greater than 0.5, indicating higher similarity. Figure 1(b) shows the distribution of cosine similarity between non-overlapping windows of four tokens.

This distribution also follows a bimodal shape but is more uniform compared to Figure 1(a), indicating lower similarity scores. Finally, Figure 1(c) presents the distribution of cosine similarity values across all tokens in the sequence, which conforms to a normal distribution. These results demonstrate that tokens exhibit higher similarity when they are spatially closer.

Based on this observation, we propose to restrict the merging of tokens to pairs of adjacent tokens to minimize information loss. To this end, we introduce LoTM, a local token merging strategy that progressively merges adjacent tokens based on their similarity scores. Notably, LoTM does not require any additional training; it is an off-the-shelf approach that can be easily deployed. Our contributions are as follows:

- We present LoTM, a local token merging strategy that leverages local information by restricting the token merging strategy to pairs of adjacent tokens.
- We highlight through statistical analysis the value of considering adjacent token pairs as merging candidates.
- We empirically evaluate LoTM on ImageNet-1K dataset, where it achieves state-of-the-art performance against several methods.

2 Related works

2.1 Vision Transformers

The first work that introduced Transformers to vision tasks is Dosovitskiy et al. [7]. They have achieved state-of-the-art performance after pre-training on large datasets like JFT-300M [24] and ImageNet-21K [5]. However, its effectiveness diminishes on mid-sized datasets such as ImageNet-1K [5], where it slightly underperforms compared to ResNet [11] models of similar size. This is primarily due to transformers lacking inherent image priors like locality and translation equivariance, which are crucial for generalization with limited data [21]. DeiT [25] addressed this issue by modifying the Transformer architecture and employing Knowledge Distillation (KD) [12], achieving improved accuracy on ImageNet-1K. Some work [3, 16] has focused on leveraging local image information to enhance performance, while other approaches [18, 20] have explored deep-narrow architectures resembling CNNs to extract multi-scale features for downstream tasks.

Given the quadratic complexity $O(N^2)$ inherent in Transformers, researchers have sought to optimize computations through various strategies, such as pruning tokens based on their importance score or merging tokens based on their similarity scores. The following sections will provide an overview of these state-of-the-art approaches.

2.2 Token pruning

Token pruning progressively discards tokens, given a pruning strategy. Top-k was first used, where only k tokens with the highest attention scores are kept at each reduction stage [9]. EViT [17] extends Top-k pruning by creating a "fused" token at each stage, computed by averaging the pruned tokens weighted by their CLS token attention scores. Evo-ViT [29] introduces a slow-fast token evolution method to retain more image information during the pruning process. DynamicViT [22] utilizes an MLP predictor to dynamically sample important tokens, trained using continuous relaxation [13] and knowledge distillation [12].

2.3 Token merging

Token merging progressively combines tokens based on a similarity or a distance metric. ToMe [1] constructs a bipartite graph from the tokens, dividing them into two equal-sized sets A and B, then connects each node from set A to the most similar node in set B, and then merges the top-k most similar nodes by averaging their tokens. K-Medoids [19] is an iterative hard-clustering method where cluster centers minimize the Euclidean distance within clusters, updating clusters iteratively based on the closest center. The method initializes the cluster centers based on the CLS token attention scores. DPC-KNN [32] computes each token's density and minimum distance to a higher density point to define cluster centers, averaging assigned elements. SiT [34] uses a small network to predict an assignment matrix for a convex combination of input tokens, forming clusters. Sinkhorn [10] uses randomly initialized learnable vectors as queries, applying the Sinkhorn-Knopp algorithm to similarities between tokens and queries to form an assignment matrix. PatchMerger [23], similar to the approach of Haurum et al. [10], constructs the assignment matrix by calculating the dot product between queries and tokens, followed by a softmax operation to ensure a convex combination.

However, clustering methods such as DP-KNN [32] and Sinkhorn [10] consider a window of top-k tokens when merging around a centroid. Other methods, such as PatchMerger [23], consider the similarity between a query token and all remaining tokens, resulting in an unconstrained merging window that fuses adjacent and distant tokens alike. Following our observation in Figure 1, LoTM differs from the previous works, where we exclusively merge adjacent token pairs according to their cosine similarity scores. This contributes to minimizing information loss when merging, as experimentally shown in Section 5.

3 Transformer architecture

In Vision Transformers (ViTs), a 2D image $I \in \mathbb{R}^{H \times W \times C}$ is divided into N independent patches I_{patch} at a resolution of $p \times p$. Each patch is then projected into a d_e-dimensional embedding, creating a visual sequence $X \in \mathbb{R}^{N \times d_e}$. Similar to BERT [6], ViT introduces a learnable class token $X_{cls} \in \mathbb{R}^{1 \times d_e}$ into the input sequence, combining it with X to form $X_{in} = [X_{cls}, X]$. ViT uses a Transformer encoder with L layers to learn a representation of the image, utilizing Multi-head Self-Attention (MSA) and Multi-Layer Perceptron (MLP) modules alternatively, as described below.

3.1 Multi-head Self-Attention (MSA)

Let $X \in \mathbb{R}^{N \times d_e}$ denote the input sentence[3], where N represents the sequence length and d_e the embedding dimension. Initially, in a self-attention layer, query (Q), key (K), and value (V) matrices are computed from X using linear transformations:

$$[Q, K, V] = XW_{qkv}, \tag{1}$$

where $W_{qkv} \in \mathbb{R}^{d_e \times 3d_h}$ is a parameter that can be learned, and d_h represents the dimensionality of each self-attention head. Subsequently, the attention map A is generated by scaling the inner product of Q and K, followed by normalization using a softmax function:

$$A = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_h}}\right), \tag{2}$$

where $A \in \mathbb{R}^{N \times N}$ and A_{ij} denote the attention score between the i-th query Q_i and the j-th key K_j. The self-attention mechanism then operates on the value vectors V to produce an output matrix:

$$O = AV, \tag{3}$$

where $O \in \mathbb{R}^{N \times d_h}$. In a Multi-head Self-Attention layer with d_e/d_h heads, the final outputs are computed by linearly projecting the concatenated self-attention outputs:

$$Z = [O_1; O_2; \ldots; O_{d_e/d_h}]W_{proj}, \tag{4}$$

where $W_{proj} \in \mathbb{R}^{d_e \times d_e}$ is another learnable parameter, and $[\cdot]$ denotes the concatenation operation.

3.2 Multi-Layer Perceptron (MLP)

Let Z represent the output from the MSA layer. The MLP layer consists of two fully-connected layers with a Gaussian Error Linear Unit (GELU) non-linearity, it can be represented as:

$$Z_{mlp} = \text{GELU}(ZW_{fc1})W_{fc2}, \tag{5}$$

where $W_{fc1} \in \mathbb{R}^{d_e \times 4d_e}$ and $W_{fc2} \in \mathbb{R}^{4d_e \times d_e}$ are learnable parameters.

3.3 Complexity of a Transformer layer

Let $\varPhi(N, d_e)$ be the number of Floating Point Operations (FLOPs) with respect to the sequence length N and the embedding dimension d_e. The computational burden in an MSA layer is primarily due to the projection of Q, K, V matrices,

[3] For simplicity, we omit the learnable class token (CLS).

the calculation of the attention map A, the self-attention operation O, and a linear projection W_{proj} for the concatenated self-attention outputs. The overall FLOPs for an MSA layer amount to:

$$\begin{aligned} \Phi_{\mathrm{MSA}}(N, d_e) &= \Phi_{qkv}(N, d_e) + \Phi_A(N, d_e) + \Phi_O(N, d_e) + \Phi_{\mathrm{proj}}(N, d_e) \\ &= 3Nd_e^2 + N^2 d_e + N^2 d_e + Nd_e^2 \\ &= 4Nd_e^2 + 2N^2 d_e \end{aligned} \tag{6}$$

In an MLP layer, the majority of FLOPs are attributed to the two fully-connected (FC) layers. The first FC layer, f_{c1}, projects each token from $\mathbb{R}^{d_e}$ to $\mathbb{R}^{4d_e}$, while the second FC layer, f_{c2}, maps each token back to $\mathbb{R}^{d_e}$. Consequently, the total FLOPs for an MLP layer can be expressed as:

$$\Phi_{\mathrm{MLP}}(N, d_e) = \Phi_{f_{c1}}(N, d_e) + \Phi_{f_{c2}}(N, d_e) = 4Nd_e^2 + 4Nd_e^2 = 8Nd_e^2 \tag{7}$$

By integrating Eq. (6) and Eq. (7), we can derive the overall FLOPs for a single Transformer layer:

$$\Phi_{\mathrm{L}}(N, d_e) = \Phi_{\mathrm{MSA}}(N, d_e) + \Phi_{\mathrm{MLP}}(N, d_e) = 12Nd_e^2 + 2N^2 d_e \tag{8}$$

Note that the complexity in Eq (8) does not take into account the projection of image patches into token embeddings.

4 Method

4.1 Token merging

$\mathbf{Z} \in \mathbb{R}^{N \times d_e}$ represents the out token sequence of MSA. As defined in Section 2, Token merging progressively combines tokens based on a similarity measure.

Given a similarity metric $s : \mathbb{R}^{d_e} \times \mathbb{R}^{d_e} \rightarrow \mathbb{R}$:

$$s(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|} \tag{9}$$

Where (z_i, z_j) is a pair of tokens. Let $\mathcal{I}$ be a set of indices such as: $\mathcal{I} \subseteq \{1, \ldots, N\}$ representing tokens to be merged, token merging computes the mean of these tokens:

$$\mathbf{z}_{\mathrm{merged}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathbf{z}_i \tag{10}$$

This merged token $\mathbf{z}_{\mathrm{merged}}$ then replaces the original tokens in $\mathcal{I}$, reducing the sequence length. For instance, if we have two sets of indices $\mathcal{I}_1$ and $\mathcal{I}_2$, the process is formalized as:

$$\mathbf{z}_{\text{merged}}^{(1)} = \frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} \mathbf{z}_i, \quad \mathbf{z}_{\text{merged}}^{(2)} = \frac{1}{|\mathcal{I}_2|} \sum_{i \in \mathcal{I}_2} \mathbf{z}_i \tag{11}$$

The resultant token sequence after merging becomes:

$$\mathbf{Z}' = [\mathbf{Z}_j \mid j \notin \mathcal{I}_1 \cup \mathcal{I}_2] \cup \{\mathbf{z}_{\text{merged}}^{(1)}, \mathbf{z}_{\text{merged}}^{(2)}\} \tag{12}$$

where $\mathbf{Z}'$ has a reduced number of tokens after the merging operation, which can either be performed at each layer, or at specific layers.

4.2 Local Token Merging (LoTM)

In Local Token Merging, depicted in Figure 2, we restrict the merging candidates only to adjacent tokens, i.e., spatially related tokens, and merge the top-k most similar pairs. This method reduces the overall token count while preserving local spatial information.

The similarity metric $s : \mathbb{R}^{d_e} \times \mathbb{R}^{d_e} \to \mathbb{R}$ between each pair of adjacent tokens is defined as:

$$s(z_i, z_{i+1}) = \frac{z_i \cdot z_{i+1}}{\|z_i\| \|z_{i+1}\|}, \quad \text{for } i \in \{1, 3, 5, \ldots, N-1\} \tag{13}$$

Let k be a hyperparameter representing the number of pairs to merge. Given the previous similarity scores, we pick the top-k most similar adjacent token pairs:

$$\{(i_1, i_1+1), (i_2, i_2+1), \ldots, (i_k, i_k+1)\} = \text{Top-}k\{s(z_i, z_{i+1})\} \tag{14}$$

For each selected pair $(i, i+1)$, we merge the tokens by averaging their representations:

$$\mathbf{z}_{\text{merged}}^{(i)} = \frac{1}{2}(\mathbf{z}_i + \mathbf{z}_{i+1}) \tag{15}$$

At last , we replace the original tokens in each selected pair with the merged token $\mathbf{z}_{\text{merged}}^{(i)}$. The resultant token sequence after merging becomes:

$$\mathbf{Z}' = \left[\mathbf{Z}_j \mid j \notin \bigcup_{n=1}^{k} \{i_n, i_n+1\}\right] \cup \{\mathbf{z}_{\text{merged}}^{(i_1)}, \mathbf{z}_{\text{merged}}^{(i_2)}, \ldots, \mathbf{z}_{\text{merged}}^{(i_k)}\} \tag{16}$$

Where $\mathbf{Z}'$ has a reduced number of tokens after merging k adjacent pairs. In our proposal, we perform the merging operation on three layers: $\{4^{th}, 7^{th}, 11^{th}\}$.

5 Experiments

5.1 Dataset, models and evaluation metrics

We evaluate the effectiveness of our local token merging method on the ImageNet-1K [5] dataset, which contains 50k samples in the validation set. We use three

Table 1: LoTM performance comparison on DeiT [25] models at different k values for *topk*.

DeiT-T			**DeiT-S**			**DeiT-B**		
topk	Top-1(%)	FLOPs(G)	*topk*	Top-1(%)	FLOPs(G)	*topk*	Top-1(%)	FLOPs(G)
0	72.20	1.26	0	79.82	4.65	0	81.85	17.60
10	72.16	1.16	10	79.82	4.21	10	81.85	16.54
20	71.96	1.07	20	79.82	3.92	20	81.70	15.48
30	71.57	0.97	30	79.62	3.63	30	81.44	14.43
40	70.76	0.88	40	79.19	3.26	40	80.82	13.97
44	70.00	0.85	44	78.87	3.11	44	80.46	12.90
48	68.70	0.81	48	78.81	2.90	48	80.01	12.34

Vision Transformer backbones proposed by Touvron et al. [25]: DeiT-Tiny, DeiT-Small, and DeiT-Base, all of which are pre-trained. Our best results are obtained by applying LoTM to the $4^{th}, 7^{th}, 11^{th}$ layers across all DeiT variants.

For comparison, we use two main metrics: top-1 accuracy (%) and computational cost in terms of FLOPs. The latter, described in Section 3, represents the number of operations performed by the model for a single inference. In our experiments, FLOPs are calculated using the *fvcore*[4] toolkit.

5.2 Implementation details

Our method operates during inference, reducing the number of tokens in a pre-trained model where images are initially split into 14×14 tokens, without requiring any additional training. We conduct experiments on the ImageNet-1K dataset with a batch size of 128. The hyperparameter k represents the number of tokens to merge at each selected layer. We compare LoTM to several state-of-the-art token reduction methods, including SiT [34], Sinkhorn [10], PatchMerger [23], K-Medoids [19], DPC-KNN [32], and ToMe [1].

5.3 Experiment results

Table 1 shows that LoTM results in a small accuracy drop while achieving a significant reduction in FLOPs. For the DeiT-T model, the most resource-constrained variant, we observe that the computational cost can be reduced to as low as 0.88 GFLOPs (k=40) while maintaining an accuracy above 70%, with only a 1.44% decrease in accuracy. This represents considerable computational savings, particularly in ultra resource-constrained environments. In the case of the DeiT-S model, which serves as a mid-tier option, the accuracy remains stable, decreasing slightly from 79.82% to 79.19% (k=40). Meanwhile, the computational cost is reduced from 4.65 GFLOPs to 3.26 GFLOPs, i.e., 30% reduction in complexity. Lastly, the DeiT-B model, intended for scenarios where

[4] https://github.com/facebookresearch/fvcore

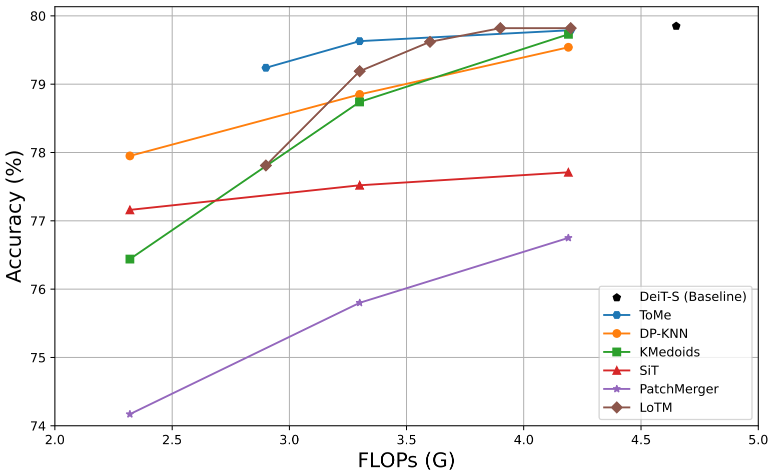


Fig. 3: Comparison of LoTM to other state-of-the-art methods on DeiT-S model. This figure shows the accuracy of various strategies given a computational budget. Our model minimizes information loss when merging adjacent tokens, this translates into higher accuracy compared to most state-of-the-art approaches.

accuracy is critical, shows a baseline accuracy of 81.85%. Our method reduces its complexity by a third, maintaining accuracy above 80% for k=48.

In Figure 3, we compare LoTM to other state-of-the-art token merging approaches on the DeiT-S backbone across various computational budgets. LoTM outperforms most of these approaches, demonstrating its effectiveness by minimizing information loss through the merging of highly similar tokens. In Table 2, we evaluate the same backbone with a reduction ratio of 30% in terms of FLOPs. Our model performs better than Sinkhorn [10], PatchMerger [23], and SiT [34], which show accuracy drops of 14.8%, 3.0%, and 2.3%, respectively. Additionally, our model slightly surpasses DPC-KNN [32] and K-Medoids [19], which have accuracy drops of 0.95% and 1.06%, respectively. The only method outperforming LoTM is ToMe [1], with a mere 0.17% drop in accuracy. However, it is important to note that ToMe [1] requires training from scratch for 300 epochs on the DeiT model [5], demanding significantly more computational resources and GPU power compared to LoTM, which does not require additional training.

In Table 3, we provide a comparative analysis of LoTM against several state-of-the-art token merging approaches implemented on the DeiT-B backbone, using a fixed reduction ratio of 30% in terms of FLOPs. The results highlight LoTM's effectiveness in maintaining high accuracy while significantly reducing

[5] The reported results for ToMe [1] on DeiT [25] models are cited from the paper and were not independently reproduced.

Table 2: Comparison of LoTM with state-of-the-art approaches in token merging on DeiT-S. The reduction ratio is set to 30%.

Models	**Top-1(%)**	**Top-1 ↓(%)**	**FLOPs(G)**
Baseline [25]	79.82	-	4.65
Sinkhorn [10]	64.02	14.8	3.26
PatchMerger [23]	75.80	3.0	3.26
SiT [34]	77.52	2.3	3.26
DPC-KNN [32]	78.85	0.95	3.26
K-Medoids [19]	78.74	1.06	3.26
ToMe [1]	79.63	0.17	3.26
LoTM	79.19	0.63	3.26

computational demands. Specifically, LoTM achieves an accuracy of 80.01%, representing a modest 1.84% drop from the baseline accuracy of 81.85%. This minimal decrease in accuracy is notable given the substantial reduction in FLOPs from 17.60 to 12.34 GFLOPs.

Table 3: Comparison of LoTM with state-of-the-art approaches in token merging on DeiT-B. The reduction ratio is set to 30%.

Models	**Top-1(%)**	**Top-1 ↓(%)**	**FLOPs(G)**
Baseline [25]	81.85	-	17.60
Sinkhorn [10]	63.36	18.49	12.34
PatchMerger [23]	74.52	7.33	12.34
SiT [34]	76.63	5.22	12.34
DPC-KNN [32]	79.06	2.79	12.34
K-Medoids [19]	79.98	1.87	12.34
ToMe [1]	81.05	0.80	12.34
LoTM	80.01	1.84	12.34

When compared to other token merging methods, LoTM notably outperforms Sinkhorn [10], PatchMerger [23], and SiT [34], which exhibit accuracy drops of 18.49%, 7.33%, and 5.22%, respectively. Additionally, LoTM demonstrates competitive performance relative to DPC-KNN and K-Medoids, which have accuracy drops of 2.79% and 1.87%, respectively. Although LoTM's accuracy reduction is slightly higher than K-Medoids [19], it remains on par with DPC-KNN [32].

Despite the 1.84% accuracy drop observed with LoTM on the DeiT-B model, our method is only outperformed by ToMe [1]. However, LoTM has a key advantage: it requires no additional training, making it more flexible and easily deployable off-the-shelf compared ToMe on DeiT [25] model, which requires retraining.

Table 4: Comparison of LoTM with state-of-the-art approaches in token merging on DeiT-T. The reduction ratio is set to 30%.

Models	**Top-1(%)**	**Top-1 ↓(%)**	**FLOPs(G)**
Baseline [25]	72.20	-	1.26
Sinkhorn [10]	53.19	19.01	0.88
PatchMerger [23]	66.81	5.39	0.88
SiT [34]	68.99	3.21	0.88
DPC-KNN [32]	70.10	2.10	0.88
K-Medoids [19]	69.90	2.30	0.88
ToMe [1]	71.74	0.46	0.88
LoTM	70.76	1.44	0.88

Finally, Table 4 presents the results of LoTM on the DeiT-T model. Compared to state-of-the-art methods, LoTM exhibits an accuracy drop of 1.44%, which significantly outperforms Sinkhorn [10], PatchMerger [23], and SiT [34]. It also slightly surpasses K-Medoids [19] and DPC-KNN [32] in terms of accuracy. Although LoTM underperforms compared to ToMe [1] (that requires training) by 0.98%, it is important to note that ToMe requires retraining, which, as stated before, comes at the expense of a reduced flexibility and also additional energy resources, for a gain of less than 1% in accuracy compared to our method.

6 Conclusion

In this paper, we introduced LoTM, a novel token merging strategy designed to leverage local similarity for enhancing computational efficiency in Vision Transformers (ViTs). Unlike traditional methods that compute similarities between all tokens to guide the merging decision, our approach focuses on spatially related, contiguous tokens. This localized strategy allows for more effective token merging, which significantly minimizes information loss, hence maintaining high accuracy while reducing computational complexity. Besides, our empirical evaluation of LoTM on the ImageNet-1K dataset demonstrates that LoTM achieves state-of-the-art performance across DeiT [25] variants that include DeiT-T, DeiT-S, and DeiT-B. The results highlight the efficiency of our method, which achieves higher or comparable performance than most other state-of-the-art techniques without requiring further training.

Future work will extend local merging beyond adjacent tokens, considering a wider pool of candidates for improved flexibility. We also aim to conduct an experimental analysis to understand the effectiveness of local token merging. Lastly, we plan to apply LoTM to semantic segmentation, leveraging its ability to preserve spatial information with reduced computational cost.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=JroZRaRw7Eu
2. Bonnaerens, M., Dambre, J.: Learned thresholds token merging and pruning for vision transformers. Transactions on Machine Learning Research (2023), https://openreview.net/forum?id=WYKTCKpImz
3. Chen, C.F.R., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 357–366 (2021)
4. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. Advances in neural information processing systems **34**, 9355–9366 (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019), https://api.semanticscholar.org/CorpusID:52967399
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. Advances in neural information processing systems **34**, 15908–15919 (2021)
9. Haurum, J.B., Escalera, S., Taylor, G.W., Moeslund, T.B.: Which tokens to use? investigating token reduction in vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 773–783 (2023)
10. Haurum, J.B., Madadi, M., Escalera, S., Moeslund, T.B.: Multi-scale hybrid vision transformer and sinkhorn tokenizer for sewer defect classification. Automation in Construction **144**, 104614 (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
13. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: International Conference on Learning Representations (2017)
14. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1383–1392 (2024)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
16. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)
17. Liang, Y., GE, C., Tong, Z., Song, Y., Wang, J., Xie, P.: EVit: Expediting vision transformers via token reorganizations. In: International Conference on Learning Representations (2022)

18. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
19. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers for image classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 12–21 (2023)
20. Pan, B., Panda, R., Jiang, Y., Wang, Z., Feris, R., Oliva, A.: Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. Advances in Neural Information Processing Systems **34**, 24898–24911 (2021)
21. Park, N., Kim, S.: How do vision transformers work? In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=D78Go4hVcxO
22. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021)
23. Renggli, C., Pinto, A.S., Houlsby, N., Mustafa, B., Puigcerver, J., Riquelme, C.: Learning to merge tokens in vision transformers. arXiv preprint arXiv:2202.12015 (2022)
24. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)
25. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
27. Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., Huang, G.: Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. Advances in Neural Information Processing Systems **33**, 2432–2444 (2020)
28. Wang, Y., Yue, Y., Lin, Y., Jiang, H., Lai, Z., Kulikov, V., Orlov, N., Shi, H., Huang, G.: Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20030–20040. IEEE (2022)
29. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2964–2972 (2022)
30. Yang, L., Han, Y., Chen, X., Song, S., Dai, J., Huang, G.: Resolution adaptive networks for efficient inference. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2369–2378 (2020)
31. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 558–567 (2021)
32. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11101–11111 (2022)

33. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations (2022)
34. Zong, Z., Li, K., Song, G., Wang, Y., Qiao, Y., Leng, B., Liu, Y.: Self-slimmed vision transformer. In: European Conference on Computer Vision. pp. 432–448. Springer (2022)