Training-Free Visual Token Compression via Delayed Spatial Merging

Jung Hwan Heo Seyedarmin Azizi Arash Fayyazi Massoud Pedram
University of Southern California

Abstract

Token compression is an emerging paradigm that accelerates the inference of Vision Transformers (ViTs) without any retraining or fine-tuning. To push the frontier of **training-free** acceleration in ViTs, we improve token merging by adding the perspectives of 1) activation outliers and 2) hierarchical representations. Through a careful analysis of the attention behavior in ViTs, we characterize a delayed onset of the *convergent attention phenomenon*, which makes token merging undesirable in the bottom blocks of ViTs. Moreover, we augment token merging with a hierarchical processing scheme to capture *multi-scale redundancy* between visual tokens. Combining these two insights, we build a unified inference framework called **DSM**: **D**elayed **S**patial **M**erging. We extensively evaluate DSM on various ViT model scales (Tiny to Huge) and tasks (ImageNet-1k and transfer learning), achieving up to 1.8× FLOP reduction and 1.6× throughput speedup at a negligible loss while being *two orders of magnitude faster* than existing methods.

1 Introduction

Transformers (Vaswani et al., 2017) has become a general-purpose backbone architecture that drove great progress in language modeling (Devlin et al., 2019), speech recognition (Tian et al., 2020), to computer vision (Dosovitskiy et al., 2020). Compared to Convolutional Neural Networks (CNNs), ViTs have minimal inductive bias, benefiting from large-scale pretraining. Modern self-supervised models such as MAE obtain up to 90.94% top-1 accuracy on ImageNet-1k (Wortsman et al., 2022).

However, efficient deployment of ViTs remains a challenge due to the large model size. A major line of work has focused on pruning task-irrelevant tokens with various importance metrics such as token embeddings (Yin et al., 2022), attention scores (Liang et al., 2022), and lightweight neural network predictors (Rao et al., 2021). Recently, a newly proposed token merging scheme enabled a *training-free* approach to token reduction (Bolya et al., 2023). While prior work can effectively accelerate ViT inference, using these techniques in practice is still challenging. Previous approaches train from scratch (Liang et al., 2022), fine-tune with extra parameters (Rao et al., 2021), and optimize with additional loss functions that increase the wall-clock training time (Yin et al., 2022). Such complexities introduce extra computational budgets and engineering efforts. Token merging scheme can avoid this via the training-free mode (Bolya et al., 2023), but it incurs nontrivial accuracy loss, which ultimately necessitates training from scratch to achieve competitive performance.

To push the frontier of training-free ViT acceleration via token merging, we turn to recent findings of activation outliers in large Transformers (Darcet et al., 2023; Xiao et al., 2023) as well as a principled hierarchical processing technique (Jarrett et al., 2009; Lee et al., 2009; Krizhevsky et al., 2009). Augmented by our delayed and hierarchical merging schemes, DSM yields a strong token merging technique that is aware of both Transformer attention mechanics and multi-scale redundancies. Our contributions are summarized as follows:

Workshop on Machine Learning and Compression @ NeurIPS 2024.

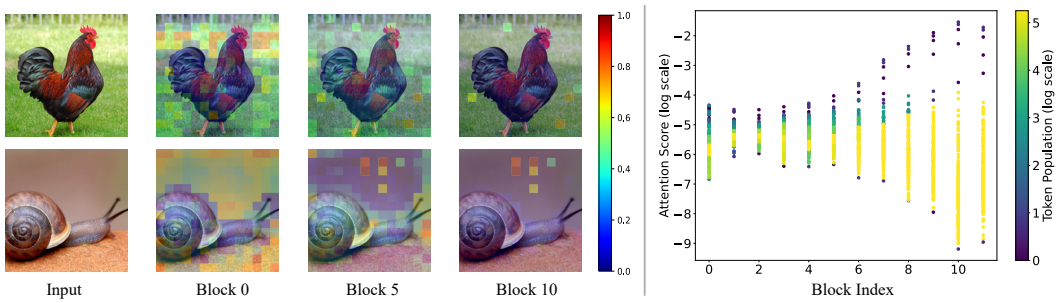

Figure 1: Connection between high-norm activation outliers and Attention Sinks on ViT-S. **Left:** Low-information background tokens progressively collect most of the attention scores across the network. **Right:** Such outlier tokens receive orders of magnitude higher attention values. Critically, we observe that there is a *delay* in which the outlier tokens begin to emerge, which inspires further investigation of the attention behavior.

- We find that the recently discovered high-norm token outliers in ViTs (Darcet et al., 2023) are attributed to the Attention Sink behavior in language models (Xiao et al., 2023). By carefully studying the attention mechanics in ViTs, we identify an intriguing phenomenon that we call *delayed convergent attention*.
- Motivated by the observation that 1) token merging is undesirable in the bottom Transformer blocks and 2) hierarchical image processing captures multi-scale interactions, we present a unified inference framework called Delayed Spatial Merging (DSM).
- We extensively evaluate DSM on ViT and DeiT models of various scales (Tiny $\sim$ Huge) on ImageNet-1k and transfer learning tasks. With no more than a 1% drop in accuracy, our framework achieves 1.8$\times$ FLOP reduction and 1.6$\times$ speedup on NVIDIA A6000 GPU.

2 Delayed Spatial Merging

Tracing Attention Sinks in ViTs. Recently, high-norm activation outliers have been observed in ViTs, which act as registers that pool global information (Darcet et al., 2023; Bondarenko et al., 2023). We find inspiration from the Attention Sink behavior from language models (Xiao et al., 2023) to trace the source of such outliers. As in Figure 1, we first verify that the high-norm outlier tokens in ViTs are related to the Attention Sink behavior. Although initialized to a Gaussian-like distribution, attention scores are progressively accumulated on only a few background tokens, leading to orders of magnitude differences between scores of the outlier sink tokens and the other token. Interestingly, we observe an initial delay before the attention sinks emerge. This naturally raises two questions: *Why does this delay exist, and how does it affect token merging?*

2.1 Delayed Merging

Vanilla Token Merging. A typical transformer block in a ViT consists of a multi-head attention (MHA) layer and a Feedforward Network (FFN) layer. For the l-th transformer block in a network of depth L, the forward pass may be expressed as follows:

$$\bar{\mathbf{X}}^l = \mathbf{X}^l + \mathrm{MHA}(\mathbf{X}^l), \mathbf{X}^{l+1} = \bar{\mathbf{X}}^l + \mathrm{FFN}(\bar{\mathbf{X}}^l), \tag{1}$$

where $\mathbf{X}^l \in \mathbb{R}^{N \times C}$ is the input sequence with N tokens, each with an embedding size of C. Token merging is applied within each transformer block between MHA and FFN (Bolya et al., 2023). Given a sequence of n tokens (MHA layer output), denoted by $\bar{\mathbf{X}}^l = [x_1, ..., x_n]$, a weighted complete bipartite graph comprising two sets of nodes (tokens): $\mathbb{A} = [x_1, x_3, ..., x_{n-1}]$ and $\mathbb{B} = [x_2, x_4, ..., x_n]$ is constructed. An edge between token $a \in \mathbb{A}$ and token $b \in \mathbb{B}$ captures the cosine similarity between embeddings of a and b. A weighted bipartite graph matching algorithm is then applied to identify the set of $r \leq n/2$ edges that have the maximum weighted sum. The tokens associated with each of these r edges are then merged using a channel-wise weighted average of the embeddings. Finally, the two sets $\mathbb{A}$ and $\mathbb{B}$ are combined to yield a truncated sequence of tokens with r fewer tokens.

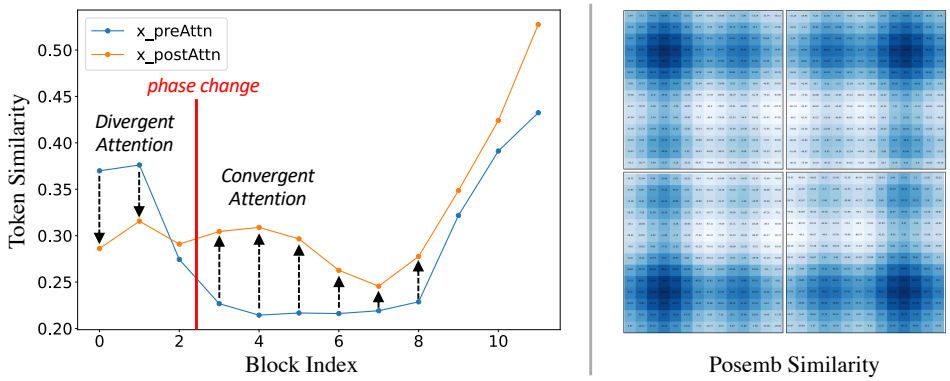

Figure 2: Token similarity and spatial similarity (via positional embedding) computed using Equation 2 on DeiT-S. **Left:** The first few attention blocks have decreasing similarity, and then it increases for the rest of the network. It is desirable to merge tokens when they are becoming similar (convergent), motivating the delayed merging scheme. **Right:** Learned positional embeddings suggest that spatial proximity correlates with embedding similarity. We use this insight to partition the tokens into local windows and thus exploit multi-scale redundancies.

Characterizing Convergent Attention. We now investigate how the delay in Attention Sinks affects token similarity distribution. Intuitively, an ideal scenario to conduct token merging would be when tokens are most similar to each other (i.e., avoid forced merging of tokens when they are dissimilar.) To quantify the degree of similarity among tokens, we adopt the token similarity metric which has been widely used in text generation (Zhang et al., 2019):

$$Sim = \frac{1}{n\,(n-1)} \sum_{i \neq j} \frac{x_i^T x_j}{||x_i||_2 ||x_j||_2}, \tag{2}$$

A higher Sim score indicates that the tokens in the layer have similar embeddings. Interestingly, as in Figure 2, the initial blocks have tokens become less similar during attention (*divergent* attention), while after a certain point in the ViT (a phase change starting at block 2), tokens consistently become more similar (*convergent* attention). Since it is counterproductive to merge tokens that exhibit diversifying embeddings, we delay merging until the embeddings stabilize to convergent attention.

2.2 Spatial Merging

Hierarchical image processing is a fundamental technique that spans a wide range of computer vision modeling from semantic segmentation (Long et al., 2015), object detection (Jarrett et al., 2009; Lin et al., 2017), to 3D rendering via Neural Radiance Fields (Barron et al., 2021). We introduce the principle of hierarchical representations to token merging for the first time. The intuition is to capture multi-scale interactions between visual tokens such that the similarity (feature redundancy) search process can be done in finer granularity.

Neighboring pixels in an image having stronger semantic relationships with each other; for example, a picture of an animal has contiguous body parts where *spatial proximity* correlates well with *semantic similarity*. Instead of globally searching for similar tokens, we constrain the search space to local windows. The input tokens can be represented as a 2D grid with dimensions (H, W), which we partition into four equally-sized windows with dimension w. To minimize the complexity, we set initial window size to $w = 7$ in all of our experiments as it nicely divides 14×14 grid of tokens (224×224 resolution w/ common patch size of 16). When the number of tokens is not divisible by w, we apply padding in the bottom right to retain the 2D formation.

Rather than a static window size w, we progressively increment the window size in every block. This is based on the intuition that positional similarity, as positional embeddings are added right before block 0, is most relevant in the earlier part of the network. Thus, we increment the windows every block until it equivalently reduces to global merging (where the window is as big as the remaining 2D grid of tokens). Windows can be stacked to efficiently merge tokens in parallel. This

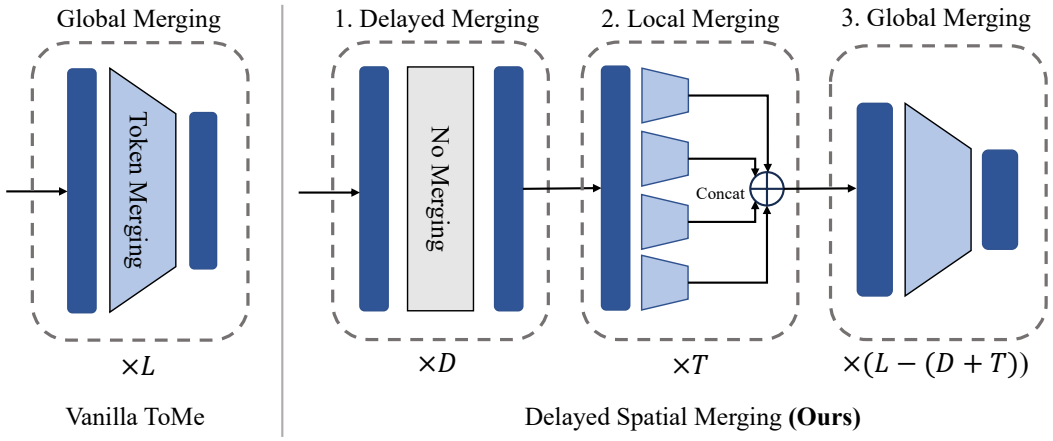

Figure 3: **Left:** The original Token Merging (ToMe) by Bolya et al. is globally applied to all tokens for all L transformer blocks. **Right:** Motivated by the principles of convergent attention and spatial awareness, our proposed Delayed Spatial Merging (DSM) augments ToMe by not merging in the initial D blocks, locally merging for T blocks, then globally merging for the rest of the network.

is possible because token merging is applied independently for each example in a batch, and the window dimension can be fused into the batch dimension B: (B, H, W) $\rightarrow$ (B, H/w, w, W/w, w) $\rightarrow$ (B * H/w * W/w, w, w). Efficient kernel implementation of the window stack operation is possible as demonstrated in Liu et al. (2021).

Unified Inference Framework. As in Figure 3, DSM augments the vanilla token merging technique with delayed merging and localized merging. For a network with depth L, we delay for D blocks, apply localized merging for T blocks with a window size of w, and execute global merging for the rest of the network. The only hyperparameter we tune is r, which is the number of tokens to reduce in a single token merging layer; we further discuss hyperparameter settings in Section B.

3 Experiments

We conduct our experiments in the ImageNet-1K dataset (Russakovsky et al., 2015) and four different transfer learning tasks (Pets (Parkhi et al., 2012), Flowers (Nilsback & Zisserman, 2008), Aircraft (Maji et al., 2013), CIFAR-100 (Krizhevsky, 2009)) to evaluate the effectiveness of our method in accelerating off-the-shelf ViTs on classification tasks. The computational cost is measured in FLOP with the Torchprofiler library. Inference throughput is measured on an Nvidia RTX A6000 GPU with a fixed batch size of 32 averaged over 50 runs. End-to-end (E2E) training time is measured in a single 8 GPU node.

	Top-1	GFLOP	Epochs	E2E (hrs)
DeiT-S	79.8	4.6	0	0
DynamicViT (Rao et al., 2021)	79.3	2.9	30	44.8
SPViT (Kong et al., 2021)	79.3	2.6	60	–
A-ViT (Yin et al., 2022)	78.6	2.9	100	76.4
E-ViT (Liang et al., 2022)	79.1	2.6	300	154.4
ATS (Fayyaz et al., 2022)	79.7	2.9	30	–
ToMe (Bolya et al., 2023)	79.4	2.7	300	102.2
Spatial Merging **(Ours)**	79.3	2.8	0	0
DSM **(Ours)**	78.6	2.5	0	0

Table 1: **Comparison to Prior Work.** Our framework provides competitive performance while being two orders of magnitude faster. E2E training time is measured in a single 8 GPU node.

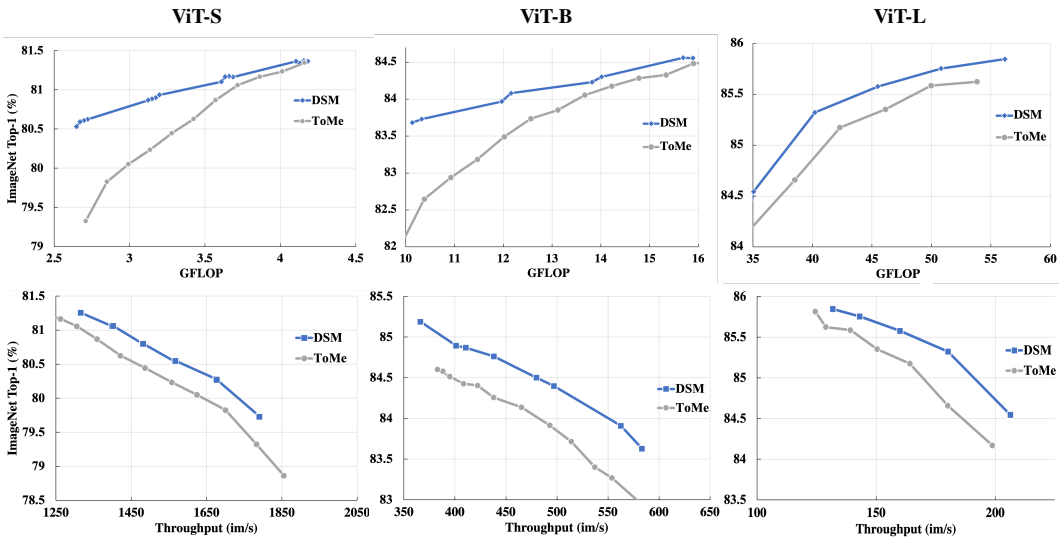

Figure 4: **Comparison to ToMe.** We apply our inference framework to several state-of-the-art ViT models in a *training-free* fashion. DSM's hyperparameters are fixed via network architecture, only varying parameter r to produce Top-1 Acc. vs. GFLOP curves on ImageNet-1k.

	Oxford-IIIT Pet			Flowers102			FGVC-Aircraft			CIFAR-100		
	acc@1	GFLOP	im/s	acc@1	GFLOP	im/s	acc@1	GFLOP	im/s	acc@1	GFLOP	im/s
Baseline	92.12	17.57	404.53	98.13	17.57	407.27	81.12	17.57	408.88	91.10	17.57	402.51
$r = 4$	92.04	16.10	384.51	98.19	16.10	384.54	80.98	16.10	388.25	90.96	16.10	383.1
$r = 8$	92.01	14.46	431.37	97.95	14.46	430.18	80.95	14.46	436.20	91.01	14.46	431.72
$r = 12$	91.74	13.27	468.78	97.69	13.27	466.17	81.07	13.27	471.10	90.90	13.27	469.10
$r = 16$	91.55	11.96	517.05	97.45	11.96	513.75	80.38	11.96	518.86	90.76	11.96	517.76
$r = 20$	91.32	10.16	612.50	97.14	10.16	609.61	80.20	10.16	612.11	90.25	10.16	613.92

Table 2: **Transfer Learning.** Fine-tuned ViT-B accelerated with DSM consistently achieves 1.5× speedup across various datasets.

Comparison to Training-based Approaches. As in Table 1, our DSM achieves competitive performance while being *two orders of magnitude faster* than existing approaches thanks to the training-free approach. For example, E-ViT Liang et al. (2022) takes around 154 single GPU hours for one run. Since it requires running the method for each target speedup, the cost of deploying to various resource constraints can become quickly intractable.

Comparison to Training-free Approaches. In Figure 4, we apply our framework to ViT-[S, B, L] *off-the-shelf* with 224px and patch size 16. For each model, we benchmark DSM against ToMe. We vary r to construct two Pareto curves that compare Top-1 accuracy to #MACs and throughput. Note, we sweep with higher r values with the DSM to match the computational load of ToMe. We can see that our framework consistently gives better results than ToMe, especially for smaller models. Remarkably, we can save 45% and 42% of the FLOP within a 1% loss for ViT-S and ViT-B, respectively. Relative to vanilla ToMe, it can improve the accuracy by more than 1%. Yet, the success of DSM inversely scales with model size, showing a negligible gain for ViT-L. Compared to the success of DSM in saving FLOP, the throughput gains are relatively marginal. We think this is because the additional data movements, such as sorting, padding, and modifying tensor dimensions, cause a nontrivial overhead. Interestingly, this overhead is less obvious for larger models, as the DSM curve shifts to the right with better trade-off margins. With larger models that execute heavy loads of matrix multiplications, computing becomes a bottleneck rather than data movement. This makes the memory I/O overhead from localized merging less evident for larger models. For transfer learning experiments as in Table 2, we consistently observe a 1.6× speedup and 1.74× FLOP reduction across four different downstream tasks, which shows for the first time that DSM is generalizable to task-specific datasets as well.

Figure 5: **Qualitative comparison** of DSM to ToMe using a ViT-L_{384} model. Merged tokens share the same border and filling color. DSM merges more contiguous patches that are semantically similar, leading to more interpretable outcomes that retain the original features. For example, the skier's legs are merged with the background in ToMe, while DSM provides correct segmentations.

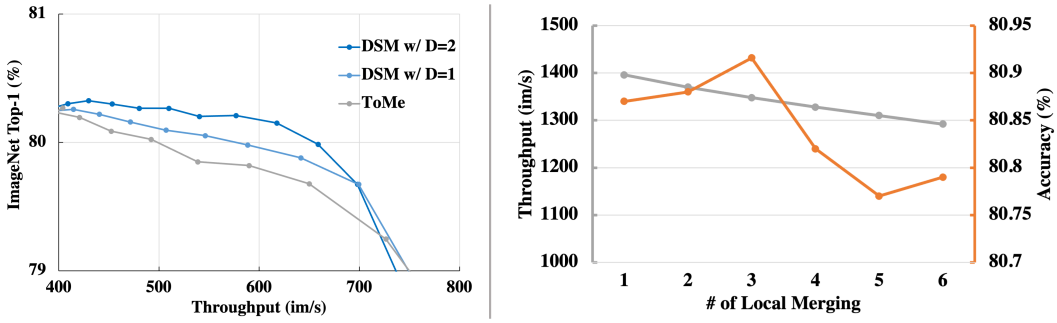

Figure 6: Ablations for Delayed Merging and Spatial Merging to verify our hyperparameters and their efficacy in advancing the overall DSM framework. **Left:** Using Delayed Merging can achieve 1.6× lossless acceleration in throughput, with the effect most pronounced when skipping two blocks $D = 2$. **Right:** Spatial Merging has a throughput cost due to the additional data movements. We find a general sweet spot for $L = 3$, the default setting used in the increasing window technique.

3.1 Analysis

In Figure 5, we show the input tokens belonging to the final merged token. We use $r = 24$ for ToMe and $r = 28$ with $D = 4$ and $w = 8$ for our framework. Note that the parameters are different since the resolution is higher. To match the final token count, we do not merge the last block in our framework. We see that in the second image, the face of the Maltese is contiguously merged into a single token for us, while ToMe separates out the nose. The same is true for the body of a Huskey in the first photo and the people in the center of the third photo, where our framework tends to merge more contiguous tokens.

Through ablation studies in Figure 6, we test our hypothesis that merging is undesirable in the initial blocks due to the delayed convergent attention phenomenon. By skipping the first two blocks with the $D = 2$ configuration, DSM achieves a remarkable 1.6× speedup with 44% FLOP savings with a mere **0.2%** loss. We also determine the optimal number of blocks to apply Spatial Merging by plotting the throughput vs. accuracy tradeoff, where $L = 3$ is a sweet spot.

References

Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P. P. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., and Kolesnikov, A. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10925–10934, 2022.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Bondarenko, Y., Nagel, M., and Blankevoort, T. Quantizable transformers: Removing outliers by helping attention heads do nothing. *arXiv preprint arXiv:2306.12929*, 2023.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

Devlin, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019.

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.

Fayyaz, M., Koohpayegani, S. A., Jafari, F. R., Sengupta, S., Joze, H. R. V., Sommerlade, E., Pirsiavash, H., and Gall, J. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, pp. 396–414. Springer, 2022.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heo, J. H., Kim, J., Kwon, B., Kim, B., Kwon, S. J., and Lee, D. Rethinking channel dimensions to isolate outliers for low-bit weight quantization of large language models. *arXiv preprint arXiv:2309.15531*, 2023.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pp. 2146–2153. IEEE, 2009.

Kim, S., Gholami, A., Yao, Z., Mahoney, M. W., and Keutzer, K. I-BERT: integer-only BERT quantization. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5506–5518. PMLR, 2021. URL `http://proceedings.mlr.press/v139/kim21d.html`.

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Ren, B., Qin, M., Tang, H., and Wang, Y. Spvit: Enabling faster vision transformers via soft token pruning. *arXiv preprint arXiv:2112.13890*, 2021.

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., Qin, M., and Wang, Y. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T. (eds.), *Computer Vision – ECCV 2022*, pp. 620–640, Cham, 2022. Springer Nature Switzerland.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Technical report, Univ. Toronto, 2009.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., and Alistarh, D. The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models. *CoRR*, abs/2203.07259, 2022. doi: 10.48550/arXiv.2203.07259. URL https://doi.org/10.48550/arXiv.2203.07259.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pp. 609–616, 2009.

Liang, W., Yuan, Y., Ding, H., Luo, X., Lin, W., Jia, D., Zhang, Z., Zhang, C., and Hu, H. Expediting large-scale vision transformer for dense prediction without fine-tuning. *Advances in Neural Information Processing Systems*, 35:35462–35477, 2022.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and 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.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M. B., and Vedaldi, A. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. URL http://arxiv.org/abs/1306.5151.

Marin, D., Chang, J. R., Ranjan, A., Prabhu, A., Rastegari, M., and Tuzel, O. Token pooling in vision transformers. *CoRR*, abs/2110.03860, 2021. URL https://arxiv.org/abs/2110.03860.

Miller, E. Attention is off by one. 2023. URL https://www.evanmiller.org/attention-is-off-by-one.html.

Nilsback, M. and Zisserman, A. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pp. 722–729. IEEE Computer Society, 2008. doi: 10.1109/ICVGIP.2008.47. URL https://doi.org/10.1109/ICVGIP.2008.47.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pp. 3498–3505. IEEE Computer Society, 2012. doi: 10.1109/CVPR.2012.6248092. URL https://doi.org/10.1109/CVPR.2012.6248092.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Ryoo, M. S., Piergiovanni, A. J., Arnab, A., Dehghani, M., and Angelova, A. Tokenlearner: Adaptive space-time tokenization for videos. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12786–12797, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html.

Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Q-BERT: hessian based ultra low precision quantization of BERT. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8815–8821. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6409.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *CoRR*, abs/2106.10270, 2021. URL https://arxiv.org/abs/2106.10270.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tian, Z., Yi, J., Bai, Y., Tao, J., Zhang, S., and Wen, Z. Synchronous transformers for end-to-end speech recognition. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pp. 7884–7888. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054260. URL https://doi.org/10.1109/ICASSP40776.2020.9054260.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1580. URL https://doi.org/10.18653/v1/p19-1580.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Xiao, G., Lin, J., Seznec, M., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. *arXiv preprint arXiv:2211.10438*, 2022.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. *arXiv*, 2023.

Yin, H., Vahdat, A., Alvarez, J. M., Mallya, A., Kautz, J., and Molchanov, P. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10809–10818, 2022.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

A Related Work

A.1 Efficient Transformers

Notable progress has been made to reduce the high computational cost of neural networks and enable efficient deployment to resource-constrained environments. At the algorithmic level, methods such as model quantization Shen et al. (2020); Kim et al. (2021); Xiao et al. (2022), model pruning Han et al. (2015); Voita et al. (2019); Kurtic et al. (2022), and knowledge distillation Hinton et al. (2015); Beyer et al. (2022) have gained popularity. Orthogonal to weight pruning, token compression Yin et al. (2022); Rao et al. (2021); Liang et al. (2022) has shown that transformer inputs can be dynamically pruned at inference time. In this paper, we focus on adaptive token compression techniques, where token reduction decisions are conditioned on the input image.

A.2 Token Compression

Token Pruning accelerates the inference of ViT models by discarding less important tokens. Various prior work studies have worked on identifying such token redundancies. DynamicViT Rao et al. (2021), for example, trains a token importance predictor using the Gumbel-Softmax distribution. A-ViT Yin et al. (2022) learns about the importance of tokens by introducing a loss function that penalizes unpruned tokens. E-ViT Liang et al. (2022) uses attention scores from the [CLS] token as the importance heuristic. Although these methods are effective post-deployment, they require costly retraining from scratch or fine-tuning from a model checkpoint. In contrast, our work focuses on completely bypassing such usability barriers (Table 3).

Token Merging combines tokens instead of pruning them. Prior works have attempted to fuse unimportant tokens into a single token using custom heuristics Kong et al. (2022) or learnable MLP projections such as the TokenLearner Ryoo et al. (2021). Token pooling has also been proposed as a downsampling method via merging Marin et al. (2021); however, its iterative k-means-based method is slow and incompatible with the off-the-shelf models. ToMe Bolya et al. (2023), which was recently introduced as a token merging module utilizing a bipartite graph matching algorithm, achieves comparable accuracy to token pruning without any retraining. Our work makes a case for token merging as a preferred building block for training-free acceleration and makes improvements to push the pareto frontier of the accuracy-efficiency trade-off.

A.3 Token Outliers

To improve the token merging technique, we tackle it from the perspective of a recently observed token outlier problem, which occurs in large transformer models for both vision and language tasks. Token outliers were popularized in activation quantization research, where certain tokens or channels have much higher activation magnitude than others Xiao et al. (2022); Dettmers et al. (2022); Lin et al. (2023); Heo et al. (2023). Similarly, both supervised and unsupervised ViTs have identified token outliers Bondarenko et al. (2023); Darcet et al. (2023), where they are characterized as low-information background tokens that pool global information (similar to the function of the [CLS] token).

The cause of token outliers can be traced back to the Softmax function in attention, where the attention must sum up to one Miller (2023); Xiao et al. (2023). When the attention head does not want to

Table 3: Comparison of different token compression techniques. Our Delayed Spatial Merging (DSM) framework fully embraces training-free acceleration.

	Pretrain	Finetune	Training-free
DynamicViT Rao et al. (2021)	✗	✓	✗
SPViT Kong et al. (2021)	✗	✗	✗
A-ViT Yin et al. (2022)	✗	✓	✗
E-ViT Liang et al. (2022)	✓	✓	✗
ATS Fayyaz et al. (2022)	✗	✗	✗
ToMe Bolya et al. (2023)	✓	✗	✓†
DSM (**Ours**)	✗	✗	✓

† susceptible to accuracy degradation.

update the residual stream, the head executes a "no-op" by attending heavily to a low-information token Bondarenko et al. (2023). In this work, we confirm that the "attention sinks" caused by the Softmax function are present–a fact that is subsequently used as a foundation to explore the unique attention behavior in ViTs.

A.4 Training-Free Acceleration

For ViTs, most of the models used in classification tasks are small (or tiny) variants in the ViT and DeiT model families. Prior training-based token reduction techniques have experimented with a focus on small models due to the high training cost of larger models Rao et al. (2021); Liang et al. (2022). When considering the training and hyperparameter tuning costs, the total computations can become unwieldy for many researchers and practitioners Steiner et al. (2021). Motivated by the fact that the highest-performing models are too expensive to compress, we propose **a training-free framework for compressing large ViTs**. We take the method's speed as an equally important figure of merit as the final model performance and constrain our solution to be training-free. Our work addresses the following question: How do we compress ViTs without expensive training to realize high-accuracy inference models?

B Detailed Methodology

DSM Hyperparameters. We fix the delay parameter D to be the transition point where the convergent attention behavior begins to emerge. That is, for a DeiT-S model with a depth of 12, we choose $D = 2$ as the convergent attention appears in the second block (Figure 2). We visualize additional networks in Figure 7 and 8, where the 1/6th point of the network is generally the point at which the attention behavior switches from divergent to convergent.

The localized merging parameter T can be fixed as a function of the window size w and the reduction rate r. This is because localized merging with progressively increasing window size naturally degenerates into global merging. With gradual token merging, the sequence length becomes smaller than the window size itself. Thus, increasing the window size yields a partial localized merging that smoothly transitions to global merging.

C More Experiments

C.1 The Case for Merging

Before delving into our DSM evaluation, we first make a more fundamental case that token merging is the right building block over token pruning for training-free ViT acceleration. Off-the-shelf ViT models are commonly pre-trained with a dense token distribution (no token dropping). Thus, a trained model "expects" to see not only task-relevant tokens but also less relevant ones like background tokens. Less informative tokens can also function as regularization, which makes it risky to assume that less important tokens can be removed without degrading the prediction performance. Thus, token pruning may not be the optimal design choice for the training-free setting.

Heuristics Ablation for Vanilla Token Compression. As in Table 4, we empirically support the case for token merging by comparing pruning to merging with various importance criteria. For pruning, the lowest L2 norm is dropped; for merging, the highest cosine similarity score is merged. We observe that the output of attention block X is a surprisingly good heuristic for pruning, but it lags merging options by 2%. The K embedding criteria yield the highest performance for merging. It best represents the tokens, even more than X, which has a larger embedding size per token. This may be due to of overparameterized embeddings, where having more channels can result in noise. Since K has less number of channels through the multi-head attention, its compact representation can resolve this problem.

C.2 Main results

We also evaluate the efficacy of DSM against training-based acceleration in various vision architectures that are not transformer-based. CNNs are known to be more parameter-efficient due to the weight-sharing nature of convolutions and smaller peak memory (since it does not use quadratic

Table 4: **Prune vs. Merge** comparison using ViT-L with $r = 7$. Merging retains accuracy more effectively in training-free settings. X is inside the attention block.

	Prune		Merge	
Features	acc	im/s	acc	im/s
Random	2.96	131.3	61.89	136.5
X	**81.58**	129.1	83.41	125.7
K	71.86	**130.7**	**83.51**	**132.7**
Q	73.65	130.3	83.25	131.3
V	78.9	130.3	83.44	132.3

Table 5: Comparison to convolution-based vision architectures. We normalize the speedup by the base model DeiT-S by using the FLOP count, which is a standard metric reported by the efficient convolution models.

Model	Top-1	Speedup (↑)
DeiT-S	79.8	1×
EfficientNet-B2 Tan & Le (2019)	80.1	1.33×
EfficientNet-B3 Tan & Le (2019)	81.6	0.78×
ResNet-152 He et al. (2016)	78.3	0.56×
RegNetY-4GF Radosavovic et al. (2020)	80.0	1.23×
DeiT-S w/ DSM (r16)	79.6	1.5×
DeiT-S w/ DSM (r18)	79.4	1.6×

Table 6: Comparison of ViT-H and DeiT-T @ ImageNet-1k

	Δ Accuracy(%)	Throughput (image/sec)	# MACs (G)
ViT-Huge			
ToMe	-0.2	50.18	145.84
DSM	-0.2	**56.90**	**129.64**
ToMe	-0.6	64.01	113.90
DSM	-0.6	**72.60**	**101.79**
ToMe	-0.8	70.38	103.36
DSM	-0.8	**79.30**	**92.59**
DeiT-Tiny			
ToMe	-0.1	2457	1.18
DSM	-0.1	**2722**	**1.09**
ToMe	-0.5	3020	0.93
DSM	-0.5	**3257**	**0.86**
ToMe	-2.0	3881	0.69
DSM	-2.0	**4001**	**0.71**

self-attention). As in Table 5, we observe that DeiT-S w/ DSM performs comparably in the accuracy-compute trade-off with much more expensive methodologies such as EfficientNet via Neural Architecture Search Tan & Le (2019).

Moreover, we conduct additional experiments for both larger (ViT-H) and smaller (DeiT-Ti) models. Table 6 compares DSM against ToMe using ViT-H and DeiT-T, respectively.

D Visualizations

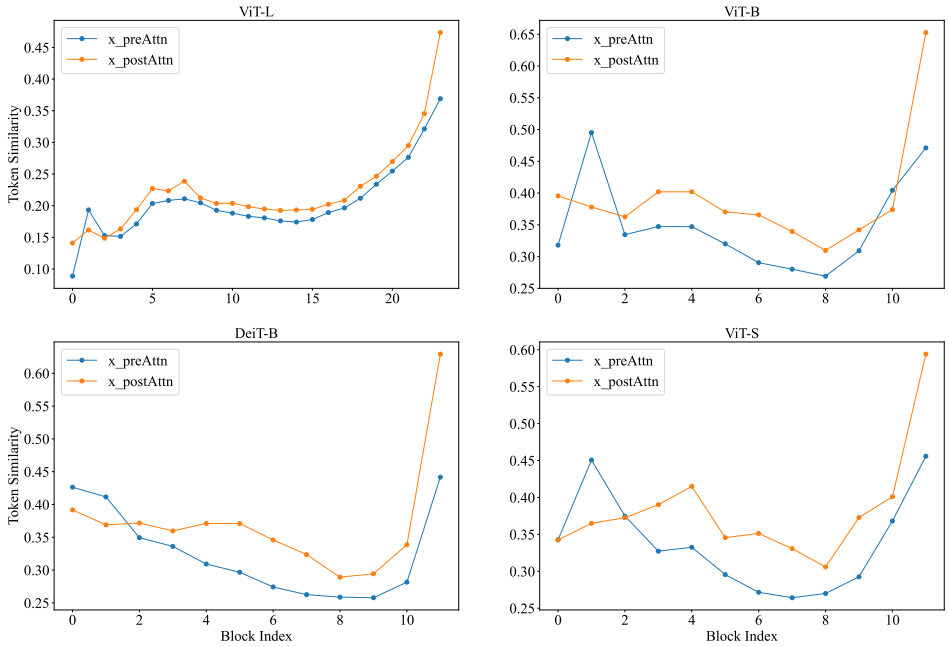

Figure 7: **Delayed convergent attention** phenomena is observed for various pretrained visual transformers. Attention block consistently makes the tokens more similar after a certain threshold layer, which is around 1/6th of the network.

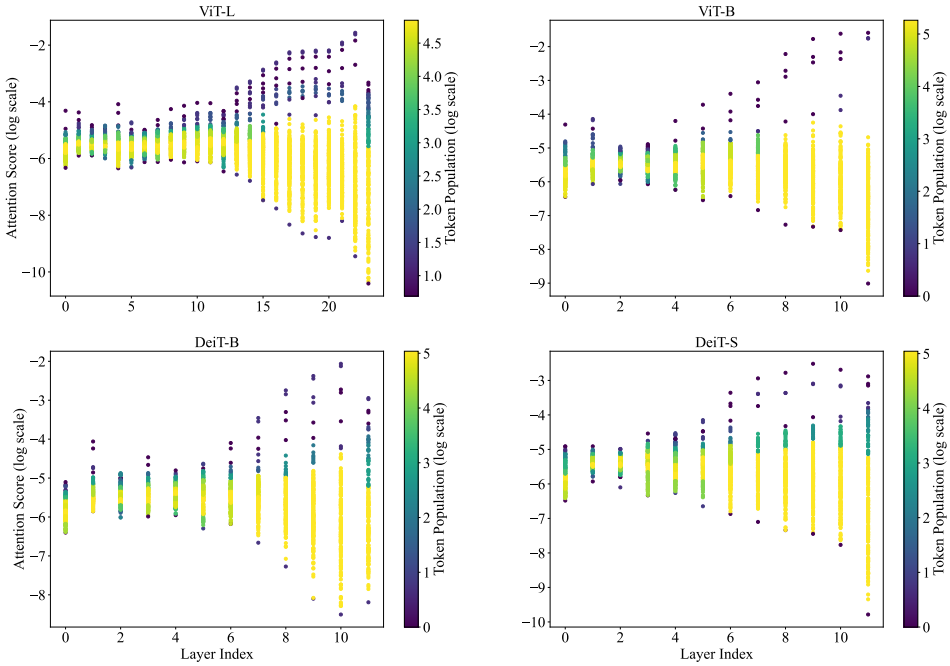

Figure 8: Outlier tokens observed in different ViT architectures.

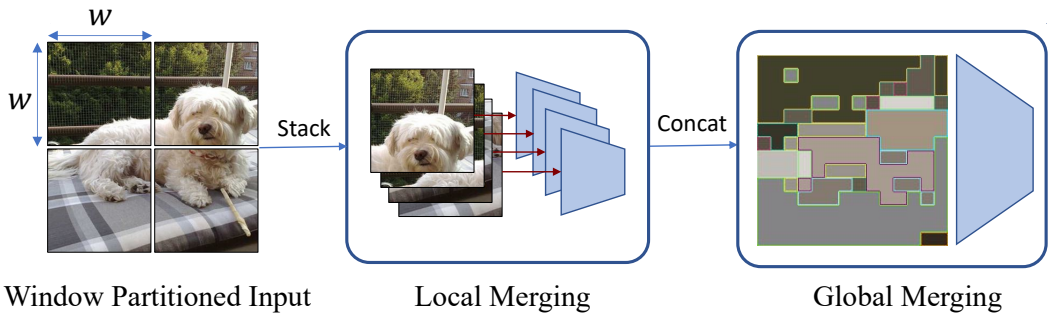

Figure 9: Local merging with window partitioning is illustrated with a visual input.