TRIPS: Efficient Vision-and-Language Pre-training with Text-Relevant Image Patch Selection

Chaoya Jiang[1], **Haiyang Xu** [2], **Chenliang Li** [2], **Ming Yan** [2]
Wei Ye [1*], **Shikun Zhang** [1*], **Bin Bi** [2], **Songfang Huang** [2]

[1] National Engineering Research Center for Software Engineering, Peking University
[2] DAMO Academy, Alibaba Group
{jiangchaoya,wye, zhangsk}@pku.edu.cn,
{shuofeng.xhy,ym119608,lcl193798,b.bi, songfang.hsf}@alibaba-inc.com

Abstract

Vision Transformers (ViTs) have been widely used in large-scale Vision and Language Pre-training (VLP) models. Though previous VLP works have proved the effectiveness of ViTs, they still suffer from computational efficiency brought by the long visual sequence. To tackle this problem, in this paper, we propose an efficient vision-and-language pre-training model with **T**ext-**R**elevant **I**mage **P**atch **S**election, namely TRIPS, which reduces the visual sequence progressively with a text-guided patch-selection layer in the visual backbone for efficient training and inference. The patch-selection layer can dynamically compute text-dependent visual attention to identify the attentive image tokens with text guidance and fuse inattentive ones in an end-to-end manner. Meanwhile, TRIPS does not introduce extra parameters to ViTs. Experimental results on a variety of popular benchmark datasets demonstrate that TRIPS gain a speedup of 40% over previous similar VLP models, yet with competitive or better downstream task performance.

1 Introduction

In recent years, Vision-Language Pre-training (VLP) (Tan and Bansal, 2019; Chen et al., 2019; Lu et al., 2019; Huang et al., 2020; Su et al., 2020; Li et al., 2020; Chen et al., 2020; Zhou et al., 2020; Li et al., 2021b; Yu et al., 2021; Li et al., 2022) has developed at an astonishing rate and become a prevalent paradigm to tackle VL tasks. Traditional VLP models (Tan and Bansal, 2019; Chen et al., 2019; Lu et al., 2019; Li et al., 2020) utilize pre-trained object detectors (Ren et al., 2015; Redmon et al., 2016; He et al., 2017) to extract region-based image features but suffer from extensive annotation and expensive computation of object detector training. Inspired by the success of the Vision Transformer (ViT) (Dosovitskiy et al., 2021) and its variants (Liu et al., 2021; Wu et al., 2021; Wang

et al., 2021a) in computer vision field, more recent VLP models (Li et al., 2021b; Radford et al., 2021; Kim et al., 2021; Wang et al., 2021b; Singh et al., 2021) have adopted ViT as the visual encoder or cross-modal fusion encoder without using region features from the pre-trained object detector.

However, these ViT-based VLP methods are required to model long visual sequences from high-resolution images for good vision understanding, with quadratic computational complexity to the length of the visual sequence. Moreover, recent efforts (Dosovitskiy et al., 2021; Touvron et al., 2021) have also begun to explore vision-language foundation models, which scale up the model and data size. This raises the necessity to decrease the high computational cost of ViT-based VLP models. As shown in Figure 1(a), we find that removing the inattentive patch tokens of the image [CLS] token will generally not affect the result of Visual Question Answering (VQA) (Agrawal et al., 2015) prediction. Based on the observation, we conjecture that not all image tokens in the visual encoder contribute positively to the final prediction result of VLP models, and large numbers of redundant image tokens exist.

There have been some recent studies (Rao et al., 2021; Xu et al., 2021b; Zong et al., 2021; Liang et al., 2022) focusing on ViT model accelerations by reducing unrelated image tokens. However, these methods are specially designed for computer vision tasks (e.g., image recognition) and remove the redundant tokens based on visual semantics, ignoring the aligned knowledge in text modality, and thus are not suitable for VL tasks. As shown in Figure 1(b), without the guidance of text knowledge, directly removing patch tokens based on the image [CLS] token will lead to a wrong answer. This observation motivates us to reduce the image tokens by fusing the less informative patch tokens with the guidance of aligned text context.

In this work, we propose an efficient VLP model
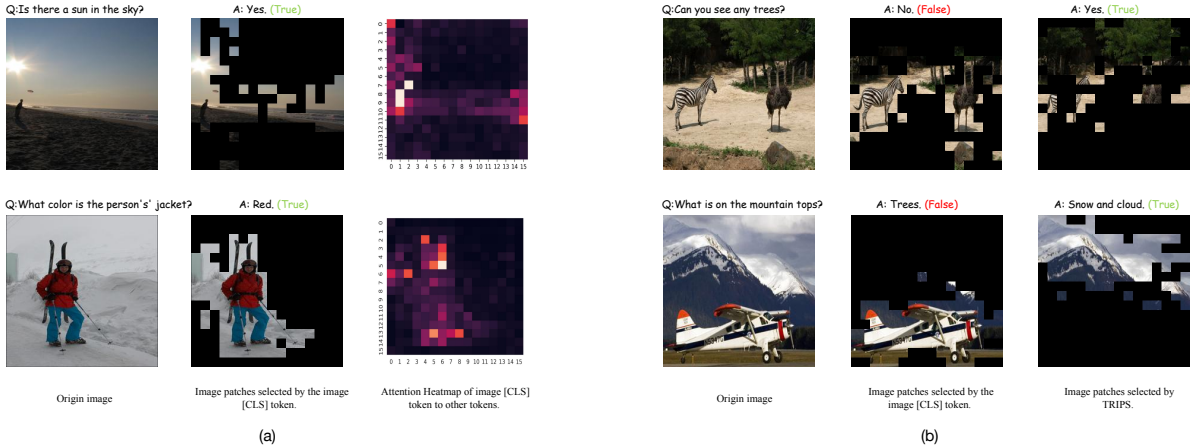
---

*corresponding author.

Figure 1: The sub-figure (a) shows the VQA cases of the ALBEF (Li et al., 2021b) finetuned on the VQA task, in which the input image tokens are directly selected with the guidance of attention weight of the image [CLS] token to other image tokens. We visualize the attention distribution of the image [CLS] token. As we can see, the [CLS] token attention naturally focuses on the objects in images and ignores the image backgrounds. If the question is about the objects in the images, we can still get the correct prediction. The sub-figure (b) shows the VQA predictions comparison between ALBEF, which directly selects image tokens with the guidance of the image [CLS] token, and our model TRIPS. As we can see, the questions are about the image backgrounds, and the former predicts the wrong answers, yet the latter can give the correct answer as it preserves the text-relevant image tokens.

with Text-Relevant Image Patch Selection (TRIPS), to progressively reduce the redundant image tokens with text guidance. TRIPS selects text-consistent image tokens through the text-aware patch-selection layer, reducing the computational cost of the visual encoding and cross-modal fusion. The patch-selection layer can preserve the attentive image tokens with text guidance and fuse the inattentive tokens into a single one by dynamically computing text-dependent visual attention, in an end-to-end way. In this way, we gradually decrease the number of image tokens as the visual backbone goes deeper to alleviate the computational cost of the visual encoder while improving the efficiency of cross-modal fusion due to the reduction of visual sequences. Besides, the model efficiency can be flexibly controlled via the keep rate of image tokens in the patch-selection layer where no additional parameters are introduced. We evaluate TRIPS on various representative VL tasks, including visual question answering, natural language visual reasoning, and cross-modal retrieval. The efficiency of TRIPS can be increased by 40% compared to previous similar VLP models, yet with competitive or better downstream task performance. For instance, TRIPS can speed up the baseline model 40.98% and even improve by 0.1 on the VQA test-dev and 0.2 on the NLVR Dev with the same experimental settings (see Table 6). Furthermore, by increasing the input image resolution with the same compu-

tational cost, TRIPS can improve 0.4 on the VQA test-dev and 0.6 on the NLVR Dev. Our contributions can be summarized as three-fold:

- We propose an efficient vision-and-language pre-training model with Text-Relevant Image Patch Selection (TRIPS). As far as we know, this is the first exploration that decreases the computational cost of VLP models by reducing image tokens with the help of linguistic context.

- We propose a text-relevant patch-selection layer, which can dynamically compute text-dependent visual attention to identify the attentive image tokens and fuse inattentive ones with text guidance in an end-to-end manner.

- Extensive experiments indicate that our TRIPS can boost the VLP model efficiency at a lower computational cost than the un-accelerated baseline model. Furthermore, by increasing the input image resolution, TRIPS benefits from taking more image tokens to achieve better performance without increasing computational costs.

2 Related Work

2.1 Vision-Language pre-training

Previous Vision-Language pre-training (VLP) methods (Lu et al., 2019; Li et al., 2019; Tan and

Bansal, 2019; Li et al., 2020; Chen et al., 2019; Yu et al., 2021) mainly take a two-step training pipeline approach, which first extracts visual features by a pre-trained object detector and then trains the cross-modal pre-training model to align text and visual features. Some region-based methods reduce the computation cost with the lightweight model architecture (Wang et al., 2020a; Gan et al.) and knowledge distillation (Fang et al., 2021). The main challenge for these methods is to balance effectiveness and efficiency. More recent CNN-based (Huang et al., 2020; Xu et al., 2021a) and ViTs-based (Li et al., 2021a; Kim et al., 2021; Radford et al., 2021; Wang et al., 2021b) methods (especially the patch-based ViT) removes the complicated object detector in feature extraction to conducts end-to-end VL learning. However, there is no work decreasing the high computational cost of ViT-based VLP models. In this work, we propose a novel method to decrease the computation cost of VLP models, which reduces the visual sequence progressively with a text-guided patch-selection layer in the visual backbone for efficient training and inference.

2.2 ViTs Acceleration

To accelerate the computation of the transformer(Vaswani et al., 2017) based model, many studies focus on proposing more efficient attention mechanisms (Wang et al., 2020b; Kitaev et al., 2020; Choromanski et al., 2021) or compress Transformer structures (Liu et al., 2021; Heo et al., 2021; Wang et al., 2021a). Recently, some approaches have focused on accelerating ViTs by reducing the number of tokens involved in the inference of ViTs. For example, to expedite ViTs, Ryoo et al. (2021) proposed TokenLearner, in which a relatively small amount of tokens are learned by aggregating the entire feature map weighted by dynamic attention. Rao et al. (2021) introduces a method to reduce tokens for a fully trained ViT, where an extra learnable neural network is added to ViT to select a subset of tokens. Liang et al. (2022) propose to reduce the computational overhead of inference by proposing a token reorganization method to reduce and reorganize image tokens progressively. However, those methods are unsuitable for VLP as they reduce the image tokens without considering the text context.

3 Method

In this section, we will first introduce TRIPS with the acceleration module of the text-relevant patch-selection layer, and then give the details of the pre-training objectives.

3.1 Model Architecture

As shown in Figure 2 (a), TRIPS contains a visual encoder with text-relevant patch-selection layer, a text encoder, and a multimodal fusion encoder. The visual encoder takes a Vision Transformer (ViT), where text-relevant patch-selection layers are used to progressively reduce and reorganize image tokens, namely ViT-TRIPS. The text encoder adopts $BERT_{base}$ transformer (Devlin et al., 2019). Similar with (Li et al., 2021b), the multimodal fusion encoder is a transformer encoder that performs the cross-modal interaction and fusion through a cross-attention mechanism.

Formally, given an input image-text pair, we first feed the input text to the text encoder and represent it as a sequence of embeddings $T = \{t_{cls}, t_1, t_2, \cdots, t_m\}$, where t_{cls} is the embedding of the text [CLS] token to summarize the input text. Then, we divide the input image into patches $P = \{p_{cls}, p_1, p_2, \cdots, p_u\}$, and encode them with the image encoder ViT-TRIPS. It takes the text [CLS] embedding t_{cls} and image patches $\{p_{cls}, p_1, p_2, \cdots, p_u\}$ as input, and outputs the image sequence $V = \{v_{cls}, v_1, v_2, \cdots, v_e\}$. Note that $e < u$, since we apply the text-relevant patch-selection layer to select the text-aware image tokens and fuse the redundant tokens, allowing us to reduce total visual sequence length for efficiency. Finally, the text features $\{t_{cls}, t_1, t_2, \cdots, t_m\}$ and the image features $\{v_{cls}, v_1, v_2, \cdots, v_e\}$ encoded by the image encoder are fused by cross attention at each layer of the multimodal encoder as in AL-BEF (Li et al., 2021b). The output of the multimodal encoder is used to pre-train and finetune downstream tasks.

3.2 Text-Relevant Image Patch Selection

Existing works in computer vision (Rao et al., 2021; Liang et al., 2022) select patches by using only the image [CLS] token from the ViT backbone. However, as shown in Figure 1 (b), the selection of image tokens in cross-modal tasks is closely related to textual context, and different texts for a single image may focus on different parts of the visual content. Selecting the image tokens with the guid-
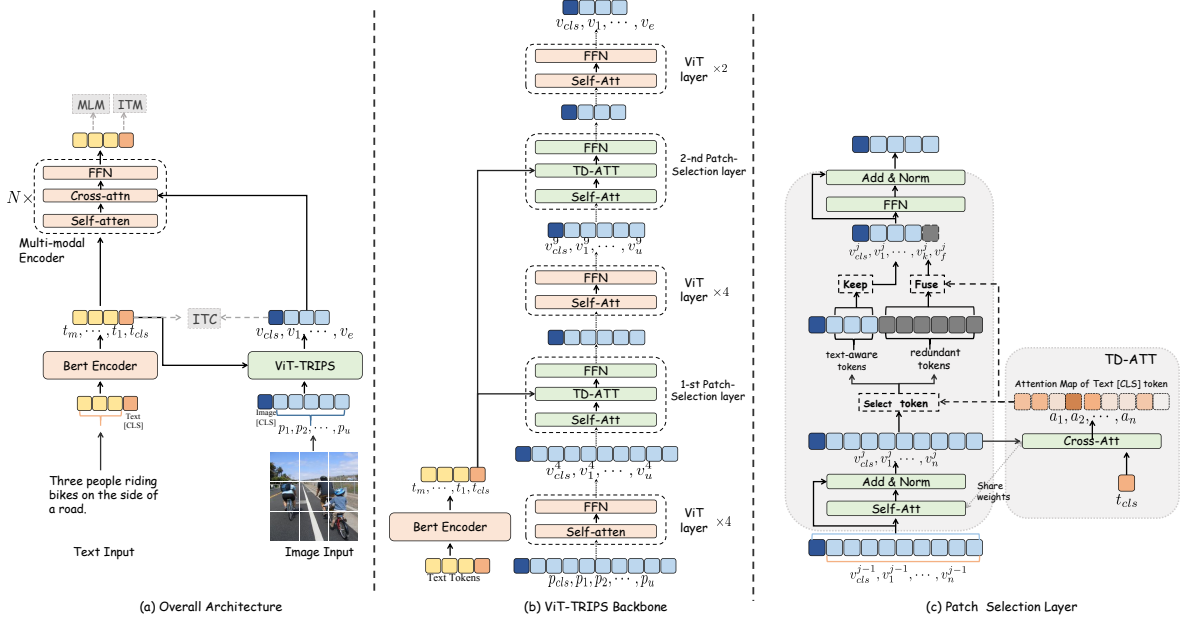
Figure 2: (a) The overall architecture of our VLP model (TRIPS) in this paper. (b) An overview of the ViT applied with a Text-Relevant Image Patch Selection module (ViT-TRIPS). Suppose the ViT-TRIPS contain 12 layers, with the 5-th and 10-th layer as the patch selection layers. (c) Illustration of Text-Relevant Image Patch-Selection layer.

ance of aligned textual content can help the VLP model focus on the key parts of the image for more effective and efficient cross-modal fusion. Here, we present a text-relevant patch-selection layer that can dynamically select the image patches with the guidance of textual input, yet with no additional parameters introduced.

As shown in Figure 2 (b), for a ViT with L standard Transformer layers and t patch-selection layers in total, the interval length is obtained as $s = L/(t + 1)$. Then, we choose the layer index $j = i * s + 1$ as the i_{th} patch-selection layer, so that patch-selection layers are uniformly inserted into the ViT-TRIPS backbone. In each patch-selection layer, as shown in Figure 2(c), we adopt standard self-attention (SA), Text-aware Dynamic Attention (TD-ATT), and Inattentive Token Fusion (ITF) modules to progressively reduce image tokens.

Specifically, for the i_{th} patch-selection layer, image features $v^{j-1} = \{v_{cls}^{j-1}, v_1^{j-1}, \cdots, v_n^{j-1}\}$ are first fed to the j_{th} visual self-attention layer:

$$v^j = LN(SA(v^{j-1}) + v^{j-1}) \qquad (1)$$

where LN is short for layer normalization, and n is the number of patch tokens in $j-1$ visual transformer layer.

Next, we will illustrate how to use the Text-aware Dynamic Attention mechanism (TD-ATT) to select the text-aware image patch tokens. The text [CLS] embedding t_{cls} is linearly projected to the

query vector denoted as q_{text} by the shared query linear layer of the j_{th} visual self-attention layer. We compute the text-to-image attention feature map excluding the image [CLS] token as follows:

$$a_{cls} = softmax(\frac{q_{text} \cdot v^j[1:]^T}{\sqrt{d}}) \qquad (2)$$

We identify and preserve the attentive image tokens corresponding to the k largest elements in the attention map $a_{cls} = \{a_1, ..a_n\}$, where $k = n \times r$, and r is the keep rate of this layer. The selected image tokens are kept and the un-selected image tokens are further fused by an inattentive token fusion operation ITF.

The remaining inattentive patch tokens $\{v_{z_1}, v_{z_2}, \cdots, v_{z_{n-k}}\}$ are treated as text-irrelevant tokens. However, the fixed keep rate may remove some useful tokens, so we fuse inattentive tokens to one token v_f by a weighted sum operation to supplement attentive ones as follow:

$$v_f = \sum_{i=1}^{n-k} a_{cls,z_i} \cdot \hat{v}_{z_i} \qquad (3)$$

After fusing the inattentive patch tokens, we reconstruct the j_{th} visual sequence as $v^j = \left[v_{cls}^j, v_1^j, \cdots, v_k^j, v_f^j\right]$, which consists of the image [CLS] token embedding, the selected text-aware image patch embedding, and fused inattentive patch embedding. Then the new visual sequence is fed to the feed-forward network (FFN).

Extension to Single-stream Model. The proposed Text-Relevant Image Patch Selection layer can be also extended to the single-stream model (TRIPS-S), which employs the [CLS] token of the multimodal encoder to preserve the attentive image tokens and fuses the inattentive tokens to speed up the training and inference. While parameter-efficient, it may be difficult to learn uni-modal and multi-modal interactions concurrently. Therefore, its performance lags behind two-stream performance on downstream VL tasks.

3.3 Pre-training Objectives

We pre-train our model with three standard objectives: Image-Text Contrastive learning (ITC), Image-Text Matching (ITM), and Masked Language Modeling (MLM). These pre-training tasks are optimized jointly.

Image-text Contrastive (ITC) For TRIPS, We follow the (Li et al., 2021b) and apply ITC to align the image representation and text representation from the unimodal encoders. For the image, the image feature corresponding to the image [CLS] token is chosen as the image representation. For the text, the text token feature corresponding to the text [CLS] token is the text representation.

Image-Text Matching (ITM) The goal of image-text matching is to predict whether the input image and text are matched. We follow the design of (Li et al., 2021b) and select hard negative image-text pairs based on the contrastive text-image similarity. We take the text [CLS] embedding of the multi-modal encoder's output as the joint representation, followed by a Multi-Layer Perceptron (MLP) layer for prediction.

Masked Language Modeling (MLM) The task setup is basically the same as in BERT (Devlin et al., 2019), where we randomly mask 15% of tokens in text and the model is asked to predict these masked words with the cross-modal representations.

4 Experiment Settings

4.1 Implementation Details

We pre-train the TRIPS for 30 epochs with a total batch size of 512 on 8 NVIDIA V100 GPUs. We initialize the visual encoder by CLIP (ViT-B/16) (Radford et al., 2021) pre-trained on 400M noisy image-text pairs and we use the AdamW (Loshchilov and Hutter, 2019) optimizer with a weight decay of 1e-2. The learning rate is warmed-up to 1e-5 (ViT-B/16) and 1e-4 (BERT_{base}) in the first 1000 iterations. During pre-training, we take the image with the resolution of 256×256 as input and increase the image resolution during finetuning. We use a 6-layer Transformer for both the text encoder and the cross-modal fusion network. As (Li et al., 2021b), the text encoder is initialized using the first 6 layers of the BERT_{base} (Devlin et al., 2019) model and the cross-modal network is initialized using the last 6 layers of the BERT_{base}. For the image-text contrastive learning, the queue size is set as 65,536 and the momentum coefficient is set as 0.995.

4.2 Pre-training Datasets

We construct our pre-training data using two web datasets (Conceptual Captions (Sharma et al., 2018), SBU Captions (Ordonez et al., 2011)) and two in-domain datasets (MSCOCO (Lin et al., 2014) and Visual Genome (Krishna et al., 2016)). The total number of unique images is 4.0M, and the number of image-text pairs is 5.1M.

5 Experiment Results

5.1 Main Result

We evaluate our model TRIPS on three widely explored vision-language downstream tasks: Visual Question Answering (VQA), Cross-modal Retrieval, and Natural Language for Visual Reasoning (NLVR). For the proposed model TRIPS, we take the 5-th and 10-th as the patch-selection layer in ViT encoder and set the keep rate of each layer to 70%, achieving the desired trade-off between the downstream task performance and the model inference speed. Details of the datasets and fine-tuning hyperparameters are in Appendix B. Details of the comparison methods are in Appendix A.

5.1.1 Visual Question Answering

The VQA task (Agrawal et al., 2015) requires the model to answer natural language questions given an image. We follow (Li et al., 2021b) and consider VQA as an answer generation problem. We report test-dev and test-std scores by submitting our results to the evaluation server[1] in Table 2. Compared with the VLP baselines, TRIPS can improve the performance on the VQA task. The results demonstrate the effectiveness of TRIPS.

---

[1]https://eval.ai/web/challenges/challenge-page/830/overview

| Models | Flickr30K (1K test set) | | | | | | MSCOCO (5K test set) | | | | | |
| | TR | | | IR | | | TR | | | IR | | |
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
E2E-VLP	86.2	97.5	98.92	73.6	92.4	96.0	-	-	-	-	-	-
UNITER	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
OSCAR	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
VinVL	-	-	-	-	-	-	74.6	92.6	96.3	58.1	83.2	90.1
ViLT	83.5	96.7	98.6	64.4	88.7	93.8	61.5	86.3	92.7	42.7	72.9	83.1
ALBEF	94.3	99.4	99.8	82.8	96.7	98.4	73.1	91.4	96.0	56.8	81.5	89.2
ALBEF-C	96.1	99.8	**100.0**	**86.1**	97.8	98.9	77.8	94.3	97.4	60.3	**84.7**	91.0
TRIPS	**96.3**	99.8	**100.0**	85.8	**98.1**	**99.0**	**78.1**	**94.8**	**97.6**	**61.3**	84.3	**91.4**

Table 1: Image-text retrieval results on Flickr30K and COCO datasets. −indicate that the result is unreported.
.

| Models | VQA | | NLVR | |
	Test-dev	Test-std	dev	Test-P
ViLBERT	70.55	-	-	-
LXMER	72.42	-	74.90	74.50
UNITER	72.70	72.91	77.18	77.85
OSCAR	73.16	73.44	78.07	78.36
VinVL	75.95	76.12	82.05	83.08
E2E-VLP	73.25	73.67	77.25	77.96
ViLT	71.26	-	75.70	76.13
ALBEF	74.54	74.70	80.24	80.50
ALBEF-C	76.12	76.32	82.21	83.11
TRIPS	**76.23**	**76.48**	**82.35**	**83.34**

Table 2: Evaluation Results on VQA test set and NLVR2. − indicates that the result is unreported. ALBEF-C is our implementation of ALBEF with a visual encoder initialized by CLIP (ViT-B/16). TRIPS takes the same model architecture and experimental setting as ALBEF-C.

Models	FLOPs	Throughput	Latency
UNITER	949.90	6.42	870ms
OSCAR	956.40	6.35	860ms
VinVL	1023.30	7.32	640ms
E2E-VLP	144.3	80.23	70ms
ViLT	55.40	247.530	15ms
ALBEF-C	36.63	197.52	21ms
TRIPS	**20.89**	**343.05**	**11ms**

Table 3: The comparison of the efficiency of different models. Here, we report FLOPs, throughput and latency. The FLOPs results of the baselines come from (Kim et al., 2021). Since FLOPs are proportional to input size, for a fair comparison, we keep same the input size with (Kim et al., 2021) which is 197 for image patches length and 40 for text tokens length. We keep the same setting when calculating throughput and latency.

5.1.2 Natural Language for Visual Reasoning

The NLVR2 (Suhr et al., 2019) task requires the model to predict whether a sentence describes a pair of images which is a binary classification task. We follow (Li et al., 2021b) and use two cross-attention layers to process the two input images, and their outputs are merged and fed to a Feed Forward Network (FFN). An MLP classifier is then applied to the output embedding of the text [CLS] token. As shown in Table 2, TRIPS has a better performance than existing VLP methods.

5.1.3 Image-Text Retrieval

We conduct experiments for both image-to-text retrieval (TR) and text-to-image retrieval (IR) on MSCOCO (Lin et al., 2014) and Flickr30K (Plum-

mer et al., 2015) datasets. We jointly optimize the ITC loss and the ITM loss during fine-tuning. The results are reported in Table 1. As shown in Table 1, the experimental results show that our model gets comparable performance with other VLP baselines.

5.2 Efficiency of TRIPS

To investigate the efficiency of Text-Relevant Image Patch Selection, we first compare the computational complexity of different models. We report the Floating Point Operations Per second (FLOPs), a widely used evaluation metric for model computational complexity. Besides, to evaluate the computational speed of our model, we compare the throughput and latency of different models. We use a Xeon Platinum 8163 CPU and an NVIDIA V100 GPU to calculate the latency and throughput. As shown in Table 3, TRIPS not only has the

Locations	Keep rates	Overall Keep rate	VQA test-dev	NLVR dev	FLOPs (G)	Throughput
-	-	100%	76.12	82.35	76.03	79.32
[2]	50%	50%	75.60	81.26	42.17	161.26
[10]	50%	50%	76.19	81.84	63.84	96.66
[2,4]	50%	%25	74.21	79.93	**28.00**	**238.41**
[4,8]	50%	%25	74.93	80.65	38.82	165.30
[5,10]	50%	%25	75.29	81.17	44.22	143.37
[6,12]	50%	%25	75.48	81.04	49.63	126.46
[2,4]	70%	%49	74.87	80.45	43.96	153.92
[4,8]	70%	%49	75.94	81.96	51.72	125.55
[5,10]	70%	%49	76.23	82.35	55.60	115.01
[6,12]	70%	%49	**76.24**	**82.48**	59.48	106.07
[2,6,10]	70%	%34	74.92	80.61	42.66	156.13
[3,6,9]	70%	%34	75.09	80.75	43.49	151.40
[4,8,12]	70%	%34	75.23	81.12	49.74	129.81

Table 4: Results of pre-training and finetuning TRIPS with different locations and keep rates. we report the text-dev score results of VQA, FLOPs and Throughput. In this table, we set the input image size to 384×384 and the length of input text is 40. The throughput (image-text/s) is measured on an NVIDIA V100 GPU using the largest possible batch size for our model.

lowest computational complexity (e.g., 20.89 of FLOPs) but also the fastest computational speed (e.g., 343.05 of Throughput and 11ms of Latency).

5.3 The Impact of Location and Keep Rate

To validate the impact of patch-selection layer location and keep rate on the efficiency and effectiveness of the model, we train TRIPS with different patch-selection locations and numbers of selected tokens. The results in Table 4 demonstrate two findings. First, moving the patch-selection layer into shallower layers reduces computational complexity but deteriorates the accuracy. For example, the accuracy drops considerably with the remarkable throughput increasing when the patch-selection layer is placed before the third layer (i.e., at the second layer). A possible explanation is that the attention maps between the text [CLS] embedding and the input tokens can be unreliable during the early processing of input tokens in shallow layers. Second, too many image tokens fused in the patch-selection layer will considerably drop the downstream task performance. For example, if we take the 2-nd and 4-th layers in ViT as the patch-selection layer and set the keep rate to 50%, the performance will decrease to 74.21 on the VQA task, compared to 76.12 of the model without the patch-selection layer.

5.4 Finetuning on Higher Resolution Images

We can control the computational cost by fusing different numbers of inattentive tokens. Therefore, we finetune TRIPS on the VQA and NLVR tasks, which take images with varying resolutions as input. We report the results in Table 5. The experimental results show that by increasing the input image resolution, we can facilitate the model by taking more image tokens to gain better performance. For example, by finetuning TRIPS with the images of 456×456, we can achieve the score of 76.54 on VQA, outperforming the baseline finetuned with images of 384×384 yet keeping similar computational complexity.

5.5 Effectiveness of Text-Relevant Image Patch Selection

To verify the effectiveness of Text-Relevant Image Patch Selection, we first implement the single-stream model TRIPS-S as we present in subsection 3.2. Then, we examine the downstream task performance, computational complexity, and inference speed of TRIPS and TRIPS-S (both with and without Text-Relevant Image Patch Selection). The results are shown in Table 6, and we find that for both TRIPS and TRIPS-S, we can see a consistent improvement in the inference speed and downstream task performance by incorporating the text-relevant image patch selection mechanism. These results suggest that the proposed image patch se-

Selection Layer	Keep rate	image size	VQA test-dev	NLVR dev	FLOPs(G)	Troughout
-	-	384×384	76.12	82.35	76.03	79.32
[5,10]	70%	224×224	75.23	80.83	**20.89**	**343.05**
[5,10]	70%	256×256	75.84	81.24	26.61	258.03
[5,10]	70%	304×304	76.13	81.72	36.62	189.07
[5,10]	70%	384×384	76.23	82.35	55.60	115.01
[5,10]	70%	456×456	**76.54**	**83.02**	74.83	81.0

Table 5: Results of TRIPS finetuning on VQA and NLVR task with different resolution images. When calculating FLOPs, the input length of the text is kept at 40 and the settings for calculating throughput are the same as Table4.

Models	Selection Layer	Keep rate	FLOPs(G)	Throughput	VQA		NLVR	
					Test-dev	Test-std	dev	Test-P
TRIPS-S	[5,10]	70%	59.42	**216.74**	71.48	71.52	75.89	76.47
-w/o PS	-	-	104.42	135.25	71.26	71.29	75.18	76.23
TRIPS	[5,10]	70%	**55.60**	115.01	**76.23**	**76.48**	**82.35**	**83.34**
-w/o PS	-	-	76.03	79.32	76.12	76.32	82.21	83.11

Table 6: Ablation results of TRIPS and TRIPS-S on VQA test set and NLVR2.w/o PS indicates that we remove the patch-selection module in ViT. The setting for calculating FLOPs and throughput is the same as Table4.

lection mechanism is not only efficient but also effective. Notably, compared with the dual-stream model TRIPS, TRIPS-S is faster in inference due to the parameter efficiency of the single-stream model. However, its performance lags behind state-of-the-art performance on downstream VL tasks.

5.6 Ablation Study

model	VQA	FLOPs(G)	Throughput
TRIPS	76.23	57.20	111.83
-w/o ITF	75.92	57.15	112.04
-w/o TD-ATT	75.23	56.4	117.21

Table 7: The result of ablations. We finetune TRIPS on VQA and report test-dev results. The setting for calculating FLOPs and throughput is the same as Table 4. The same with the settings of TRIPS we present in the main results, we select the 5-th and 10-th as the patch selection layer, and each layer will keep 70% image tokens.

We also perform ablation studies to investigate the effects of inattentive image token fusing and Text-aware Dynamic Attention. In Table 7, w/o ITF indicates the inattentive tokens are directly discarded without fusing. As shown in Table 7, fusing inattentive tokens outperforms the model without inattentive tokens. Although the improvement is small, there is no additional computational over-

head introduced. We also verify the impacts of Text-aware Dynamic Attention (TD-ATT). Specifically, w/o TD-ATT indicates that we remove the TD-ATT in the patch-selection layer and select the image tokens based on the image [CLS] token. As shown in Table 7, selecting the image patch tokens with the image [CLS] token without considering the linguistic context will degrade the model's performance. This result supports our motivation that directly removing patch tokens based on image [CLS] without incorporating the text knowledge is unsuitable for VLP models.

5.7 Visualization

The proposed TRIPS accelerates VLP by a novel patch selection module that selects the text-consistent image tokens in the vision backbone and preserves the attentive image tokens. To further investigate the interpretability of our model, we visualize the procedure of text-relevant image path selection in Figure 3. It can be seen that as the network deepens, the inattentive tokens are gradually removed or fused, while the text-relevant tokens are selected and preserved. Besides, we present more visualization results in Figure 4 to show the effectiveness of the text-relevant image patch selection module. We take different text words as the input and visualize the text-aware image patches selected by the text-relevant image patch selection

module. As shown in Figure 4, the selected image patches are highly relevant to the query texts and thus enable our model to make a correct prediction.

6 Conclusion

We have presented TRIPS, an efficient VLP model with **T**ext-**R**elevant **I**mage **P**atch **S**election to progressively reduce the redundant image tokens with text guidance. TRIPS introduces a novel patch selection module to select the text-consistent image tokens in the vision backbone, which preserve the attentive image tokens with text guidance and fuses the inattentive tokens into one token by dynamically computing text-dependent visual attention in an end-to-end way. The experiment shows our method not only decreases the computation cost of VLP but also improves the efficiency of cross-modal fusion due to the reduction of visual sequences, while keeping or even improving the performance of downstream image-text tasks.

7 Limitations

Despite the effectiveness and efficiency of TRIPS across a wide range of downstream image-text tasks, our model still has several limitations. First, in our current settings, we pre-train TRIPS with only 4M image-text pairs. It is unclear how well the performance will be if we pre-train TRIPS on a larger pre-training dataset with other available data types, such as text-only, image-only data, and some labeled data. Second, we provide a novel perspective for the efficient training and inference of VLP models, which reduces the visual sequence progressively with a text-guided patch-selection layer in the visual backbone. In our method, the number of selected text-aware image patches in each patch-selection layer is fixed, and there can be a more ingenious technical design that can dynamically select different numbers of image patches.

8 Acknowledgement

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4–31.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *ArXiv*, abs/1909.11740.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. 2021. Rethinking attention with performers. *ArXiv*, abs/2009.14794.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Compressing visual-linguistic model via knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1428–1438.

Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. Playing lottery tickets with vision and language.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. 2021. Rethinking spatial dimensions of vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11916–11925.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *ArXiv*, abs/2004.00849.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *ArXiv*, abs/2001.04451.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73.

Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. 2022. mplug: Effective and efficient vision-language learning by cross-modal skip-connections.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. 2021b. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

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

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *NIPS*.

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74–93.

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. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. 2021. Tokenlearner: Adaptive space-time tokenization for videos. In *NeurIPS*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2021. Flava: A foundational language and vision alignment model. *ArXiv*, abs/2112.04482.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. *ArXiv*, abs/1908.08530.

Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. *ArXiv*, abs/1811.00491.

Hao Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *ArXiv*, abs/1908.07490.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020a. Minivlm: A smaller and faster vision-language model. *arXiv preprint arXiv:2012.06946*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021a. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558.

Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. 2021b. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *ArXiv*, abs/2111.02358.

Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31.

Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. 2021a. E2e-vlp: End-to-end vision-language pre-training enhanced by visual learning. *ArXiv*, abs/2106.01804.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. 2021b. Evo-vit: Slow-fast token evolution for dynamic vision transformer. *ArXiv*, abs/2108.01390.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. In *AAAI*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Making visual representations matter in vision-language models. *CoRR*, abs/2101.00529.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. *ArXiv*, abs/1909.11059.

Zhuofan Zong, Kunchang Li, Guanglu Song, Yali Wang, Y. Qiao, Biao Leng, and Yu Liu. 2021. Self-slimmed vision transformer. *ArXiv*, abs/2111.12624.

Caption	Input Image	Selected Patches-1	Selected Patches-2
There was a man at the beach			
The boy performs a trick with the bicycle			
A woman standing beside a clean kitchen island			
A nun riding a very tallk bike next to a purple bus			
Woman wearing scarf around neck			

Figure 3: The visualization of the selected text-aware image patches in different selection layers. We set the 5-th and 10-th layers in the vision backbone as the patch-selection layer and we keep 70% image patches in each layer.
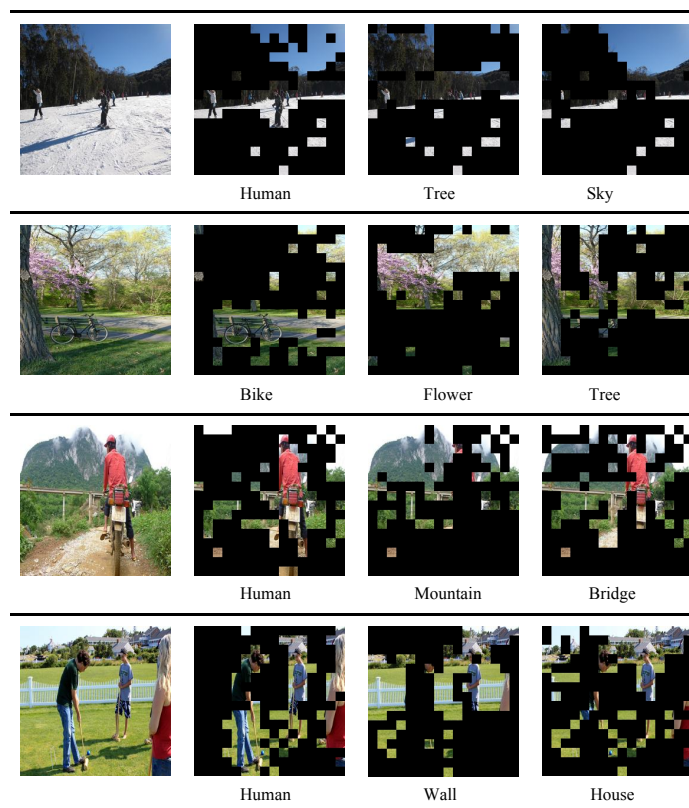

Figure 4: The visualization of the selected text-aware image patches with different text words. We set the 5-th and 10-th layers in the vision backbone as the patch-selection layer and we keep 70% image patches in each layer. We visualize the selected text-aware image patches output by the 10-th layer.

A Comparison Methods

LXMERT (Tan and Bansal, 2019): is the first two-stream region-based VLP model, which consists of an object relationship encoder, a language encoder and a cross-modality encoder.

E2E-VLP (Xu et al., 2021a): proposes the first end-to-end VLP method for both V+L understanding and generation, with a unified Transformer encoder-decoder architecture.

VILT (Kim et al., 2021): adopts linear projection and word embedding as the visual and textual encoders, and uses the visual transformer as the cross-modal encoder to align and fuse the features of both modalities in an end-to-end manner.

OSCAR (Li et al., 2020): proposes to use object tags detected in images as anchor points to the learning of cross-modal alignments.

VinVL (Zhang et al., 2021): pre-trains a large-scale object-attribute detection model with much larger amounts of supervised data to extract better region-based visual features.

ALBEF (Li et al., 2021b): adopts a contrastive loss to align the image and text representations, then fuses them through cross-modal attention in an end-to-end manner.

UNITER (Chen et al., 2019): proposes a new word-region alignment pre-training task via the use of optimal transport to help fine-grained alignment between words and image regions.

ViLBERT (Lu et al., 2019): proposes one of the first work that extend the BERT architecture to a multi-modal two-stream region-based VLP model.

B Downstream Task Details

We evaluate TRIPS on the three downstream vision-language tasks. The hyperparameters that we use for finetuning on the downstream tasks are listed in Table 8. Following (Li et al., 2021a), all tasks adopt RandAugment, AdamW optimizer with a weight decay of 0.05 and a cosine learning rate schedule. Next we introduce the dataset settings in detail.

Task	LR (ViT-B/BERT$_{base}$)	batch size	epochs
VQA	2e-5/5e-6	1024	8
Retrieval	1e-5/2e-6	256	5
NLVR2	5e-5/5e-6	256	15

Table 8: Finetuning hyperparameters for downstream tasks.

VQA. The VQA task (Agrawal et al., 2015) requires the model to answer natural language questions given an image. We conduct experiment on the VQA2.0 dataset (Agrawal et al., 2015), which contains 83k/41k/81k images for training/validation/test. Following (Li et al., 2021a), we use both training and validation splits for training, and incorporate additional training data from Visual Genome (Suhr et al., 2019).

Image-Text Retrieval. We conduct experiments for both image-to-text retrieval (TR) and text-to-image retrieval (IR) on COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015) datasets. We take the widely-used Karpathy split (Karpathy and Fei-Fei, 2015) for both COCO and Flickr30K. COCO contains 113k/5k/5k images for train/validation/test, and Flickr30K contains 29k/1k/1k images for train/validation/test.

NLVR2. The NLVR2 (Suhr et al., 2019) task requires the model to predict whether a sentence. We conduct experiments following the original train/val/test split in (Suhr et al., 2019).