

# Patch Ranking: Token Pruning as Ranking Prediction for Efficient CLIP

Cheng-En Wu\* Jinhong Lin\* Yu Hen Hu Pedro Morgado

University of Wisconsin–Madison

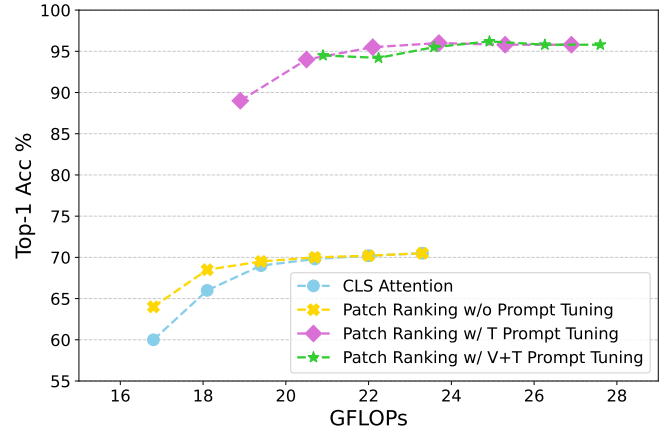
{cwu356, jlin522, yhhu, pmorgado}@wisc.edu

## Abstract

Contrastive image-text pre-trained models such as CLIP have shown remarkable adaptability to downstream tasks. However, they face challenges due to the high computational requirements of the Vision Transformer (ViT) backbone. Current strategies to boost ViT efficiency focus on pruning patch tokens but fall short in addressing the multimodal nature of CLIP and identifying the optimal subset of tokens for maximum performance. To address this, we propose greedy search methods to establish a “Golden Ranking” and introduce a lightweight predictor specifically trained to approximate this Ranking. To compensate for any performance degradation resulting from token pruning, we incorporate learnable visual tokens that aid in restoring and potentially enhancing the model’s performance. Our work presents a comprehensive and systematic investigation of pruning tokens within the ViT backbone of CLIP models. Through our framework, we successfully reduced 40% of patch tokens in CLIP’s ViT while only suffering a minimal average accuracy loss of 0.3% across seven datasets. Our study lays the groundwork for building more computationally efficient multimodal models without sacrificing their performance, addressing a key challenge in the application of advanced vision-language models.

## 1. Introduction

Contrastive Language-Image Pretraining (CLIP) [24] has emerged as a paradigm shift in the field of visual recognition, demonstrating remarkable transferability across a wide array of downstream tasks through zero-shot inference. By training models to align representations of images with text descriptions at scale (400 million text-image pairs in [24]), language-image pretraining enables zero-shot (or open-dictionary) recognition by matching the learned visual embeddings to class embeddings constructed from hand-crafted text prompts such as “a photo of a [class]”. Although



**Figure 1.** Accuracy vs. complexity for various token pruning strategies in pre-trained CLIP models is evaluated on Caltech101. Six points represent models with token keep-rates from 100% to 50%. The CLS Attention method prunes image patches by measuring similarity between CLS tokens and others in the 4th layer of CLIP’s ViT. Patch Ranking, using our Preservation-based ranking strategy, outperforms the traditional CLS method. Patch Ranking w/ T Prompt Tuning and Patch Ranking w/ V+T Prompt Tuning extend this by adding learnable tokens to the Text Encoder or both the ViT and Text Encoder, fine-tuned with 16 shots per class. Prompt-tuning boosts performance, and tuning both prompts (green line) shows no significant degradation up to a 50% keep rate.

CLIP’s success is undeniable, commonly used CLIP backbones like the Vision Transformer (ViT) [7] can be computationally intensive during inference. The complexity of this process increases quadratically with the length of the tokens in the self-attention layer, posing significant challenges for practical deployment.

One of the most direct and effective strategies to alleviate this computational burden is token pruning, a method that has received considerable attention recently [4, 8, 13, 15, 18,

\*Equal Contribution

20, 25, 27, 29–31]. Current studies in this domain focus mainly on the design of various metrics to assess token importance to eliminate those deemed redundant. However, these approaches [15, 18, 30] typically rely on the attention between the CLS and patch tokens. However, it is unclear what these attention weights capture, particularly in the early stages of the model.

To address this issue, we propose a framework in which pruning is guided by well-defined and interpretable scoring functions. Our method follows three phases: Phase I ranks each token based on three scoring functions that measure the usefulness of each token for either (1) optimal classification, (2) maximum confidence in the model’s prediction, or (3) minimal impact on the model’s output representation. These scoring functions establish the “Golden Ranking” of tokens. While useful, the Golden Ranking does not necessarily speed up inference, as the full sequence must be evaluated to compute it. Thus, in Phase II, we introduce a lightweight predictor, trained to closely approximate the Golden Ranking, and thus to determine which tokens to prune during inference. Finally, Phase III addresses potential performance drops due to token pruning by tuning the model to operate on pruned sequences. Given our focus on resource efficiency (both during inference and training), we demonstrate that prompt tuning techniques, *i.e.*, by integrating both learnable visual and text tokens, effectively recover the model performance with minimal training budget requirements. Through systematic experiments, on a large number of datasets, we validated the effectiveness of our method. Our results demonstrate that the proposed framework significantly reduces computational complexity without compromising the model’s classification accuracy. As a result, this work presents a compelling solution that balances model efficiency and effectiveness, opening new avenues for the practical application of CLIP models in various real-world settings. Our contributions are summarized as follows:

- We propose the “Golden Ranking” which ranks patch tokens in CLIP models based on their usefulness to the model’s predictive capabilities.
- We introduce a lightweight predictor, specifically trained to approximate the Golden Ranking, to guide the token pruning process in CLIP models during inference.
- We demonstrate that the integration of learnable tokens into the CLIP model compensates for the inevitable performance degradation due to token pruning, effectively enhancing the model’s accuracy post-pruning.

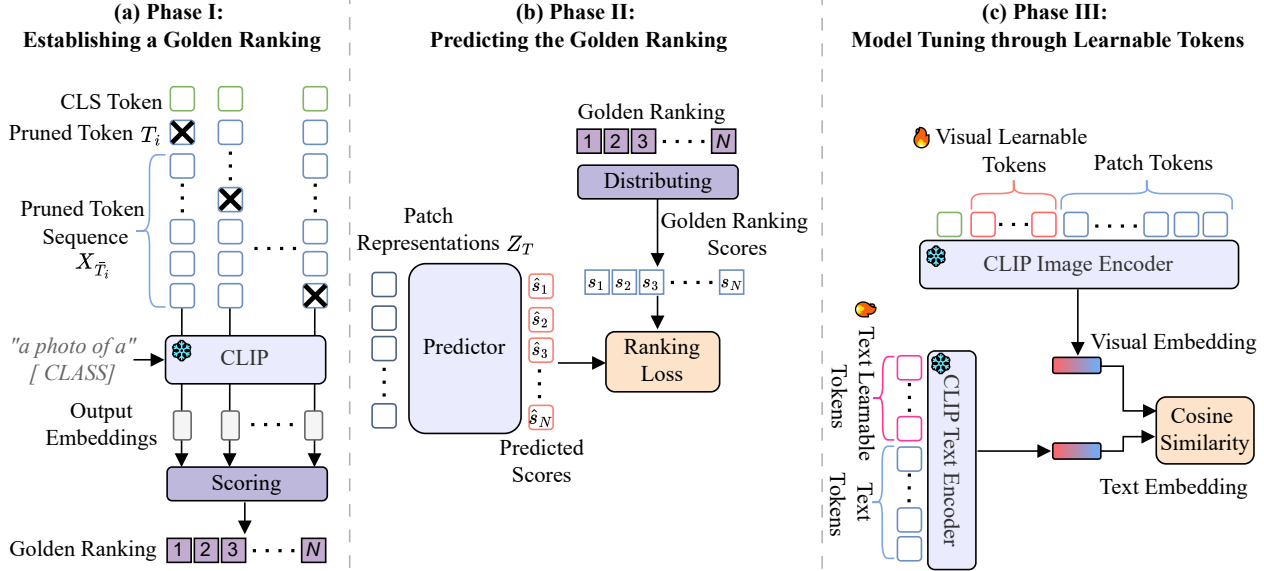
## 2. Related Work

**Token Pruning.** The efficiency of ViT is crucial, especially because their attention mechanisms require a lot of computa-

tional resources. A direct and intuitive approach to enhance ViT efficiency is the reduction of patch tokens, especially considering that some of these tokens may be redundant. Various studies have proposed approaches to evaluate and prune less informative tokens. These approaches broadly fall into two categories. The first category leverages the weights of patch tokens as attended by the CLS token, effectively identifying tokens with lesser contributions to the overall model prediction [15, 18, 30]. The second category involves the integration of additional learnable modules within the ViT architecture [20, 22, 25]. However, neither of these methods conclusively demonstrates that the tokens pruned are indeed the optimal subset, which would allow the model to achieve the highest possible accuracy for a given pruning ratio. In our token pruning approach, we introduce “Golden Ranking” — an optimal ranking of tokens determined by our proposed metrics. This ranking acts as a ground truth for a newly introduced lightweight predictor within ViTs. By learning from the Golden Ranking, the predictor can efficiently discern which tokens to prune, striking a balance between maintaining model accuracy and enhancing computational efficiency.

**Prompt Tuning.** Prompt tuning is a new paradigm for adapting pre-trained models for various tasks and domains. It includes text prompt tuning in natural language processing (NLP), which has progressed from using handcrafted prompts in models like GPT-3 [3] to learnable prompts for better unimodal task performance [14, 17]. In computer vision, VPT [10] introduces visual learnable tokens for ViT to enhance the transfer performance of downstream tasks. Recently, prompt tuning has been extended to vision-language (V-L) pre-trained models. For example, CoOp [33] adapts CLIP for downstream tasks by optimizing learnable text prompts. Co-CoOp [32] builds on this by introducing a meta-net that integrates image-conditional context with text prompts. Despite significant progress in prompt learning for VL models, most methods primarily target the text encoder, neglecting the image encoder’s adaptation, which can result in less optimal visual features. To overcome this, MaPLe [12] aims to simultaneously learn both vision and language prompts.

**Learning to Rank.** Learning-to-rank has gained prominence in machine learning, primarily utilizing a score-and-sort strategy to solve ranking problems. The main aim of these approaches is to create scoring functions that determine the relevance of individual items, which are then used to derive a ranking order. Learning-to-rank has found widespread application in various fields, notably in information retrieval [16] and recommendation systems [11]. In this paper, we introduce an innovative application within this field. To our knowledge, it is the first work to adapt the learning-to-rank approach for pruning patch tokens in ViT. We develop a lightweight module that evaluates and ranks the importance



**Figure 2.** This diagram presents an overview of our pruning framework for patch tokens in CLIP’s ViT. The framework comprises three main phases: **(a) Phase I: Establishing a Golden Ranking**, which involves assigning scores to each token based on their importance, as discussed in Section 3.1; **(b) Phase II: Predicting the Golden Ranking**, which focuses on training a predictor to approximate the Golden Ranking, as elaborated in Section 3.2; and **(c) Phase III: Model Tuning through Learnable Tokens**, a process where additional visual learnable tokens are added to mitigate accuracy loss resulting from the removal of patch tokens, detailed in Section 3.3.

of patch tokens. This ranking then guides the pruning process, removing tokens of lesser significance. Such an approach significantly boosts the efficiency of ViT models.

### 3. Method

Unlike existing methods that predominantly leverage CLS attention for token ranking — a technique not ideally suited for CLIP due to its multi-modal embedding architecture — we introduce a novel token pruning strategy, illustrated in Fig. 2. The proposed framework unfolds into three distinct stages: (a) Searching for the “Golden Ranking” of patch tokens; (b) Learning to rank patch tokens by training a predictor to approximate the Golden Ranking; and (c) Compensating for the potential performance degradation incurred after removal of uninformative tokens. These three stages, elaborated in the subsequent sections, collectively form the foundation of our approach and enable the deployment of CLIP models with significant speed enhancements and minimal performance degradation.

#### 3.1. Phase I: Establishing a Golden Ranking

Token pruning can be conceptualized as a problem of identifying the optimal subset of tokens that maximizes the model’s accuracy. While a brute force search could in principle identify this optimal subset, it is impractically time-

consuming. Consequently, most existing methods resort to heuristic approaches, such as utilizing CLS attention weights, to determine suitable subsets for pruning. However, these methods often fall short of truly approximating the optimal pruning subset. To bridge this gap, we introduce three scoring metrics designed to rank patch tokens based on their impact on CLIP’s predictions, forming what we term the ‘Golden Ranking’.

Fig. 2 (a) illustrates how the Golden Ranking is obtained. We start with a set of class prompts and an image that has been subdivided and encoded into a token sequence  $X$  of length  $N$ . Normally, CLIP’s ViT encoder  $f_i$  would process this token sequence  $Z$ ,  $Z^{cls} = f_i(X)$ , to obtain a set of visual embeddings  $Z$  and a CLS token embedding  $Z_{cls}$ . However, to determine the Golden Ranking, we apply the CLIP model to a series of pruned token sequences  $X_{T_i}$ , each of which is obtained by removing a small set of tokens  $T_i$  from  $X$ . The resulting visual embeddings  $Z_{T_i}$ ,  $Z_{T_i}^{cls} = f_i(X_{T_i})$  can then be used to determine the importance of each token in  $T_i$ . To measure this score, denoted  $s(T_i)$ , we experimented with three distinct metrics:

**(1) Label-Driven Ranking Score** The pruned tokens  $T_i$  are scored based on CLIP’s zero-shot posterior probability of assigning the pruned sequence  $X_{T_i}$  to the ground-

truth label  $y_{\text{gt}}$

$$s(\mathcal{T}_i) = P(y_{\text{gt}} | \mathbf{X}_{\bar{\mathcal{T}}_i}) \quad (1)$$

A high score of  $s(\mathcal{T}_i)$  suggests that the removed tokens are not required for accurate classification, and some tokens might even be misleading, resulting in more accurate classification after removal.

- (2) **Maximum Confidence Score** Label-Driven Ranking Score requires prior knowledge of the ground-truth class. To avoid this assumption, we can alternatively assess the pruned tokens  $\mathcal{T}_i$  based on the maximum confidence across all classes.

$$s(\mathcal{T}_i) = \max_y P(y | \mathbf{X}_{\bar{\mathcal{T}}_i}) \quad (2)$$

A high score of  $s(\mathcal{T}_i)$  indicates that the removal of tokens in  $\mathcal{T}_i$  did not reduce (or even increase) the model’s overall confidence in its prediction.

- (3) **Feature Preservation Score** Finally, instead of searching for tokens that optimize classification performance, which makes the golden ranking task-specific, feature preservation seeks to identify the tokens that, when removed, do not alter the image representation, as expressed by the CLS token embedding. This score is quantified using cosine similarity:

$$s(\mathcal{T}_i) = \frac{\mathbf{Z}^{cls} \cdot \mathbf{Z}_{\bar{\mathcal{T}}_i}^{cls}}{\|\mathbf{Z}^{cls}\| \|\mathbf{Z}_{\bar{\mathcal{T}}_i}^{cls}\|} \quad (3)$$

where  $\mathbf{Z}^{cls}$  denotes the CLS embedding obtained from the full sequence  $\mathbf{X}$  and  $\mathbf{Z}_{\bar{\mathcal{T}}_i}^{cls}$  denotes the embedding obtained with a pruned sequence.

The metrics delineated above measure the importance of a set of tokens  $\mathcal{T}_i$ . To determine the importance of individual tokens, a straightforward approach is to prune one at a time, *i.e.*,  $\mathcal{T}_i = t_i$ . However, this results in small changes in the model’s output, making it challenging to discern the relative importance of different tokens. Instead, we remove a larger  $r \times r$  block of tokens, resulting in more noticeable changes. As the removal block  $\mathcal{T}_i$  slides over the image, each token is removed and assessed multiple times, thus stabilizing the final average score of each token  $t$

$$s(t) = \frac{1}{|\mathcal{T}_i|} \sum_{i:t \in \mathcal{T}_i} s(\mathcal{T}_i). \quad (4)$$

The time complexity of estimating the golden ranking scores  $s(t)$  using the sliding window approach is  $O(L)$ , where  $L$  is the number of patches in the sequence. However, in practice, we can significantly reduce the time required to score all tokens, by creating a batch with all pruned sequences  $\mathbf{X}_{\bar{\mathcal{T}}_i}$ , and processing them simultaneously through the model.

### 3.2. Phase II: Predicting the Golden Ranking

After establishing the Golden Ranking using one of the three metrics above, we train a lightweight predictor,  $\hat{s} = h(\mathbf{Z}; \Theta) \in \mathbb{R}^N$ , to efficiently approximate it, and thus identify the least useful tokens from their representations  $\mathbf{Z}$ . Since token removal must occur early on to reduce the computational complexity, we deployed the predictor on internal representations  $\mathbf{Z}^{(i)}$  obtained at an early layer  $i$ .

**Predictor Design.** The predictor architecture is a single Mix-MLP layer [28], chosen for its ability to efficiently capture contextual dependencies among tokens. The Mix-MLP performs two main functions: token mixing and channel mixing, implemented using MLP layers. Given a sequence of patch representations  $\mathbf{Z}_{\mathcal{T}} \in \mathbb{R}^{|\mathcal{T}| \times d}$ , the channel and token mixing processes are computed as

$$\mathbf{Z}_{\mathcal{T}}^{\text{channel}} = \text{MLP}_{\text{channel}}(\mathbf{Z}_{\mathcal{T}}) + \mathbf{Z}_{\mathcal{T}} \quad (5)$$

$$\mathbf{Z}_{\mathcal{T}}^{\text{token}} = \text{MLP}_{\text{token}}(\mathbf{Z}_{\mathcal{T}}^{\text{channel}}) + \mathbf{Z}_{\mathcal{T}}^{\text{channel}} \quad (6)$$

$$(7)$$

where  $\text{MLP}_{\text{channel}}$  and  $\text{MLP}_{\text{token}}$  operate on the channel and token dimensions, respectively. The final score  $\hat{s}(\mathcal{T}) \in \mathbb{R}^{|\mathcal{T}|}$  is obtained by average pooling of the token representations across the channel dimension,  $\hat{s}(\mathcal{T}) = \text{Avg}_{\text{token}}(\mathbf{Z}_{\mathcal{T}}^{\text{token}})$ . Each MLP layer consists of a fully connected layer accompanied by layer normalization and a GELU activation function. **Loss Function.** The predictor  $\hat{s} = h(\mathbf{Z}^{(i)}; \Theta)$  is trained to regress normalized golden ranking scores  $s_t = \frac{s(t) - \mu_s}{\sigma_s}$ , where  $\mu_s$  and  $\sigma_s$  denote the mean and standard deviation of  $s(t)$  among all patch tokens  $t$ . Specifically, we minimize

$$\mathcal{L}(\Theta) = - \sum_t \sigma(s_t) \log \sigma(\hat{s}_t) \quad (8)$$

where  $\sigma(\cdot)$  denotes the sigmoid function. Although this loss does not aim to directly predict the ranking between patches (which is too difficult to accomplish from early-stage representations), it encourages the predictor to assign the highest scores to tokens at the top of the ranking and the lowest scores to those at the bottom.

**Token Removal.** During inference, a predetermined number of patch tokens is removed. Since the Golden Ranking predictor operates on CLIP’s intermediate representations, the full sequence  $\mathcal{T}$  is maintained until the predictor is applied. The tokens with the lowest predicted scores are then removed, and the compressed sequence is passed through the rest of the model. We experiment with different removal rates to determine the optimal trade-off between speed and accuracy. We also explore the impact of progressively removing tokens in a layer-wise manner, starting from the predictor layer. Our results show that progressive pruning is more effective, as it allows the model to remove tokens in a more controlled manner.



Dataset	Golden Ranking Score	0-Shot Acc No Pruning	Golden Ranking Keep rate				Matching Rate@100	Predicted Ranking Keep rate			
			80	70	60	50		80	70	60	50
Caltech101	Label	93.5	<b>94.9</b>	<b>94.6</b>	<b>94.0</b>	<b>92.4</b>	56.7	91.6	89.1	86.4	83.9
	Confidence		91.6	90.9	90.4	89.1	57.0	91.5	88.4	85.3	83.4
	Preservation		92.8	92.4	92.2	91.4	<b>78.0</b>	<b>93.6</b>	<b>93.2</b>	<b>93.4</b>	<b>91.0</b>
OxfordPets	Label	89.5	<b>93.9</b>	<b>94.6</b>	<b>94.4</b>	<b>93.8</b>	59.8	87.7	87.1	85.9	<b>84.5</b>
	Confidence		88.1	88.2	88.0	87.2	60.4	86.8	86.0	84.6	83.1
	Preservation		88.9	88.4	88.0	85.4	<b>78.2</b>	<b>88.6</b>	<b>88.4</b>	<b>88.1</b>	83.5
Flowers102	Label	70.5	<b>80.6</b>	<b>82.3</b>	<b>83.2</b>	<b>80.8</b>	56.8	68.8	66.7	65.7	63.3
	Confidence		70.6	70.6	70.3	68.5	57.8	68.5	67.3	65.6	62.5
	Preservation		70.8	70.4	69.1	66.9	<b>71.9</b>	<b>69.6</b>	<b>68.8</b>	<b>67.9</b>	<b>63.9</b>
Food101	Label	86.0	<b>95.9</b>	<b>96.4</b>	<b>96.2</b>	<b>95.0</b>	55.5	83.6	80.6	77.4	72.1
	Confidence		86.1	86.0	85.7	84.7	55.5	83.2	80.3	76.9	71.7
	Preservation		85.8	85.4	84.1	79.5	<b>71.9</b>	<b>85.1</b>	<b>84.2</b>	<b>82.8</b>	<b>75.8</b>
FGVCAircraft	Label	23.4	<b>43.5</b>	<b>43.6</b>	<b>44.2</b>	<b>43.5</b>	60.1	17.6	17.3	17.1	17.1
	Confidence		24.2	23.8	23.9	23.5	61.4	18.6	18.7	18.2	17.9
	Preservation		25.0	24.8	24.5	23.4	<b>89.3</b>	<b>23.2</b>	<b>23.2</b>	<b>22.9</b>	<b>22.4</b>
DTD	Label	45.1	<b>53.1</b>	<b>54.0</b>	<b>54.7</b>	<b>55.1</b>	55.7	44.6	<b>44.6</b>	<b>45.0</b>	<b>43.3</b>
	Confidence		44.3	44.1	44.4	44.3	56.1	44.9	44.6	44.7	43.0
	Preservation		44.1	44.0	43.9	43.3	<b>72.3</b>	<b>45.0</b>	44.2	43.7	43.0
UCF101	Label	67.0	<b>79.0</b>	<b>79.1</b>	<b>78.7</b>	<b>77.1</b>	55.9	65.2	63.6	60.0	56.4
	Confidence		66.2	65.0	64.6	63.5	55.9	64.7	62.3	58.7	54.6
	Preservation		66.6	67.4	67.0	65.2	<b>79.7</b>	<b>66.6</b>	<b>65.9</b>	<b>65.2</b>	<b>60.9</b>

**Table 1.** Pruning effectiveness leveraging either the ground-truth or predicted golden ranking using three scoring functions: Label, Confidence, and Preservation. “Matching Rate” measures the agreement of the top 100 tokens in the ground-truth and predicted rankings. In all cases, token pruning was applied at the 4th layer of CLIP’s ViT, and classification was conducted without any additional model tuning.

### 3.3. Phase III: Model Tuning through Learnable Tokens

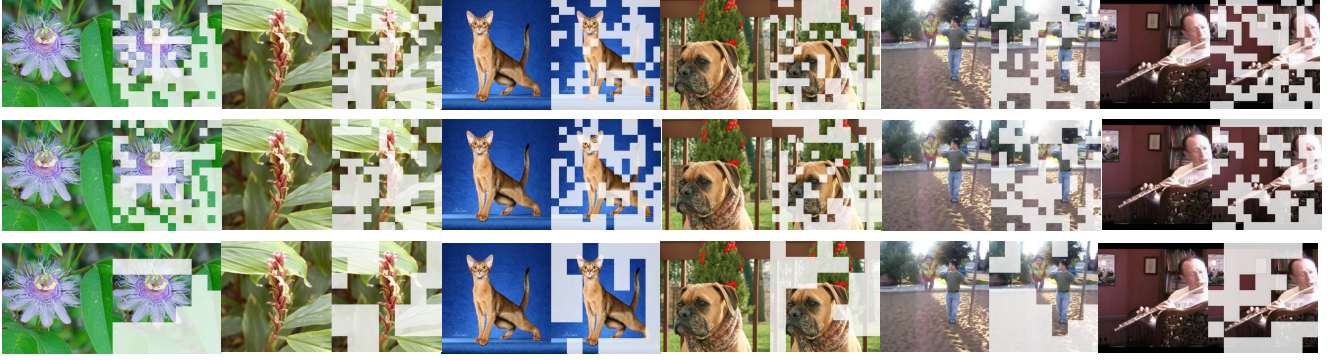
Although the predictor can identify the least useful tokens, removing them can still degrade performance, as the model is forced to operate outside of its training distribution. A variety of fine-tuning methods could be employed to recover the lost performance. For example, one could simply fine-tune the CLIP model on the original 400M image-text pairs using pruned visual input sequences. Although likely to succeed, this approach would be computationally expensive and data-intensive. Instead, inspired by prompt-tuning strategies [12, 33], we introduce a set of learnable tokens into the CLIP model and train them using a small dataset to compensate for the performance degradation. As demonstrated in [33] and [12], adjusting a model through learnable tokens is significantly more data and compute efficient than fine-tuning.

Similarly to CoOp [33], we augment the input sequence to the CLIP’s text encoder,  $\mathbf{W}_t$ , with a set of  $b$  learnable tokens  $\{P_t^i \in \mathcal{R}^{d_t}\}_{i=1}^b$ , where  $d_t$  is the text embedding dimension. We also introduce additional visual tokens to the CLIP image encoder. Following [12], the new visual tokens  $\{P_v^i\}_{i=1}^b$  are

obtained from the text learnable tokens  $\{P_t^i\}_{i=1}^b$  through a linear transformation  $P_v^i = MP_t^i$ , where  $M \in \mathcal{R}^{d_v \times d_t}$  is a learnable projection matrix and  $d_v$  is the visual embedding dimension. The new input sequences,  $\mathbf{W}_t^* = \{P_t^i\}_{i=1}^b \cup \mathbf{W}_t$  and  $\mathbf{W}_v^* = \{P_v^i\}_{i=1}^b \cup \mathbf{W}_v$ , allow the model to dynamically adapt the text and visual representations to better align with each other, compensating for the representational shifts introduced by removing image patches.

## 4. Experiments

In this section, we present our experimental results, demonstrating the effectiveness of our approach in improving CLIP’s efficiency of inference. We first describe the experimental settings, followed by an investigation of the Golden Ranking. Subsequently, we present the results of prompt tuning, which integrates learnable tokens into the CLIP model to recover the performance loss resulting from token pruning. Finally, we conduct an ablation study to evaluate the effectiveness of our approach.



**Figure 3.** Visualization of Scoring Functions for Patch Token Pruning: The scoring functions for patch token pruning are visualized as follows: **Top row** – Label-Driven Ranking Score, **middle row** – Maximum Confidence Score, and **bottom row** – Feature Preservation Score.

#### 4.1. Experimental Setting

We conduct experiments using the ViT-B/16 model as the pre-trained CLIP’s visual encoder on seven datasets: Caltech101 [9], OxfordPets [23], Flowers102 [21], Food101 [2], FGVCAircraft [19], DTD [5], UCF101 [26] and ImageNet [6]. These datasets are chosen to represent a diverse range of image classification tasks, including object recognition, scene classification, and fine-grained classification. The official train/test splits were used for all datasets. While no further training data is used for zero-shot experiments (*i.e.*, without prompt tuning), 16 images per class are used for prompt-tuning experiments, following the training splits of [33].

#### 4.2. Golden Ranking

We begin by evaluating the effectiveness of the Golden Ranking in a Zero-Shot setting. In this experiment, we present our investigation into the Golden Ranking. We focus on four aspects: (1) evaluating its effectiveness in classification performance; (2) examining the predictor’s ability to accurately approximate the Golden Ranking; (3) assessing the classification performance using predictor-estimated rankings; and (4) evaluating the generalizability of the predictor across different datasets.

**Effectiveness of Golden Ranking.** Section 3.1 introduces three distinct scoring functions to establish the utility of each token: *Label-Driven Ranking Score*, *Maximum Confidence Score*, and *Feature Preservation Score*. Table 1 (columns 3 to 7) shows the classification performance of the pruned CLIP model while utilizing the ground truth Golden Rankings at various pruning rates. Interestingly, pruning a significant portion of patch tokens using the label-driven ranking scores results in substantial accuracy improvements over the original 0-shot performance without pruning. This can be attributed to the ability of the label information to effectively identify and eliminate distractor patch tokens, which might mislead the CLIP model. As for the maximum confidence

and feature preservation scores, the overall performance remains relatively stable even after significant pruning, but their effectiveness varies depending on the dataset.

**Golden Ranking Predictability.** Although the label-driven ranking score shows the highest accuracy in Table 1, ground truth rankings are not available during inference. Thus, high performance can only be achieved if the predictor can accurately estimate the Golden Ranking. To quantify this, we measured the percentage of the top 100 predicted tokens that match the top 100 tokens from the Golden Ranking. As shown in Table 1 (column 8), the label-driven and maximum confidence ranking scores are more challenging to predict accurately than the feature preservation score.



**Figure 4.** This figure compares token pruning methods at the 50% keep rate: the **middle column** shows CLS attention weight-based pruning, and the **right column** features our Feature Preservation Score method.

For a deeper understanding of the predictor’s performance, we visualized the tokens pruned by each of the scoring functions in Fig. 3. As can be seen, the tokens pruned according to the label-driven and maximum confidence scoring function seem to be less intuitive (from a human perspective) than those pruned by the feature preservation score.

Test Dataset ↓	Predictor Training Dataset →						
	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	DTD	UCF101
Caltech101	92.0	92.4	91.3	92.5	92.5	92.0	92.0
OxfordPets	86.2	86.7	84.7	85.3	86.4	85.6	86.2
Flowers102	67.9	67.9	67.2	67.6	68.3	68.2	67.0
Food101	82.3	82.0	81.5	81.9	83.4	82.0	82.7
FGVCAircraft	19.9	20.8	18.9	20.5	21.7	19.4	19.1
DTD	45.3	44.3	44.7	43.9	44.0	44.3	45.1
UCF101	61.2	60.6	59.8	60.2	61.9	60.6	60.8

**Table 2.** Cross-dataset generalizability of the golden ranking predictor. Columns indicate the predictor training dataset and rows the testing distribution. Models are evaluated through 0-shot recognition with a keep rate of 50%.

Methods	GFLOPs	Time (ms)
CLIP	23.4	3.5
↪ w/ pruning (60% keep rate)	16.8	1.6
CLIP+Prompt Tuning	27.6	4.0
↪ w/ pruning (60% keep rate)	20.9	2.0

**Table 3.** Computational efficiency in terms of GFLOPs and inference time per image. Measured on an NVIDIA A4500 GPU.

This suggests that the “distractor” tokens identified by the label-driven ranking score are likely to be caused by subtle visual patterns that are challenging for the predictor to learn. The feature preservation score, on the other hand, seems to be more effective in identifying redundant tokens, such as background elements, which are less likely to be crucial for classification.

**Predictor-Based Pruning.** To assess the predictor’s performance, we conduct zero-shot inference utilizing the predictor for token pruning. As shown in Table 1 (last 4 columns), a predictor trained to regress the feature preservation score generally achieves superior accuracy compared to the other two scoring functions across all keep rates. These findings suggest that the higher predictability of the feature preservation score makes it more suitable for token pruning in practice.

**Comparison to CLS Attention Pruning.** We also compared our pruned tokens against the more common CLS attention pruning. As shown in Fig. 4, the proposed scoring function seems to provide a more stable pruning set. Since, unlike CLS attention, the predictor is trained to identify tokens that do not affect the output embeddings, crucial tokens for object identification are more likely to be preserved. As a result, predictor-based pruning is more effective in maintaining classification performance at lower keep rates, as shown in Fig. 1 and expanded in Supplementary Material for a

variety of datasets.

**Generalizability of Predictor.** Despite the encouraging results of 1, one potential drawback of the proposed approach is the fact that the predictor is directly trained on images of the target task. However, we found that the predictor generalizes well across datasets, as highlighted in Table 2. Each predictor, despite being trained on a singular dataset (the columns in the table) to approximate the feature preservation score, can be applied to prune tokens from other datasets without significant drops in performance. Thus, despite requiring additional training data, these results show that our predictor does not require training data from the downstream task and thus retains CLIP’s capability for open-dictionary recognition.

**Runtime.** To better assess computational efficiency, we measured the GFLOPs and inference time of the pruned CLIP model with a keep rate of 60%. The results, presented in Table 3, show that our method significantly reduces the computational cost of existing V-L methods like CLIP, showing a decrease of over 30% in GFLOPs and approximately 50% in inference time.

### 4.3. Comparison with Existing Works

Our work aims to enhance the efficiency of Vision-Language models like CLIP by introducing a lightweight token pruning predictor. This approach differs from traditional methods that utilize CLS attention weights, as it approximates the Golden Ranking, demonstrated in Fig. 1 to be more effective at maintaining classification performance with lower keep rates. We compare our method against existing token pruning techniques for ViT models, such as EViT [15], AViT [31], and ToMe [1], which preserve CLIP’s pre-trained weights without additional training. As shown in Table 4, our method consistently outperforms these alternatives across various datasets. Furthermore, our assessment using both dataset-specific and agnostic predictors reveals that token pruning with a general predictor trained on ImageNet achieves comparable accuracy to dataset-specific predictors, confirming its effectiveness and versatility even

Method	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	DTD	UCF101	ImageNet	Avg	GFLOPs
EViT [15]	92.5	87.1	67.1	80.3	23.3	43.3	63.3	58.1	64.3	16.8
A-ViT [31]	91.4	83.2	67.7	82.3	21.7	43.5	63.3	57.6	63.8	16.8
ToMe [1]	91.5	87.2	67.7	82.4	20.4	41.5	64.9	58.3	64.2	16.8
Ours	93.4	88.1	67.9	82.8	22.9	43.7	65.2	59.5	<b>65.4</b>	16.8
Ours-IN	93.6	87.7	69.6	84.0	21.6	44.3	63.0	59.5	<b>65.4</b>	16.8

**Table 4.** Comparison with prior token pruning methods. We apply these methods to CLIP’s ViT. To ensure the same level of computational cost, we prune tokens at the 4th, 6th, 8th, and 10th layers of the CLIP’s ViT, eliminating 20 tokens at each specified layer. For fair comparison to prior work, we evaluate our token pruning without further prompt tuning (Phase III). **Ours-IN** refers to the predictor trained solely on ImageNet and then applied to all eight datasets.

T-tuning	V-tuning	Pruning	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	DTD	UCF101	Average	GFLOPs
			93.5	89.5	70.5	86.0	23.4	45.1	67.0	67.9	23.4
		✓	93.4	88.1	67.9	82.8	22.9	43.7	65.2	66.3	16.8
✓			95.2	92.5	95.7	84.8	37.3	70.0	81.7	79.6	26.9
✓		✓	94.5	91.7	93.8	82.3	34.7	67.4	81.0	77.9	18.9
✓	✓		95.6	92.3	95.5	84.9	37.6	69.7	82.7	79.8	27.6
✓	✓	✓	95.1	92.2	95.4	84.2	38.8	68.9	81.8	79.5	20.9

**Table 5.** Accuracy evaluation with learnable tokens. Results from applying prompt tuning to CLIP’s ViT to recover accuracy loss from pruning 40% of patch tokens. ‘T-tuning’ uses 16 text prompts, and ‘V-tuning’ uses 16 visual prompts, both shared across dual encoders. Pruning occurs progressively at the 4th, 6th, 8th, and 10th layers, removing 20 tokens each.

in zero-shot settings.

#### 4.4. Prompt Tuning

**Prompt Tuning Ablation.** As discussed in Section 3.3, regardless of the predictor accuracy in identifying redundant patches, the integration of learnable tokens is crucial for (1) enhancing the zero-shot prediction by better aligning visual and class embeddings and (2) recovering the performance loss due to token pruning. To evaluate the effectiveness of prompt tuning, we ablated the integration of text and visual prompts across all datasets. The results, shown in Table 5, support several conclusions. First, as shown in the first two rows, pruning without tuning results in a 1.6% accuracy drop on average across all datasets. While acceptable, this drop can be mitigated with learnable tokens. Second, the integration of text prompts (T-tuning) significantly improves the performance of the baseline method (without pruning), from 67.9% to 79.6% on average. However, text prompts do not help mitigate the performance gap after pruning. Finally, while visual prompts (V-tuning) do not help with baseline performance significantly, they are crucial for recovering the performance loss due to the pruning of visual tokens.

**Prompt Tuning with Varying Keep Rates.** Fig. 1 compares the accuracy of token pruning, with and without prompt tuning for varying keep rates. Fig. 1 also compares the proposed predictor-based tuning with a CLS attention-based method. As can be seen, the proposed method is more effective in maintaining classification performance at lower keep rates.

This improved efficiency is particularly pronounced when learnable visual tokens are integrated into the model, showing nearly unchanged performance even after pruning as much as 50% of the tokens on the Caltech101 dataset.

## 5. Conclusion

In this work, we introduce a novel framework designed for pruning patch tokens in CLIP’s ViT, effectively addressing the computational intensity typically associated with these models. At the heart of our approach is the “Golden Ranking” concept, which methodically ranks patch tokens based on scoring functions. A key innovation in our method is the deployment of a lightweight predictor, trained to closely approximate this Golden Ranking. Furthermore, to mitigate the inevitable performance loss resulting from the pruning process, we integrate learnable text and visual tokens into our framework. These tokens, especially visual tokens, were shown to play a pivotal role in compensating for potential performance degradation, ensuring the model’s output remains accurate post-pruning. Our extensive experiments across a variety of datasets have demonstrated that our framework can achieve a substantial reduction in patch tokens, by up to 40% in CLIP’s ViT, while maintaining comparable performance (only 0.3% lower accuracy).



## Acknowledgement

This work was partially supported by the National Science Foundation under Grant No. 2006394.

## References

- [1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2022. 7, 8
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *ECCV*, 2014. 6
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 2
- [4] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Y. Qiao, and Ping Luo. DiffRate: Differentiable compression rate for efficient vision transformers. 1
- [5] Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 6
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 6
- [7] 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 *ICLR*, 2020. 1
- [8] Mohsen Fayyaz, Soroush Abbasi Kouhpayegani, Farnoush Rezaei Jafari, Eric Sommerlade, Hamid Reza Vaezi Joze, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers. *ECCV*, 2022. 1
- [9] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR-W*, 2004. 6
- [10] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2
- [11] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. Learning to rank for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013. 2
- [12] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, 2023. 2, 5
- [13] 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 *ECCV*, 2022. 1
- [14] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 2
- [15] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Evit: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. 1, 2, 7, 8
- [16] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009. 2
- [17] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *ACL*, 2022. 2
- [18] Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In *CVPR*, 2023. 1, 2
- [19] Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint*, 2013. 6
- [20] 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 *CVPR*, 2022. 1, 2
- [21] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 6
- [22] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red2: Interpretability-aware redundancy reduction for vision transformers. 2021. 2
- [23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 6
- [24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1
- [25] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 1, 2
- [26] Khuram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint*, 2012. 6
- [27] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *CVPR*, 2022. 1
- [28] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 4
- [29] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *CVPR*, 2023. 1
- [30] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *AAAI*, 2022. 1, 2

- [31] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022. 1, 7, 8
- [32] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 2
- [33] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 2, 5, 6

## A. Additional Results Comparison

**Comparison with CLS Attention** In prior works, using CLS attention weight to rank the importance of patch tokens has been a prevalent method for enhancing the efficiency of Vision Transformers (ViTs). However, this approach is less effective for CLIP’s ViT due to its dual-modality structure. Addressing this limitation, we introduce ‘Patch Rank,’ a novel framework tailored for CLIP’s ViT. To assess the efficacy of Patch Rank, we conduct a comparative analysis with the CLS attention method across seven datasets, evaluating performance at keep rates ranging from 100% to 50%. Token pruning was executed at the first layer of CLIP’s ViT to optimize computational savings. Importantly, neither method performs fine-tuning after token pruning. As shown in Figure 5, our Patch Ranking consistently demonstrates higher accuracy than CLS attention across all keep rates and datasets. Notably, our method shows a significant advantage over CLS attention, especially at lower keep rates (60% and 50%). This outcome indicates the ability of Patch Rank to precisely eliminate less informative patch tokens while minimizing the loss in accuracy, thereby affirming its effectiveness in the nature of CLIP’s ViT.

## B. Ablation study

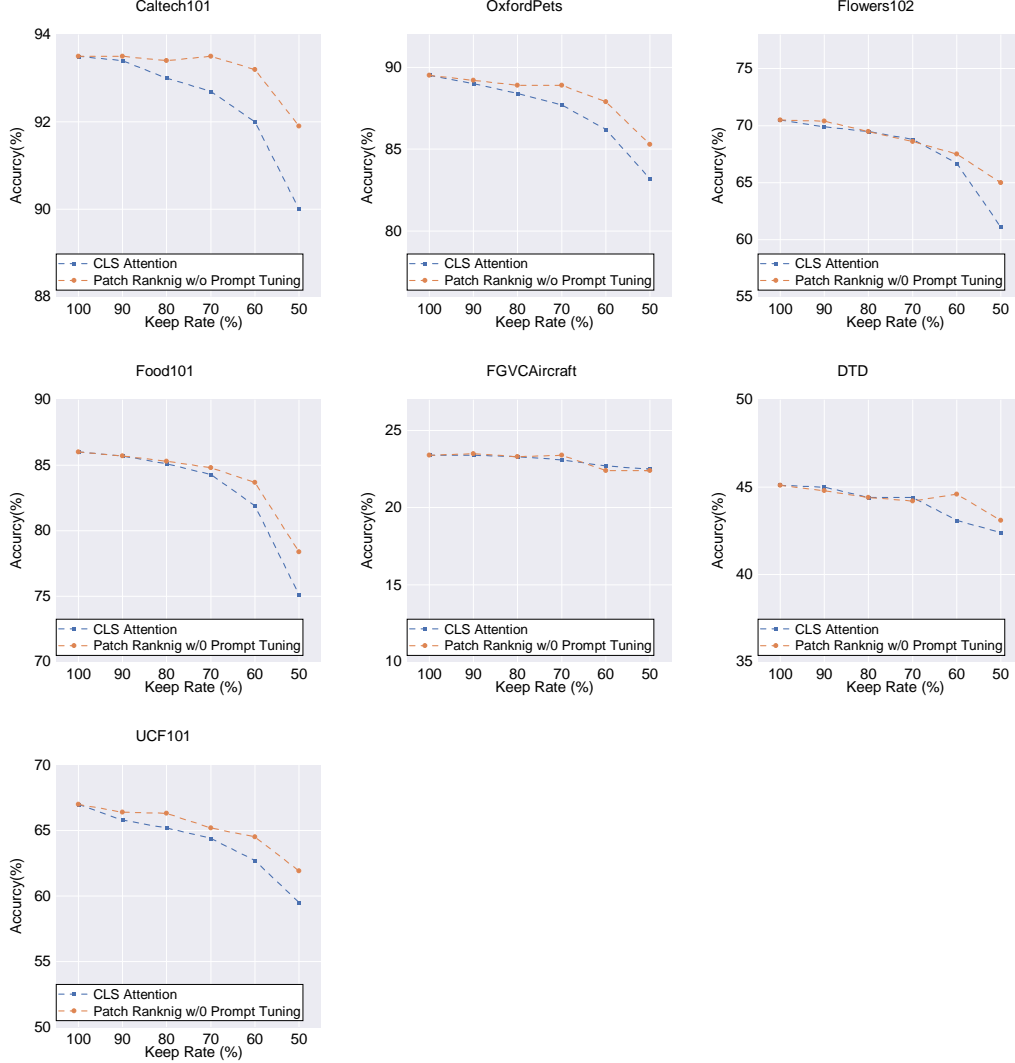
**Architecture of Predictor** To construct our predictor, we selected three different architectures: (1) MLP, which consists of a 256-dimensional hidden layer, layer normalization, GLUE, and a 196-dimensional hidden layer; (2) Transformer, specifically a Transformer-encoder block; and (3) Mix-MLP, which is a single block configuration. To assess the performance of these architectures, we evaluated their top-100 matching rates and pruning effectiveness across various keep rates, from 80% to 50%. As depicted in Table 6, Mix-MLP emerges as the most effective, achieving the highest matching rate. Regarding the performance in token pruning, Mix-MLP demonstrates stable results across all datasets, and notably, it significantly outperforms the other architectures in the UCF101 dataset. This superiority of Mix-MLP can be attributed to its optimal capacity for learning and applying the Golden Ranking, coupled with its ability to avoid overfitting the training dataset. **Token Pruning Locations**

In our exploration of token pruning locations within CLIP’s Vision Transformer, we conducted an in-depth analysis to determine the impact of varying pruning depths on model performance. This involved progressively pruning an equal number of patch tokens at different layers while maintaining a consistent keep rate of 60%. The results are shown in Table 7. It focuses on four distinct combinations of pruning locations, ranging from shallower to deeper layers within the network. Despite a slight margin favoring pruning patch tokens at deeper layers, the overall average performance across all datasets remains notably consistent. This suggests that

Dataset	Arch.	Matching rate	Predictor				
			100	80	70	60	50
Caltech101	MLP	76.5	93.5	93.3	93.2	92.7	91.0
	Trans.	73.4	93.5	93.4	<b>93.3</b>	92.8	<b>91.2</b>
	Mix-MLP	<b>78.0</b>	93.5	<b>93.6</b>	93.2	<b>93.4</b>	91.0
OxfordPets	MLP	75.7	89.5	<b>89.2</b>	88.5	88.0	84.5
	Trans.	72.9	89.5	89.0	<b>89.0</b>	88.0	<b>85.5</b>
	Mix-MLP	<b>78.2</b>	89.5	88.6	88.4	<b>88.1</b>	83.5
Flowers102	MLP	69.2	70.5	69.5	<b>69.2</b>	67.4	<b>64.6</b>
	Trans.	56.2	70.5	<b>69.8</b>	69.0	67.5	60.9
	Mix-MLP	<b>71.9</b>	70.5	69.6	68.8	<b>67.9</b>	63.9
Food101	MLP	70.4	86.0	85.5	84.8	<b>83.7</b>	78.3
	Trans.	69.7	86.0	<b>85.7</b>	<b>85.0</b>	83.8	<b>78.5</b>
	Mix-MLP	<b>71.9</b>	86.0	85.1	84.2	82.8	75.8
FGVCAircraft	MLP	85.2	23.4	23.5	23.1	<b>23.1</b>	<b>22.8</b>
	Trans.	84.6	23.4	<b>23.6</b>	<b>23.6</b>	22.9	<b>22.8</b>
	Mix-MLP	<b>89.3</b>	23.4	23.2	23.2	22.9	22.4
DTD	MLP	<b>77.5</b>	45.1	44.6	44.3	43.7	41.9
	Trans.	62.1	45.1	44.5	<b>44.9</b>	<b>43.8</b>	42.1
	Mix-MLP	72.3	45.1	<b>45.0</b>	44.2	43.7	<b>43.0</b>
UCF101	MLP	77.5	67.0	61.8	58.2	53.6	43.2
	Trans.	51.2	67.0	62.2	60.0	53.5	43.5
	Mix-MLP	<b>79.7</b>	67.0	<b>66.6</b>	<b>65.9</b>	<b>65.2</b>	<b>60.9</b>

**Table 6.** Design Choices for the Predictor: This table explores three different architectures employed as predictors: Multilayer Perceptron (MLP), Transformer-encoder block (Trans.), and Mix-MLP. We evaluate these architectures based on their top-100 matching rates and classification accuracy across various keep rates, ranging from 100% to 50%. Token pruning is executed at the 4th layer of CLIP’s ViT, aiming to assess the effectiveness of each architecture in maintaining accuracy while managing token redundancy.

our predictor can adapt to different layers within the network, accurately estimating rankings, and identifying redundant tokens across various depths. Specifically, the minimal variation in performance across different pruning configurations indicates that our approach maintains the predictor’s ability regardless of the specific layers targeted for token reduction.



**Figure 5.** This figure compares the classification accuracy between the CLS attention method and our Patch Ranking approach, both without fine-tuning post-token pruning. CLS attention employs CLS attention weights to rank tokens, whereas Patch Ranking utilizes the Feature Preservation Score for this purpose. Token removal occurs at the first layer of CLIP’s ViT. We present classification accuracy across different keep rates, ranging from 100% to 50%, highlighting the differential impact of each method on model performance as the number of pruned tokens increases.

Pruning Locations	Caltech101	OxfordPets	Flowers102	Food101	FGVCAircraft	DTD	UCF101	Avg.
2, 3, 4, 5	94.4	92.3	94.3	82.0	37.9	67.6	81.7	78.6
4, 5, 6, 7	94.3	92.1	95.3	82.2	39.5	68.1	82.0	79.1
1, 3, 5, 7	94.8	91.6	94.4	82.0	39.0	67.8	81.8	78.9
4, 6, 8, 10	95.3	91.4	94.5	83.0	40.0	68.4	83.0	<b>79.2</b>

**Table 7.** Performance analysis across different pruning locations: In this experiment, we maintained a keep rate of 60% and progressively pruned equal quantities of patch tokens at four distinct layers within CLIP’s ViT. We examined four different combinations of pruning locations to evaluate how varying the pruning layers within the network layers affects overall model performance.