Agglomerative Token Clustering

Joakim Bruslund Haurum[1], Sergio Escalera[2,1], Graham W. Taylor[3], and Thomas B. Moeslund[1]

[1] Visual Analysis and Perception (VAP) Laboratory, Aalborg University & Pioneer Centre for AI, Denmark
[2] Universitat de Barcelona & Computer Vision Center, Spain
[3] University of Guelph & Vector Institute for AI, Canada
{joha, tbm}@create.aau.dk, sescalera@ub.edu, gwtaylor@uoguelph.ca
https://vap.aau.dk/atc/

Abstract. We present Agglomerative Token Clustering (ATC), a novel token merging method that consistently outperforms previous token merging and pruning methods across image classification, image synthesis, and object detection & segmentation tasks. ATC merges clusters through bottom-up hierarchical clustering, without the introduction of extra learnable parameters. We find that ATC achieves state-of-the-art performance across all tasks, and can even perform on par with prior state-of-the-art when applied *off-the-shelf*, *i.e.* without fine-tuning. ATC is particularly effective when applied with low keep rates, where only a small fraction of tokens are kept and retaining task performance is especially difficult.

1 Introduction

Since their introduction in 2020, Vision Transformers (ViTs) [14] have been utilized for a wide variety of computer vision tasks such as image classification, synthesis, segmentation and more with great success [10, 32, 53]. Unlike Convolutional Neural Networks (CNNs), which require a structured representation throughout the network, ViTs can process variable length input sequences, even allowing for the sequences to be modified throughout the network. This gives ViTs stronger expressive powers than CNNs [49], but comes at a computational cost due to the quadratic scaling of the self-attention computation. Therefore, there has been increasing research interest in making ViTs more efficient while retaining their task performance. *Token reduction* has shown to be a promising subfield which directly decreases model complexity by reducing the input sequence through pruning or merging [20], thereby decreasing the computation cost of the self-attention operation. Haurum *et al.* [20] conducted an in-depth study of 13 different methods, and found that the baseline Top-K pruning-based method, and its extension EViT [34], outperformed the vast majority of more complex methods on four image classification tasks. However, token pruning has the disadvantage that it typically requires the backbone to be fine-tuned in order to maintain good performance even at high keep rates, which by design leads to information loss as tokens are removed, and performs poorly on image synthesis

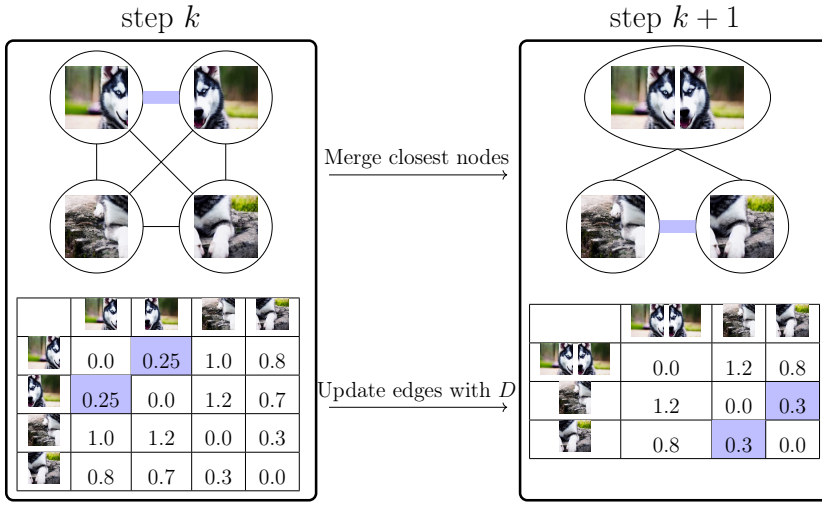

Fig. 1: Illustration of the Agglomerative Clustering Method. Prior hard merging-based methodologies have focused on using either partition-based approaches (*e.g.* DPC-KNN [73] or K-Medoids [41]) or graph-based (*i.e.* ToMe). All of these methods globally cluster the input tokens through the use of cluster centers. In contrast, our **Agglomerative Token Clustering (ATC)** method builds clusters locally, *i.e.* by iteratively combining the most similar tokens, until the desired amount of tokens remain. A step of this process is shown here, where a graph of nodes (in this case, tokens) are connected with edges based on their similarity. The most similar pair of nodes are combined, and the edges are updated using linkage function D, in this case D^{complete} (Eq. 2).

tasks [2, 3]. Motivated by these observations, we focus on merging-based token reduction methods as they show the most versatility.

We present a novel merging-based token reduction method, Agglomerative Token Clustering (ATC), which outperforms all prior merging-based and pruning-based token reduction methods on both classification tasks and dense computer vision tasks such as image synthesis and object detection & segmentation, and achieve comparable performance without any fine-tuning, *i.e. off-the-shelf*. This is the most diverse set of experiments considered within token reduction, where previous methods have been evaluated just on classification [2, 20, 34], image synthesis [3], or detection and segmentation [38]. Through this suite of diverse tasks we demonstrate the efficacy of the proposed ATC method.

ATC is motivated by the observation that prior merging-based methods such as K-Medoids [41], DPC-KNN [73], and ToMe [2] all perform merging globally, which may lead to redundant clusters. Our hypothesis is that hierarchically merging similar, and thus redundant, observations results in more informative groupings. A natural and robust methodology for including this notion into token reduction is via agglomerative clustering [16], where tokens are iteratively clustered in a bottom-up hierarchical way, see Figure 1.

Our contributions are the following:

- We propose Agglomerative Token Clustering (ATC), a novel parameter-free hierarchical merging-based token reduction method.
- Using ATC we achieve state-of-the-art performance on image classification, image synthesis, and object detection & segmentation tasks, outperforming all other token reduction methods, including both merging-based and pruning-based token reduction methods.
- We show ATC can reach comparable performance to the prior fine-tuned state-of-the-art in image classification and object detection & segmentation when applied off-the-shelf, *i.e.* without any fine-tuning.

2 Related Work

Efficient Transformers. As ViTs have become widely adapted by the computer vision community, there have been several attempts at making ViTs more efficient. These attempts range from model pruning [5, 8, 9, 42, 56, 71], quantization [33, 37], structured downsampling [19, 25, 39, 46], sparsification of the attention module [4, 65], part selection modules [22, 27, 28, 61, 62], and dynamically adjusting the size of input patches [1,7,12,35,63,64,72,74]. Lastly, the field of token reduction has emerged, which is the topic of this paper.

Token Reduction. The goal of token reduction is to sparsify the sequence of patches, also referred to as tokens, processed by the ViT. This has been achieved through either token pruning or token merging [20]. Token pruning focuses on removing tokens either through keeping the tokens with the highest attention from the class (CLS) token [17, 20, 34, 40, 66, 67, 69], introducing a gating mechanism based on the Gumbel-Softmax trick [29,30,38,50,66], sampling based approaches [17, 70] modifying the training loop [8, 31] or reinforcement learning [45].

Token merging, on the other hand, combines tokens instead of explicitly pruning them. This can be done either through hard or soft merging of tokens. Hard merging includes techniques such as the partition-based K-Medoids [20,41] and DPC-KNN [73], as well as the bipartite graph-based approach Token Merging (ToMe) [2, 3]. Hard merging-based approaches are characterized by having the tokens assigned to clusters in a mutually exclusive manner. In contrast, soft merging techniques let tokens be assigned to multiple clusters resulting in cluster centers being a convex combination of tokens. This combination is either based on similarity between the spatial tokens [41], or the similarity between the spatial tokens and a set of explicit queries which are optimized [21, 52, 68, 75].

The token reduction field has been moving at an immense speed, resulting in little to no comparisons between methods. This was rectified by Haurum *et al.* [20], who compared 13 different token reduction methods over four image classification datasets. The study provided insights into the token reduction process through extensive experiments, showing that the Top-K and EViT pruning-based methods consistently outperformed all other token reduction methods on the considered classification datasets. However, Bolya and Hoffmann found that

merging-based methods outperform pruning-based methods for image synthesis [3], while Bolya *et al.* showed that the merging-based ToMe [2] can perform well on several classification tasks without any fine-tuning.

Despite its impressive performance, a core component of ToMe is the bipartite matching algorithm, where tokens are split into two exclusive sets between which a bipartite graph is created. This inherently limits which tokens can be merged as there is no within-set comparison and limits the keep rate, r, such that it must be 50% or higher. Therefore, we make a single, but important, modification to the ToMe method, replacing the bipartite matching algorithm with the classical agglomerative clustering method [16].

3 Agglomerative Token Clustering

Similar to previous token merging methods, the objective of ATC is to merge redundant tokens, while preserving or enhancing the performance of the ViT model. We insert the token merging operation between the self-attention and Multi Layer Perceptron (MLP) modules in a ViT block. This is consistent with prior merging-based methods, such as ToMe [2].

We believe the agglomerative clustering algorithm is a more appropriate choice as it builds the clusters in a bottom-up manner, such that redundant features are clustered early on, while more diverse features are kept unmodified for as long as possible. Prior partition-based (*e.g.* DPC-KNN and K-Medoids) and graph-based (*i.e.* ToMe) merging methods are limited by having to create clusters globally, necessitating the selection of cluster centers which may be redundant. In contrast, ATC creates clusters in a sequential manner, resulting in a local merging approach which leads to the most redundant feature to be clustered at any step in the process. We revisit the ToMe method in Section 3.1 and agglomerative clustering in Section 3.2.

3.1 Token Merging

ToMe was designed for seamless integration into ViTs, allowing for minimal performance loss without necessitating fine-tuning. Its key feature is a fast bipartite merging operation, placed between the self-attention and MLP modules in the ViT block. A bipartite graph is then constructed by setting the edge between nodes in token subsets A and B equal to their similarity, where the t highest valued edges are kept while allowing only a single edge for each node in subset A. Bolya *et al.* investigated how to construct A and B, and found the best performance was achieved by assigning tokens in an alternating manner. This inherently limits which tokens can be merged, which can lead to redundant clusters as spatially co-located patches contain semantically similar information.

3.2 Agglomerative Clustering

Agglomerative Clustering is a classical method for bottom-up hierarchical clustering, where each element is initially its own cluster [16]. The elements are com-

bined by iteratively comparing the clusters according to some *linkage* function with distance metric $D(\cdot)$, combining the two closest clusters in each iteration. This is repeated until a certain stopping criteria is met, such as the number of desired clusters, leading to a static reduction method, or a minimum distance between clusters, leading to a dynamic reduction method. This is illustrated in Figure 1. In this paper we consider the static reduction scenario. Similar to Bolya *et al.*, we use the cosine distance as our distance metric $D(\cdot)$ and the keys from the self-attention module as token features. The choice of linkage function can have a large impact on how the elements are clustered. We consider the three most common ones: single (Eq. 1), complete (Eq. 2), and average (Eq. 3) [43].

$$D(I, J)^{\text{single}} = \min_{i \in I,\ j \in J} D(i, j) \tag{1}$$

$$D(I, J)^{\text{complete}} = \max_{i \in I,\ j \in J} D(i, j) \tag{2}$$

$$D(I, J)^{\text{average}} = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} D(i, j) \tag{3}$$

where I and J are clusters with elements $i \in I$ and $j \in J$.

After the stopping criteria has been reached we average the tokens in each cluster to get an updated cluster representation. However, as tokens are merged they represent more than one input patch. In order to advantage tokens that capture a larger spatial extent, we use the weighted average for the cluster representation and proportional attention in the self-attention module as proposed by Bolya *et al.*

4 Experiments

To evaluate the versatility and applicability of ATC, we conduct assessments across a diverse set of tasks (image classification, image synthesis, and object detection & segmentation) and datasets.

For the image classification task we follow the experimental protocol of Haurum *et al.* [20], evaluating multi-class and multi-label classification performance across four classification datasets using three DeiT models and token reduction performed at three discrete stages. We also follow the MAE experiments of Bolya *et al.* [2], evaluating on the ImageNet-1K dataset with token reduction at all stages following a constant and linearly decreasing schedule. For the image synthesis task, we follow the proposed protocol of Bolya & Hoffman [3], incorporating the token reduction method into the Stable Diffusion image generation model [53]. Lastly, for the object detection and segmentation task we follow the experimental protocol of Liu *et al.* [38], evaluating on the COCO-2017 dataset [36]. Two versions of the ViT-Adapter model are used with the token reduction method incorporated at three discrete stages.

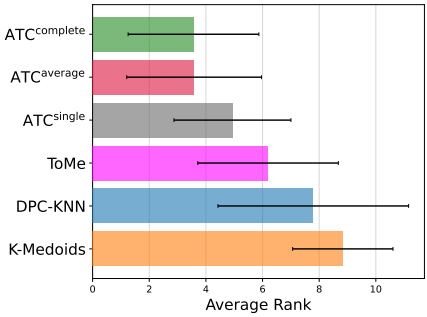

Fig. 2: Average Token Reduction Rank (Lower is better). We compare our proposed ATC method with the hard-merging based token reduction methods investigated by Haurum *et al.* [20]. We average across four keep rates, three model capacities, and four datasets, and plot with ±1 standard deviation similar to Haurum *et al.* We test three versions of ATC, varying the linkage function, and find that the three variants all outperform the prior merging-based methods.

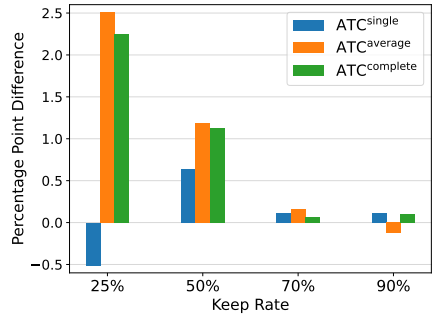

Fig. 3: Percentage Point Difference per Keep Rate. We compute the average difference between our proposed ATC method and the best prior merging-based methods investigated by Haurum *et al.* [20] for each keep rate, measured in percentage points. We average across the three model capacities and four datasets. We find that for high keep rates $r = \{70, 90\}\%$ ATC is comparable to the prior best merging method, while for $r = \{25, 50\}\%$ our proposed ATC method leads to significant performance gains.

All experiments are conducted using ATC with the single, complete, and average linkage functions, respectively. The different linkage functions are indicated using a superscript, such as $\text{ATC}^{\text{single}}$ for the single linkage function. For the image classification and object detection & segmentation tasks we report both *off-the-shelf* results, where ATC is inserted into the pre-trained model and evaluated without any fine-tuning, and results after fine-tuning. For the image synthesis task we report off-the-shelf results, similarly to Bolya & Hoffman [3].

4.1 Cross-Dataset Classification Performance

Following the experimental protocol proposed by Haurum *et al.* [20], we evaluate the performance of ATC across four classification datasets covering multi-class and multi-label classification: ImageNet-1K [13], NABirds, [59] COCO 2014 [36], and NUS-WIDE [11] datasets. ImageNet-1K and NABirds are evaluated with the accuracy metric, while COCO and NUS-WIDE are evaluated with the mean Average Precision (mAP) metric. Token reduction is applied at the 4th, 7th, and 10th ViT block, where at each stage only $r \in \{25, 50, 70, 90\}\%$ of the available tokens are kept. The backbone model is a DeiT [58] model trained without distillation, across three model capacities: DeiT-Tiny, DeiT-Small, and DeiT-Base. The models are denoted DeiT-T, DeiT-S, and DeiT-B, respectively. We consider

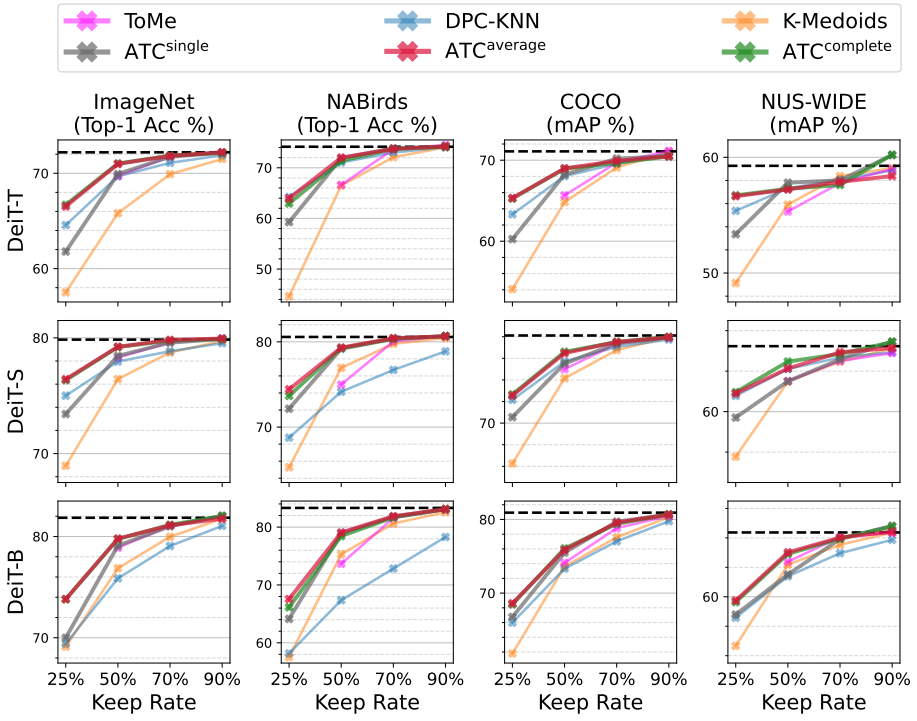

Fig. 4: Hard Token Merging Method Comparison with the DeiT Backbone. We compare the hard-merging token reduction methods considered by Haurum *et al.* [20] with the proposed ATC method. All methods have been fine-tuned. Model performance is measured across keep rates, r, denoted in percentage of tokens kept at each reduction stage, and with the DeiT-{Tiny, Small, Base} models. Comparison with all 13 token reduction methods considered by Haurum *et al.* can be found in the supplementary materials. ImageNet and NABirds performance is measured with top-1 accuracy, whereas COCO and NUS-WIDE is measured with mAP. The baseline DeiT performance is noted with a dashed black line. Note that ToMe is limited to $r \geq 50\%$, and that ATC$^{\text{average}}$ and ATC$^{\text{complete}}$ often overlap.

both the off-the-shelf scenario, where ATC is inserted into the pre-trained DeiT models with no further training, as well as the fine-tuning scenario, where the model is fine-tuned for 30 epochs [20]. We conduct a hyperparameter sweep following the setup used by Haurum *et al.* [20]. The final hyperparameters can be found in the supplementary materials.

We find that across all three backbone capacities, the fine-tuned ATC methods consistently outperform or match the performance of the previously proposed merging-based methods using all three proposed linkage functions, see Figure 2 and Figure 4 for details. We also investigate the average improvement in performance between ATC and the prior best merging-based methods, see Figure 3.

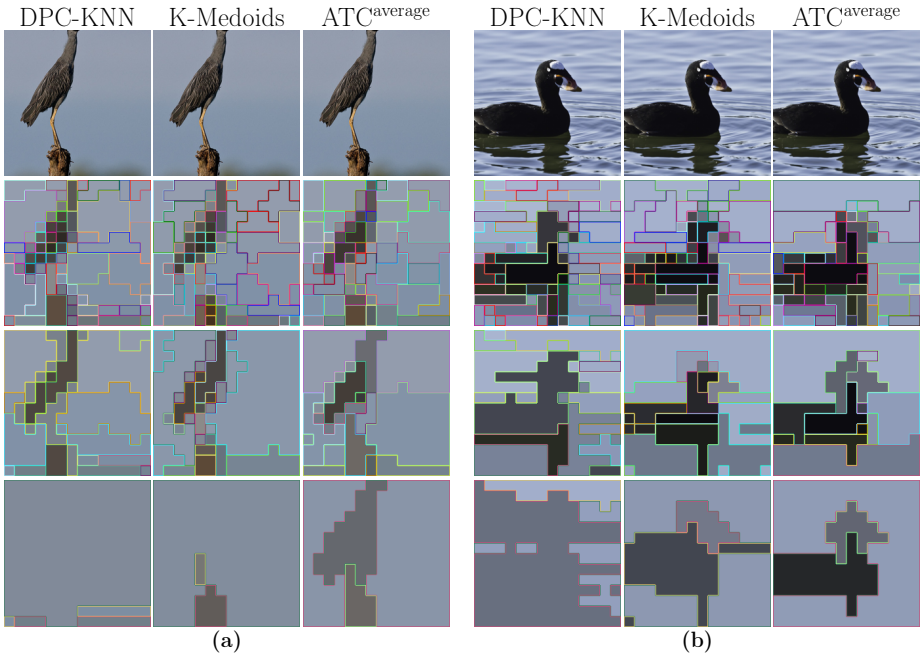

Fig. 5: Token Merging Visualization with $r = 25\%$. We visualize the three token merging steps for DPC-KNN [73], K-Medoids [41] and our ATC$^{\text{average}}$ on two examples from NABirds [59] with a DeiT-B backbone. The first row is the input image, and each subsequent row is the constructed clusters after the first, second, and third reduction stage. In subfigure **(a)** we find that there is a major difference in the final clustering of the data, where our ATC method creates separate clusters for the bird, wood pole, and background. In contrast, DPC-KNN and K-Medoids create mostly arbitrary clusters. Similarly in subfigure **(b)** we see that the DPC-KNN method creates very arbitrary clusters, while K-Medoids and ATC create more meaningful clusters. However, the ATC clusters still better contain the bird in the image, while the K-Medoids clusters have background patches in all clusters. We find this to be a repeating occurrence and believe this is the reason for the large improvement by ATC on NABirds at $r = 25\%$.

We find that at keep rates of 70% and 90%, all three linkage functions achieve comparable results, while at lower keep rates the single linkage function drops in performance. We believe this is due to the *chaining phenomenon* where two distinct clusters are merged due to outliers within the clusters [16].

However, at keep rates of 25% and 50% we find that both the average and complete lead to large performance improvements compared to the prior best merging methods (up to 2.5 percentage points as per Figure 3). In some cases we even find that at keep rates of 25% we can improve performance significantly, such as with the DeiT-S and DeiT-B backbones on NABirds, where ATC with the average linkage function results in a 5.7 and 9.6 percentage point increase in accuracy over the prior state-of-the-art, respectively. Through qualitative eval-

uation, as seen in Figure 5, we can provide intuition for why ATC outperforms prior merging-based methods. We find that DPC-KNN and K-Medoids create arbitrary clusters whereas ATC creates more meaningful clusters even at the third reduction stage with $r = 25\%$.

Lastly, we find that applying ATC off-the-shelf on the DeiT backbone leads to good performance at high keep rates, matching or even outperforming the prior best performing merging methods. However, when the keep rate is lowered the performance drops dramatically, which can be rectified through fine-tuning. For full details on the off-the-shelf results, we refer to the supplementary materials.

4.2 Classification with Self-Supervised Pretraining

We follow the protocol of Bolya *et al.* [2] and compare the performance on ImageNet-1K when applying token reduction on pretrained ViT models, specifically the MAE models trained initially with masked image modelling [23] and fine-tuned on ImageNet [13]. We consider the off-the-shelf performance using the publicly available checkpoints, as well as fine-tuning using the original fine-tuning protocol by He *et al.* [23]. We consider the ViT-Base, ViT-Large, and ViT-Huge models, where token reduction is applied at every block, following the constant and linear decreasing schedules proposed by Bolya *et al.*:

$$t^l = t \tag{4}$$

$$t^l = \left\lfloor 2t - \frac{2tl}{L-1} \right\rfloor, \tag{5}$$

where L is the total number of ViT blocks, t^l is the number of tokens to be removed at ViT block $l = \{0, 1, \ldots, L-1\}$, and t is a hyperparameter controlling the aggressiveness of the reduction. Both schedules result in tL tokens being removed in total, with the linear schedule removing more tokens at early layers.

We fine-tune the ViT block models using both schedules in the most aggressive settings considered by Bolya *et al.*: ViT-B with $t = 16$, ViT-L with $t = 8$, and ViT-H with $t = 7$. Note that we also fine-tune ToMe, leading to a second set of values that are a small improvement compared to the original paper. We also replicate the larger sweep over t values on off-the-shelf ViT models with weights from different training protocols [23, 55, 57], originally performed by Bolya *et al.* These are found in the supplementary materials. We find that ATC consistently outperforms ToMe across both token reduction schedules and when using off-the-shelf and fine-tuned models, see Table 1. We find that ATC drastically outperforms ToMe when using the linear token scheduler, and when using models with lower backbone capacity. This is especially observable on the ViT-Base backbone where using a linear schedule off-the-shelf leads to a 10.5 percentage point improvement when using ATC instead of ToMe. When fine-tuning the backbone using the same token scheduler this gap is reduced to 1.6 percentage points when fine-tuning, with ATC still outperforming ToMe. This illustrates the clear general benefit of using ATC for adapting already trained

Table 1: MAE Pretrained Backbone Comparison. We compare ToMe and ATC with a self-supervised MAE backbone on ImageNet-1K. We consider three backbones: ViT-Base, ViT-Large, and ViT-Huge. Tokens are removed at each ViT block, following the constant (Eq. 4) or linear (Eq. 5) token schedules. The best performing method per column is denoted in **bold**. A blue background indicates that the token reduction method was applied off-the-shelf on the ViT backbone, whereas all other models have have been fine-tuned. For each ATC method we write the performance improvement over ToMe in parenthesis after the model accuracy.

	ViT-B ($t = 16$)		ViT-L ($t = 8$)		ViT-H ($t = 7$)	
Schedule	Constant	Linear	Constant	Linear	Constant	Linear
No Reduction	83.6		85.9		86.9	
ToMe	78.5	56.6	84.2	80.1	86.0	85.0
$\text{ATC}^{\text{single}}$ (ours)	79.7 (+1.2)	65.8 (+9.2)	84.8 (+0.6)	82.5 (+2.4)	86.4 (+0.4)	85.6 (+0.6)
$\text{ATC}^{\text{average}}$ (ours)	80.1 (+1.6)	67.1 (+10.5)	84.8 (+0.6)	82.6 (+2.5)	86.4 (+0.4)	85.6 (+0.6)
$\text{ATC}^{\text{complete}}$ (ours)	80.2 (+1.7)	67.1 (+10.5)	84.9 (+0.7)	82.6 (+2.5)	86.4 (+0.4)	85.8 (+0.8)
ToMe	81.9	78.6	85.1	83.8	86.4	86.1
$\text{ATC}^{\text{single}}$ (ours)	82.2 (+0.3)	80.0 (+1.4)	85.3 (+0.2)	**84.5** (+0.7)	86.6 (+0.2)	86.3 (+0.2)
$\text{ATC}^{\text{average}}$ (ours)	82.3 (+0.4)	**80.2** (+1.6)	85.3 (+0.2)	**84.5** (+0.7)	**86.7** (+0.3)	86.3 (+0.2)
$\text{ATC}^{\text{complete}}$ (ours)	**82.5** (+0.6)	80.1 (+1.5)	**85.4** (+0.3)	84.3 (+0.5)	**86.7** (+0.3)	**86.4** (+0.3)

Table 2: Stable Diffusion FID Comparison. We apply ToMe and ATC over the self-attention blocks to an off-the-shelf (*i.e.* frozen) Stable Diffusion model, denoted with a blue background. We compare the FID score across different keep rates (Note that a lower FID score is better). The best FID score per column is denoted in **bold**, and the best per row is underlined.

r (%)	100	90	80	70	60	50	40
ToMe	**33.80**	33.77	33.61	33.59	33.57	33.63	33.67
$\text{ATC}^{\text{single}}$ (ours)	**33.80**	33.63	33.48	33.53	33.43	34.01	36.95
$\text{ATC}^{\text{average}}$ (ours)	**33.80**	**33.51**	**33.45**	**33.33**	**33.40**	**33.36**	**33.22**
$\text{ATC}^{\text{complete}}$ (ours)	**33.80**	33.67	33.71	33.70	33.74	33.56	33.65

ViT backbones, even when using the more aggressive linear token scheduler and without applying any fine-tuning.

4.3 Image Synthesis with Stable Diffusion

Bolya & Hoffman [3] demonstrated how a modified ToMe algorithm can be incorporated into an image generation model, specifically Stable Diffusion [53], without any fine-tuning, resulting in improved quality of the generated images as well as generation speed. We follow the experimental setup of Bolya & Hoffman, inserting the token reduction method only over the self-attention block. We generate two images with a resolution of 512×512 pixels for each class in the ImageNet-1K dataset, following the exact setting from Bolya & Hoffman [3].

Table 3: Object Detection & Segmentation Results. We compare the performance of ATC and ToMe on the dense object detection & segmentation task using the COCO 2017 dataset, following Liu *et al*. [38]. The ViT-Adapter method is used, with DeiT-T and DeiT-S backbones. The best performing method per column is denoted in **bold**. A blue background indicates that ATC and ToMe were applied off-the-shelf on the frozen ViT-Adapter backbone, whereas all other models have been fine-tuned.

(a) ViT-Adapter-T Backbone

	mAP^{box}				mAP^{mask}			
ViT-Adapter-T	45.8				40.9			
r (%)	25	50	70	90	25	50	70	90
ToMe	-	39.9	45.3	45.8	-	36.2	40.5	40.8
$\text{ATC}^{\text{single}}$ (ours)	13.9	38.5	44.8	45.7	12.6	34.6	39.9	40.8
$\text{ATC}^{\text{average}}$ (ours)	33.8	43.7	45.5	45.8	31.5	39.2	40.7	40.8
$\text{ATC}^{\text{complete}}$ (ours)	34.9	43.9	45.6	45.8	32.3	39.3	40.6	40.8
ToMe	-	43.7	45.7	**46.0**	-	39.3	40.9	41.0
$\text{ATC}^{\text{single}}$ (ours)	36.9	43.5	45.5	45.9	33.6	39.1	40.7	40.9
$\text{ATC}^{\text{average}}$ (ours)	42.4	45.2	45.8	45.9	38.5	**40.5**	40.8	**41.1**
$\text{ATC}^{\text{complete}}$ (ours)	**42.6**	**45.3**	**45.9**	**46.0**	**38.7**	**40.5**	**41.0**	**41.1**

(b) ViT-Adapter-S Backbone

	mAP^{box}				mAP^{mask}			
ViT-Adapter-S	48.5				42.8			
r (%)	25	50	70	90	25	50	70	90
ToMe	-	42.4	47.8	48.4	-	38.2	42.3	42.7
$\text{ATC}^{\text{single}}$ (ours)	16.9	41.9	47.4	48.4	14.4	36.8	41.8	42.7
$\text{ATC}^{\text{average}}$ (ours)	37.2	46.5	48.0	48.4	33.7	41.2	42.4	42.7
$\text{ATC}^{\text{complete}}$ (ours)	38.3	46.6	48.0	48.4	34.7	41.3	42.4	42.7
ToMe	-	46.4	48.2	48.3	-	41.4	**42.6**	42.7
$\text{ATC}^{\text{single}}$ (ours)	40.3	46.4	47.9	48.3	36.2	41.1	42.3	42.6
$\text{ATC}^{\text{average}}$ (ours)	45.1	**47.8**	**48.3**	48.5	40.2	**42.4**	**42.6**	**42.9**
$\text{ATC}^{\text{complete}}$ (ours)	**45.3**	**47.8**	48.2	**48.6**	**40.5**	42.3	**42.6**	42.7

We implement the setup using the HuggingFace Diffusers framework [48], use the STABLE-DIFFUSION-V1-5 model, and use the exact same seed for each model. We investigate the effect when using keep rates of $r \in \{40, 50, 60, 70, 80, 90\}\%$. In order to evaluate the quality of the generated images, we measure the Fréchet Inception Distance (FID) score [26] between the generated images and a reference dataset consisting of 5000 images, created by taking the first five validation images per ImageNet-1K class.

We compare the FID scores of the ToMe and ATC models in Table 2. The ToMe results are computed using the same setup as the ATC models, and therefore differ from the original paper. Even though our generated images lead to a generally higher (thus worse) FID for ToMe, we find that the general trends observed in the original paper still hold. We find that across all linkage functions the ATC model outperforms or matches the ToMe model. While ToMe achieves a minimum FID of 33.57, this is outperformed by both the single and average linkage functions with FID scores of 33.43 and 33.22, respectively. The complete linkage function in general performs worse than both the single and average linkage functions, but we also find that the results with single linkage diverges when $r \leq 50\%$. We observe that the average linkage function leads to better results as the keep rate is reduced, achieving the best performance when $r = 40\%$, whereas the other methods peak at $r = 50\%$ and $r = 60\%$. By looking at the generated images, see Figure 6, we find that the average linkage function manages to keep a lot more of the distinctive patterns and contextual background. In comparison, the single linkage function loses the head pattern, and all methods except $\text{ATC}^{\text{average}}$ convert the tree branch to a tree pole resulting in a change in pose of the generated magpie.

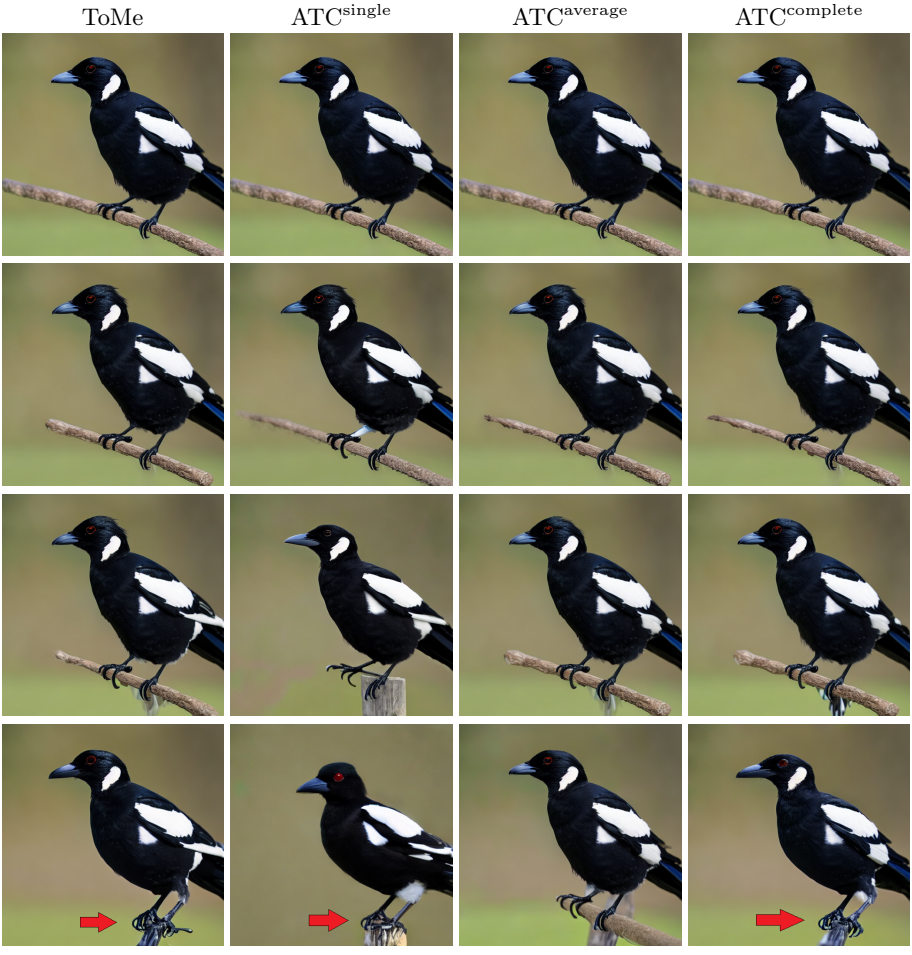

Fig. 6: Image Synthesis Visualization. We visualize image synthesis results for the "magpie" ImageNet class. The first row is the standard Stable Diffusion result, with each subsequent row having token merging applied with $r \in \{80, 60, 40\}\%$. Examples for all considered keep rates and more classes can be found in the supplementary materials. We find that as the keep rate is lowered the ATC$^{\text{single}}$ method drastically changes the image, specifically the head and patterns of the magpie, as well as having a more monochrome background. We see that even at $r = 40\%$ ATC$^{\text{average}}$ manages to keep most of the details such as the branch the magpie is sitting on, whereas ToMe and ATC$^{\text{complete}}$ keep the general patterns but switches from a branch to a wooden pole (highlighted with a red arrow), leading to a change in pose of the generated magpie.

4.4 Object Detection and Segmentation

Liu *et al*. [38] conducted a systematic comparison of several token pruning methods for object detection and segmentation on the COCO 2017 dataset [36]. The Mask-RCNN model [24] is used for predicting bounding boxes and segmentation masks with a ViT-Adapter backbone model [10], building upon an ImageNet pretrained DeiT model [58]. Instead of the typical windowed self-attention used in ViT-Adapter, Liu *et al*. use global self-attention and train using the original ViT-Adapter settings for 36 epochs. Tokens are reduced at the 4th, 7th, and 10th ViT block, and a DeiT-Tiny and DeiT-Small backbone are used. When applying the token reduction method, the ViT-Adapter-Tiny and ViT-Adapter-Small models are fine-tuned for 6 and 4 epochs, respectively, using the MMDetection framework [6] and following the training protocol of Liu *et al*. Unlike the original protocol by Liu *et al*. which only considered a keep rate of $r = 70\%$, we extend the considered keep rates to $r \in \{25, 50, 70, 90\}\%$, similar to the protocol used by Haurum *et al*. [20]. We evaluate both ToMe and ATC in the off-the-shelf and fine-tuned scenarios. As the Injector and Extractor modules in the ViT-Adapter expect the original number of tokens, we back-project through the clustering when relevant, leading to the original number of patches.

As is apparent in Table 3, we find that both ToMe and ATC are very strong token reduction methods for detection and segmentation. When applying ToMe and ATC off-the-shelf (*i.e*. without fine-tuning the ViT-Adapter backbone) with keep rates $r \in \{70, 90\}\%$, both methods can match the detection and segmentation performance of the baseline ViT-Adapter backbones. When the keep rate is lowered to $r = 50\%$ we find that our ATC method with the average and complete linkage function outperforms ToMe by 4 percentage points in detection mAP and 3 percentage points in segmentation mAP.

When fine-tuning the Tiny and Small backbones we see major improvements at keep rates of 25% and 50%, such as a 26 and 22 percentage points improvement in bounding box mAP for $\text{ATC}^{\text{single}}$ with the Tiny and Small backbones and $r = 25\%$, respectively. In comparison, fine-tuning has less of an effect on $\text{ATC}^{\text{average}}$ and $\text{ATC}^{\text{complete}}$ as the off-the-shelf performance is already high. We also observe that after fine-tuning, our ATC method still outperforms ToMe, though with a smaller margin, and in several cases outperforms the baseline ViT-Adapter performance. Lastly, we see that by fine-tuning, we can get comparable performance to the baseline method at keep rates of 50%, whereas ToMe is several percentage points worse than the baseline ViT-Adapter.

5 Discussion

Through our experiments we have demonstrated how our proposed ATC method consistently outperforms prior merging-based token reduction methods across a diverse set of tasks. This includes the prior best merging-based method ToMe, which we confidently outperform across all considered tasks. While our analysis have been focused on comparing ATC with prior merging-based approaches,

we also find that ATC outperforms pruning-based methods on the image classification task (Following the setup from Sec. 4.1) and object detection and segmentation, where ATC outperforms the prior state-of-the-art pruning-based SViT method from Liu *et al.* [38]. Detailed experimental results including the pruning-based methods are available in the supplementary materials, and establishes that our ATC method is the current best token reduction method.

6 Limitations

We show that ATC is a very adaptable method, achieving state-of-the-art performance across the different classification, image synthesis, and object detection & segmentation tasks considered. However, ATC is not without its limitations. Firstly, ATC requires the selection of a linkage function, adding an extra hyperparameter. While there is not a specific linkage function that is the best at all tasks, we find that the average or complete linkage functions are good general choices, whereas the single linkage function often underperforms when working with more aggressive keep rates, matching prior linkage function recommendations [16]. Secondly, we find that the inference throughput of the current implementation of ATC is hampered by the available implementations of the agglomerative clustering functions, which are all non-batched and often CPU-bounded. This is discussed and analyzed at lengths in the supplementary materials. However, there are also clear indications that these limitations can be lifted if the current frameworks are slightly modified.

7 Conclusion

In this work, we introduce Agglomerative Token Clustering (ATC), a novel hard-merging token reduction approach grounded in the principles of classical bottom-up hierarchical clustering. ATC distinguishes itself by efficiently merging redundant observations early on, thus preserving the semantic richness of more diverse observations. We evaluate our method across the most diverse sets of tasks considered in the literature, covering both classification, synthesis, detection, and segmentation tasks. In image classification, image synthesis, and object detection & segmentation, ATC sets a new state-of-the-art, highlighting its significant contribution to the field. We also demonstrate that ATC can achieve comprable performance to the prior state-of-the-art without any fine-tuning, *i.e.* when applied *off-the-shelf*. We are optimistic that ATC will inspire subsequent advancements, leveraging classical clustering methods to enhance modern neural architectures.

Acknowledgements

This work was supported by the Pioneer Centre for AI (DNRF grant number P1), partially supported by the Spanish project PID2022-136436NB-I00 and by ICREA under the ICREA Academia programme, and the Canada CIFAR AI Chairs.