ACCELERATING VISION TRANSFORMERS WITH ADAPTIVE PATCH SIZES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers (ViTs) partition input images into uniformly sized patches regardless of their content, resulting in long input sequence lengths for high-resolution images. We present Adaptive Patch Transformers (APT), which addresses this by using multiple different patch sizes within the same image. APT reduces the total number of input tokens by allocating larger patch sizes in more homogeneous areas and smaller patches in more complex ones. APT achieves a drastic speedup in ViT inference and training, increasing throughput by 40% on ViT-L and 50% on ViT-H while maintaining downstream performance. It can be applied to a previously fine-tuned ViT and converges in as little as 1 epoch. It also significantly reduces training and inference time without loss of performance in high-resolution dense visual tasks, achieving up to 30% faster training and inference in visual QA, object detection, and semantic segmentation. We will release all code and trained models.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2020) have become the dominant paradigm for visual recognition, but their scalability is limited by the quadratic cost of self-attention with respect to sequence length. Since inputs are divided into fixed-size patches, image resolution directly determines sequence length: higher resolution images yield disproportionately long token sequences despite much higher redundancy.

Many prior works have proposed solutions to this issue, typically by merging a fixed proportion of similar tokens (Bolya et al., 2022) or pruning uninformative ones with auxiliary predictors (Rao et al., 2021; Yin et al., 2022). While these reduce theoretical FLOPs, they face two drawbacks. Firstly, a fixed reduction ratio is mismatched to image complexity: merging only half the tokens in a pure white image is insufficient, while merging half the tokens in a busy cityscape is harmful. Secondly, pruning during the forward pass introduces padding and irregular shapes, often negating speedups in practice (Dehghani et al., 2021). In contrast to vision transformers, language models rely on adaptive tokenizers such as Byte-Pair Encoding (Sennrich et al., 2016) and SentencePiece Kudo & Richardson (2018), which flexibly assign tokens of varying lengths depending on subword frequency. This reduces input sequence size while improving performance, suggesting that variable-granularity tokenization can be more efficient than fixed-size splits.

Our key insight is that a similar idea can be applied to vision transformers. As illustrated in Figure 1, ViTs use the same amount of computation on a uniform green background as on the complex patches on the head of the bird, despite the significant difference in visual complexity. We introduce the Adaptive Patch Transformer (APT), which addresses this mismatch by varying patch sizes *within a single image*. Regions that are smooth and redundant can be represented with large patches, while regions rich in detail are allocated smaller patches. This content-aware patchification preserves important information where it matters while reducing redundancy elsewhere. To do this, APT computes entropy at multiple scales and assigns larger patch sizes to regions with the lowest entropy, resulting in significantly fewer input tokens. We then down-sample the larger patches and combine their patch embeddings with the information from the original large patch using a zero-initialized MLP, allowing APT to converge without harming the network.

APT speeds up ViT inference *and* training by almost 40%, with even larger boosts for higher resolution images and larger models. When initialized from a self-supervised or large-scale pretrained

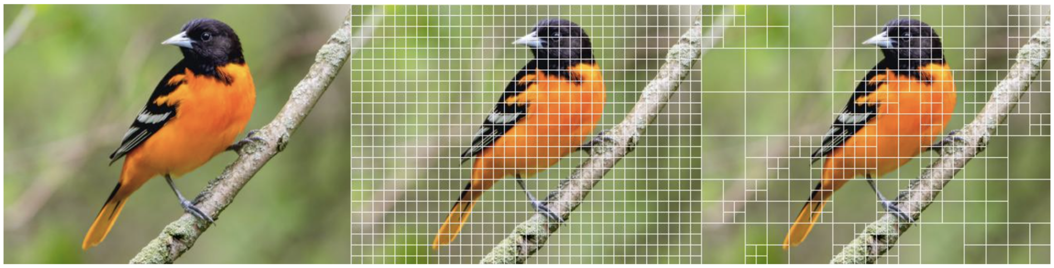


Figure 1: **Adaptive Patch Sizing.** We present APT, Adaptive Patch Transformers, which significantly accelerate vision transformer training and inference by patchifying images based on their content. Complex regions receive more, smaller tokens, while simpler, homogeneous regions receive fewer.

checkpoint, APT reaches the same performance as the original ViT after fine-tuning. If applied directly to an already fine-tuned ImageNet checkpoint, APT incurs only a small accuracy drop without additional training. With our zero-initialized MLP, this gap can be closed in as little as a single epoch of fine-tuning. We also find that unlike most prior token reduction works, APT can successfully accelerate vision transformers on a wide range of image understanding tasks, such as visual question answering, object detection, and semantic segmentation, while matching the baseline performance.

In summary, we (1) introduce the Adaptive Patch Transformer (APT), which accelerates Vision Transformers by up to 40% through content-aware patch sizes, with larger gains at higher resolutions and model scales; (2) show that APT preserves the accuracy of standard pretrained models across resolutions and scales; and (3) demonstrate that APT extends beyond ImageNet, performing well on dense prediction and vision-language tasks.

2 Related Work

Vision Transformers and Patchification. Vision Transformers (ViTs) (Dosovitskiy et al., 2020) are currently the *de facto* standard architecture for computer vision backbones (Xu et al., 2022; Kirillov et al., 2023; Peebles & Xie, 2022). In contrast to language models, which typically use subword tokenizers (Sennrich et al., 2016; Kudo & Richardson, 2018) with varying numbers of bits per token, ViTs *patchify* images into equally sized patches, each becoming a token. This can result in an enormous number of tokens, especially at high resolution. Transformer-based generative models (Peebles & Xie, 2022; Esser et al., 2020) use visual tokenizers, typically using a variational auto-encoder (Kingma et al., 2013; Van Den Oord et al., 2017), to project images into a compressed latent space, reducing the input size significantly. Some recent works explore adaptive visual tokenizers (Yan et al., 2024; Duggal et al., 2024), which dynamically allocate more tokens to more complex visual inputs, but do not meaningfully speed up training or generation. As a result, image understanding tasks are typically limited to lower resolution.

Reducing ViT Tokens. Accelerating ViTs by removing tokens is a rich area of research. Methods such as pruning (Yu & Xiang, 2023; Yang et al., 2023; Zheng et al., 2022), compressed representations (Wu et al., 2018; Park & Johnson, 2023), or quantization (Liu et al., 2021b; Li et al., 2022c; Moon et al., 2024) remove redundancies or compactly encode parameters, reducing inference time and memory usage. Alternative attention mechanisms, such as linearized (Katharopoulos et al., 2020; Lu et al., 2021) or local window attention (Liu et al., 2021a; Wei et al., 2023; Chen et al., 2023b), improve efficiency by limiting token interactions. More related to our work are methods that exploit the inherent redundancy of images (Meng et al., 2022; Yin et al., 2022; Kong et al., 2022; Rao et al., 2021) and videos (Choudhury et al., 2025; Ding et al., 2023) by pruning uninformative tokens. While these works are content-aware, most require learning which tokens are unhelpful, negating any training speedup and preventing inference on batch sizes greater than 1. Several methods instead merge tokens based on similarity (Bolya et al., 2022; Bolya & Hoffman, 2023; Liang et al., 2022b; Shang et al., 2024; Cao et al., 2023; Liang et al., 2022a; Tran et al., 2024; Kallini et al., 2024; Lee & Hong, 2024), which does accelerate training. However, merging methods

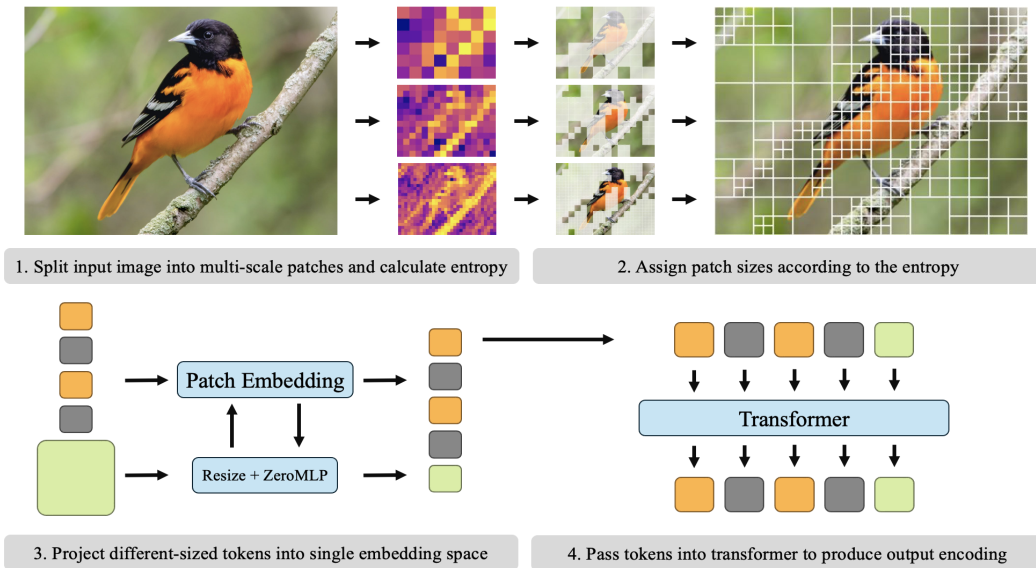


Figure 2: **APT overview.** APT works by measuring the entropy at multiple scales and assigning large patch sizes to low entropy patches. All patches are projected to the same size token embedding, and the reduced size input sequence is passed to the transformer.

typically combine a constant number of tokens for each input, which can be suboptimal for inputs with varying complexities. APT strikes a balance between these two lines of work by providing significant acceleration to training and inference while maintaining content-awareness.

Adaptive Patch Sizing for Efficient ViTs. Our work is not the first to propose using multiple patch sizes for faster ViTs. Early attempts in this direction (Chen et al., 2021; Beyer et al., 2023; Wang et al., 2024; 2021; Zhou & Zhu, 2023; Hu et al., 2024) train models that are capable of using different patch sizes, but still require a single patch size for each image. Closer to APT are works that allow for varying patch sizes within a single image (An et al., 2024; Ronen et al., 2023; Chen et al., 2023a; Bai et al., 2024). However, CF-ViT (Chen et al., 2023a) and Quadformer (Ronen et al., 2023) rely on a fixed number of patches, neglecting the variability of semantic information across images, which can lead to suboptimal performance. Closest to our work is MS-ViT (Havtorn et al., 2023), which, like DynamicViT (Rao et al., 2021) learns a gating network to determine patch sizes and defines separate patch embedding networks for each size. However, it requires significant fine-tuning on pre-trained networks and does not speed up training. APT resolves this issue while demonstrating dramatically larger speedups at higher resolutions and on larger models.

3 Method

Our goal is to achieve a significant wall-clock speedup during *both* training and inference by using different-sized patches in different regions of the image. We first describe how we allocate different patch sizes within an image (Section 3.1) and then how we process different-sized regions into the same embedding space (Section 3.2). We then explain how we efficiently handle different input sizes and how we can adapt APT to work on dense visual prediction tasks like object detection.

3.1 Deciding Patch Sizes

Consider a vision transformer that takes an $H \times W \times C$ image as input. The standard ViT partitions the image into a set of $p \times p$ patches. A linear layer $\mathcal{E}$ is applied to each patch to convert it into a token, with size d_{embed}, yielding a sequence of $N = (HW/p^2)$ tokens.

In contrast, our goal is to decide patch size based on the image content, instead of using a constant number of patches. Concretely, we define a fixed number of patch scales S, where the set of patches

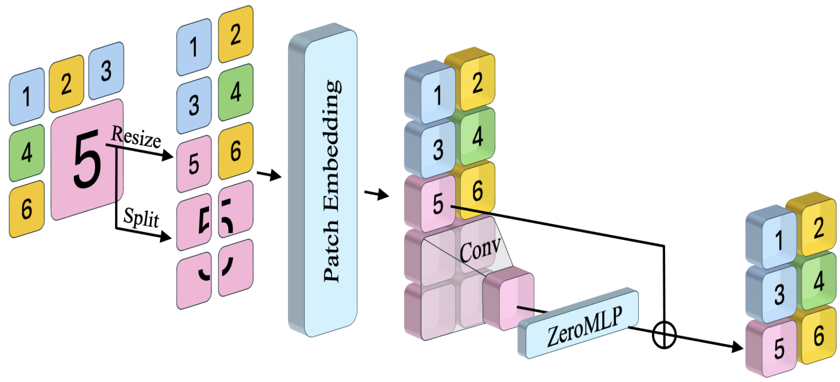


Figure 3: **Embedding Different Patch Sizes.** The smallest size patches are projected with the patch embedding. Larger patches are both split into their sub-patches and resized; the sub-patches are embedded, aggregated with a convolution layer. These are combined with the resized embedding with a zero-initialized MLP (Zhang et al., 2023).

consists of $\mathcal{P} = P_1 \cup P_2 \cup \dots P_S$, with each patch in P_i having size $2^i p \times 2^i p$. For example, if $S = 3, p = 16$, we are trying to find a smaller set of $16 \times 16, 32 \times 32$ and 64×64 patches while maximizing 'information' conveyed. For simplicity, we also impose the constraint that all patches follow a quadtree-like structure, following a regular grid.

We use *entropy* H as a measure of a patch's compressibility, given by:

$$H(P) = -\sum_{i=0}^{L-1} p_i \log_2 p_i, \tag{1}$$

where p_i is the probability of pixel intensity i. Since patches contain discrete pixel values, we approximate this by binning pixel intensities and computing entropy from the resulting distribution. Entropy quantifies the unpredictability and thus information content of a patch, making it a useful predictor of compressibility—lower entropy indicates higher redundancy. A large patch with low entropy should therefore be efficiently representable by a d_{embed}-dimensional vector. We discuss alternative measures further in the Appendix.

We obtain the patchification of the image hierarchically, as illustrated in Figure 3. We first divide the image into patches at the coarsest scale $2^S p \times 2^S p$ and compute their entropies. We then retain all such patches with entropy below a fixed threshold τ_i, which is a tunable hyperparameter for each level. We repeat this process until we reach the smallest possible patch size $p \times p$, to which we assign all remaining patches.

3.2 Patch Aggregation

After dividing the image into different patch sizes, we need to convert these patches into embeddings with dimension d_{embed}; in standard ViTs, this is done with a single linear layer $\mathcal{E}$. Prior work on vision transformers with varying patch sizes either resize every patch to $p \times p$ (Ronen et al., 2023), resize $\mathcal{E}$ for each size (Beyer et al., 2023), or train S separate patch embedding layers $\mathcal{E}_i$ for each possible patch size (Havtorn et al., 2023), adding overhead. Resizing allows reasonable performance with no training but can be improved upon—it uses strictly less information than if we applied $\mathcal{E}$ to the higher-resolution sub-patches.

We combine these strategies, as shown in Figure 3. We resize patches to a uniform size $p \times p$ but retain copies of larger original patches. For a given patch P_i of size $2^i p \times 2^i p$, we define its constituent $p \times p$ sub-patches as the set $\{P_j\}$. Each sub-patch P_j is embedded using the standard embedding layer $\mathcal{E}$. The final embedding for patch P_i is then computed as:

$$\mathcal{E}(P_i) = \text{ZeroMLP}\left(\text{Conv2d}^{(i)}(\{\mathcal{E}(P_j) \mid P_j \subset P_i\})\right) + \mathcal{E}(\text{Resize}_p(P_i)), \tag{2}$$

where $\text{Conv2d}^{(i)}$ indicates applying a convolutional downsampling layer i times, aggregating embeddings from sub-patches back to size $p \times p$. The ZeroMLP, a single linear layer initialized with zero weights inspired by ControlNet (Zhang et al., 2023), allows the model to gradually incorporate high-resolution details without initially degrading performance, facilitating faster convergence during fine-tuning. In particular, this enables APT to be applied to any pre-trained ViT and matches the performance of the initial model with a single epoch of accelerated fine-tuning.

3.3 Dynamic Input Sizes

Since APT is content-aware, the number of tokens for each image can vary widely. However, in contrast to token pruning works (Rao et al., 2021; Liang et al., 2022b), we do not reduce the size of the input at each layer, but *before* running the model. While most vision works use a fixed resolution, our setting is closer to that of language modeling, and in vision to RLT (Choudhury et al., 2025) and NaViT (Dehghani et al., 2023), where the number of tokens varies, but is predictably dictated by the input data. We follow these methods and employ sequence packing. For a batch of input images with sequence lengths $\{N_1, N_2, \dots N_B\}$, we concatenate the tokens into a single sequence with length $\sum_{i=1}^{B} N_i$ and construct a block-diagonal mask that ensures tokens only attend to tokens from the same example. This is natively implemented in commonly available attention backends such as FlashAttention (Dao et al., 2022; Dao, 2024) or xFormers (Lefaudeux et al., 2022), and adds no overhead to the network itself as the mask does not change. After running the network, we split the resulting sequence into its constituent subsequences and either extract the class token or compute a pooled representation for each subsequence.

Adaptation to Downstream Tasks. Standard methods for dense visual tasks like object detection or semantic segmentation often rely on a feature map that has the same aspect ratio as the image. This is required for methods that rely on transposed convolutions to upsample an input feature map for per-pixel predictions (Li et al., 2022a). In contrast, APT produces a different number of tokens per image, which cannot be simply reshaped into a rectangular feature map. To handle this, we rely on the assumption that the tokens representing larger patches encode simpler features and simply repeat them 2^{2i} times, as in (Havtorn et al., 2023; Bolya & Hoffman, 2023). This yields a fully differentiable feature map that can be upsampled by transposed convolutions and seamlessly applied to downstream tasks. Furthermore, tasks requiring high-resolution dense predictions such as object detection often rely on *window attention* (Liu et al., 2021a; Yuan et al., 2021; Fang et al., 2024), where the image is subdivided into multiple window regions to localize attention and increase efficiency. APT can still be applied even with window-attention; we simply divide the image into windows that are multiples of the largest patch-size, and use variable numbers of tokens within each window; this can again be straightforwardly implemented using sequence packing and attention masks with light overhead.

4 Experiments

4.1 Baselines

We categorize token merging approaches into two groups: *input-level* and *layer-level*. Input-level merging reduces tokens directly from image patches before entering the model, which is the category our method belongs to. In contrast, layer-level merging performs reduction within the network during feature propagation. We adopt input-level merging as our main baseline for fairest comparison, but compare to layer-level methods as well.

Input-level Merging Baselines. We use three main baselines: *random-masking*, *resizing-only* (He et al., 2021; Li et al., 2023) and the original optimized *Vanilla* implementation from `timm`. *Resizing* refers to disabling the zero initialized layer; this represents a stronger version of Quadformer (Ronen et al., 2023), which used a constant, nonadaptive number of patches per image. *Random* is a stronger version of FLIP (Li et al., 2023); we compute the token reduction obtained from APT and set it as the random patch dropping rate.

Layer-level Merging Baselines. We benchmark against three baselines: EViT (Liang et al., 2022a), ToMe (Bolya et al., 2022), and DTEM (Lee & Hong, 2024). All three baselines perform token merging across the ViT layers, removing a constant number of tokens regardless of image content.

Model	Res/Patch	Acc ↑	Img/s ↑	GFLOPS ↓	WC Time ↓	Speedup ↑
ViT-B^{MAE}	384/16	84.2	1151	49.4	11.6h	-
Random	384/16	83.4	1401	21.5	8.8h	+32%
Resizing	384/16	83.9	1390	21.5	9.0h	+29%
APT (Ours)	384/16	84.2	1390	21.9	9.0h	+29%
ViT-L^{MAE}	336/14	86.1	395	174.7	15.9h	-
Random	336/14	85.5	550	76.2	9.6h	+66%
Resizing	336/14	85.9	527	76.2	9.9h	+61%
APT (Ours)	336/14	86.1	527	76.8	9.9h	+61%
ViT-L^{MAE}	448/14	86.4	190	645	31.4h	-
Random	448/14	85.8	314	267	16.2h	+94%
Resizing	448/14	86.0	302	267	16.9h	+86%
APT (Ours)	448/14	86.3	302	268	16.9h	+86%

Table 1: **Full Fine-Tuning on ImageNet.** APT significantly reduces the wall-clock time to fine-tune a pre-trained backbone on ImageNet with no degradation in accuracy. We use the MAE (He et al., 2021) training recipe for all cases. Note that ViT-B is trained for 2× more epochs than ViT-L.

Model	Res/Patch	Acc ↑	GFLOPS ↓	Img/s ↑	Speedup ↑	Res/Patch	Acc ↑	GFLOPS ↓	Img/s ↑	Speedup ↑
ViT-B	224/16	85.1	16.9	3310	-	384/16	86.1	49.4	1151	-
Random	224/16	83.7	12.5	3751	+13%	384/16	85.0	21.5	1401	+22%
Resizing	224/16	84.6	12.5	3540	+7%	384/16	85.7	21.5	1390	+21%
APT-B (Ours)	224/16	85.1	12.7	3540	+7%	384/16	86.1	21.9	1390	+21%
ViT-L	224/14	87.9	59.7	883	-	336/14	88.2	174.7	395	-
Random	224/14	86.9	44.3	1049	+19%	336/14	87.3	76.2	550	+39%
Resizing	224/14	87.4	44.3	993	+12%	336/14	87.9	76.2	527	+33%
APT-L (Ours)	224/14	87.8	44.5	993	+12%	336/14	88.1	76.8	527	+33%
ViT-H	224/14	88.3	162.0	441	-	336/14	88.5	363.7	175	-
Random	224/14	87.4	92.1	568	+29%	336/14	87.0	158.3	272	+55%
Resizing	224/14	88.0	92.1	542	+23%	336/14	88.0	158.3	263	+50%
APT-H (Ours)	224/14	88.3	92.3	542	+23%	336/14	88.4	158.9	263	+50%

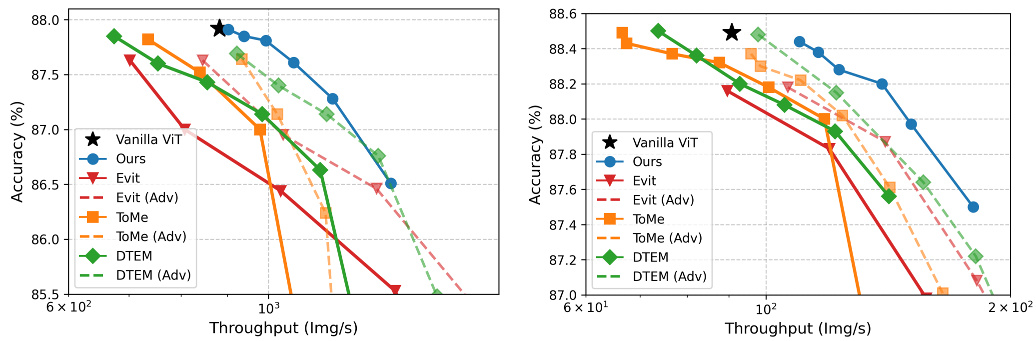
Table 2: **1-epoch Fine-Tuning on ImageNet.** APT consistently achieves large speedups while matching or sometimes exceeding the original network's performance after fine-tuning for 1 more epoch. Compared to only random masking or only resizing, APT offers the best tradeoff between speed and accuracy.

However, they share key limitations: none of them is implemented with FlashAttention, which makes them even slower than the Vanilla ViT equipped with FlashAttention. To provide a fairer and stronger comparison, we re-implement advanced versions of these baselines with FlashAttention. When weighted attention is involved, we disable it to enable FlashAttention.

4.2 Image Classification

Full Fine-Tuning. We provide the results of fine-tuning a vision transformer with APT on two different model scales and resolutions in Table 1. In all experiments, we use the official MAE (He et al., 2021) pre-trained checkpoint, interpolated to match the target resolution. At lower resolution, APT provides a ∼10% speedup over the baseline with a ∼14% token reduction. On the other hand, the speedup dramatically increases for higher resolutions—on 336×336, the speedup doubles. APT also reduces training time more at larger model scales, likely due to the fact that the attention computation dominates much more of the training time per iteration. Across all cases, APT matches the baseline while using the exact same training recipe—it can be considered an absolute improvement over standard patchification.

Short Fine-Tuning. We next present results from training with APT for 1 epoch from a fully fine-tuned ImageNet (Deng et al., 2009) model. Compared to other methods like DynamicVIT (Rao et al., 2021) or MS-VIT (Havtorn et al., 2023) which require 50 or more epochs of fine-tuning to learn a scoring function, thanks to our use of a zero-initialized layer, APT models make high-quality predictions from initialization. With no training, APT only resizes the larger patches to the base size, which is a stronger version of Quadformer (Ronen et al., 2023). However, we observe that just



(a) Trade-off comparison on ViT-L

(b) Trade-off comparison on ViT-H

Figure 4: **Accuracy vs. Throughput under different compute budgets.** Comparison between APT and layer-level merging methods on ViT-L and ViT-H. For a fairer evaluation, we also include their re-implemented Advanced (Adv) versions with FlashAttention, shown with a dashed line. APT consistently outperforms the baselines in both throughput and accuracy across all compute budgets.

one epoch is sufficient to "heal" the degradation from the new patchification scheme and match the original performance of the model, as shown in Table 2.

We provide a comparison with representative baselines in Figure 4. We compare throughput and ImageNet accuracy to our short fine-tuning results on ViT-L/14 with a resolution of 224 and ViT-H/14 with a resolution of 336. APT consistently outperforms all baselines, including their original versions as well as our improved reproductions using FlashAttention. The results confirm that input-level merging is inherently more efficient and reliable than layer-level merging.

4.3 VQA and Dense Visual Tasks

Vision transformers are used for a wide range of tasks beyond image classification; we evaluate how APT affects downstream performance in vision-language understanding tasks as well as dense prediction. Building on our short fine-tuning experiment for image classification, we start with a fully fine-tuned model, freeze its parameters, and train only the newly added components for 5% of the total iterations used in each model's fine-tuning scheme.

Visual QA. We first apply APT to the vision backbone of LLaVA (Liu et al., 2023; 2024a). LLaVA is a vision language model (VLM) that combines a vision transformer backbone with a language backbone via a projection layer. In the original paper (Liu et al., 2023), the vision encoder was completely frozen, and only the projection layer was updated. APT matches the original model performance while reducing image tokens and increasing throughput by 23%. Note that APT provides no speedup to the language component, but by reducing the number of visual tokens, it accelerates both the vision backbone and cross attention layers. We find that APT exceeds the original performance of the LLaVA model on a range of vision-language benchmarks (Goyal et al., 2017; Hudson & Manning, 2019; Lu et al., 2022; Singh et al., 2019; Schwenk et al., 2022; Fu et al., 2024; Liu et al., 2024b; Yu et al., 2023).

Object Detection. One might expect APT to degrade performance for tasks that require pixel-level understanding, such as object detection. To investigate this, we trained an object detector using the EVA-02 (Fang et al., 2024) backbone with window attention, with a ViTDet (Li et al., 2022b) style detection head. We conduct experiments on the COCO (Lin et al., 2014) dataset at 1536 $\times$ 1536 resolution. APT is able to reduce an impressive 30% of input tokens, drastically speeding up training and inference, while matching the final performance on mAP and AP50. Furthermore, these results demonstrate that APT remains effective under window attention beyond naive full attention, broadening the scope of its application.

Semantic Segmentation. We conduct another experiment on semantic segmentation, which requires fine-grained understanding of object boundaries. We again use the protocol of EVA-02 (Fang et al., 2024), using it as a backbone with a UperNet (Xiao et al., 2018) segmentation model on top.

Model	Img/s	Speedup	VQA^{v_2}	GQA	SQA^{I}	VQA^{T}	POPE	MME	MMB	MMB^{C}	MMV
LLaVA-1.5-7B	3.70	-	78.5	61.0	67.8	58.2	86.9	1510.1	64.6	58.1	30.7
Random	4.58	+24%	76.9	60.9	67.2	54.1	86.1	1460.5	62.7	57.6	30.5
Resizing	4.51	+22%	77.5	61.1	66.8	56.5	86.6	1473.8	63.2	58.1	30.2
APT (Ours)	4.51	+22%	77.9	61.4	67.5	56.9	86.4	1474.0	63.8	58.2	30.8
LLaVA-1.5-13B	2.22	-	80.0	63.2	72.7	61.2	87.1	1530.6	68.5	63.4	35.4
Random	2.79	+26%	78.0	60.7	72.0	55.7	86.5	1484.0	64.7	60.8	32.0
Resizing	2.72	+23%	78.9	61.1	72.0	59.1	86.8	1496.9	65.8	62.5	33.9
APT (Ours)	2.72	+23%	79.4	63.0	72.4	59.5	87.2	1511.2	66.5	63.7	34.7

Table 3: **Transfer to VQA**. APT enables significant throughput increase while matching or exceeding performance to the baseline.

Model	Res	Img/s	Speedup	mAP	AP50
EVA-02-B	1536	3.86	-	58.93	77.85
Resizing	1536	4.41	+14%	58.43	77.22
APT (Ours)	1536	4.41	+14%	58.79	77.65
EVA-02-L	1536	1.62	-	62.28	80.80
Resizing	1536	2.17	+30%	61.75	80.27
APT (Ours)	1536	2.17	+30%	62.07	80.64

Table 4: **Transfer to Object Detection.** APT can be scaled to high-resolution dense image tasks supporting window attention.

Model	Res	Img/s	Speedup	aAcc	mIoU
EVA-02-L	512	4.40	-	86.67	59.77
Resizing	512	4.87	+11%	86.09	58.81
APT (Ours)	512	4.87	+11%	86.68	59.70
EVA-02-L	640	2.55	-	86.83	60.05
Resizing	640	2.83	+11%	86.06	58.83
APT (Ours)	640	2.83	+11%	86.82	60.01

Table 5: **Transfer to Semantic Segmentation.** APT can handle pixel-level fine-grained tasks without compromising visual acuity.

When tested on ADE20K (Zhou et al., 2019; 2017), APT attains baseline performance while reducing 28∼32% of the input tokens depending on image resolution, thereby substantially accelerating inference. APT's success at semantic segmentation is particularly encouraging, since it implies that it reduces compute while not sacrificing visual acuity at the pixel level.

4.4 Ablations

We ablate components of APT to evaluate their effect on speed and accuracy.

Measuring APT overhead. Next, we measure the computational overhead introduced by APT. Re-arranging the input patches and using masks does not have zero computational cost, and given that GPUs are highly optimized for constant input shapes, understanding the cost of adding APT is important. The results of this analysis are in Table 6.

Zero-initialization. Finally, we ablate the use of our *zero-initialized* connection for incorporating higher-resolution details in larger patches. In Table 7, we compare with a simple residual connection, a non-zero initialized connection, and resizing. We find that initializing to zero offers the best off-the-shelf accuracy as well as the strongest performance after an epoch of training. This matches the finding of ControlNet (Zhang et al., 2023), which showed that zero-initialization works well for adding new capabilities without adding harmful noise to the original model.

Thresholds. The main tunable parameter in APT is the entropy threshold, which can differ per scale and controls how compressible a region must be in order to be retained. The speed–accuracy trade-off resulting from threshold adjustment is shown in Figure 4, and additional analysis and visualizations are provided in Appendix.

4.5 Example Visualizations

We provide qualitative visualizations of the patchification produced by APT with 3 patch scales in Figure 5. As desired, APT consistently assigned larger patches to more homogeneous regions of the image. Dark backgrounds, blue sky, and blurry backdrops are all covered by the largest (64×64) patches, while smaller regions that are still simple are given the second largest (32×32). Regions with more detail or high frequency receive smaller patches: people's faces or objects in focus are allocated the most. Each image has a different number of patches, depending on its inherent complexity—the cityscape in the bottom right has significantly more than the simpler cartoon image

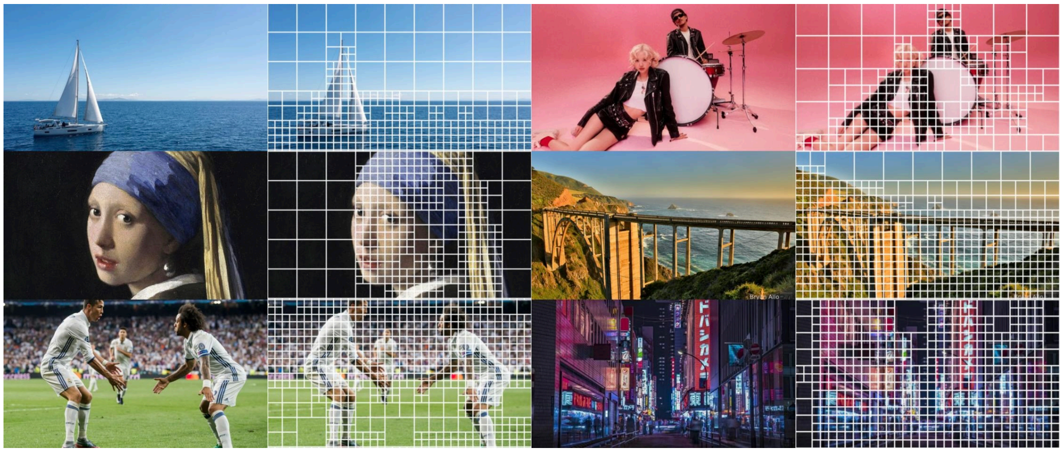


Figure 5: **Visualized Examples.** APT consistently places large patches on more homogenous regions and smaller patches on more complex ones. We use conservative thresholds to limit information loss. Images are best viewed zoomed in. More visualizations are in Appendix.

Res/Patch	Base (Img/s)	$\text{APT}_{\tau=-1}$
ViT-B 224/16	3310	3090
ViT-B 384/16	1151	1030
ViT-L 224/16	883	811
ViT-L 336/14	395	360
ViT-H 224/14	441	418
ViT-H 336/14	190	180

Table 6: **APT overhead with no reduction.** With no token reduction, APT incurs nontrivial overhead. However, token reduction gives 20%+ speedups relative to the standard implementation, more than covering the discrepancy.

Model	w/o training	w/ training
Base	88.15	88.15
Residual	87.40	87.52
NonZero	87.50	87.81
Zero (Ours)	87.98	88.13

Table 7: **Ablating Zero-initialization.** Using a zero-initialized connection works the best for training APT networks to properly incorporate higher resolution details from the original image, while preserving good-quality predictions before training.

in the top left. By design, the patches produced by APT are agnostic to downstream goals and unaware of what a user might desire from an image. If the top right image were input to a VLM along with the question "What color is the background?", APT would still assign extremely coarse patches to the pink wall due to its textural simplicity. Additional visualizations are provided in Appendix.

5 Conclusion

We presented Adaptive Patch Transformer (APT), a method to accelerate ViTs that uses larger patches in simpler areas and smaller patches in more complex ones. It significantly improves training and inference speeds, especially for larger models and higher resolutions. APT can be applied to any pretrained ViT backbone and converges in 1 epoch or less, enabling users to quickly train their models to be faster on a wide range of vision tasks. Our results suggest that APT will benefit the broader vision community by reducing the compute budget required to train state-of-the-art models.

Limitations. Although APT provides significant speedups, it still relies on a hand-crafted heuristic to determine patch sizes, which may not always align with downstream users' preferences and could likely be improved. Furthermore, while APT works for image understanding tasks, it does not currently support image generation. It also requires extremely high-resolution images and large models, making it an ideal application for our work. Future work will be required to overcome these limitations, and we hope that APT can inspire further research on efficient ViTs.

REFERENCES

Xiaoqi An, Lin Zhao, Chen Gong, Nannan Wang, Di Wang, and Jian Yang. Sharpose: Sparse high-resolution representation for human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 691–699, 2024.

Zhuhua Bai, Weiqing Li, Guolin Yang, Fantong Meng, Renke Kang, and Zhigang Dong. A coarse-to-fine framework for point voxel transformer. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 205–211. IEEE, 2024.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14496–14506, 2023.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4599–4603, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. *arXiv preprint arXiv:2305.17530*, 2023.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. 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.

Mengzhao Chen, Mingbao Lin, Ke Li, Yunhang Shen, Yongjian Wu, Fei Chao, and Rongrong Ji. Cf-vit: A general coarse-to-fine method for vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 7042–7052, 2023a.

Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2061–2070, 2023b.

Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems*, 37:28127–28149, 2025.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. *arXiv preprint arXiv:2110.12894*, 2021.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Shuangrui Ding, Peisen Zhao, Xiaopeng Zhang, Rui Qian, Hongkai Xiong, and Qi Tian. Prune spatio-temporal tokens by semantic-aware temporal accumulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16945–16956, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. *arXiv preprint arXiv:2411.02393*, 2024.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. URL https://arxiv.org/abs/2306.13394.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Jakob Drachmann Havtorn, Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Msvit: Dynamic mixed-scale tokenization for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 838–848, 2023.

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

Youbing Hu, Yun Cheng, Anqi Lu, Zhiqiang Cao, Dawei Wei, Jie Liu, and Zhijun Li. Lf-vit: Reducing spatial redundancy in vision transformer for efficient image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2274–2284, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Julie Kallini, Shikhar Murty, Christopher D Manning, Christopher Potts, and Róbert Csordás. Mrt5: Dynamic token merging for efficient byte-level language models. *arXiv preprint arXiv:2410.20771*, 2024.

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

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

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

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Dong Hoon Lee and Seunghoon Hong. Learning to merge tokens via decoupled embedding for efficient vision transformers. *Advances in Neural Information Processing Systems*, 37:54079–54104, 2024.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pp. 280–296. Springer, 2022a.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pp. 280–296. Springer, 2022b.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *CVPR*, 2023.

Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in neural information processing systems*, 35:34451–34463, 2022c.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022a.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740–755. Springer, 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021a.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021b.

Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

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

Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16132–16141, 2024.

Jeongsoo Park and Justin Johnson. Rgb no more: Minimally-decoded jpeg vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22334–22346, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

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

Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4613–4622, 2023.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Chau Tran, Duy MH Nguyen, Manh-Duy Nguyen, TrungTin Nguyen, Ngan Le, Pengtao Xie, Daniel Sonntag, James Y Zou, Binh Nguyen, and Mathias Niepert. Accelerating transformers with spectrum-preserving token merging. *Advances in Neural Information Processing Systems*, 37: 30772–30810, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in neural information processing systems*, 34:11960–11973, 2021.

Yunke Wang, Bo Du, Wenyuan Wang, and Chang Xu. Multi-tailed vision transformer for efficient inference. *Neural Networks*, 174:106235, 2024.

Cong Wei, Brendan Duke, Ruowei Jiang, Parham Aarabi, Graham W Taylor, and Florian Shkurti. Sparsifiner: Learning sparse instance-dependent attention for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22680–22689, 2023.

Ross Wightman. Pytorch image models, 2019. URL https://github.com/rwightman/pytorch-image-models.

Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6026–6035, 2018.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022.

Wilson Yan, Matei Zaharia, Volodymyr Mnih, Pieter Abbeel, Aleksandra Faust, and Hao Liu. Elastictok: Adaptive tokenization for image and video. *arXiv preprint arXiv:2410.08368*, 2024.

Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18547–18557, 2023.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10809–10818, 2022.

Lu Yu and Wei Xiang. X-pruner: explainable pruning for vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24355–24363, 2023.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Chuanyang Zheng, Kai Zhang, Zhi Yang, Wenming Tan, Jun Xiao, Ye Ren, Shiliang Pu, et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

Qiqi Zhou and Yichen Zhu. Make a long image short: Adaptive token length for vision transformers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 69–85. Springer, 2023.