TiP: Text-Informed Image Pruning for Efficient and Interpretable Vision Language Models

Anonymous EMNLP submission

Abstract

Large-scale vision language (VL) models use Transformers to perform cross-modal interactions between the input text and image. These cross-modal interactions are computationally expensive and memory-intensive due to the quadratic complexity of processing the input image. We present **TiP**: a **T**ext-informed **I**mage **P**runing method that progressively removes text-irrelevant portions of the input image, improving model inference speed and reducing memory footprint. We design several lightweight modules — token pruners — and add them to the cross-modal layers in a VL model to predict which image portions are salient. To train **TiP**, we introduce a text-informed contrastive learning technique that optimizes the representation similarity between the text and the salient text-relevant image portions predicted by the token pruners. Our neighbor-based continuity regularization loss encourages the pruners to select contiguous segments of the image as relevant. Our evaluation for two vision language models on three downstream VL tasks shows **TiP** prunes over 87% of input image data, thus increasing inference throughput by over 1.5x and reducing memory footprint by over 36%, while incurring less than a 1% accuracy drop. **TiP** is also interpretable by construction. [1]

1 Introduction

Large-scale vision language (VL) models (Dou et al., 2022; Wang et al., 2022; Zeng et al., 2021; Kim et al., 2021; Wang et al., 2021; Zhang et al., 2021) have shown substantial progress on many vision language tasks such as visual question answering, natural language visual reasoning, and visual entailment. However, state-of-the-art language and vision models are memory intensive and computationally expensive because they use multi-layer self-attention between many language and vision input tokens [2] with quadratic complexity. This inefficiency limits high-throughput cloud deployments and makes it infeasible to run on resource-constrained devices.

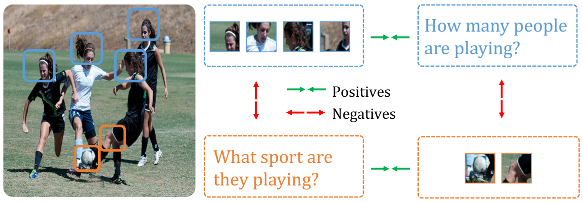

Figure 1: **TiP** applies image pruning to VL models via text-informed contrastive learning. **TiP** makes VL models run faster by progressively removing text-irrelevant image portions and extracts text-informed image portions to contrast with negative examples.

The key challenge in deep VL models is that these models need to process the entire image over all the layers. On the contrary, humans process natural language and visual world in a coarse-to-fine grained manner (Hegdé, 2008) and are able to selectively pay attention to key parts in a visual scene (Rensink, 2000). This selective attention mechanism helps summarize key information and makes humans process complex visual scenes more efficiently. For example, for the visual question "What sport are they playing?" in Figure 1, humans easily answer "soccer" by focusing on the bottom-center region of the image.

In this paper, we design **TiP**, a Text-informed Image Pruning framework, that progressively prunes image regions that are not related to the text and are unimportant to the VL task predictions. At the core of **TiP** is a set of simple and lightweight token pruners that predict which image tokens are pruned as the VL model forward computation proceeds. We jointly train the token pruner and the underlying VL model using a multi-task training objective that optimizes for the end task performance, learns

[1] Code is available at `anonymized_url`.

[2] small image patches

the task logits by distilling from the original VL model, and encourages the model to only keep text-grounded, continuous image regions.

To preserve text-grounded information in the input image, we design a contrastive loss to align the pruned image to the reference text as opposed to other texts. Since salient objects in images can span contiguous tokens of the image, we introduce a continuity prior to reduce the discontinuity of salient image tokens and encourage them to form continuous image regions. To reduce abrupt image information loss and improve the computational efficiency, we scatter the token pruners at different cross-modal layers in the VL model and prune the image tokens in a cascaded fashion. Fewer tokens are pruned in earlier layers. During inference too, we prune image tokens progressively and use salient tokens for cross-modal attention in subsequent layers. This leads to improved inference efficiency and inherent interpretability.

We evaluate **TiP** over two recent VL models ViLT (Kim et al., 2021) and METER (Dou et al., 2022) across three visual reasoning tasks: visual question answering (VQAv2; Goyal et al. 2017), natural language visual reasoing (NLVR2; Suhr et al. 2019), and visual entailement (SNLI-VE; Xie et al. 2019). Compared to baselines, **TiP** improves the model inference throughput by over 1.5x and reduces memory footprint by over 30% with minimal (less than 1%) accuracy loss. We quantitatively and qualitatively show **TiP** maintains grounding capability and provides more interpretable results. Our analysis indicates that original model distillation, text-informed contrastive loss and continuity prior contribute to the effectiveness of **TiP**.

2 Background and Overview

Vision Language Models. Figure 2 shows the backbone of a VL model consisting of a text encoder, an image encoder and a cross-modal encoder. The input sentence (e.g. a question or a statement) is first tokenized as text tokens and fed to the text encoder to create contextualized text representations. Similarly, the input image is projected into many small image patches, referred to as "image tokens", that are further contextualized by the image encoder. Finally, the cross-modal encoder takes the concatenated text and image tokens and fuses information between image and text modalities via Transformer-style (Vaswani et al., 2017) cross-attention interactions.

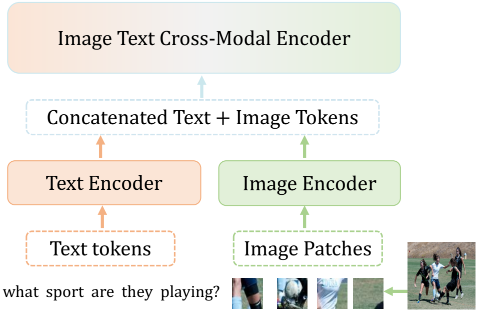

Image Text Cross-Modal Encoder
Concatenated Text + Image Tokens
Text Encoder
Image Encoder
Text tokens
Image Patches
what sport are they playing?

Figure 2: General architecture of vision language models. The input image is projected into many small image patches ("tokens") that are processed by the image encoder. Transformer-style cross-modal encoder attention between concantenated text and image tokens has quadratic time-complexity, which is computationally expensive. Both ViLT and METER models follow this pattern.

For many VL tasks, the number of tokens of the input image is an order of magnitude more than that of the input text — a visual question can have at most a dozen tokens but the associated image consists of a hundred image tokens. For example, for an image with a resolution of 384x384 and patch size of 16, the number of tokens is $(384/16)^2 = 576$.

Image Pruning for Efficiency. In this paper, we focus on pruning image tokens to improve computational efficiency of the model. However, naively removing a large percentage of the image tokens inside the cross-modal layers may cause abrupt image information loss, as the VL model is trained to build representations of the full image for the downstream task. For example, if the soccer region in Figure 1 gets pruned, the VL model is unlikely to output the answer "soccer" for the question "what sport are they playing?".

Our Solution: Text-Informed Image Pruning. Section 3.1 describes our approach. We design a set of token pruners that remove image tokens that are not *salient* to the end task, hence maintaining the task performance. We measure saliency of image tokens based on how much they are related to the input text and how important they are to the end task. **TiP** learns to align representations between the input text and text-grounded image regions via text-informed contrastive learning. Token pruners are incorporated in the cross-modal attention layers in a cascaded manner (dense to sparse image tokens across layers) to avoid an abrupt information loss.

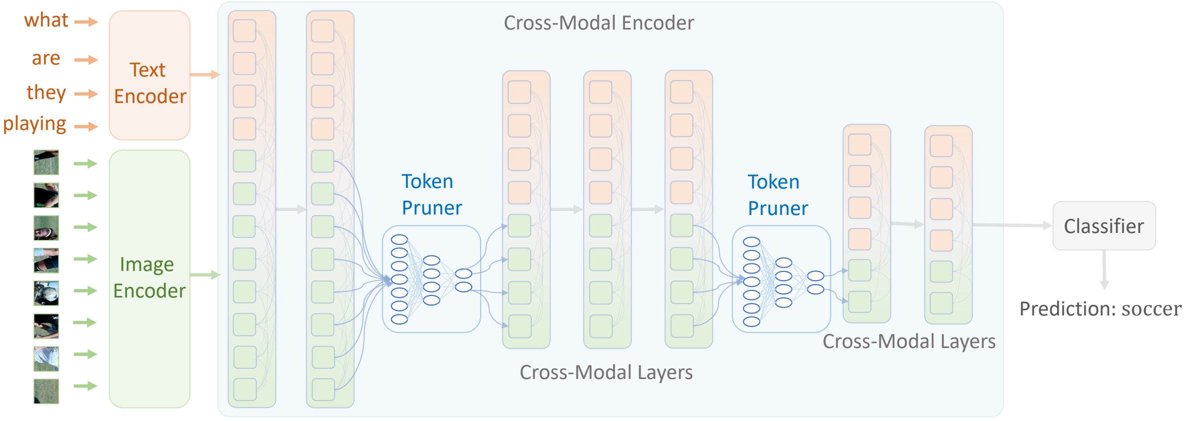

Figure 3: **TiP** applies cascaded image pruning to the cross-modal layers of a VL model via token pruners.

3 Tex-Informed Image Pruning

This section describes our text-informed image pruning approach (depicted in Figure 3). The basic building blocks of our **TiP** are lightweight learnable token pruners that remove image tokens in a cascaded manner to reduce the information loss and improve the computational efficiency of a VL model (Section 3.1). Our text-informed contrastive learning objective encourages the token pruners to keep text-relevant image tokens that are required to maintain task performance (Section 3.2). Our token pruners are jointly trained with a VL model on the downstream VL tasksa using a multi-task learning objective (Section 3.3).

3.1 TiP Design

Our goal is to improve the efficiency of VL models and keep their task performance intact. Given a VL cross-modal encoder, we design **TiP**, that progressively prunes image tokens going through the cross-modal encoder. To do so, we design lightweight token pruner modules (MLPs) and add them in different layers of the cross-modal encoder to predict which image tokens are removed (Figure 3). To prevent abrupt information loss in removing image tokens, we start to prune the image only after the first few cross-modal layers and prune the image tokens in a cascading manner.

Image Token Pruners. Figure 3 illustrates the image pruning procedure in **TiP**. For an n-layer cross-modal encoder, after the first i ($i < n$) layers, a token pruner removes $r\%$ the image tokens at any layer between i and n. The image tokens removed in layer j are not used in subsequent layers. We scatter the token pruners across the cross-modal layers to achieve a better accuracy and efficiency trade-off. Intuitively, pruning at early layers in the cross-modal encoder will have higher inference efficiency but may have bigger performance loss and vice versa. We study this trade-off in more details in §5.3. Each token pruner is a three-layer MLP (denoted as $\mathcal{P}^\ell$) followed by a two-class softmax layer to evaluate *saliency* — keeping or pruning probability — of image tokens. At each pruning layer ℓ, the token pruner takes the contextualized representation of every image token $\boldsymbol{v}_i$, and predicts the *saliency* probability $p_i^\ell = \mathcal{P}^\ell(\boldsymbol{v}_i)$.

Training and Inference. During training, we use p_i^ℓ to derive a binary pruning mask m_i that prevents text and salient image tokens from attending to the pruned image tokens.

$$m_i = \begin{cases} 1 & \text{if } p_i^\ell > 0.5 \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where 1 means the token should not be pruned. The pruning mask is a discrete binary value and not differentiable, hence we adopt the straight-through Gumbel-Softmax (Gumbel, 1954; Jang et al., 2016) to reparameterize m_i to facilitate end-to-end training. To guarantee the VL model is only conditioned on the salient tokens, we apply this pruning mask to the image self-attention and text-to-image cross-attention in the cross-modal layers. Token pruners do not require explicit supervision. Instead, we leverage a multi-task training objective that optimizes for the downstream VL task and learns the task logits by distilling from the original VL model.

During inference, we keep the top-k image tokens $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ ranked by the saliency probability.

3.2 Text-Informed Contrastive Learning

The goal of our text-informed contrastive learning is to encourage the VL model to focus more on the text-relevant image tokens and supervise the token pruners to remove text-irrelevant image tokens.

Text and Image Representations. Formally, we denote $\boldsymbol{T}^\ell = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_n\}$ for text tokens at layer ℓ, $\boldsymbol{V}^\ell = \{\overbrace{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k}^{\text{salient tokens}}, \overbrace{\boldsymbol{v}_{k+1}, \cdots, \boldsymbol{v}_m}^{\text{removed tokens}}\}$ for the image tokens[3], where $\boldsymbol{t}_i, \boldsymbol{v}_i$ are hidden state representations of layer ℓ in the cross-modal encoder. The image representation is obtained by average pooling over salient image tokens: $\overline{\boldsymbol{v}}_k^\ell =$ avgpool$(\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\})$. The text representation is similarly encoded through average pooling: $\overline{\boldsymbol{t}}^\ell =$ avgpool$(\{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_n\})$.

The goal is to enforce high similarity between salient image tokens and their corresponding text representation. Therefore, we define a similarity function for text-image pairs as $s^\ell = p_v(\overline{\boldsymbol{v}}_k^\ell)^\top p_t(\overline{\boldsymbol{t}}^\ell)$, where p_v and p_t are linear projections that transform the vectors to lower-dimensional (192-d) representations.

We form training batches by randomly sampling text-image pair examples from the training dataset of the VL task. We compute the in-batch text to image similary as:

$$\mathcal{S}^\ell = \frac{\exp\left(s^\ell/\tau\right)}{\sum_{b=1}^{B} \exp\left(s_b^\ell/\tau\right)},$$

where τ is a learnable temperature parameter.

Contrastive Loss. Given a pair of text and its aligned image $(\boldsymbol{T}, \boldsymbol{V})$, $\boldsymbol{T}$ is the positive example of $\boldsymbol{V}$ (labeled as 1), and the other $(B-1)$ texts within the mini-batch are negative examples (labeled as 0). Let $\boldsymbol{y}_{ctr}$ denote the ground-truth one-hot similarity (positives are ones and negatives are zeros), the contrastive loss is defined as the cross-entropy H between $\boldsymbol{S}^\ell$ and $\boldsymbol{y}_{ctr}$:

$$\mathcal{L}_{ctr} = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{E}_{(T,V)\sim D}[H(\boldsymbol{y}_{ctr}, \boldsymbol{S}^\ell)] \quad (2)$$

3.3 TiP Multi-Task Training

We conduct multi-task training of the VL task and the token pruning task. We minimizes the following multi-task objective:

$$\mathcal{L}_{\mathbf{TiP}} = \mathcal{L}_{task} + \mathcal{L}_{distill} + \mathcal{L}_{spr} + \mathcal{L}_{ctr} + \lambda \mathcal{L}_{cnt} \quad (3)$$

where $\mathcal{L}_{task}$ is the VL task-specific loss, $\mathcal{L}_{distill}$ is a logit distillation loss that learns the task logits from the original VL model, the sparsity loss $\mathcal{L}_{spr}$ regularizes the token pruners to learn sparse binary pruning masks, $\mathcal{L}_{ctr}$ is our text-informed contrastive loss that encourages the token pruners to keep text-relevant image tokens, and the continuity loss $\mathcal{L}_{cnt}$ regularizes the token pruners to keep continuous image tokens and the λ controls the continuity regularization effect.

Task Loss and Distillation. $\mathcal{L}_{task}$ is a standard cross-entropy loss:

$$\mathcal{L}_{task} = \mathbb{E}[H(\boldsymbol{y}_{task}, \boldsymbol{y}_{pred})] \quad (4)$$

where $\boldsymbol{y}_{task}$ is the ground truth task labels and $\boldsymbol{y}_{pred}$ is the task predictions of the **TiP** VL model.

$\mathcal{L}_{distill}$ minimizes the KL-divergence the prediction logits between **TiP** and the teacher VL model (finetuned without token pruners):

$$\mathcal{L}_{distill} = D_{KL}(\boldsymbol{y}_{pred} \| \boldsymbol{y}'_{pred}) \quad (5)$$

where $\boldsymbol{y}'_{pred}$ is the prediction logits of the teacher VL model. We observe that without distilling from the original finetuned VL model, token pruners remove image tokens that are important to the task predictions and cause significant accuracy drop, we analyze the effect in Section 5.3.

Sparsity Regularization. In order to force image token pruners to learn a sparse mask, we add a regularization loss over the learned salience probabilities p_i. Specifically, for a sparsity ratio of $r \in (0, 1)$ in layer ℓ in the cross-modal encoder, we add the following mean square error loss for *sparsity regularization*:

$$\mathcal{L}_{spr} = \frac{1}{mL} \sum_{\ell=1}^{L} \sum_{i=1}^{m} (p_i^\ell - r)^2 \quad (6)$$

Without sparsity regularization, token pruners can predict all ones for the pruning mask $\boldsymbol{m}$, which is no longer useful for image pruning.

Continuity Regularization. Humans typically focus on salient portions of the image for more efficient visual processing (Figure 1), which are often small *contiguous* regions of the image corresponding to individual objects or clusters of objects. Previous work on grounding images to text has also found that human-interpretable grounded units are often contiguous parts of the image (Li et al.,

[3] Note that the salient tokens might not be consecutive.

2021). To encourage the image token pruners to select contiguous image tokens, we add a contiguity regularization loss over the keep probability p_i. Inspired by the selection of consecutive words (Lei et al., 2016), for a given image token i in location $\{x, y\}$ in the image, we minimize the mean square error of its saliency probability with all its 8 neighbors in locations $\mathcal{N}(x, y) = [(x+e, y+e) \mid \forall e \in \{-1, 0, 1\}]$ as follows:

$$\mathcal{L}_{cnt} = \frac{\sum_{\ell=1}^{L} \sum_{(x',y') \in \mathcal{N}(x,y)} (p^{\ell}_{(x,y)} - p^{\ell}_{(x',y')})^2}{L \cdot |\mathcal{N}(x,y)|} \quad (7)$$

4 Evaluation Setup

4.1 Backbone Vision-Language Models

We evaluate **TiP** for two different VL models: ViLT (Kim et al., 2021) with 110 million parameters and a state-of-the-art VL model, METER (Dou et al., 2022) with 330 million parameters. We denote **TiP**-ViLT and **TiP**-METER as **TiP** applied for ViLT and METER respectively. More details about these models are in Appendix A.2.

4.2 Evaluation Tasks

We evaluate the models on three visual reasoning tasks: visual question answering (VQAv2; Goyal et al. 2017), natural language visual reasoing (NLVR2; Suhr et al. 2019), and visual entailement (SNLI-VE; Xie et al. 2019). More details about these datasets are in Appendix A.2.

4.3 TiP Implementation Details

We set the cascading pruning ratio to be 0.5 for three pruning layers (4th, 7th, 10th for ViLT and 2nd, 4th, 6th for METER) by default (except for ViLT-NLVR2, where we set pruning ratio to 0.4 due to a larger accuracy drop). In Section 5.3, we show ablations for pruning ratios and pruning layer locations. The token pruner implementation and more training details are in Appendix A.1.

4.4 Baselines

To compare the benefits of **TiP**, we additionally evaluate three baselines:

Random Pruning: we use the same pruning ratio and same pruning locations as **TiP** to randomly remove image tokens during inference. The random pruning uses the finetuned **TiP** model but random pruning masks instead of those predicted by the token pruners.

Smaller Resolution: We downsample the input image to smaller resolutions and finetune the VL models. Using smaller input images directly reduces the computation of VL models.

Pruning via Grounding: text-grounded image regions are supposed to provide more important information for the VL task. We compare with a grounded pruning baseline that can approximate the upper bound of the task performance under image pruning. This baseline uses the grounded image regions as the gold labels for salient tokens. We remove the image tokens at the same layers as **TiP**. However, it is often challenging to obtain text-grounded image regions because the datasets we study do not come with annotated grounded image data. Instead, we use a recent grounding model ALBEF (Li et al., 2021) to generate text-grounded image regions for the visual questions in the VQAv2 dataset. We show in Appendix A.4 the image regions generated by the ALBEF grounding model has high overlaps with human attention data and can serve as a reasonable tool to automatically generating grounded image regions.

4.5 Evaluation Metrics

Accuracy Metrics. We measure *VQA accuracy* (Goyal et al., 2017) for the VQAv2 dataset and *accuracy* for both the VE and NLVR2 datasets. Unlike previous works (Kim et al., 2021; Dou et al., 2022), where their models are trained on the combined training and validation sets, our focus is not to obtain the state-of-the-art results, so we train the two VL models on the training set and report the results on the test set.

Resource Consumption. We measure the actual inference throughput (examples per second) of the VL models on the GPU hardware and compare them to the finetuned models with no image pruning. We also measure the peak memory consumed during the model inference phase and report *memory reduction* ratio compared to the original model without image pruning. These two runtime metrics are found to be more accurate to compare resource consumption instead of using the FLOPs complexity metric (Graham et al., 2021). We describe the detailed setup in Appendix A.5.

5 Experimental Results

5.1 Main Results

TiP is faster and remains accurate. Table 1 shows the main results comparing performance, in-

Model	Datasets	Original Accuracy	**TiP** Accuracy	Throughput Increase	Memory Reduction
METER (SoTA)	VQAv2	77.5	77.2 (-0.3)	1.57x	32%
	VE	81.2	80.8 (-0.4)	1.52x	33%
	NLVR2	82.8	82.4 (-0.4)	1.56x	32%
ViLT	VQAv2	69.5	68.9 (-0.6)	1.51x	36%
	VE	75.9	75.2 (-0.7)	1.54x	35%
	NLVR2	75.6	74.9 (-0.7)	1.43x	29%

Table 1: Performance and efficiency comparison between the original fine-tuned vs **TiP** fine-tuned models for the ViLT and METER over 3 downstream visual reasoning tasks.

Model	Image Resolution	VQAv2 Accuracy	Throughput Increase	Memory Reduction
Resolution	192x192	72.7 (-3.0)	4.23x	75%
	224x224	73.7 (-2.0)	3.48x	66%
	256x256	74.4 (-1.3)	2.67x	54%
	320x320	75.2 (-0.5)	1.62x	40%
TiP (Ours)	320x320	74.8 (-0.9)	2.32x	56%
Random Pruning	384x384	69.4 (-6.3)	1.57x	32%
Pruning via Grounding	384x384	75.8 (+0.1)	0.53x	25%
TiP (Ours)	384x384	75.4 (-0.3)	1.57x	32%
METER	384x384	75.7	1x	0%

Table 2: Performance and efficiency comparison between the baselines and **TiP** for the METER model on VQAv2 dev set.

ference speed and memory reduction of **TiP**. Overall, we observe over **1.5x speedup** in inference throughput and over **30% reduction** in memory footprint for both ViLT and METER models on all three datasets. Importantly, the task performance of **TiP** remains competitive compared to the original finetuned VL models with only <1% drop in accuracy.

Baseline pruning methods incur large accuracy drops. Table 2 shows downsampling the input image to smaller resolution improves the inference throughput and reduces memory footprint but comes with larger accuracy drops. The closest resolution is 320x320 which provides slightly more (8%) memory reduction than **TiP**, but it still has lower inference throughput and bigger accuracy gap. Meanwhile, **TiP** is orthogonal to downsampling strategies, and applying **TiP** to smaller images could provide additional efficiency gains (see 3rd row numbers in Table 2).

Compared to **TiP**, the random pruning baseline (4th row in Table 2) has a much higher (6.3%) accuracy drop. This indicates the token pruners in **TiP** are effective in identifying salient image tokens for the VL tasks.

Grounded pruning baseline (5th row in Table 2) shows the METER VQA model becomes even more accurate when giving the grounded image tokens. This validates that the grounded image regions are sufficient for the VL model to achieve high accuracy on the VL tasks. On the other hand, the inference efficiency is slowed down by 53% due to the use of the grounding model. Unlike these models, **TiP** does not rely on explicit supervision signals of grounded image regions, learning to prune image tokens that are not related to text by solely using end task supervision.

5.2 Interpretability Analysis

TiP keeps image regions that overlap with human attention. We evaluate whether salient tokens identified by **TiP** align with human attention using the VQA-HAT (Das et al., 2016) dataset that contains human attention maps for the images of visual questions from the VQAv1 (Antol et al., 2015) dataset. We apply the **TiP**-METER VQA model on the VQA-HAT validation dataset and extract the kept image portions (patches) (see details in Appendix A.3). We then extract image portions from the human attended regions. For each image patch kept by **TiP**-METER, if it overlaps with human attention region, we count it as correct and calculate the *alignment precision* as the total number of correct patches divided by the total number of kept patches. We average the precision score across three human annotators on the validation set.

TiP-METER has an average of 69.2% precision. In contrast, a random pruning baseline only achieves 39.8% average precision, indicating **TiP**-METER has a high overlap with human attention. We find the precision for **TiP**-ViLT model is lower but still 2.7% higher than a random baseline. This is likely because the ViLT model does not learn fine-grained alignments like the METER model.

TiP keeps faithful and interpretable image regions. In order to measure the faithfulness of the salient image tokens as an explanation (Wiegreffe and Marasović, 2021) – whether the model only uses the salient tokens for prediction, we compute how *comprehensive* and *sufficient* this explanation is (DeYoung et al., 2019). Sufficiency measures difference in prediction accuracy (compared to using the full image) when only the salient tokens are used for prediction. A low sufficiency score implies that the salient tokens are indeed *necessary* for the VL model to make a prediction. Comprehensiveness measures difference in accuracy computed by using the pruned tokens for prediction. A high comprehensiveness score indicates that the salient tokens were *sufficient* for prediction. To obtain comprehensiveness scores for **TiP**, we invert the learned mask in the first pruning layer on the validation set.

Both **TiP**-METER and **TiP**-ViLT have low sufficiency and high comprehensiveness compared to the random baseline (6.3%).[4] Specifically, **TiP**-METER has a sufficiency accuracy of 0.3% (the lower the better) and has a comprehensiveness accuracy of 21.0% (the higher the better). **TiP**-ViLT has a sufficiency accuracy of 0.6% and has a low comprehensiveness accuracy of 13.4% which is lower than **TiP**-METER, indicating that METER produces more *faithful* explanations.

We visualize the salient image regions generated by **TiP**-METER in Figure 4 by randomly sampling visual questions from the VQA-HAT validation set. **TiP**-METER prunes question-irrelevant image portions and keeps portions that align with human attention regions. More visualization of **TiP**-METER images are in Appendix A.6.

5.3 Ablation Study

Learning Components in TiP. Table 3 indicates the impact of different components of **TiP**. It shows the performance drop of **TiP** ViLT by removing the distillation loss (w/o distillation), text-informed contrastive learning (w/o text contrastive) and the continuity prior for fine-tuning (w/o continuity loss) on the VQAv2 dataset. The distillation loss is a key ingredient in **TiP** as the performance drops by 5.2% without it. The results also show that both the contrastive learning and the continuity prior significantly contribute to **TiP** and the original fintuned VQA model.

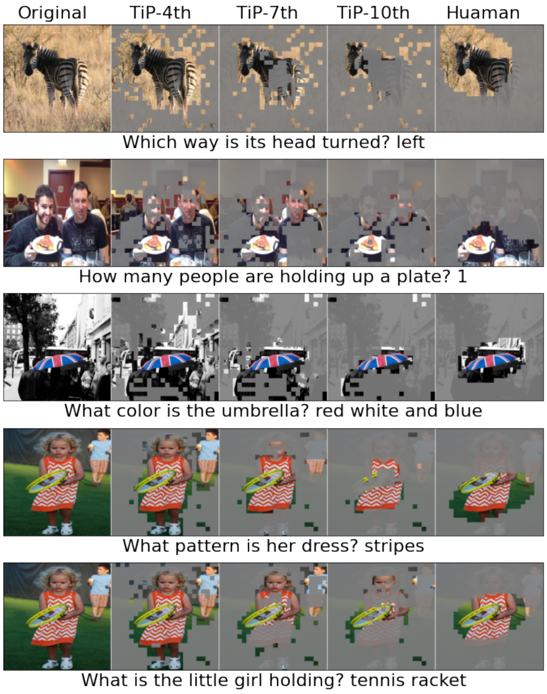
Original
TiP-4th
TiP-7th
TiP-10th
Huaman
Which way is its head turned? left
How many people are holding up a plate? 1
What color is the umbrella? red white and blue
What pattern is her dress? stripes
What is the little girl holding? tennis racket

Figure 4: Visualization of **TiP**-METER salient image portions for visual questions in the VQA-HAT validation set. For each row, the first image is the original input image, the last is the human attention regions, three images in between are generated by **TiP**-METER with cascaded pruning. Note that with different input text, **TiP**-METER removes different image regions for the same image.

Model	VQA Accuracy
ViLT	69.5
TiP-ViLT	68.9 (-0.7)
w/o distillation	63.7 (-5.2)
w/o text contrastive loss	67.9 (-1.6)
w/o continuity loss	68.4 (-1.1)
w/o text contrastive & continuity loss	67.6 (-1.9)
TiP-ViLT (MLP)	68.9 (-0.7)
w/ linear layer	67.3 (-2.2)
w/ mean attention scores	67.4 (-2.1)
w/ single-head attention scores	67.1 (-2.4)

Table 3: Ablation analysis for applying text-informed contrastive learning (text contrastive) and continuity loss on the VQAv2 dataset for ViLT; and for different token pruner architectures in **TiP**-ViLT on the VQAv2 dataset.

Token Pruner Architectures. **TiP** uses a 3-layer MLP architecture for the token pruners (described in Appendix A.1) that predict the pruning masks taking the image representations as inputs. We com-

[4] Note that the comprehensiveness for the random baseline is low because the inverse of random tokens still contains a lot of useful information for the task prediction.

pare three alternative lightweight architectures for the token pruner modules: (1) *linear layer* - we replace the MLP with a single linear layer; (2) *mean attention scores* - we use the average text to image attention scores across all attention heads as the token pruner inputs; (3) *single-head attention scores* - we feed the text to image attention scores from the first attention head to the token pruners. All of these architectures added little ($< 0.1\%$) runtime overhead to the VL models. Table 3 shows that the MLP architecture achieves the best performance.

Group	Pruning Locations	Pruning Ratios	VE Test Accuracy	Throughput Increase
ratios	4,7,10 4,7,10	0.7 0.3	72.2 (-3.7) 75.7 (-0.2)	1.77x 1.26x
# of layers	4 4,7	0.7 0.6	74.5 (-1.4) 75.0 (-0.9)	1.56x 1.55x
locations	4,5,6 8,9,10	0.5 0.5	74.2 (-1.7) 75.3 (-0.6)	1.70x 1.21x
TiP (Ours)	4,7,10	0.5	75.2 (-0.7)	1.54x
ViLT	-	-	75.9	-

Table 4: Ablation analysis of pruning ratios, # of pruning layers, and pruning locations for the ViLT model on VE task.

Design Choices of Pruning Ratios and Locations. Given a 12 layer VL cross-modal encoder like ViLT, many combinations of pruning locations and ratios achieve similar inference speedups. Pruning at earlier layers with lower ratios has similar computation efficiency to pruning at later layers with higher ratios. For comparing the accuracy with different number of pruning layers, we control the inference throughput to be similar to **TiP** by selecting the pruning ratios and locations. Table 4 shows cascaded pruning at 3 layers (4th,7th,10th) has higher accuracy than pruning at one (4th) or two layers (4th,7th) while having similar speedups.

The ratios row in Table 4 shows pruning more image tokens (with ratio=0.7) leads to bigger throughput increase but has significantly lower (-3.7%) accuracy, while pruning fewer image tokens (with ratio=0.3) is more accurate but causes lower throughput. We find that pruning in the earlier layers leads to bigger throughput but drops accuracy by 1.7%, while pruning in the later layers is slightly more accurate but provides fewer benefits in throughput. Overall, we choose 3-layer cascaded pruning strategy with a pruning ratio of 0.5 and scatter the pruning locations more evenly to balance accuracy and speed trade-offs.

6 Related work

Data Pruning. Prior work in data pruning (Rao et al., 2021; Yin et al., 2021; Liang et al., 2021; Goyal et al., 2020) focus on single-modality models by either pruning input text or image alone. DynamicViT (Rao et al., 2021) and AdaViT (Yin et al., 2021) both progressively remove the uninformative content and keep salient regions in the input image. This type of pruning does not apply to language and vision tasks where the salient regions depend on the input text. Our work shows different input texts lead to pruning different image regions even for the same input image. EViT (Liang et al., 2021) reduces the image tokens progressively but requires expensive pretraining. PoWER-BERT (Goyal et al., 2020) speeds up the inference of text-based Transformers like BERT (Devlin et al., 2019) by removing the input text tokens, which are not the main computation bottlenecks for most vision and language tasks.

Efficient Inference. Many techniques have focused on model pruning (Lagunas et al., 2021; Yu and Wu, 2021; Yu et al., 2022; TPr), dynamic computation by early exiting (Xin et al., 2020; Zhou et al., 2020; Schwartz et al., 2020; Liu et al., 2020) or designing small and efficient VL models (Fang et al., 2021; Wang et al., 2020). Combining these orthogonal optimizations with our text-informed image-pruning method could further accelerate the inference in VL models. Meanwhile, our method provides more interpretability power for VL tasks.

7 Conclusion

Large vision language models have been effective at visual reasoning tasks due to their complex cross-modal interactions between the text and image information across multiple Transformer self-attention layers, which is computationally expensive. We introduce a text-informed image pruning approach -**TiP**- that progressively removes the redundant image information and make VL models run faster. **TiP** maintains task performance of the original VL models while also providing model interpretability without explicit supervision.

8 Limitations

While we show our text-informed image pruning method is interpretable both quantitatively and qualitatively, there are still cases where the VL models rely on the counter-intuitive image regions (which humans never use) to predict the task label. We suspect this is because the models overfit to spurious correlations. Improvements in pretrained models that overcome model biases might mitigate these issues.

Our method does not apply to VL models where the cross modal encoder layers are relatively lightweight. For example, the vision encoder is much more computationally expensive than the cross-modal encoder for VL models like ALBEF (Li et al., 2021) and X-VLM (Zeng et al., 2021), therefore, the end to end inference speed improvement is marginal. Pruning the image inside the vision encoder could further improve the model efficiency, we leave this exploration to future work.

References

TPrune: Efficient Transformer Pruning for Mobile Devices: ACM Transactions on Cyber-Physical Systems: Vol 5, No 3.

2022. Deepspeed.

2022. huggingface/accelerate.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. pages 2425–2433.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2016. Human Attention in Visual Question Answering: Do Humans and Deep Networks look at the same regions? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 932–937, Austin, Texas. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. 2022. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18166–18176.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Compressing Visual-Linguistic Model via Knowledge Distillation. pages 1428–1438.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. PoWER-BERT: Accelerating BERT Inference via Progressive Word-vector Elimination. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3690–3699. PMLR. ISSN: 2640-3498.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. pages 6904–6913.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. 2021. LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference. pages 12259–12269.

Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office.

Jay Hegdé. 2008. Time course of visual perception: Coarse-to-fine processing and beyond. *Progress in Neurobiology*, 84(4):405–439.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5583–5594. PMLR. ISSN: 2640-3498.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block Pruning For Faster Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in neural information processing systems*, volume 34, pages 9694–9705. Curran Associates, Inc.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2021. EViT: Expediting Vision Transformers via Token Reorganizations.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 740–755, Cham. Springer International Publishing.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a Self-distilling BERT with Adaptive Inference Time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Number: arXiv:1907.11692 arXiv:1907.11692 [cs].

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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR. ISSN: 2640-3498.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. *arXiv:2106.02034 [cs]*. ArXiv: 2106.02034.

Ronald A. Rensink. 2000. The dynamic representation of scenes. *Visual Cognition*, 7(1-3):17–42. Place: United Kingdom Publisher: Taylor & Francis.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020. The Right Tool for the Job: Matching Model and Instance Complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651, Online. Association for Computational Linguistics.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626. ISSN: 2380-7504.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020. MiniVLM: A Smaller and Faster Vision-Language Model. *arXiv:2012.06946 [cs]*. ArXiv: 2012.06946.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. Technical Report arXiv:2202.03052, arXiv. ArXiv:2202.03052 [cs] version: 2 type: article.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. SimVLM: Simple Visual Language Model Pretraining with Weak Supervision.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. *arXiv preprint arXiv:2102.12060*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual Entailment Task for Visually-Grounded Language Learning. Technical Report arXiv:1811.10582, arXiv. ArXiv:1811.10582 [cs] type: article.

Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020. Early Exiting BERT for Efficient Document Ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, Online. Association for Computational Linguistics.

Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2021. AdaViT: Adaptive Tokens for Efficient Vision Transformer.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78. Place: Cambridge, MA Publisher: MIT Press.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. 2022. Width & depth pruning for vision transformers. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 2022.

Hao Yu and Jianxin Wu. 2021. A Unified Pruning Framework for Vision Transformers. *arXiv:2111.15127 [cs]*. ArXiv: 2111.15127.

Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. VinVL: Revisiting Visual Representations in Vision-Language Models. pages 5579–5588.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. BERT loses patience: Fast and robust inference with early exit. In *Advances in neural information processing systems*, volume 33, pages 18330–18341. Curran Associates, Inc.

A Appendix

A.1 TiP Details

Implementation. We implement the token pruner architecture as a 3-layer MLP: $LayerNorm$(H) -> $Linear$(H, H) -> $gelu$ -> $Linear$(H, H/2) -> $gelu$ -> $Linear$(H/2, H/4) -> $gelu$ -> $Linear$(2) -> $LogSoftmax$(), where H is hidden size set to 768.

Training. We use the Transformers (Wolf et al., 2020) and Accelerate (Hug, 2022) with DeepSpeed (Dee, 2022) library to implement the training tasks. We conduct training jobs on 4 Nvidia A40 GPUs. For both ViLT and METER model, we first follow the training hyperparameters in their original papers and finetune the pretrained model to obtain task-specific models. These models are used as baselines for measuring accuracy drop and also used as the teacher model for **TiP** distillation.

For each task in **TiP**, we follow a two-stage finetuning procedure. In the first stage, we first initialize the model weights from the finetuned task-specific VL model, then we freeze the backbone VL model and only finetune the token pruners and the task-specific classifier for 5 epochs with all the multi-task losses except the continuity regulation loss (i.e, setting λ to 0). In the second stage, we unfreeze the cross-modal encoder weights of the finetuned model in the first stage and finetune the model with all the multi-task losses (setting λ to 1) for 60 epochs METER models and 200 epochs for ViLT models. We apply early stopping for the training and setting a stopping patience of 10 epochs (i.e. the training stops if the validation performance does not improve for 10 epochs). For both the VQAv2 and VE tasks, training for takes roughly 2 days for **TiP**-METER models and 1 day for **TiP**-ViLT models. For NLVR2 task, training **TiP**-ViLT and **TiP**-METER takes 4 hours and 6 hours respectively. The token pruner learning rate is set to the same as the cross-modal encoder of the VL model. The warmup ratio is 0.1 for the first stage and 0.01 for the second stage for all tasks and models.

We list all training hyperparameters in Table 5.

A.2 Evaluation setup

We evaluate **TiP** for two different VL models: ViLT (Kim et al., 2021) and METER (Dou et al., 2022) on three visual reasoning tasks: Visual Question Answering (VQA) (Goyal et al., 2017), Visual Entailment (VE) (Xie et al., 2019), and Natural Language for Visual Reasoning (NLVR2) (Suhr et al., 2019).

	METER		ViLT	
	VE,VQAv2	NLVR2	VE,VQAv2	NLVR2
token pruner lr	2.5e-5	5e-5	1e-4	1e-4
cross-modal lr	2.5e-5	5e-5	1e-4	1e-4
task classifier lr	2.5e-4	1e-4	1e-3	1e-4
end lr	1e-7	1e-7	1e-6	1e-6
batch size per gpu	32	16	64	64
total batch size	256	128	256	128
image size	384	288	384	384
patch size	16	16	32	32

Table 5: Hyperparameters for training **TiP**.

ViLT is a recent efficient VL model that uses BERT (Devlin et al., 2019) embeddings to encode text and a linear layer to project image patches. ViLT then concatenates the text and image tokens and uses 12-layer Transformer encoder to perform cross-modal fusion. ViLT is a relatively lightweight model and has 110 million parameters.

METER is a state-of-the-art VL model that uses RoBERTa (Liu et al., 2019) as the text encoder and CLIP (Radford et al., 2021) as the image encoder, and 12 BERT-like cross-attention layers to fuse the text and image modalities. METER is a large model and has 330 million parameters.

Visual Question Answering dataset (VQAv2) contains over 1 million diverse open-ended questions about images both from the MSCOCO (Lin et al., 2014) and real world scenes. Answering these questions requires an understanding of vision, language and commonsense knowledge.

Visual Entailment is a visual inference task that consists of 570K sentence image pairs constructed from the Stanford Natural Language Inference corpus (Bowman et al., 2015) and Flickr30k (Young et al., 2014). The goal is to predict to predict whether the image premise semantically entails the text.

Natural Language for Visual Reasoning corpora (NLVR2) have over 100K examples of linguistically diverse English sentences written by human and are grounded in pairs of visually complex images. The goal is to predict whether a sentence is true about two input images.

A.3 Computing Overlap with Human Attention

We filter the human attention maps by setting a threshold of 0.2 as attention values smaller than 0.2 do not form a focus region on the image. We then resize the attention map image into 384x384, divide it into patches of 16x16 and 32x32 for the comparing with METER and ViLT respectively.

A.4 Grounded Pruning Implementation

To implement grounded pruning, we need annotated text-grounded image regions. Ideally, human attention data like in VQA-HAT (Das et al., 2016) can be used as the gold grounded image regions. However, human attention data does not exist for the three datasets we evaluate. Instead, we focus on a recent grounding model – ALBEF (Li et al., 2021). To make sure ALBEF can ground similar image regions as human attention, we evaluate ALBEF grounding precision on the validation set of the VQA-HAT (Das et al., 2016) dataset. ALBEF uses the Grad-CAM (Selvaraju et al., 2017) of the text to image cross-attention maps in the cross-modal encoder. We take the grounding model weakly supervised on the RefCOCO+ (Kazemzadeh et al., 2014) dataset.We append the answer label to the question, extract Grad-CAM data and resize it into the same size as the input image (384x384). We compute the precision of ALBEF grounding model as follows: an image is first divided into 32x32 patches, an image patch from the grounded region is count as correct if it overlaps with human attention patch. ALBEF grounding model has an average precision of 78.4% across three human annotators. The high precision indicates it can be used to generate grounded image regions similar to the areas where humans pay attention.

We use the ALBEF grounding model to automatically generate grounded image regions for the training and dev sets of VQAv2 dataset and use them as gold labels for the image pruning during training and inference (i.e., an image token that does not appear in the grounded regions is removed). To fairly compare with **TiP**, we gradually expand the grounded image regions to their neighbor patch by a radius of 2 (24 patches) and 1 (8 patches) for the pruning at 4th and 7th layers. We use the grounded image regions for pruning at 10th layer.

A.5 Resource Measurement Details

For inference throughput measurements, we increase the batch size until the model gets out of GPU memory, and run the inference with the largest batch size for 30 seconds on a GTX 1080 Ti with 12 GiB memory. For inference memory footprint, we use the same batch size for the original VL model and **TiP** version and report the peak memory difference.

A.6 TiP-METER Visualization

We show in Figure 5 more visualizations of the **TiP**-METER VQA model on the VQA-HAT validation set.

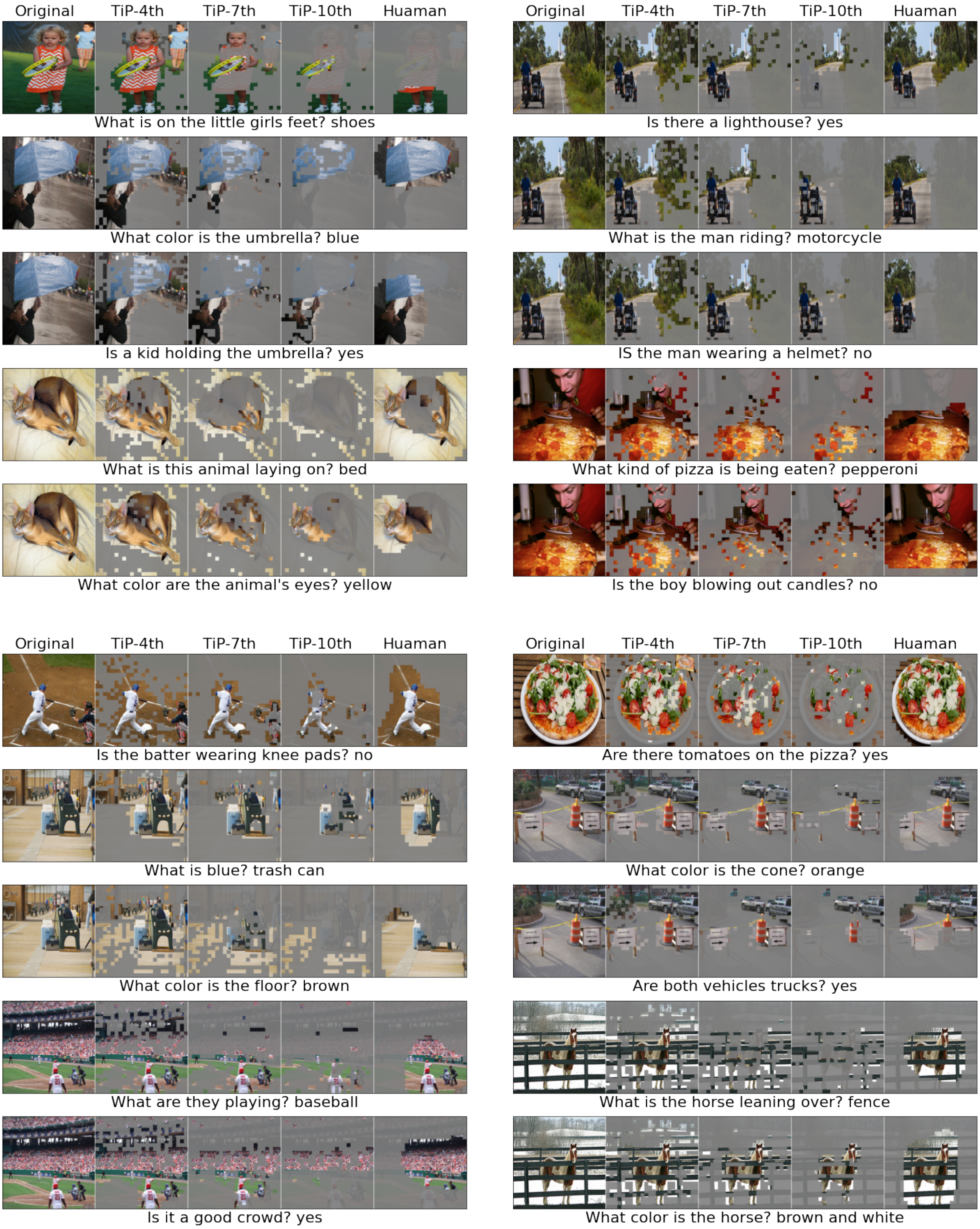
Original
TiP-4th
TiP-7th
TiP-10th
Huaman
What is on the little girls feet? shoes
What color is the umbrella? blue
Is a kid holding the umbrella? yes
What is this animal laying on? bed
What color are the animal's eyes? yellow
Original
TiP-4th
TiP-7th
TiP-10th
Huaman
Is there a lighthouse? yes
What is the man riding? motorcycle
IS the man wearing a helmet? no
What kind of pizza is being eaten? pepperoni
Is the boy blowing out candles? no
Original
TiP-4th
TiP-7th
TiP-10th
Huaman
Is the batter wearing knee pads? no
What is blue? trash can
What color is the floor? brown
What are they playing? baseball
Is it a good crowd? yes
Original
TiP-4th
TiP-7th
TiP-10th
Huaman
Are there tomatoes on the pizza? yes
What color is the cone? orange
Are both vehicles trucks? yes
What is the horse leaning over? fence
What color is the horse? brown and white

Figure 5: More visualization of **TiP**-METER kept image portions for visual questions in the VQAv1 validation set. For each row, the first image is the original input image, the last one is the input image only revealing the human attention regions, the middle three ones are **TiP**-METER images with removed regions (shaded grey).