SlowFormer: Adversarial Attack on Compute and Energy Consumption of Efficient Vision Transformers

K L Navaneet [*1] Soroush Abbasi Koohpayegani [*1] Essam Sleiman [*12†] Hamed Pirsiavash[1]
[1] University of California, Davis [2] Harvard University

Abstract

Recently, there has been a lot of progress in reducing the computation of deep models at inference time. These methods can reduce both the computational needs and power usage of deep models. Some of these approaches adaptively scale the compute based on the input instance. We show that such models can be vulnerable to a universal adversarial patch attack, where the attacker optimizes for a patch that when pasted on any image, can increase the compute and power consumption of the model. We run experiments with three different efficient vision transformer methods showing that in some cases, the attacker can increase the computation to the maximum possible level by simply pasting a patch that occupies only 8% of the image area. We also show that a standard adversarial training defense method can reduce some of the attack's success. We believe adaptive efficient methods will be necessary in the future to lower the power usage of expensive deep models, so we hope our paper encourages the community to study the robustness of these methods and develop better defense methods for the proposed attack. Code is available at:
https://github.com/UCDvision/SlowFormer

1. Introduction

The field of deep learning has recently made significant progress in improving model efficiency for inference. Two broad categories of methods can be distinguished: 1) those that reduce computation regardless of input, and 2) those that reduce the computation depending on the input (adaptively). Most methods, such as weight pruning or model quantization, belong to the first category which reduces computation by a constant factor regardless of the input. However, in many applications, the complexity of the perception task may differ depending on the input. For example, when a self-driving car is driving between lanes in an empty street, the perception may be simpler and require less computation when compared to driving in a busy city street scene. Interestingly, in some applications, simple scenes such as highway driving may account for the majority of the time. Therefore, we believe that adaptive computation reduction will become an increasingly important research area in the future, especially when non-adaptive methods reach the lower bound of computation.

We argue that reduction of compute usually reduces power usage, which is crucial, particularly in mobile devices that run on battery, e.g., AR/VR headsets, humanoid robots, and drones. For instance, increasing the size of the battery for a drone may lead to a drastic reduction in its range due to the increased battery weight. This is important since the improvement in battery technology is much slower than compute technology. For instance, the battery capacity from iPhone [1] (1st generation) in 2007 to iPhone 15 Pro Max in 2023 improved from 5.18 watt-hour to 17.32 watt-hour (less than 4 times) while the compute has increased by a much larger factor. As an example, a delivery robot like Starship uses a 1,200Wh battery and can run for 12 hours [2], so it uses almost 100 watts for compute and mobility. Hence, an adversary increasing the power consumption of the perception unit by 20 watts, will reduce the battery life by almost 20%, which can be significant. Note that 20 watts increase in power is realistic assuming that it uses two NVIDIA Jetson Xavier NX cards (almost 20 watts each).

Key idea: Assuming that a perception method is reducing the computation adaptively with the input, an adversary can attack the model by modifying the input to increase the computation and power consumption. **Our goal** is to design a universal adversarial patch that when pasted on any input image, it will increase the computation of the model leading to increased power consumption. We believe this is an important vulnerability, particularly for safety-critical mobile systems that run on battery.

Please note that in this paper, we do not experiment with real hardware to measure the power consumption. Instead, we report the change in FLOPs of the inference time assuming that the power consumption is positively correlated with the number of FLOPs, as studied in [73].

* Equal contribution
† Work done while he was an undergraduate student at UC Davis.

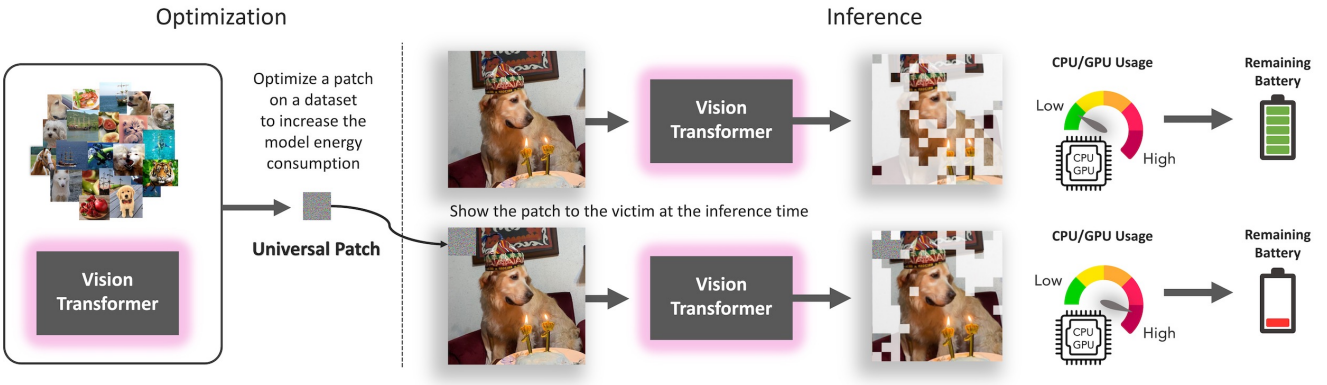


Figure 1. **Computation and Energy Attack on Vision Transformers:** Given a pre-trained input-dependent computation efficient model, the adversary first attaches an adversarial patch to all images in a dataset and optimizes this patch with our method such that it maximizes the model's computation for each sample. During inference, the adversary modifies the input of the victim's model by applying the learnt patch to it. This results in an increase in compute in the victim's model. The attack will thus potentially slowdown and also lead to increased energy consumption and CPU/GPU usage on the victim's device.

We design our attack, SlowFormer, for three different methods (A-ViT [88], ATS [25], and Ada-VIT [63]) that reduce the computation of vision transformers. These methods generally identify the importance of each token for the final task and drop the insignificant ones to reduce the computation. We show that in all three cases, our attack can increase the computation by a large margin, returning it to the full-compute level (non-efficient baseline) for all images in some settings. Our threat model is agnostic to the accuracy of the model and attacks the computation and power consumption only. Figure 1 shows our attack.

There are some prior works that design a pixel-level perturbation attack to increase the compute of the model. We believe universal patch-based attacks that do not change with the input image (generalize from training data to test data) are much more practical in real applications. Note that to modify the pixel values on a real robot, the attacker needs to access and manipulate the image between the camera and compute modules, which is impossible in many applications.

Contributions: We show that efficient vision transformer methods are vulnerable to a universal patch attack that can increase their compute and power usage. We demonstrate this through experiments on three different efficient transformer methods. We show that an adversarial training defense can reduce attack success to some extent.

2. Related Work

Vision Transformers: The popularity of transformers [79] in vision has grown rapidly since the introduction of the first vision transformer [20, 78]. Recent works demonstrate the strength of vision transformers on a variety of computer vision tasks [11, 18, 21, 56, 69, 77, 90, 93, 94, 96]. Moreover, transformers are the backbone of recent Self-Supervised Learning (SSL) models [12, 37], and vision-language models [67]. In our work, we design an attack to target the computation and energy efficiency of vision transformers.

Efficient Vision Transformers: Due to the recent importance and popularity of vision transformers, many works have started to study the efficiency of vision transformers [7, 46, 89]. To accomplish this, some lines of work study token pruning with the goal of removing uninformative tokens in each layer [25, 62, 63, 68, 88]. ToMe [6] merges similar tokens in each layer to decrease the computation. Some works address quadratic computation of self-attention module by introducing linear attention [3, 45, 48, 58, 74]. Efficient architectures [40, 55] that limit the attention span of each token have been proposed to improve efficiency. In our paper, we attack token pruning based efficient transformers where the computation varies based on the input samples [25, 63, 88].

Dynamic Computation: There are different approaches to reducing the computation of vision models, including knowledge distillation to lighter network [39, 59], model quantization [54, 70] and model pruning [51]. In these methods, the computation is fixed during inference. In contrast to the above models, some works address efficiency by having variable computation based on the input. The intuition behind this direction is that not all samples require the same amount of computation. Several recent works have developed models that dynamically exit early or skip layers [5, 22, 28, 33, 34, 44, 76, 80, 82] and selectively activate neurons, channels or branches for dynamic width [4, 9, 13, 26, 31, 38, 43, 91] depending on the complexity of the input sample. Zhou et al. show that not all locations in an image contribute equally to the predictions of a CNN model [95], encouraging a new line of work to make CNNs more

efficient through spatially dynamic computation. Pixel-Wise dynamic architectures [10, 14, 23, 47, 71, 81, 85] learn to focus on the significant pixels for the required task while Region-Level dynamic architectures perform adaptive inference on the regions or patches of the input [29, 53]. Finally, lowering the resolution of inputs decreases computation, but at the cost of performance. Conventional CNNs process all regions of an image equally, however, this can be inefficient if some regions are "easier" to process than others [42]. Correspondingly, [86, 87] develop methods to adaptively scale the resolution of images.

Transformers have recently become extremely popular for vision tasks, resulting in the release of a few input-dynamic transformer architectures [24, 63, 88]. Fayyaz et al. [24] introduce a differentiable parameter-free Adaptive Token Sampler (ATS) module which scores and adaptively samples significant tokens. ATS can be plugged into any existing vision transformer architecture. A-ViT [88] reduces the number of tokens in vision transformers by discarding redundant spatial tokens. Meng et al. [63] propose AdaViT, which trains a decision network to dynamically choose which patch, head, and block to keep/activate throughout the backbone.

Adversarial Attack: Adversarial attacks are designed to fool models by applying a targeted perturbation or patch on an image sample during inference [32, 50, 75]. These methods can be incorporated into the training set and optimized to fool the model. Correspondingly, defenses have been proposed to mitigate the effects of these attacks [27, 52, 65, 84]. Patch-Fool [30] considers adversarial patch-based attacks on transformers. Some recent works [61, 83, 92] also study and design methods for the transferability of adversarial attacks on vision transformers. However, most prior adversarial attacks target model accuracy, ignoring model efficiency.

Energy Attack: Very recently, there have been a few works on energy adversarial attacks on neural networks. In ILFO [35], Haque et al. attack two CNN-based input-dynamic methods: SkipNet [82] and SACT [28] using image specific perturbation. DeepSloth [41] attack focuses on slowing down early-exit methods, reducing their energy efficiency by 90-100%. GradAuto [64] successfully attacks methods that are both dynamic width and dynamic depth. NICGSlowDown and TransSlowDown [16, 17] attack neural image caption generation and neural machine translation methods, respectively. All these methods primarily employ image specific perturbation based adversarial attack. SlothBomb injects efficiency backdoors to input-adaptive dynamic neural networks [15] and NodeAttack [36] attacks Neural Ordinary Differential Equation models, which use ordinary differential equation solving to dynamically predict the output of a neural network. Our work is closely related to ILFO [35], DeepSloth [41] and GradAuto [64] in that we attack the computational efficiency of networks. However, unlike these methods, we focus on designing an adversarial patch-based attack that is universal and on vision transformers. We additionally provide a potential defense for our attack. We use a patch that generalizes from train to test set and thus we do not optimize per sample during inference. Our patch-based attack is especially suited for transformer architectures [30].

3. Computation and Energy Attack

3.1. Threat Model:

We consider a scenario where the adversary has access to the victim's trained deep model and modifies its input such that the energy consumption and computational demand of the model is increased. The attack is agnostic to model accuracy. To make the setting more practical, instead of perturbing the entire image, we assume that the adversary can modify the input image by only pasting a patch [8, 72] on it and that the patch is universal, that is, image independent. During inference, a pretrained patch is pasted on the test image before propagating it through the network.

In this paper, we attack three state-of-the-art efficient transformers. Since the attacker manipulates only the input image and not the network parameters, the attacked model must have dynamic computation that depends on the input image. As stated earlier, several recent works have developed such adaptive efficient models and we believe that they will be more popular in the future due to the limits of non-adaptive efficiency improvement.

3.2. Attack on Efficient Vision Transformers:

Universal Adversarial Patch: We use an adversarial patch to attack the computational efficiency of transforms. The learned patch is universal, that is, a single patch is trained and is used during inference on all test images. The patch generalizes across images but not across models. The patch optimization is performed only on the train set. The patch is pasted on an image by replacing the image pixels using the patch. We assume the patch location does not change from train to test. The patch pixels are initialized using i.i.d. samples from a uniform distribution over $[0, 255]$. During each training iteration, the patch is pasted on the mini-batch samples and is updated to increase the computation of the attacked network. The patch values are projected onto $[0, 255]$ and quantized to 256 uniform levels after each iteration. Note that we use a pretrained network and do not update its parameters either in the training or in the evaluation of our attack. During inference, the trained patch is pasted on the test images and the computational efficiency of the network on the adversarial image is measured.

Here, we focus on three methods employing vision transformers for the task of image classification. All these methods modify the computational flow of the network based on the input image for faster inference. A pretrained model is used for the attack and is not modified during

our adversarial patch training. We first provide a brief background of each method before describing our attack.

Attacking A-ViT :

Background: A-ViT [88] adaptively prunes image tokens to achieve speed-up in inference with minimal loss in accuracy. For a given image, a dropped token will not be used again in the succeeding layers of the network. Let x be the input image and $\{t^l\}_{1:K}$ be the corresponding K tokens at layer l. An input-dependent halting score h_k^l for a token k at layer l is calculated and the token is dropped at layer N_k where its cumulative halting score exceeds a fixed threshold value $1-\epsilon$ for the first time. The token is propagated until the final layer if its score never exceeds the threshold. Instead of introducing a new parameter for h_k^l, the first dimension of each token is used to predict the halting score for the corresponding token. The network is trained to maximize the cumulative halting score at each layer and thus drop the tokens earlier. The loss, termed ponder loss, is given by:

$$\mathcal{L}_{\text{ponder}} = \frac{1}{K}\sum_{k=1}^{K}(N_k + r_k), \qquad r_k = 1 - \sum_{l=1}^{N_{k-1}} h_k^l \quad (1)$$

Additionally, A-ViT enforces a Gaussian prior on the expected halting scores of all tokens via KL-divergence based distribution loss, $\mathcal{L}_{\text{distr.}}$. These loss terms are minimized along with the task-specific loss $\mathcal{L}_{\text{task}}$. Thus, the overall training objective is $\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha_d \mathcal{L}_{\text{distr.}} + \alpha_p \mathcal{L}_{\text{ponder}}$ where α_d and α_p are hyperparameters.

Attack: Here, we train the patch to increase the inference compute of a trained A-ViT model. Since we are interested in the compute and not task-specific performance, we simply use $-(\alpha_d \mathcal{L}_{\text{distr.}} + \alpha_p \mathcal{L}_{\text{ponder}})$ as our loss. It is possible to preserve (or hurt) the task performance by additionally using $+\mathcal{L}_{\text{task}}$ (or $-\mathcal{L}_{\text{task}}$) in the loss formulation.

Attacking AdaViT:

Background: To improve the inference efficiency of vision transformers, AdaViT [63] inserts and trains a decision network before each transformer block to dynamically decide which patches, self-attention heads, and transformer blocks to keep/activate throughout the backbone. The l^{th} block's decision network consists of three linear layers with parameters $W_l = W_l^p, W_l^h, W_l^b$ which are then multiplied by each block's input Z_l to get m.

$$(m_l^p, m_l^h, m_l^b) = (W_l^p, W_l^h, W_l^b) Z_l \quad (2)$$

The value m is then passed to sigmoid function to convert it to a probability value used to make the binary decision of keep/discard. Gumbel-Softmax trick [60] is used to make this decision differentiable during training. Let M be the keep/discard mask after applying Gumbel-Softmax on m. The loss on computation is given by:

$$\mathcal{L}_{usage} = (\frac{1}{D_p}\sum_{d=1}^{D_p} M_d^p - \gamma_p)^2 + (\frac{1}{D_h}\sum_{d=1}^{D_h} M_d^h - \gamma_h)^2 + (\frac{1}{D_b}\sum_{d=1}^{D_b} M_d^b - \gamma_b)^2 \quad (3)$$

where D_p, D_h, D_b represent the number of total patches, heads, and blocks of the entire transformer, respectively. γ_p, γ_h, γ_b denote the target computation budgets i.e. the percentage of patches/heads/blocks to keep. The total loss is a combination of task loss (cross-entropy) and computation loss: $\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{usage}$.

Attack: To attack this model, we train the patch to maximize the computation loss $\mathcal{L}_{usage}$. More specifically, we set the computation-target γ values to 0 and negate the $\mathcal{L}_{usage}$ term in Eq. 3. As a result, the patch is optimized to maximize the probability of keeping the corresponding patch (p), attention head (h), and transformer block (b). We can also choose to attack the prediction performance by selectively including or excluding the $\mathcal{L}_{ce}$ term.

Attacking ATS:

Background: Given N tokens with the first one as the classification token, the transformer attention matrix $\mathcal{A}$ is calculated by the following dot product: $\mathcal{A} = \text{Softmax}\left(\mathcal{Q}\mathcal{K}^T/\sqrt{d}\right)$ where $\sqrt{d}$ is a scaling coefficient, d is the dimension of tokens, $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ are the query, key and value matrices, respectively. The value $\mathcal{A}_{1,j}$ denotes the attention of the classification token to token j. ATS [25] assigns importance score S_j for each token j by measuring how much the classification token attends to it:

$$\mathcal{S}_j = \frac{\mathcal{A}_{1,j} \times ||\mathcal{V}_j||}{\sum_{i=2} \mathcal{A}_{1,i} \times ||\mathcal{V}_i||} \quad (4)$$

The importance scores are converted to probabilities and are used to sample tokens, where tokens with a lower score have more of a chance of being dropped.

Attack: Since ATS uses inverse transform sampling, it results in fewer samples if the importance distribution is sharp. To maximize the computation in ATS, we aim to obtain a distribution of scores with high entropy to maximize the number of retained tokens. Therefore, we optimize the patch so that the attention of the classification token over other tokens is a uniform distribution using the following MSE loss:

$$\mathcal{L} = \sum_{i=2}^{N} ||\mathcal{A}_{1,i} - \frac{1}{N}||_2^2 \quad (5)$$

Note that one can optimize $\mathcal{S}$ to be uniform, but we found the above loss to be easier to optimize. For a multi-head

attention layer, we calculate the loss for each head and then sum the loss over all heads. Moreover, ATS can be applied to any layer of a vision transformer. For a given model, we apply our loss at all ATS layers and use a weighted summation for optimization.

4. Defense

An obvious defense, although weak, will be to use non-dynamic efficient methods only, e.g., weight pruning, where the reduction in compute is deterministic and does not depend on the input. However, most such methods do not achieve high levels of computation efficiency since they do not take advantage of the simplicity of images.

We adopt standard adversarial training as a better defense method for our attack. In the standard way, at each iteration of training the model, one would load an image, attack it, and then use it with correct labels in training the model. We cannot adopt this out-of-the-box since our attack generalizes across images and is not dependent on a single image only. To do this, we maintain a set of adversarial patches, and at each iteration sample one of them randomly (uniformly), and use it at the input while optimizing the original loss of the efficient model to train a robust model. To adapt the set of adversarial patches to the model being trained, we interrupt the training at every 20% mark of each epoch and optimize for a new patch to be added to the set of patches. To limit the computational cost of training, we use only 500 iterations to optimize for a new patch, which results in an attack with reasonable accuracy compared to our main results.

5. Experiments

5.1. Attack on Efficient Vision Transformers

Dataset: We evaluate the effectiveness of our attack on two datasets: ImageNet-1K [19] and CIFAR-10 [49]. ImageNet-1K contains 1.3M images in the train set and 50K images in the validation set with 1000 total categories. CIFAR-10 has 50K images for training and 10K images for validation with 10 total categories.

Metrics: We report Top-1 accuracy and average computation in terms of GFLOPs for both attacked and unattacked models. Similar to Attack Success Rate in a standard adversarial attack, we introduce a metric: Attack Success to quantify the efficacy of the attack. We define Attack Success as the number of FLOPs increased by the attack divided by the number of FLOPs decreased by the efficient method. Attack Success $= \frac{(\text{FLOPs}_{\text{attack}} - \text{FLOPs}_{\text{min}})}{(\text{FLOPs}_{\text{max}} - \text{FLOPs}_{\text{min}})}$ where $\text{FLOPs}_{\text{min}}$ is the compute of the efficient model and $\text{FLOPs}_{\text{max}}$ is that of the original inefficient model. Attack Success is thus capped at 100% while a negative value denotes a reduction in FLOPs. Note that our Attack Success metric illustrates the effectiveness of an attack in reversing the FLOPs reduction of a particular method.

Baselines: We propose three alternative approaches to SlowFormer (ours) to generate the patch.

Random Patch: A simple baseline is to generate a randomly initialized patch. We sample IID pixel values from a uniform distribution between 0 and 255 to create the patch.

NTAP: We consider a standard adversarial patch that is trained to attack the model task performance instead of compute. We use a non-targeted universal adversarial patch (NTAP) to attack the model. We train the patch to fool the model by misclassifying the image it is pasted on. We use the negative of the cross-entropy loss with the predicted and ground-truth labels as the loss to optimize the patch.

TAP: We train a universal targeted adversarial patch (TAP). The patch is optimized to classify all images in the train set to a single fixed category. Similar to NTAP, the adversarial attack here is on task performance and not computation. We experiment with ten randomly generated target category labels and report the averaged metrics.

Implementation Details: We use PyTorch [66] for all experiments. Unless specified, we use a patch of size 64×64, train and test on 224×224 images, and we paste the patch on the top-left corner. Note that our patch occupies just 8% of the total area of an input image. We use AdamW [57] optimizer to optimize the patches and use 4 NVIDIA RTX 3090 GPUs for each experiment. We use varying batch sizes and learning rates for each of the computation-efficient methods.

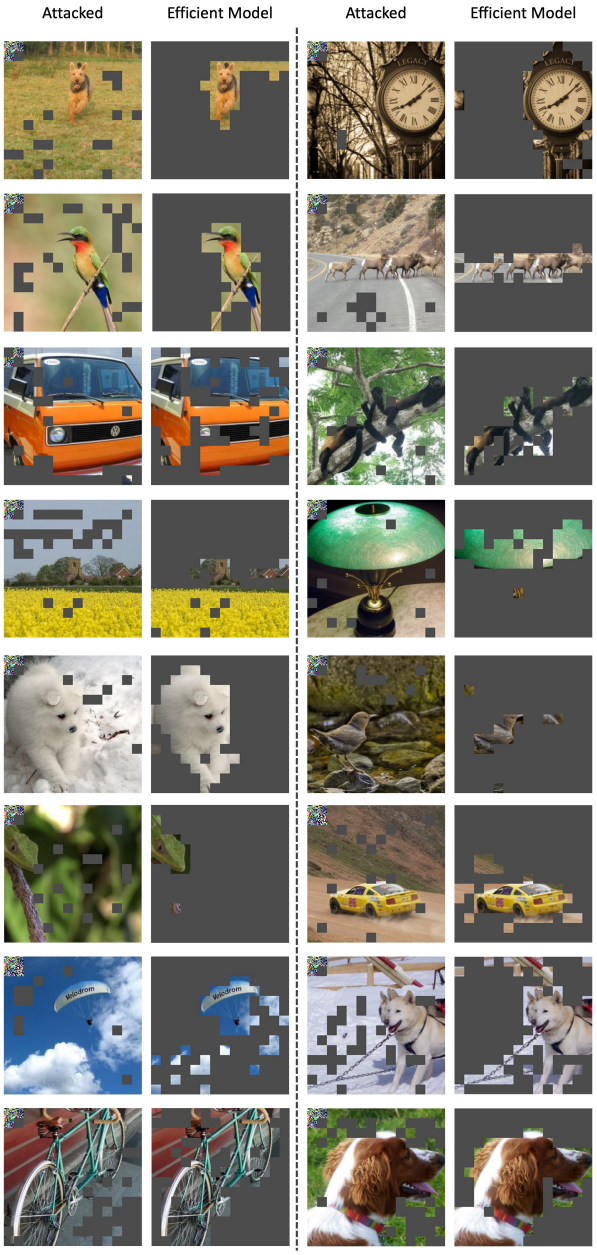
ATS Details: As in ATS [25], we replace layers 3 through 9 of ViT networks with the ATS block and set the maximum limit for the number of tokens sampled to 197 for each layer. We train the patch for 2 epochs with a learning rate of 0.4 for ViT-Tiny and $lr = 0.2$ for ViT-Base and ViT-Small. We use a batch size of 1024 and different loss coefficients for each layer of ATS. For DeiT-Tiny we use $[1.0, 0.2, 0.2, 0.2, 0.01, 0.01, 0.01]$, for DeiT-Small we use $[1.0, 0.2, 0.05, 0.01, 0.005, 0.005, 0.005]$, and for DeiT-Base we use $[2.0, 0.1, 0.02, 0.01, 0.005, 0.005, 0.005]$. The weights are vastly different at initial and final layers to account for the difference in loss magnitudes across layers.

A-ViT Details: When attacking A-VIT[88], the patches are optimized for one epoch with a learning rate of 0.2 and a batch size of 512 (128×4GPUs) using AdamW [57] optimizer. We optimize the patches for 4 epochs for patch length 32 and below. For CIFAR-10 experiments, the images are resized from 32×32 to 256×256 and a 224×224 crop is used as the input. For the training of adversarial defense, we generate 5 patches per epoch of adversarial training and limit the number of iterations for patch generation to 500. The learning rate for patch optimization is increased to 0.8 for faster convergence.

Table 1. **Computation and Energy Attack on Efficient Vision Transformers:** Comparison of the effect of our attack with baselines: No Attack, Random Patch, targeted (TAP), and non-targeted (NTAP) adversarial patches applied to three input-dynamic computation efficient pre-trained models of varying architectures. The maximum possible compute for a given architecture is provided in bold. On A-ViT , we completely undo the efficiency gains obtained by the efficient method through our attack, achieving Attack Success of 100%. We achieve high Attack Success on all approaches while the baselines expectedly do not contribute to increase in compute.

Method	Attack	Model GFLOPs	Top-1 Acc	Attack Success
	ViT-Tiny	**1.3**	-	-
	No attack	0.87	71.4%	-
	Random Patch	0.87	70.8%	-1%
A-ViT	TAP	0.85	0.1%	-5%
	NTAP	0.83	0.1%	-10%
	SlowFormer (ours)	1.3	4.7%	100%
	ViT-Small	**4.6**	-	-
	No attack	3.7	78.8%	-
	Random Patch	3.7	78.4%	-2%
A-ViT	TAP	3.6	0.1%	-12%
	NTAP	3.6	0.1%	-7%
	SlowFormer (ours)	4.6	2.3%	99%
	ViT-Tiny	**1.3**	-	-
	No attack	0.84	70.3%	-
	Random Patch	0.83	69.8%	-2%
ATS	TAP	0.76	0.1%	-17%
	NTAP	0.61	0.1%	-50%
	SlowFormer (ours)	1.0	1.2%	35%
	ViT-Small	**4.6**	-	-
	No attack	3.1	79.2%	-
	Random Patch	3.1	78.6%	-1%
ATS	TAP	3.0	0.1%	-7%
	NTAP	2.4	0.1%	-47%
	SlowFormer (ours)	4.0	1.0%	60%
	ViT-Base	**17.6**	-	-
	No attack	12.6	81.3%	-
	Random Patch	12.5	81.2%	-2%
ATS	TAP	12.0	0.1%	-12%
	NTAP	11.0	0.1%	-32%
	SlowFormer (ours)	15.4	0.2%	52%
	ViT-Small	**4.6**	-	-
	No attack	2.25	77.3%	-
	Random Patch	2.20	76.9%	-2%
AdaViT	TAP	2.28	0.1%	1%
	NTAP	2.15	0.1%	-4%
	SlowFormer (ours)	3.2	0.4%	40%

Figure 2. **Visualization of our Energy Attack on Vision Transformers:** We visualize the A-ViT-Small with and without our attack. We use patch size of 32 for the attack (on the top-left corner). We show pruned tokens at layer 8 of A-ViT-Small. Our attack can recover most of the pruned tokens, resulting in increased computation and power consumption. Note that although the patch is reasonably small and is in the corner of the view, it can affect the whole computational flow of the network. This is probably due to the global attention mechanism in transformers.



AdaViT Details: For AdaViT[63], we first freeze the weights and use a learning rate of 0.2 and a batch size of 128 with 4 GPUs for patch optimization. We use AdamW [57] optimizer with no decay and train for 2 epochs with a patch size of 64 x 64. We train on the ImageNet-1k train dataset and evaluate it on the test set.

Table 2. **Results on CIFAR10 dataset.** We report results on CIFAR10 dataset to show that our attack is not specific to ImageNet alone. CIFAR-10 is a small dataset compared to ImageNet and thus results in an extremely efficient A-ViT model. Our attack increases the FLOPs from 0.11 to 0.58 which restores nearly 41% of the original reduction in the FLOPs.

Method	Model FLOPs	Top-1 Acc	Attack Success
ViT-Tiny	1.26	95.9%	-
A-ViT-Tiny	0.11	95.8%	-
SlowFormer (ours)	0.58	60.2%	41%
ATS-Tiny	0.85	94.7%	-
SlowFormer (ours)	0.99	24.7%	34.1%

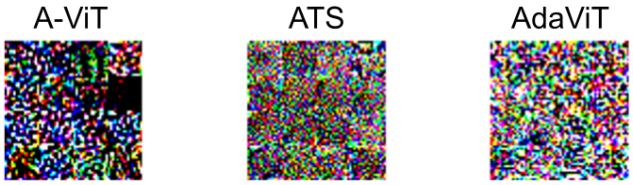


Figure 3. **Visualization of optimized patch:** We show the learned universal patches for each of the three efficient methods.

Table 4. **Effect of patch size:** Analysis of the effect of adversarial patch size on the attack success rate on A-ViT. Our attach is reasonably successful even using a small patch size (32×32), which is only 2% of the image area. Interestingly, a small patch on the corner of the view affects the computational flow of the entire transformer model. This might be due to the global attention mechanism in transformers.

Patch Size (Area)	Model GFLOPs	Top-1 Accuracy	Attack Success
ViT-Tiny	1.26	-	-
A-ViT-Tiny	0.87	71.4%	-
64 (8%)	1.26	4.7%	100%
48 (5%)	1.26	1.8%	99%
32 (2%)	1.22	17.4%	90%
16 (0.5%)	0.98	63.3%	27%
ViT-Small	4.6	-	-
A-ViT-Small	3.7	78.8%	-
64 (8%)	4.6	2.3%	99%
48 (5%)	4.6	5.1%	98%
32 (2%)	4.4	39.5%	78%
16 (0.5%)	3.8	78.2%	16%

Results. The results of our attack, SlowFormer , on various methods on ImageNet dataset are shown in table 1. In A-ViT, we successfully recover 100% of the computation reduced by A-ViT . Our attack has an Attack Success of 60% on ATS and 40% on AdaViT with ViT-Small. A random patch attack has little effect on both the accuracy and computation of the method. Both standard adversarial attack baselines, TAP and NTAP, reduce the accuracy to nearly 0%. Interestingly, these patches further decrease the computation of the efficient model being attacked. This might be because of the increased importance of adversarial patch tokens to the task and thus reduced importance of other tokens. Targeted patch (TAP) has a significant reduction in FLOPs on the ATS method. Since the token dropping in ATS relies on the distribution of attention values of classification tokens, a sharper distribution due to the increased importance of a token can result in a reduction in computation. The

Table 3. **Accuracy controlled compute adversarial attack:** We attack the the efficiency of A-ViT while either maintaining or destroying its classification performance. We observe that our attack can achieve a huge variation in task performance without affecting the Attack Success. The ability to attack the computation without affecting the task performance might be crucial in some applications.

Attack	Model GFLOPs	Attack Success	Top-1 Acc
ViT-Tiny	1.26	-	-
No attack	0.87	-	71.4%
Acc agnostic	1.26	100%	4.7%
Preserve acc	1.23	92%	68.5%
Destroy acc	1.26	100%	0.1%

Table 5. **Attack with adversarial perturbation on ImageNet**. The efficient methods are also susceptible universal perturbation based attacks. We use an ℓ_∞ bound on the perturbation.

Method	Epsilon (/255.)	Attack	Model GFLOPs	Top-1 Accuracy	Attack Success
		ViT-Tiny	**1.3**	-	-
A-ViT	-	No attack	0.87	71.4%	-
	16	SlowFormer	1.15	6.1%	73%
	32	SlowFormer	1.25	0.5%	98.4%
ATS	-	No attack	0.84	70.3%	-
	16	SlowFormer	0.98	15.6%	30.4%
	32	SlowFormer	1.04	0.8%	43.5%
		ViT-Small	**4.6**	-	-
A-ViT	-	No attack	3.7	78.8%	-
	16	SlowFormer	4.48	20%	86.4%
	32	SlowFormer	4.59	1%	98.1%
ATS	-	No attack	3.1	79.2%	-
	16	SlowFormer	3.6	31.0%	33.3%
	32	SlowFormer	3.8	3.6%	46.7%
AdaVit	-	No attack	2.25	77.3%	-
	16	SlowFormer	3.0	26.1%	31.9%
	32	SlowFormer	3.2	2.8%	40.4%

computation increase with SlowFormer for AdaViT is comparatively low. To investigate, we ran a further experiment using a patch size of 224×224 (entire image size) to find the maximum possible computation for an image. This resulted in 4.18 GFLOPs on the ImageNet-1K validation set, which is markedly lower than the limit of 4.6. Using this as an upper-bound of GFLOPs increase, SlowFormer achieves a 49% Attack Success.

We report the results on CIFAR-10 dataset in Table 2. The efficient model (A-ViT) drastically reduces the computation from 1.26 GFLOPs to 0.11 GFLOPs. Most of the tokens are dropped as early as layer two in the efficient model. SlowFormer is able to effectively attack even in such extreme scenarios, achieving an Attack Success of 40% and increasing the mean depth of tokens from nearly one to five. SlowFormer is similarly effective on ATS with an Attack Success of 34%.

We additionally visualize the effectiveness of our attack in Figure 2. The un-attacked efficient method retains only highly relevant tokens at the latter layers of the network. However, our attack results in nearly the entire image being passed through all layers of the model for all inputs. In Fig. 3, we visualize the optimized patches for each of the three efficient methods.

5.2. Ablations:

We perform all ablations on the A-ViT approach using their pretrained ViT-Tiny architecture model.

Accuracy controlled compute adversarial attack: As seen in Table 1, our attack can not only increase the computation, but also reduce the model accuracy. This can be desirable or hurtful based on the attacker's goals. A low-accuracy model might be an added benefit, similar to regular adversaries, but might also lead to the victim detecting the attack. We show that it is possible to attack the computation of the model while either preserving or destroying the task performance by additionally employing a task loss in the patch optimization. Table 3 indicates that the accuracy can be significantly modified while maintaining a high Attack Success.

Effect of patch size: We vary the patch size from 64×64 to 16×16 (just a single token) and report the results in Table 4. Interestingly, our attack with ViT-Small has a 73% Attack Success with a 32×32 patch size, which occupies only 2% of the input image area.

Effect of patch location: We vary the location of the patch to study its effect on the Attack Success of the model. We randomly sample a location in the image for where we paste the patch on. We perform five such experiments and observe an Attack Success of 100% for all patch locations.

Perturbation attack: While we focus on patch based attacks in this paper, efficient transformers are also susceptible to perturbation based attacks (table 5). In perturbation attacks, all pixels in the image can be modified, but with an upper bound on the ℓ_∞ norm of the perturbation.

Table 6. **Defense using adversarial training:** We propose and show the impact of our defense for our adversarial attack on A-ViT. Our defense is simply maintaining a set of universal patches and training the model to be robust to a random sample of those at each iteration. The defense reduces the computation to some extent (1.26 to 1.01), but is still far from the unattacked model (0.87).

Method	GFLOPs	Top-1 Acc.	Attack Success
No attack	0.87	71.4	-
SlowFormer	1.26	4.7%	100%
Adv Defense + SlowFormer	1.01	65.8%	34%

5.3. Adversarial training based defense

Our simple defense that is adopted from standard adversarial training is explained in Section 4. The results for defending against attacking A-ViT are shown in Table 6. The original A-ViT reduces the GFLOPs from 1.26 to 0.87, our attack increases it back to 1.26 with 100% attack success. The proposed defense reduces the GFLOPs to 1.01 which is still higher than the original 0.87. We hope our paper encourages the community to develop better defense methods to reduce the vulnerability of efficient vision transformers.

6. Conclusion

Recently, we have seen efficient vision transformer models in which the computation is adaptively modified based on the input. We argue that this is an important research direction and that there will be more progress in this direction in the future. However, we show that the current methods are vulnerable to a universal adversarial patch that increases the computation and thus power consumption at inference time. Our experiments show promising results for three SOTA efficient transformer models, where a small patch that is optimized on the training data can increase the computation to the maximum possible level in the testing data in some settings. We also propose a defense that reduces the effectiveness of our attack. We hope that our paper will encourage the community to study such attacks and develop better defense methods on various machine learning methods, including generative models, that reduce the computation adaptively with the input.

Acknowledgment: This work was partially supported by DARPA under Contract No. HR00112190135 and HR00112290115 and NSF grant 1845216.

References

[1] iphone battery capacity. https://bigthink.com/the-future/battery-technology-lags/. 1

[2] Starship robot. https://www.wevolver.com/specs/starship-technologies-starship-robot. 1

[3] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34, 2021. 2

[4] Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. Batch-shaping for learning conditional channel gated networks. *arXiv preprint arXiv:1907.06627*, 2019. 2

[5] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017. 2

[6] 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. 2

[7] Jason Ross Brown, Yiren Zhao, Ilia Shumailov, and Robert D Mullins. Dartformer: Finding the best type of attention. *arXiv preprint arXiv:2210.00641*, 2022. 2

[8] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 3

[9] Shaofeng Cai, Yao Shu, and Wei Wang. Dynamic routing networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3588–3597, 2021. 2

[10] Shijie Cao, Lingxiao Ma, Wencong Xiao, Chen Zhang, Yunxin Liu, Lintao Zhang, Lanshun Nie, and Zhi Yang. Seernet: Predicting convolutional neural network feature-map sparsity through low-bit quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11216–11225, 2019. 3

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2

[13] Guangyi Chen, Chunze Lin, Liangliang Ren, Jiwen Lu, and Jie Zhou. Self-critical attention learning for person re-identification. In *ICCV*, pages 9637–9646, 2019. 2

[14] Jin Chen, Xijun Wang, Zichao Guo, Xiangyu Zhang, and Jian Sun. Dynamic region-aware convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8064–8073, 2021. 3

[15] Simin Chen, Hanlin Chen, Mirazul Haque, Cong Liu, and Wei Yang. Slothbomb: Efficiency poisoning attack against dynamic neural networks. 3

[16] Simin Chen, Mirazul Haque, Zihe Song, Cong Liu, and Wei Yang. Transslowdown: Efficiency attacks on neural machine translation systems. 2021. 3

[17] Simin Chen, Zihe Song, Mirazul Haque, Cong Liu, and Wei Yang. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15365–15374, 2022. 3

[18] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[19] 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*, pages 248–255. Ieee, 2009. 5

[20] 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. 2

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[22] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. *arXiv preprint arXiv:1910.10073*, 2019. 2

[23] Quanfu Fan, Chun-Fu (Ricarhd) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More Is Less: Learning Efficient Video Representations by Temporal Aggregation Modules. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[24] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers, 2021. 3

[25] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 396–414. Springer, 2022. 2, 4, 5

[26] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res*, 23:1–40, 2021. 2

[27] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 3

[28] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov.

Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1039–1048, 2017. 2, 3

[29] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[30] Yonggan Fu, Shunyao Zhang, Shang Wu, Cheng Wan, and Yingyan Lin. Patch-fool: Are vision transformers always robust against adversarial perturbations? *arXiv preprint arXiv:2203.08392*, 2022. 3

[31] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic channel pruning: Feature boosting and suppression. *arXiv preprint arXiv:1810.05331*, 2018. 2

[32] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *arXiv preprint arXiv:1412.6572*, 2014. 3

[33] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. 2

[34] Jiaqi Guan, Yang Liu, Qiang Liu, and Jian Peng. Energy-efficient amortized inference with cascaded deep classifiers. *arXiv preprint arXiv:1710.03368*, 2017. 2

[35] Mirazul Haque, Anki Chauhan, Cong Liu, and Wei Yang. Ilfo: Adversarial attack on adaptive neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14264–14273, 2020. 3

[36] Mirazul Haque, Simin Chen, Wasif Arman Haque, Cong Liu, and Wei Yang. Nodeattack: Adversarial attack on the energy consumption of neural odes. 2021. 3

[37] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 2

[38] Charles Herrmann, Richard Strong Bowen, and Ramin Zabih. Channel selection using gumbel softmax. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII*, pages 241–257. Springer, 2020. 2

[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[40] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. 2

[41] Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. *arXiv preprint arXiv:2010.02432*, 2020. 3

[42] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[43] Weizhe Hua, Yuan Zhou, Christopher M De Sa, Zhiru Zhang, and G Edward Suh. Channel gating neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[44] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 2

[45] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 2

[46] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. *arXiv preprint arXiv:2209.04881*, 2022. 2

[47] Shu Kong and Charless Fowlkes. Pixel-wise attentional gating for scene parsing. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1024–1033. IEEE, 2019. 3

[48] Soroush Abbasi Koohpayegani and Hamed Pirsiavash. Sima: Simple softmax-free attention for vision transformers. *arXiv preprint arXiv:2206.08898*, 2022. 2

[49] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5

[50] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 3

[51] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 2

[52] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, pages 5764–5772, 2017. 3

[53] Zhichao Li, Yi Yang, Xiao Liu, Feng Zhou, Shilei Wen, and Wei Xu. Dynamic computational time for visual attention. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1199–1209, 2017. 3

[54] Jing Liu, Zizheng Pan, Haoyu He, Jianfei Cai, and Bohan Zhuang. Ecoformer: Energy-saving attention with linear complexity. *arXiv preprint arXiv:2209.09004*, 2022. 2

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[56] 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 (ICCV)*, 2021. 2

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019. 5, 6

[58] 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, 2021. 2

[59] Wenhao Lu, Jian Jiao, and Ruofei Zhang. Twinbert: Distilling knowledge to twin-structured bert models for efficient retrieval. *arXiv preprint arXiv:2002.06275*, 2020. 2

[60] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 4

[61] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7838–7847, 2021. 3

[62] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021. 2

[63] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022. 2, 3, 4, 6

[64] Jianhong Pan, Qichen Zheng, Zhipeng Fan, Hossein Rahmani, Qiuhong Ke, and Jun Liu. Gradauto: Energy-oriented attack on dynamic neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 637–653. Springer, 2022. 3

[65] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016. 3

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[68] 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, 2021. 2

[69] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[70] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks, 2016. 2

[71] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. Sbnet: Sparse blocks network for fast inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8711–8720, 2018. 3

[72] Aniruddha Saha, Akshayvarun Subramanya, Koninika B. Patil, and Hamed Pirsiavash. Adversarial patches exploiting contextual reasoning in object detection. In *ArXiv*, 2019. 3

[73] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference, 2023. 1

[74] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021. 2

[75] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 3

[76] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469. IEEE, 2016. 2

[77] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021. 2

[78] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[80] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018. 2

[81] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2320–2329, 2020. 3

[82] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–424, 2018. 2, 3

[83] Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2668–2676, 2022. 3

[84] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017. 3

[85] Zhenda Xie, Zheng Zhang, Xizhou Zhu, Gao Huang, and Stephen Lin. Spatially adaptive inference with stochastic feature sampling and interpolation. In *Computer Vision–ECCV*

2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pages 531–548. Springer, 2020. 3

[86] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2369–2378, 2020. 3

[87] Zerui Yang, Yuhui Xu, Wenrui Dai, and Hongkai Xiong. Dynamic-stride-net: Deep convolutional neural network with dynamic stride. In *Optoelectronic Imaging and Multimedia Technology VI*, pages 42–53. SPIE, 2019. 3

[88] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. 2, 3, 4, 5

[89] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 2

[90] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[91] Zhihang Yuan, Bingzhe Wu, Guangyu Sun, Zheng Liang, Shiwan Zhao, and Weichen Bi. S2dnas: Transforming static cnn model for dynamic inference via neural architecture search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 175–192. Springer, 2020. 2

[92] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16415–16424, 2023. 3

[93] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[94] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[95] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[96] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 2