# Visually-augmented Pretrained Language Models for NLP Tasks without Images

**Hangyu Guo**[1]*, **Kun Zhou**[3,4]*, **Wayne Xin Zhao**[2,4]†, **Qinyu Zhang**[1] and **Ji-Rong Wen**[2,3,4]

[1]School of Electronics and Information Engineering, Harbin Institute of Technology (Shenzhen).

[2]Gaoling School of Artificial Intelligence, Renmin University of China.

[3]School of Information, Renmin University of China.

[4]Beijing Key Laboratory of Big Data Management and Analysis Methods.

hyguo0220@gmail.com, francis_kun_zhou@163.com

batmanfly@gmail.com, zqy@hit.edu.cn, jrwen@ruc.edu.cn

## Abstract

Although pre-trained language models (PLMs) have shown impressive performance by text-only self-supervised training, they are found lack of visual semantics or commonsense. Existing solutions often rely on explicit images for visual knowledge augmentation (requiring time-consuming retrieval or generation), and they also conduct the augmentation for the whole input text, without considering whether it is actually needed in specific inputs or tasks. To address these issues, we propose a novel **V**isually-**A**ugmented fine-tuning approach that can be generally applied to various PLMs or NLP tasks, **W**ithout using any retrieved or generated **I**mages, namely **VAWI**. Experimental results show that our approach can consistently improve the performance of BERT, RoBERTa, BART, and T5 at different scales, and outperform several competitive baselines on ten tasks. Our codes and data are publicly available at https://github.com/RUCAIBox/VAWI.

## 1 Introduction

Recent years have witnessed the success of pre-trained language models (PLMs) (Qiu et al., 2020; Zhao et al., 2023), such as GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020), in a variety of natural language process (NLP) tasks. Since these PLMs are mostly trained on text-only corpus via self-supervised pre-training, they have been shown lack of visual commonsense (Liu et al., 2022) and real-world knowledge (Zhang et al., 2022). As a result, PLMs can't well solve visually related language tasks [1], *e.g.,* answering the color and size of common things, especially those requiring complex commonsense knowledge.

To alleviate this problem, existing works mainly enhance PLMs by infusing visual information. Typically, given a text input, these studies firstly augment the visual information from retrieved or generated images about the input and then leverage their visual representations to improve PLMs on NLP tasks. Such an approach leads to *visually-augmented pre-trained language models (VaLMs)*, where they adopt either visually-augmented pre-training (Tan and Bansal, 2020; Wang et al., 2022) or visually-augmented fine-tuning (Lu et al., 2022). Despite the effectiveness, there are two major shortcomings in these methods. First, these methods often rely on pre-learned complementary retrievers or generators, and also require time-consuming inference to retrieve or generate proper images that are paired with the input. The above costly conditions largely limit the applicability of these approaches. Second, the retrieved or generated images are inevitable to involve irrelevant or redundant visual information. If simply integrating them, the original text representations might be affected.Increasing evidence shows that the visual information is not always useful for NLP tasks (Dai et al., 2022), and sometimes leads to performance degradation.

Considering these issues, we aim to develop a more efficient and effective way to visually augment the PLMs and the solution is twofold:

- Firstly, we don't explicitly produce (retrieve or generate) the images but instead generate visually-aligned representations of the text on-the-fly. Recent studies (Radford et al., 2021; Jia et al., 2021) have shown that the vision-language pre-trained models (VL-PTMs) can well learn the alignment between the representations of texts and images from large-scale text-image pairs. Thus, our idea is to employ the output representations of a text from VL-PTMs' text encoders as a surrogate for the visual representations of related images.

Such a way is simple and efficient: we can only keep the text encoder of a VL-PTM to produce the visually-aligned representations of texts, getting rid of the complicated image retrieval or generation

* Equal contributions.

† Corresponding authors.

[1]In this work, we mainly focus on text-only NLP tasks that may benefit from external visual information, rather than visual-language tasks involving images.



*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 14912–14929
July 9-14, 2023 ©2023 Association for Computational Linguistics

process. It is widely recognized that there is a large semantic gap between different modalities (Liang et al., 2022). Our method can alleviate this issue to some extent since the visual augmentations are derived from the text representation itself.

• Secondly, instead of directly feeding visual augmentations into the PLM, we propose to use the augmented visual information only when it is actually required. In fact, for a text input of a NLP task, PLMs are not always hungry for the visual background knowledge to effectively understand it, especially for visually-irrelevant expressions. Unlike previous works which inject visual information into a text (Tan and Bansal, 2020; Wang et al., 2022) from the whole, we consider identifying *visually-hungry words* (those that require visual knowledge to derive complete semantics) from the text input, and only infuse the visual augmentations through these trigger words. We conduct visual augmentations at the word level, because it is more flexible and controllable, considering the augmented information is often irrelevant or noisy.

To this end, in this paper, we propose a general **V**isually-**A**ugmented fine-tuning approach to improving PLMs for NLP tasks **W**ithout **I**mages, namely **VAWI**. Our approach consists of three ingredients, namely visually-hungry words extraction, visual knowledge augmentation, and visually-enhanced fine-tuning. Given the text input from a NLP task, we first extract the visually-hungry words (VH-words) from the input sentence. As the annotations of VH-words are generally unavailable, we propose three strategies to automatically extract the VH-words, relying on the syntax trees, attention distributions of VL-PTMs, and an adaptive learnable module, respectively. Then, based on the extracted VH-words, we leverage the text encoder of CLIP (Radford et al., 2021) (being fixed in our approach), a VL-PTM that has been pre-trained on millions of text-image pairs, to encode the VH-words for obtaining their visually-aligned representations. Finally, we infuse the visually-aligned representations into PLMs, and consider the general and parameter-efficient fine-tuning strategies for small and large PLMs, respectively.

To verify the effectiveness of our framework **VAWI**, we test it on four PLMs (*i.e.,* BERT, BART, RoBERTa, and T5) at different scales (*i.e.,* 110M, 340M, 3B), and conduct extensive experiments in natural language understanding, commonsense reasoning, and text generation tasks. Experimental results show that our **VAWI** can boost the performance of these PLMs significantly, *i.e.,* 3.11%, 2.54%, and 2.16% absolute improvements on the commonsenseQA task using RoBERTa-base, RoBERTa-large, and T5-3b, respectively. Besides, **VAWI** can outperform (or be on par with) several competitive baselines that adopt complicated visually-augmented methods.

## 2 Related Work

**Pre-trained Language Models.** Recent years have witnessed the success of pre-trained language models (PLMs) (Devlin et al., 2019; Radford et al., 2019). After pre-trained on the large-scale corpus, PLMs can be fine-tuned on multiple NLP tasks and achieve remarkable performance. However, since PLMs are just pre-trained with text-only data, they may suffer from the reporting bias problem (Gordon and Van Durme, 2013; Paik et al., 2021; Zhang et al., 2022), where the frequency distribution of visual commonsense in the text may not fully reflect the real-world distribution of the commonsense. Existing works have also found that such a problem can not be well addressed by enlarging the model or pre-training corpus (Paik et al., 2021; Zhang et al., 2022). In this work, we aim to alleviate this problem by adding visual knowledge on PLMs during fine-tuning.

**Vision-Language Pre-Trained Models.** To better accomplish the vision-language tasks, vision-language pre-trained models (VL-PTMs) (Su et al., 2019; Lu et al., 2019) become a hot point in recent years, which require large-scale image-text pairs for pre-training. Existing VL-PTMs fall into two categories based on the way of modeling vision-language interaction. The first category of models (Lu et al., 2019; Li et al., 2021) adopts an explicit vision-language interaction layer to fuse the text embeddings and image features. These models are more suitable to capture fine-grained semantic interactions between vision and language.The second category of models (Radford et al., 2021; Jia et al., 2021) incorporates separate encoders to model the vision and language information, and relies on pre-training tasks (*e.g.,* cross-modal contrastive learning) to align their representations into the same latent space. Such a way is capable of producing enriched single-modal representations.

**Visually-Augmented Language Model.** To introduce visual information into PLMs, visually-augmented language model (VaLM) (Wang et al., 2022) has become an emerging research topic. Existing VaLMs can be categorized into visually-augmented pre-training and fine-tuning. Visually-augmented pre-training approaches (Tan and Bansal, 2020; Zhu et al., 2022) continually pre-train PLMs with the retrieved visual information related to input tokens or sentences and also revise the masked language model task for better capturing the visual semantics. Visually-augmented fine-tuning method (Lu et al., 2022) introduces the visual information into PLMs during fine-tuning. These methods also leverage the image retrieval or generation models to augment the visual information and design a special fusion module to inject it into PLMs. However, existing VaLM approaches mostly need to retrieve or generate visual information for utilization. Such a way is time-consuming, and may involve unrelated or noisy information into PLMs, leading to performance degradation. In this work, we aim to first detect the visually-hungry words from the text, and then utilize a VL-PTM to generate their visually-aligned representations without the usage of external images or generation models. As a comparison, our approach is more flexible and efficient to leverage visual information for enhancing text-based PLMs.

# 3 Method

In this section, we firstly introduce the task setting, and then describe our proposed visual augmentation approach for infusing visual knowledge into PLMs during fine-tuning.

## 3.1 Task Setting and Solution Overview

This work aims to improve the fine-tuning performance of pre-trained language models (PLMs) on NLP tasks by leveraging the related visual information without images. For a NLP task, a set of $n$ labeled texts $\{\langle x_i, y_i \rangle\}$ are available, where $x_i$ is the $i$-th text data consisting of a sequence of words, denoted as $x_i = \{w_1, w_2, ..., w_m\}$, and $y_i$ is the ground-truth output, which can be a discrete label (classification), a continuous value (regression) or a text sequence (generation).

To solve the target task, we assume that a text-based PLM is given (either for understanding or generation). Let $f$ denote a PLM parameterized by $\theta_{\text{PLM}}$ that has already been pre-trained on general-purpose large-scale text data. Given the labeled training data, we can train the PLM using a specific loss function (*e.g.,* cross-entropy loss) and further solve the target task. However, existing works (Tan and Bansal, 2020; Zhang et al., 2022) have revealed that PLMs may be unaware of visual knowledge that is not explicitly mentioned in the pre-trained text-only data (*e.g.,* the shape of coins and the color of the sky), leading to the lack of world commonsense and generating wrong statements.

In this work, we focus on devising an efficient and effective way to infuse such visual knowledge into PLMs during fine-tuning. Our approach is based on *visually-hungry words* (abbreviated as *VH-words*), which require visual information to derive complete semantic representations. The overall illustration of our approach is shown in Figure 1. Given the input text $x_i$ and its label $y_i$, we first detect and extract a set of VH-words. Then, we adopt a visual knowledge augmentation module to enhance the visual background knowledge of their tokens and generate their visually-aligned representations. Finally, we infuse the visually-aligned text representations into the PLM to improve its fine-tuning performance, where we consider both the general fine-tuning of small PLMs and the parameter-efficient fine-tuning of large-scale PLMs.

## 3.2 Visually-Hungry Words Extraction

In our approach, visually-hungry words (VH-words) are the trigger units for visual augmentations, requiring visual knowledge for deriving complete semantic representations (*e.g.,* color, shape, and object). Therefore, we propose to first detect the VH-words from the input text, and then inject the proper visual knowledge that they are hungry for into the PLM. However, the annotations about VH-words are generally not available in NLP datasets. To address this problem, we devise three different strategies to extract the VH-words from the input text, including two feature-based strategies based on syntax tree and attention distribution of PLMs, and a learnable model-based strategy.

**Syntax-based Strategy.** In natural language, entity words and descriptive words usually convey more visual semantics than others. For example, for the sentence "*He is eating a green apple*", where underlined words are more related to visual semantics. Such words are mostly nouns or adjectives in the input text, which can be detected by syntactic analysis. Therefore, we design a rule-based strategy that leverages the syntactic information for

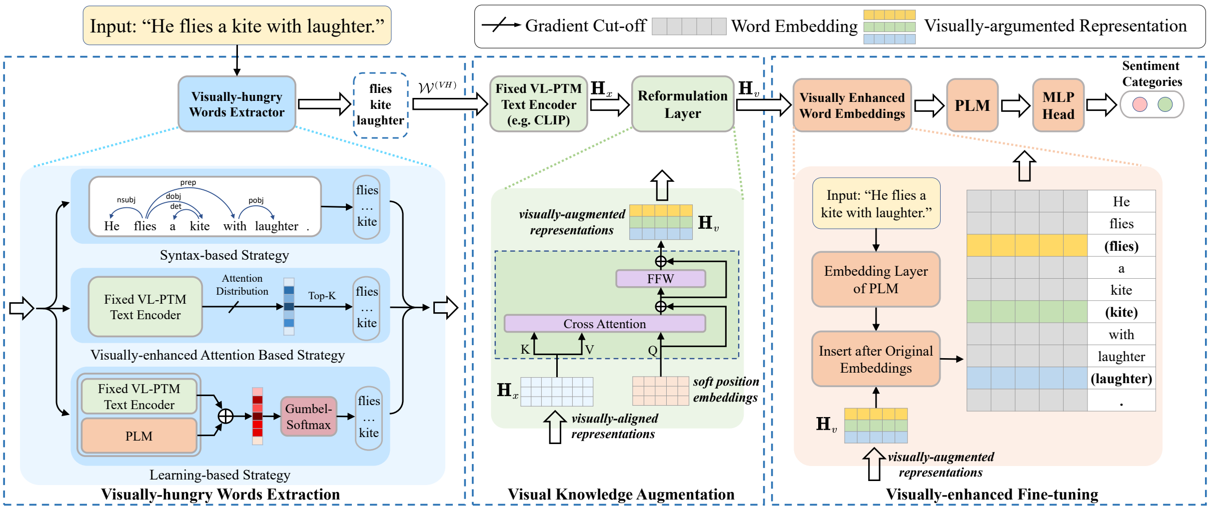


Figure 1: The illustration of our VAWI approach, consisting of visually-hungry words extraction, visual knowledge augmentation and visually-enhanced fine-tuning.

VH-words extraction. Concretely, we first delete all stop words in a text and then adopt an open-resource toolkit SPACY [2] to convert the input text into a syntax dependency tree. Based on the syntax tree, we extract the words that have a particular part of speech (POS), *e.g.,* nouns or adjectives, as the VH-words denoted by $\mathcal{W}^{(VH)}$. In this way, we can efficiently extract the VH-words from input text by using a fast parser toolkit.

**Visually-enhanced Attention Based Strategy.** The attention-based strategy utilizes the attention distribution of a VL-PTM to detect the VH-words. Since VL-PTMs (Radford et al., 2021) are pre-trained on large-scale image-text pairs, their text encoders can focus more on the words corresponding to some specific visual concepts in an image, which are likely to be VH-words. Inspired by it, we use the attention scores calculated by the text encoder of VL-PLMs to select the VH-words. Specifically, we adopt the text encoder of CLIP (Radford et al., 2021), a VL-PTM that has been pre-trained on millions of image-text pairs, to help extract the VH-words. As CLIP adopts an autoregressive GPT-2 model as the text encoder, we calculate the average attention scores between each token and the "`[EOS]`" token on the self-attention layer, denoted as $s_{w_i}$. Then, we select the top-$K$ ranked words according to $\{s_{w_i}\}$ as the VH-words $\mathcal{W}^{(VH)}$.

**Learning-based Strategy.** Considering that diverse PLMs and NLP tasks may be hungry for different complementary visual information, we devise a learning-based strategy that can adaptively extract VH-words according to task requirements. Concretely, we add a parameterized VH-words extractor layer for the PLM, which can be updated by gradient-based optimization algorithms to fit the need for some specific task. Given the input text $x_i$, we first leverage the PLM and a text encoder of a VL-PTM (*i.e.,* CLIP (Radford et al., 2021)) to produce the contextualized representations of the contained words in $x_i$. Then, we concatenate the representations of each word from the two models and utilize a MLP layer to obtain the score $s_{w_i}$:

$$s_{w_i} = \text{MLP}([\mathbf{h}_{w_i}^{(P)}; \mathbf{h}_{w_i}^{(V)}]) \tag{1}$$

where $\mathbf{h}_{w_i}^{(P)}$ and $\mathbf{h}_{w_i}^{(V)}$ are the output word representations from the PLM and VL-PTM, respectively, and scores $s_{w_i}$ are calculated by the learned model based on the supervision information from downstream tasks. Based on the scores of all words, we incorporate the gumbel-softmax function (Jang et al., 2016) to extract the top-$k$ words as the VH-words in a differentiable way. In this way, the gradients of the fine-tuned tasks can be back-propagated to the extractor layer, which learns to adaptively select the more suitable VH-words.

### 3.3 Visual Knowledge Augmentation

Existing works (Lu et al., 2022; Wang et al., 2022) mainly utilize image retrieval or generation module to augment related visual knowledge. Such a way is time-consuming and may also involve noisy images.Inspired by recent works that show the effective visual-language alignment in VL-PTMs (Rad-

[2] `https://spacy.io/`

ford et al., 2021; Li et al., 2021), we utilize the visually-aligned text encoders to generate the visual augmentation representations of VH-words. As the text encoders have been aligned to the image encoders during pre-training, their output textual representations can be used as surrogates of visual augmentations based on real images related to the input text. As will be shown in experiments (Section 4), this approach is not only efficient but very effective for downstream NLP tasks.

Based on the extracted VH-words, we first add a prefix text in the image caption style before the VH-words, *e.g.,* "*a photo of:* ", to compose the input text $x'$. Then, we utilize the text encoder of CLIP (Radford et al., 2021) to encode $x'$ and obtain the contextualized word representations as the visually-aligned representations $\mathbf{H}_x \in \mathbb{R}^{k \times d}$, where $k$ is the sequence length of $x'$ and $d$ is the embedding size. Next, we incorporate a reformulation layer to aggregate and strengthen the visually-aligned representation $\mathbf{H}_x$ into the visually-augmented representations of these VH-words. As the positions of the VH-words vary from sentence to sentence, we design a position-aware attention mechanism in the reformulation layer to inject position information into $\mathbf{H}_x$ for obtaining the visual representation of each VH-word. Specifically, we first leverage a soft position embedding matrix $\mathbf{E} \in \mathbb{R}^{l \times d}$ to reserve the position information of VH-words, where $l$ is the number of VH-words. Then, we perform the cross-attention between it and the visual representations as:

$$\mathbf{Q} = \mathbf{E}, \ \ \mathbf{K} = \mathbf{H}_x \mathbf{W}^K + \boldsymbol{b}^K, \tag{2}$$

$$\mathbf{V} = \mathbf{H}_x \mathbf{W}^V + \boldsymbol{b}^V, \tag{3}$$

$$\mathbf{H}_v = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \tag{4}$$

$$\mathbf{H}_v^\top = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_l], \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^d$, $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{k \times d}$. $\mathbf{H}_v \in \mathbb{R}^{l \times d}$ is the obtained visually-augmented representations of VH-words, which is leveraged for augmenting the visual knowledge of the PLM. $\mathbf{h}_i$ is the visual representation of the $i$-th VH-word in $\mathcal{W}^{(VH)}$. Note that in Eq. 2 and 3, we adopt an efficient way that only uses the position information to set the *query* matrix $\mathbf{Q}$, and the visual semantics are mainly captured and injected through the *key* and *value* matrices.

### 3.4 Visually-Enhanced Fine-tuning

After obtaining the visually-augmented representations of VH-words (*i.e.,* $\mathbf{H}_v$ in Eq. 5), we propose a visually-enhanced fine-tuning strategy to inject the captured visual knowledge. Here, we consider two cases: (1) full-parameter fine-tuning for small PLMs, and (2) parameter-efficient prompt-tuning for large-scale PLMs. Before introducing the learning method, we simply review the parameters of our approach, consisting of the parameters in the underlying PLM ($\Theta_{plm}$), the VL-PTM ($\Theta_{vlp}$) and the parameters of the reformulation layer ($\Theta_{ref}$). Note that we will always fix $\Theta_{vlp}$ in our approach.

**Fine-tuning for Small PLMs.** For small PLMs, we can perform full-parameter fine-tuning, which updates both $\Theta_{plm}$ and $\Theta_{ref}$. Specifically, given the visually-augmented representations $\mathbf{H}_v$ of VH-words, we directly incorporate them into the embedding layer of the PLM. For each VH-word, we insert its visually-augmented representation after the original word embedding, to leverage the visual semantics to enrich the word representations.

**Prompt-tuning for Large-Scale PLMs.** For large-scale PLMs, we fix the parameters in it, *i.e.,* $\Theta_{plm}$, and employ a parameter-efficient prompt-tuning way to optimize it on downstream NLP tasks. Concretely, given the visually-augmented representations $\mathbf{H}_v$ of VH-words, we directly insert them before the input representations of every layer of PLMs. Then, following the typical prompt-tuning paradigm (Li and Liang, 2021), we only tune the parameters of the reformulation layer (*i.e.,* $\Theta_{ref}$) as the soft prompts to adapt all the model into the fine-tuning task.

Our approach can be generally applied to various PLMs (*e.g.,* BERT (Devlin et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020)) and NLP tasks (natural language understanding and text generation). Unlike other complicated visually-augmented methods (Tan and Bansal, 2020; Wang et al., 2022), it is more efficient, without the explicit need of external images or generation model; and meanwhile, it only introduces a small number of parameters (Eq. 3), which are easier to learn.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We conduct experiments on four types of tasks. (1) Natural Language Understanding (NLU): we extract 6 datasets from the GLUE benchmark (Wang et al., 2018); (2) Commonsense reasoning: we select CommonsenseQA (Talmor et al.,

| Base Model | Method | SST-2 | QNLI | QQP | MNLI | MRPC | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|
| CLIP | +None | 73.3 | 74.5 | 72.8 | 68.4 | 74.3 | 73.8 | 72.85 |
| BLIP | +None | 76.3 | 77.4 | 78.8 | 72.5 | 77.8 | 76.4 | 76.53 |
| ALBEF$_{14M}$ | +None | 78.9 | 78.2 | 79.4 | 73.4 | 76.5 | 77.5 | 77.31 |
| BERT$_{base}$ | +None | 89.3 | 87.9 | 87.2 | 79.4 | 81.7 | 84.4 | 84.98 |
| | +VOKEN | 92.2 | 88.6 | 88.6 | 82.6 | 83.5 | 86.0 | 86.83 |
| | +iACE | 91.7 | 88.6 | 89.1 | 82.8 | 85.8 | 86.6 | 87.43 |
| | +VAWI-SBS | **92.9** | 88.4 | 89.6 | 82.2 | 85.5 | 86.9 | 87.58 |
| | +VAWI-VABS | 92.7 | 88.9 | 89.5 | 82.7 | **85.8** | **87.2** | **87.80** |
| | +VAWI-LBS | 92.4 | **89.1** | **89.7** | **83.0** | 85.6 | 86.9 | 87.78 |
| RoBERTa$_{base}$ | +None | 89.2 | 87.5 | 86.2 | 79.0 | 81.4 | 85.4 | 84.78 |
| | +VOKEN | 90.5 | 89.2 | 87.8 | 81.0 | 87.0 | 86.9 | 87.06 |
| | +iACE | 91.6 | 89.1 | **87.9** | 82.6 | 87.7 | 86.9 | 87.63 |
| | +VAWI-SBS | 91.4 | 89.4 | 87.7 | 82.2 | 88.2 | 87.7 | 87.76 |
| | +VAWI-VABS | **91.7** | 89.1 | **87.9** | **82.6** | 88.3 | 88.1 | 87.95 |
| | +VAWI-LBS | 91.6 | **90.6** | **87.9** | 82.4 | **88.5** | **88.3** | **88.21** |

Table 1: Performance comparison of different methods on NLU tasks, the **BEST** results are highlighted in bold. +*None* denotes that we directly fine-tune the backbone without adding visual information. SBS, VABS, and LBS represent using the syntax-based strategy, visually-enhanced attention based strategy, and learning-based strategy in our approach, respectively. The results of VOKEN and iACE on GLUE are reported from Lu et al. (2022).

2019), a 5-way multiple choice QA dataset that requires commonsense knowledge; (3) Text generation: we select CommonGen (Lin et al., 2019b), a constrained text generation task about generative commonsense reasoning. (4) Cross-modal reasoning: we select SNLI-VE (Xie et al., 2019), to evaluate the capacity of predicting whether the image semantically entails the text.

**Baseline Models.** We compare our approach with the following baselines, including pre-trained language models (PLMs), visual-language pre-trained models (VL-PTMs), and visually-augmented pre-trained language modes (VaLMs). (1) **PLMs**: We choose BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020) as the PLM backbones, and directly fine-tune them as baselines. (2) **VL-PTMs**: We select ALBEF (Li et al., 2021), BLIP (Li et al., 2022), and CLIP (Radford et al., 2021), which have been pre-trained on large-scale image-text pairs. (3) **VaLMs**: we select VOKEN (Tan and Bansal, 2020) and iACE (Lu et al., 2022), which introduce the visual information into PLMs by pre-training on retrieved images and fine-tuning on generated images, respectively.

**Implementation Details.** We implement all methods based on Huggingface Transformers (Wolf et al., 2020). For all baselines, we set their hyper-parameters according to their papers. In our approach, we leverage the text encoder of CLIP (ViT-B/32) to implement the learnable model-based VH-words extractor and generate the visual representations of VH-words in the visual knowledge augmentation module. The hidden size of visual representations is set to 512. For different NLP tasks, we tune the number of visually hungry words in {2, 3, 4, 5}. During fine-tuning, we perform parameter-efficient tuning on T5-3b and BART-Large, and full-parameter tuning on other PLMs. For all tasks and all backbones, we utilize Adam as the optimizer, set the learning rate to 2e-5, weight decay to 0.01, and a linear warmup for the first 6% steps. For GLUE, GommonGen, and SNLI-VE datasets, we fine-tune our model for 3 epochs with a batch size of 32. For CommonsenseQA, we tune our model for 10 epochs with a batch size of 32. We use the cross-entropy loss for classification and the mean squared error loss for regression.

## 4.2 Main Experimental Results

In this part, we conduct a series of experiments on NLU, commonsense reasoning, text generation, and cross-modal commonsense reasoning tasks.

**Evaluation on NLU Tasks.** We present the experimental results of different methods on 6 NLU tasks in Table 1. First, we observe that VL-PTMs perform worse than PLMs, a possible reason is that

| Base Model | Method | CommonsenseQA-3k | | | | CommonsenseQA | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 5% | 10% | 20% | 100% | 5% | 10% | 20% | 100% |
| RoBERTa$_{base}$ | +None | 41.88 | 46.04 | 50.58 | 61.88 | 44.88 | 50.04 | 57.08 | 67.90 |
| | +Images | 42.37 | 48.09 | 52.81 | 64.22 | 45.72 | 51.17 | 58.96 | 69.64 |
| | +VAWI-SBS | **42.94** | **49.27** | **53.97** | **65.10** | **46.51** | **52.44** | **59.87** | **71.01** |
| RoBERTa$_{large}$ | +None | 48.39 | 56.30 | 59.06 | 74.19 | 51.24 | 59.95 | 65.52 | 76.65 |
| | +Images | 49.55 | 57.78 | 61.29 | 75.61 | 52.18 | 60.93 | 66.08 | 78.39 |
| | +VAWI-SBS | **50.27** | **58.17** | **62.22** | **76.54** | **52.98** | **61.97** | **67.40** | **79.19** |
| T5-3B | +None | 70.16 | 73.02 | 75.04 | 81.81 | 71.99 | 75.27 | 77.72 | 82.40 |
| | +Images | 70.96 | 73.60 | 75.91 | 82.40 | 72.87 | 76.17 | 78.71 | 83.64 |
| | VAWI-SBS+PET | **71.52** | **74.19** | **76.49** | **83.61** | **73.58** | **73.58** | **79.66** | **84.56** |

Table 2: Performance comparison on CommonsenseQA-3k and CommonsenseQA with different amounts of training data. We report the average performance on the dev set over three runs, and the **BEST** results are highlighted in bold. *+Images* denotes that we add retrieved images about the VH-words using web search engines, and encode them via CLIP-ViT.

| Method | Base Model | BLUE-3 | BLUE-4 | METOR | Rouge-L | CIDER | SPICE |
|---|---|---|---|---|---|---|---|
| BART-large | +None | 42.80 | 32.42 | 31.36 | 57.57 | 16.56 | 32.94 |
| | +Images | 42.67 | 32.67 | 32.12 | 57.46 | 16.78 | 32.81 |
| | +VAWI-SBS | **44.56** | **34.17** | **32.47** | **58.46** | **17.23** | **33.67** |
| | +VAWI-SBS+PET | 43.12 | 33.76 | 32.20 | 58.12 | 16.91 | 33.17 |
| T5-3b | +None | 45.92 | 35.92 | 33.02 | 58.57 | 17.71 | 33.51 |
| | +Images | 45.69 | 35.50 | 33.55 | 58.94 | 17.51 | 32.91 |
| | +VAWI-SBS | **47.67** | **37.54** | 33.41 | **59.94** | **18.34** | **34.67** |
| | +VAWI-SBS+PET | 47.40 | 37.36 | **33.71** | 59.78 | 18.18 | 34.17 |

Table 3: Performance comparison on CommonGen. We also show the performance of parameter-efficient tuning of our approach, denoted as *+PET*. The **BEST** results are highlighted in bold.

they have been continually pre-trained on large-scale image-text pairs, which may cause the catastrophic forgetting problem. Second, VaLMs (*i.e.,* VOKEN, iACE, and VAWI) achieve better performance over PLMs. As VaLMs infuse external visual knowledge into the PLMs, they can help the PLMs better understand the background knowledge of some words (*e.g.,* color, shape, and size of objects). Between the two VaLM baselines, iACE is slightly better. This is because iACE is enhanced based on VOKEN and incorporates an image generation model, so it produces more visual information to utilize. However, the generated images inevitably contain noise and redundant information, which limits the performance gain of iACE.

Finally, by comparing our approach with all baselines, it is obvious that VAWI performs consistently better than them on the six datasets. In our approach, we adopt an efficient and effective way that augments the visually-augmented representations using the text encoder of CLIP to encode the VH-words from the input text. Benefiting from pre-training on large-scale image-text pairs, the text encoder of CLIP has been well aligned with the semantic space of images, so that it can generate high-quality visually-augmented representations of the VH-words to enrich them. Such a way not only saves the costs of time and computation but also reduces the influence of inevitable noise from retrieved or generated images. Additionally, among three VH-words extraction strategies, LBS slightly outperforms others in most NLU tasks. The reason is that LBS incorporates a learnable model-based strategy to select the VH-words. Such a way can adaptively extract proper VH-words with the consideration of the intrinsic knowledge of the PLMs. However, LBS will increase the computation cost due to its involved learnable VH-words extractor layer. Therefore, for efficiency, in the following experiments, we utilize the SBS strategy in our approach for comparison.

**Evaluation on Commonsense Reasoning Tasks.** Following existing works (Lin et al., 2019a), we

| Method | SNLI-VE | | | |
|---|---|---|---|---|
| | 10% | 20% | 50% | 100% |
| ALBEF | 65.46 | 67.52 | 75.47 | 80.91 |
| ALBEF+VAWI +SBS | **65.94** | **68.23** | **76.14** | **81.64** |

Table 4: Results on the test set of SNLI-VE task. The **BEST** results are highlighted in bold.

also rely on a rule-based strategy to extract the examples containing visible objects, to construct a new dataset called CommonsenseQA-3K. It consists of 2,903 and 341 examples in the training set and dev set, respectively. Based on the CommonsenseQA and CommonsenseQA-3k, we also report the results with different amounts of training data, to further evaluate the performance of different methods in the few-shot setting.

As shown in Table 2, we can also see that with the help of the visual information from either retrieved images or our VAWI-SBS, the performance of PLMs can be improved significantly. It indicates that visual information is indeed helpful to improve PLMs for understanding commonsense knowledge. Besides, our approach outperforms the method using retrieved images from search engines. Our approach omits the image retrieval process due to its inevitably involved noise, and relies on the text encoder of CLIP to augment the visual representations. Such a way can guarantee the relevance between the augmented visual knowledge and the text input, reducing the influence of retrieved noisy images and redundant information. Furthermore, we also perform parameter-efficient tuning on T5-3B-encoder with our approach and boost its performance. It shows that our approach is able to be applied to large-scale PLMs to meet their thirst for visual information.

**Evaluation on the Text Generation Task.** As shown in previous experiments, it is useful to improve the performance of VAWI on commonsense reasoning and nature language understanding tasks. Here, we would like to study the effectiveness of our approach on the text generation task (*i.e.,* CommonGen) using large PLMs. As shown in Table 3, our model VAWI also consistently boosts the performance of BART-Large and T5-3b among all metrics. It further shows that our approach can also improve PLMs on the text generation task. As a comparison, we can see that the retrieved images are not very helpful and even cause performance degradation. The reason may be that the text generation task is more sensitive to the inevitable noise from the retrieved images. Finally, the parameter-efficient tuning strategy of our approach also achieves comparable performance with the full-parameter tuning. It indicates that our parameter-efficient strategy is able to efficiently optimize the parameters of large-scale PLMs, and shows a promising future to apply our approach to much larger PLMs, *e.g.,* GPT-3.

**Evaluation on the Cross-modal Commonsense Reasoning Task.** To verify the generality of our method, we further implement our VAWI on a VL-PTM (*i.e.,* ALBEF (Li et al., 2021)), and conduct experiments on a cross-modal reasoning dataset, SNLI-VE. Concretely we implement our approach on ALBEF by inserting the visually-augmented representations after the VH-words embeddings of the text encoder before the multimodal encoder, and keeping others unchanged. As shown in Table 4, our VAWI can also improve the performance of ALBEF using different amounts of training data. It further shows the generality of our approach in VL-PTMs, as it can also provide rich information to enhance the text encoder of VL-PTM, helping it better perform cross-modal reasoning.

### 4.3 Ablation Study

In this part, we conduct a series of experiments to verify whether the improvement of our approach derives from the augmented visual knowledge about the VH-words. More ablation studies are shown in Appendix A.

**The Effect of the Source of Visual Representations.** We first propose three variants that incorporate powerful PLMs, *i.e.,* RoBERTa-base, T5-Large, and T5-3b respectively, to replace the text encoder of CLIP in our framework. We also replace the generated visual representations from the text encoder of CLIP with random noise, to investigate the importance of the visual representations. As shown in Table 5, we can see that our approach is better than all the variants, even T5-3b with billion-scale parameters. It indicates that CLIP-base is more effective to augment visual knowledge to improve the performance of PLMs. Besides, our approach also outperforms the variant using random noise as the visual representation, showing the worse performance among all the variants. It also shows the importance of visual representations, as they indeed contain the visual knowledge that the

| Source of visual representation (Params) | CSQA-3k | CSQA | SST-2 | QQP | STS-B | QNLI |
|---|---|---|---|---|---|---|
| Random Noise (0M) | 61.59 | 66.78 | 89.13 | 86.27 | 85.13 | 87.22 |
| RoBERTa-large (355M) | 61.18 | 67.17 | 89.43 | 86.53 | 85.60 | 87.77 |
| T5-large-encoder (375M) | 62.21 | 67.87 | 89.71 | 86.67 | 86.40 | 87.94 |
| T5-3b-encoder (1500M) | 63.10 | 68.42 | 90.24 | 86.96 | 86.93 | 88.21 |
| CLIP-base (52M) | **65.10** | **71.07** | **91.41** | **87.72** | **87.67** | **89.40** |

Table 5: Performance comparison of different sources of visual representation in our approach. The base model is RoBERTa-base.

| The text encoder of different VL-PTMs (Params) | CSQA-3k | SST-2 | QQP |
|---|---|---|---|
| Random Noise (0M) | 61.59 | 89.23 | 86.21 |
| ALBEF (110M) | 63.34 | 90.72 | 87.17 |
| CLIP-base (52M) | 65.10 | 91.41 | 87.72 |
| UniCL-base (52M) | 65.98 | 91.75 | 88.07 |
| CLIP-large (123M) | **66.27** | **92.10** | **88.31** |

Table 6: Performance comparison of visual representations from different VL-PTMs in our approach. The base model is RoBERTa-base.

PLM is hungry for.

**The Effect of the Stronger VL-PTMs.** In our work, we choose CLIP-base to enhance PLMs, as it has been pre-trained on a large-scale image-text dataset. Generally, a stronger VL-PTM would be more promising to further improve the performance. Here, we replace our CLIP-base model with some stronger VL-PTMs, *e.g.,* ALBEF (Li et al., 2021), UniCL-base (Yang et al., 2022), and CLIP-large. Concretely, ALBEF leverages more pre-training tasks (*e.g.,* MLM, ITM, and ITC), UniCL utilizes more high-quality pre-training data, and CLIP-large increases the scale of model parameters. We evaluate the above variations on CSQA-3k, QQP, and SST-2, and the results are shown in Table 6. We can see that UniCL and CLIP-large outperform CLIP-base. It indicates that the VL-PTMs with the larger scale of model parameters or more high-quality pre-training data are more capable of augmenting useful visual knowledge for PLMs. Considering the efficiency, CLIP-base is also a good choice in our approach, and we will investigate more proper VL-PTMs in the future.

## 5 Conclusion

In this paper, we proposed a general visually-augmented fine-tuning approach that can be applied to a variety of PLMs and NLP tasks, without using any retrieved or generated images, namely **VAWI**. Specifically, we first identified and extracted the visually-hungry words (VH-words) from input text via a token selector, where three different methods have been proposed, including syntax-, attention- and learning-based strategies. Then, we adopted a fixed VL-PTM text encoder to generate the visually-augmented representations of these VH-words. As it has been pre-trained by visual-language alignment tasks on the large-scale corpus, it is capable of injecting visual semantics into the aligned text representations. Finally, we transformed the visually-aligned features into visually-augmented features by reformulation layer based on VH-words, and inserted them into PLMs to enrich the visual semantics of word representations in PLMs. Experimental results on 10 NLP tasks show that our approach can consistently improve the performance of BERT, RoBERTa, BART, and T5 at different scales, and outperform several competitive baselines significantly. Besides, the visual prompts of our framework can also be used for parameter-efficient tuning, which can boost the performance of large language models, such as T5-3b.

## Limitations

An important limitation of our approach VAWI is the need for extracting visually-hungry words (VH-words) as the trigger to inject visual knowledge into PLMs. In real-world applications, it is hard to obtain the annotations of VH-words. Therefore, we propose three VH-words extraction strategies. However, the three strategies may be not always proper for all NLP tasks, and we rely on the experimental results to select the best one among them. Besides, we adopt the text encoder of CLIP as the

VL-PTM for generating the visually-aligned representation. As a pre-trained model, CLIP also may contain biases learned from the pre-training corpus, which may result in improper biased prediction on some NLP tasks.

## Acknowledgement

This work was partially supported by National Natural Science Foundation of China under Grant No. 62222215, Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. Xin Zhao is the corresponding author.

## References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Enabling multimodal generation on clip via vision-language knowledge distillation. *arXiv preprint arXiv:2203.06386*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.

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

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

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. *Advances in neural information processing systems*, 34:9694–9705.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *arXiv preprint arXiv:2203.02053*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019a. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019b. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. 2022. Things not written in text: Exploring spatial commonsense from visual signals. *arXiv preprint arXiv:2203.08075*.

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. *ArXiv*, abs/1907.11692.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Yujie Lu, Wanrong Zhu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2022. Imagination-augmented natural language understanding. *arXiv preprint arXiv:2204.08535*.

Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. 2021. The world of an octopus: How reporting bias influences a language model's perception of color. *arXiv preprint arXiv:2110.08182*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International*

*Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Hao Tan and Mohit Bansal. 2020. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. *arXiv preprint arXiv:2010.06775*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Weizhi Wang, Li Dong, Hao Cheng, Haoyu Song, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2022. Visually-augmented language modeling. *arXiv preprint arXiv:2205.10178*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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*.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. 2022. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173.

Chenyu Zhang, Benjamin Van Durme, Zhuowan Li, and Elias Stengel-Eskin. 2022. Visual commonsense in pretrained unimodal and multimodal models. *arXiv preprint arXiv:2205.01850*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Wanrong Zhu, An Yan, Yujie Lu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2022. Visualize before you write: Imagination-guided open-ended text generation. *arXiv preprint arXiv:2210.03765*.

# A Ablation Study

## A.1 Ablation Study on Visual Knowledge Augmentation

**The Effect of the Pre-trained Dataset of VL-PTMs.** We notice that the pre-training dataset of VL-PTMs is different from PLMs. Here, we investigate whether the captions or images from the large-scale image-text pairs contribute more to the performance gain of our approach. To verify it, we pre-train a new PLM only using the captions data. Following the setting of ALBEF, we utilize the pre-trained parameters of BERT to initialize this model and only extract the captions from the pre-training data of ALBEF (14.5M sentences in total). After pre-training on these captions until convergence, we utilize this model to replace CLIP-base in our approach and keep other settings unchanged. We conduct experiments on commonsense reasoning and NLU tasks to evaluate its effectiveness for augmenting visual knowledge. As shown in Table 7, we can see that such a variation underperforms ALBEF and our approach, and even leads to performance degradation on the CSQA task. It indicates that during pre-training the image data is an important resource for learning visual knowledge in VL-PTMs. Only text data (*i.e.,* captions) can not provide sufficient visual knowledge that PLMs are hungry for. Therefore, after pre-learned on large-scale text-image pairs, CLIP can absorb the useful visual knowledge from the images and inject them into PLMs in our approach. It further indicates that the improvement of our method is due to the involvement of the visual information about the VH-words.

## A.2 Ablation Study on Visually-enhanced Fine-tuning

**Different Insertion Positions of Visual Representations.** In our visually-enhanced fine-tuning framework, we insert the visual representation of the VH-word after its original word embedding. To verify its effectiveness, we propose three variants of it that do not insert, insert all visual representations of VH-words before and after the input text, respectively. As shown in Table 8, we can observe that all these variants would lead to a performance decrease. It demonstrates that a proper position to insert the visual representation is important for the utilization of augmented visual representations. By inserting them after the word embeddings of corresponding VH-words, PLMs can effectively aggregate the visual representations to enrich the word representations, leading to better performance on downstream NLP tasks.

# B Further Analysis

**The Frozen CLIP's Text Encoder.** In the experiment presented in Table 1, we directly fine-tuned CLIP and the results indicate that the performance of VL-PTMs' text encoder is unsatisfactory when directly fine-tuned on NLP tasks. In our VAWI, we fix the model parameters of CLIP-base's text encoder to preserve the visual knowledge. Hence we also conduct experiments on four NLU tasks from GLUE using frozen CLIP. Specially, we fix CLIP-base's text encoder and only fine-tuned added 4 transformer layers above it. As shown in Table 9, we can see that CLIP's performance under this setting is better than that of directly full-parameter fine-tuning CLIP and also underperforms RoBERTa and BERT. It indicates that fixing CLIP is more suitable for NLP tasks, and shows the rationality of VAWI settings that always fix the CLIP's text encoder in VAWI to preserve CLIP's knowledge.

**The Computation Latency of the Proposed Methods.** In our VAWI, we fix the model parameters of CLIP-base to preserve the visual knowledge. Such a way can also decrease the computation costs during training and inference. To verify it, we report the mean training and inference latency per batch on the CSQA-3k dataset of our method and baselines on RTX3090 GPU, where all these methods utilize RoBERTa-base as the backbone. As shown in Table 10, we can see that our proposed VAWI-SBS and VAWI-VABS would not increase the latency too much. For VAWI-LBS, as it requires a PLM and a VL-PTM to adaptively select the VH-words, it will relatively increase the latency. As shown in Table 1, we can see that all the three variants achieve comparable performance in 6 NLU datasets. Therefore, it is more efficient and effective to select the SBS and VABS variations in our approach. Despite it, we can see that all our variants own less latency than iACE, since our approach does not require a time-consuming image generation process. And as shown in Table 1, our approach can also achieve better performance.

**The Effect of the Improper Visually-hungry Words.** To analyze how the quality of the VH-

| The text encoder of different VL-PTMs (Params) | CSQA-3k | CSQA | SST-2 | STS-B | MNLI |
|---|---|---|---|---|---|
| None | 61.59 | 67.90 | 89.23 | 85.46 | 79.06 |
| BERT pre-trained on captions (110M) | 62.17 | 67.56 | 89.58 | 85.73 | 79.24 |
| ALBEF (110M) | 63.64 | 68.47 | 90.72 | 87.17 | 80.86 |
| CLIP-base (52M) | **65.10** | **71.07** | **91.41** | **87.73** | **82.27** |

Table 7: Performance comparison of visual representations pre-trained using different pre-training data in our approach. The base model is RoBERTa-base.

| Insert Positions | CSQA-3k | | | |
|---|---|---|---|---|
| | 5% | 10% | 20% | 100% |
| Not insert | 41.88 | 46.04 | 50.58 | 61.88 |
| Before input text | - | 39.77 | 44.86 | 57.47 |
| After input text | - | 40.23 | 45.67 | 58.08 |
| After the VH-words | **42.94** | **49.27** | **53.97** | **65.10** |

Table 8: Performance comparison w.r.t. different insertion positions of visual representations. The base model is RoBERTa-base.

| | SST-2 | QNLI | QQP | STS-B |
|---|---|---|---|---|
| CLIP-base | 73.3 | 74.5 | 72.8 | 73.8 |
| Fixed CLIP-base | 75.1 | 76.9 | 73.7 | 75.2 |

Table 9: The effect of fixed CLIP's text encoder.

| Method | Training Time (s) | Inference Time (s) |
|---|---|---|
| RoBERTa-base | 0.506 | 0.182 |
| +Voken | 0.506 | 0.182 |
| +iACE | 1.138 | 0.512 |
| +VAWI-SBS | 0.587 | 0.241 |
| +VAWI-VABS | 0.680 | 0.308 |
| +VAWI-LBS | 0.893 | 0.486 |

Table 10: The computation latency during training and inference.

| Correct VH-words proportions | CSQA-3k | SST-2 | QQP |
|---|---|---|---|
| 0 % | 61.60 | 89.57 | 87.63 |
| 20 % | 62.17 | 89.44 | 87.40 |
| 50 % | 64.22 | 91.73 | 89.20 |
| 100 % | **65.10** | **92.93** | **89.74** |
| None | 61.88 | 89.23 | 86.21 |

Table 11: The effect of the improper visually-hungry words. The base model is RoBERTa-base.

words affects the performance of our approach, we further conduct the experiments on CSQA-3K and two NLU tasks SST-2 and QQP from GLUE, to show the effect of insufficient VH-words on our model performance. After extracting the VH-words, we remove part of them and only randomly sample 0%, 20%, and 50% VH-words for augmentation. As shown in Table 11, we can see that with the decreasing of the sampling probability, the performance of our approach degrades gradually. It indicates that not enough VH-words would degrade the performance of our approach.

**The Number of VH-words.** Our approach has an important hyper-parameter required to tune, such as the number of VH-words. VH-words can supply visual knowledge that PLMs may be hungry for. Here, we would like to study whether more VH-words are better to improve performance. We conduct experiments on the QQP and CSQA-3K datasets using RoBERTa-base as the backbone, and present the results in Figure 2. We can see that with the increase of the number of VH-words, the performance gain of our approach first increases and then decreases. A possible reason is that too many VH-words may also introduce noisy or redundant information (*e.g.,* not very relevant words), which would also influence the fine-tuning performance. Instead, it is also more efficient to select a few VH-words (*e.g.,* two words for CSQA-3k) for deploying our approach in large-scale PLMs.

**Case Study of Extracted Visually-hungry Words.** In this part, we show the VH-words extracted by syntax-, attention- and learning-based strategies in Table 12, Table 13, Table 14 and Table 15. We can see that the three strategies would extract slightly different VH-words. The reason is that the three strategies are based on different techniques to identify the VH-words. As we can see, the cases show that most of the extracted VH-words by our strategies are generally related to some visual semantics, *e.g.,* spider, two eyes. Although such VH-words can not perfectly cover all the visual semantics, they actually contain most of the important words that the PLMs may be hungry for, *e.g.,* red and yellow. Besides, we can also see that the VH-words extracted by our three strategies may not perfectly align with human judgment. In fact, it is also hard for humans to determine proper rules to identify VH-words, *e.g.,* people, human, and water. In addition, as the learned knowledge of PLM is a black

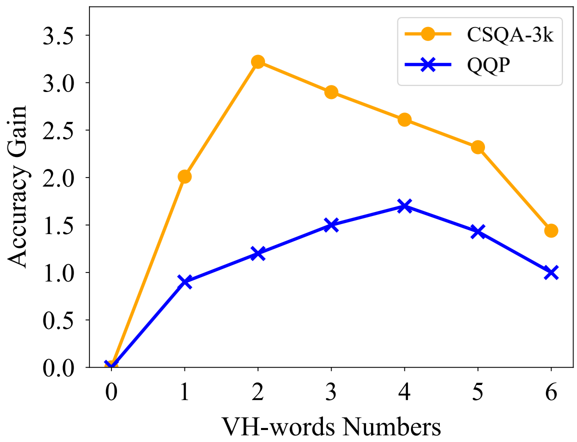


Figure 2: Performance comparison w.r.t. different numbers of VH-words.

box, it is also difficult for humans to judge the usefulness of our extracted VH-words for PLMs.

**The Interpretability of Augmented Embeddings.** In this part, we show how our augmented embeddings infuse visual knowledge into the PLM. Concretely, we show the attention distributions of a PLM (*i.e.,* RoBERTa-base) in the last few layers before and after infusing visually-augmented representations on CSQA. As shown in Table 16, we can see that the [CLS] tokens pay more attention to the VH-words and their visually-augmented representations, and the VH-words also pay more attention to their visually-augmented representations. It shows that the injected visually-augmented representations provide useful knowledge, which guides the PLM to focus on more important tokens and also improves the representations of the VH-words and the [CLS] token.

| |
|---|
| **Input**<br>*Input sentence:* Unlike a spider and his many sight seers, people only have what? two eyes. |
| **Syntax-based Strategy**<br>Unlike a spider and his many sight seers, people only have what? two eyes |
| **Visually-enhanced Attention Based Strategy**<br>Unlike a spider and his many sight seers, people only have what? two eyes. |
| **Learning-based Strategy**<br>Unlike a spider and his many sight seers, people only have what? two eyes. |

Table 12: The first instance from the CommonsenseQA dataset. The extracted visually-hungry words are highlighted in green.

| |
|---|
| **Input**<br>*Input sentence:* Where on a river can a human hold a cup upright to catch water on a sunny, clear day? waterfall. |
| **Syntax-based Strategy**<br>Where on a river can a human hold a cup upright to catch water on a sunny, clear day? waterfall. |
| **Visually-enhanced Attention Based Strategy**<br>Where on a river can a human hold a cup upright to catch water on a sunny, clear day? waterfall. |
| **Learning-based Strategy**<br>Where on a river can a human hold a cup upright to catch water on a sunny, clear day? waterfall. |

Table 13: The second instance from the CommonsenseQA dataset. The extracted visually-hungry words are highlighted in green.

| |
|---|
| **Input**<br>*Input sentence:* the mesmerizing performances of the leads keep the film grounded and keep the audience riveted. |
| **Syntax-based Strategy**<br>the mesmerizing performances of the leads keep the film grounded and keep the audience riveted. |
| **Visually-enhanced Attention Based Strategy**<br>the mesmerizing performances of the leads keep the film grounded and keep the audience riveted. |
| **Learning-based Strategy**<br>the mesmerizing performances of the leads keep the film grounded and keep the audience riveted. |

Table 14: The instance from the SST-2 dataset. The extracted visually-hungry words are highlighted in green.

| |
|---|
| **Input**<br>*Input sentence:* How do I sell dry Moringa leaves powder in Indian market? Can I use the moringa leaves that are already starting to turn yellow or yellowish? |
| **Syntax-based Strategy**<br>How do I sell dry Moringa leaves powder in Indian market? Can I use the moringa leaves that are already starting to turn yellow or yellowish? |
| **Visually-enhanced Attention Based Strategy**<br>How do I sell dry Moringa leaves powder in Indian market? Can I use the moringa leaves that are already starting to turn yellow or yellowish? |
| **Learning-based Strategy**<br>How do I sell dry Moringa leaves powder in Indian market? Can I use the moringa leaves that are already starting to turn yellow or yellowish? |

Table 15: The instance from the QQP dataset. The extracted visually-hungry words are highlighted in green.

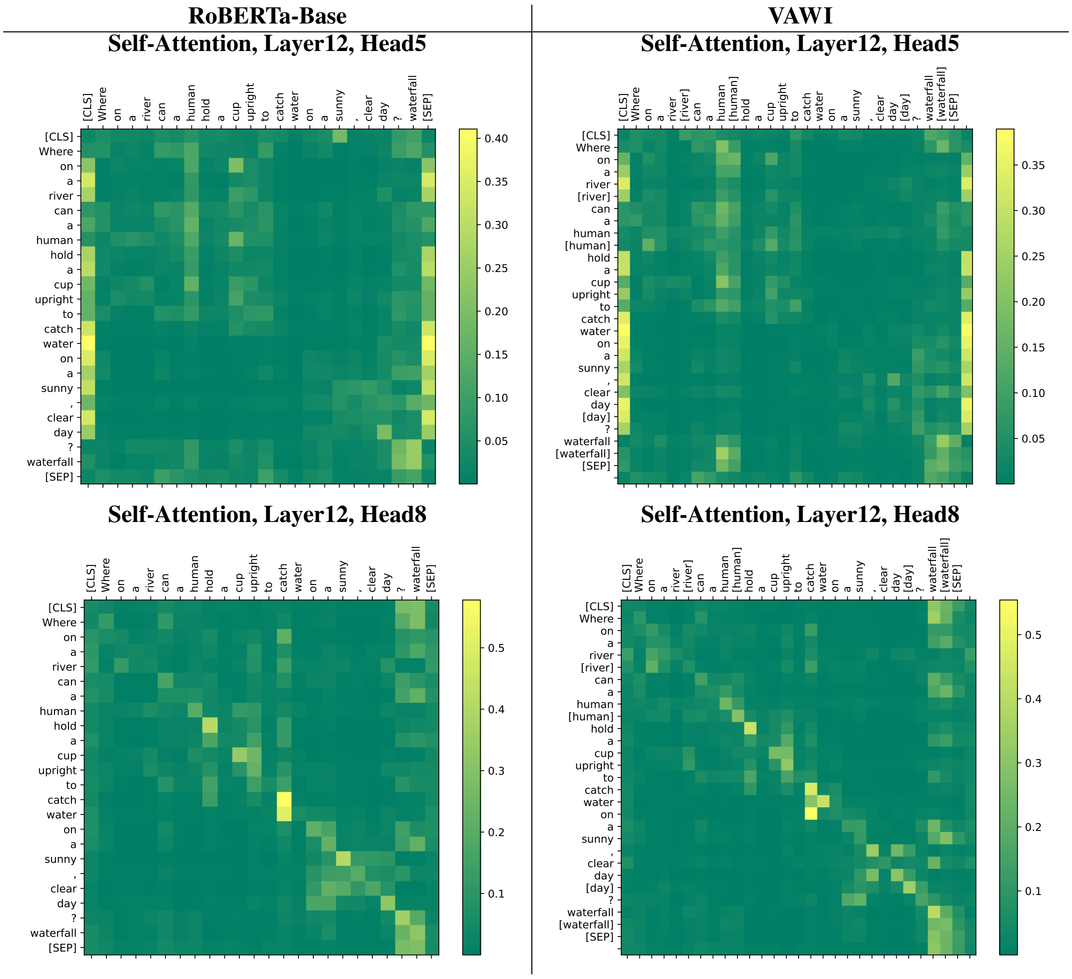


Table 16: The attention maps of the self-attention layers on RoBERTa-base and our approach.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Section 6.*

☑ A2. Did you discuss any potential risks of your work?
*The potential risks can be found in Section 6.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*The main claims can be found in Abstract and Section 5.*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*We introduce the dataset and pre-trained models used and the baselines in Section 4.*

☑ B1. Did you cite the creators of artifacts you used?
*We cite the creators of artifacts we used in Section 4.1.*

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*We use the default license of all artifacts in Section 4.*

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*We use all artifacts with their default intended use in Section 4.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Except for the dataset we created ourselves in Section 4, the relevant statistics is not reported in any other datasets we use. The dataset we use exactly follows the amount and train/test/dev splits of data in the original dataset paper.*

### C ☑ Did you run computational experiments?

*Our computational experiments for evaluating our method can be found in Section 4.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*We report the number of parameters in the models used in a few experiments, which can be found in Section 4 and Appendix. In addition, we report the total computation budget in Section C of the Appendix.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*We talk about them in Section 4.1.*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*We report descriptive statistics about our results, which can be found in Section 4.*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*We report these important settings in Section 4.*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**
*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*