VL-Match: Enhancing Vision-Language Pretraining with Token-Level and Instance-Level Matching

Junyu Bi[1,2] Daixuan Cheng[3] Ping Yao[1,2] Bochen Pang[3] Yuefeng Zhan[3]
Chuanguang Yang[1,2] Yujing Wang[3] Hao Sun[3] Weiwei Deng[3] Qi Zhang[3]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Microsoft Corporation
{bijunyu, yaoping, yangchuanguang}@ict.ac.cn daixuancheng6@gmail.com
{bopa, yuefzh, yujwang, hasun, dedeng, qizhang}@microsoft.com

Abstract

*Vision-Language Pretraining (VLP) has significantly improved the performance of various vision-language tasks with the matching of images and texts. In this paper, we propose VL-Match, a **V**ision-**L**anguage framework with Enhanced Token-level and Instance-level **Match**ing. At the token level, a Vision-Language Replaced Token Detection task is designed to boost the substantial interaction between text tokens and images, where the text encoder of VLP works as a generator to generate a corrupted text, and the multimodal encoder of VLP works as a discriminator to predict whether each text token in the corrupted text matches the image. At the instance level, in the Image-Text Matching task that judges whether an image-text pair is matched, we propose a novel bootstrapping method to generate hard negative text samples that are different from the positive ones only at the token level. In this way, we can force the network to detect fine-grained differences between images and texts. Notably, with a smaller amount of parameters, VL-Match significantly outperforms previous SOTA on all image-text retrieval tasks.*

1. Introduction

The pretrain-then-finetune paradigm has achieved great success in both natural language processing [7, 5, 24, 15] and computer vision [14, 3, 2, 27]. Vision-Language Pretraining (VLP) [22, 44, 18, 21, 39] has also attracted much attention in recent years, which aims to pretrain a model that can understand and align the semantics of images and texts through a variety of pretraining tasks based on massive image-text pairs. These models can be finetuned to adapt to various downstream vision-language tasks such as visual question answering [12] and image-text retrieval [30, 23].

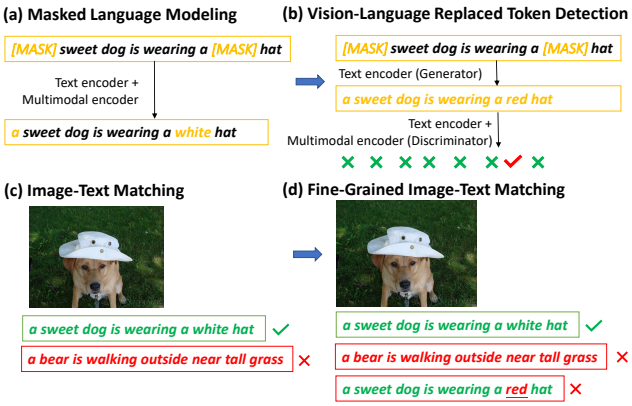

Figure 1. (a) Masked Language Modeling (MLM) predicts original tokens of the masked positions with image and text representations; (b) Vision-Language Replaced Token Detection (VLRTD) enhances token-level matching by discriminating whether each token in the generated text aligns with the image and the text context; (c) Image-Text Matching (ITM) predicts whether the given texts match the image; (d) Fine-Grained Image-Text Matching (FG-ITM) adds a fine-grained negative sample to enhance the matching ability at instance level.

To learn the matching between images and texts in vision-language pretraining, two pretraining tasks are commonly adopted to train a multimodal encoder [18, 9, 34]: Masked Language Modeling [35] tries to learn the token-level matching of different modalities by predicting original tokens of the masked positions with the image and text representations [22, 1]. The image and text representations are encoded with an image encoder and a text encoder respectively. Image-Text Matching attempts to match vision and language at instance level in a binary classification task, which predicts whether the given texts match the images [18, 22].

To further enhance vision-language matching at both token level and instance level, we propose VL-Match with two novel objectives: Vision-Language Replaced Token Detection (VL-RTD) to enhance the matching at the token level with a generator-discriminator structure, and Fine-Grained Image-Text Matching (FG-ITM) to enhance the Image-Text Matching task at instance level by introducing more hard negative samples. Inspired by the Replaced Token Detection task of ELECTRA [5] in natural language pretraining, VL-RTD is designed to discriminate whether each token in the text aligns with the image and the text context, with a generator-discriminator structure. Specifically, VL-RTD regards the multimodal encoder as a discriminator and the text encoder as a generator. Given an original text input, the generator outputs a corrupted text, and then the discriminator learns to discriminate whether each token in the corrupted text is replaced by the generator. Compared with Masked Language Modeling which corrupts the original text with `[MASK]`, VL-RTD corrupts the text with tokens selected from the vocabulary, preserving more semantic information of the original text. Different from Masked Language Modeling which only predicts on the masked tokens, VL-RTD predicts on all text tokens, thus forcing more text information to interact with the image. As shown in Figure 1 (a) and (b), with the multimodal encoder predicting on all tokens, VL-RTD can efficiently learn the connection between the unmasked "dog" in the text with the "dog" in the image.

Moreover, we also design FG-ITM to enhance the Image-Text Matching task at the instance level, by introducing more fine-grained negative samples. Previously, negative text samples of the Image-Text Matching task are sampled either randomly or according to instance-level similarities [21]. To present the differences between the positive and the negative samples in a fine-grained manner, we propose a novel data augmentation method named NegGen. In our method, we synthesize a new text instance by applying a language generator on masked tokens. The generated text is expected to be coherent in natural language, but has some fine-grained differences with the corresponding image. To ensure the synthesized pair to be negative, we further adopt a vision-language discriminator to predict image-text matching probabilities and to filter out potentially positive samples. For example, in Figure 1 (c) and (d), the term "white" in the positive sample is replaced with "red", formulating a fine-grained negative sample. In this way, the multimodal model is able to capture more fine-grained information for better image-text alignment.

In summary, our contributions include:

- We propose VL-Match to enhance the matching of images and texts at both token level and instance level, by designing two novel VLP pretraining tasks.
- We introduce a generator-discriminator structure pretrained with a Vision-Language Replaced Token Detection task to enhance the matching at the token level for vision-language pretraining.
- We are the first to bootstrap fine-grained negative samples in Image-Text Matching task to learn fine-grained representations for efficient vision and language alignment.
- As shown in experiments, on multiple cross-modal downstream tasks, VL-Match significantly outperforms previous SOTA on all retrieval tasks (up to 2.9% absolute improvement on Flickr30K dataset, and 1.9% on COCO dataset).

2. Related work

Inspired by the success of self-supervised representation learning on uni-modal tasks, Vision-Language Pretraining (VLP) has attracted much attention. VLP learns the representation and understanding of images and texts based on the training of massive image-text pairs. Model architectures, pretraining objectives and data augmentation are critical to the effectiveness of vision-language models.

Model Architectures Initial VLP methods [35, 25, 22, 44] adopt object detection models like Faster RCNN [32] to extract image region embeddings, which is expensive in both computation and data annotation. ViLT [18] removes the detector and leverages a Transformer encoder to encode both images and texts. Although these methods succeed in fusing vision and language information, a large computational effort on cross-modal retrieval tasks is resulted from the insufficiency in feature alignment of images and texts. Benefiting from the development of contrastive learning, dual-encoder models such as CLIP [31] and ALIGN [17] encode images and texts separately and try to align features of different modalities in the Image-Text Contrastive Learning task. However, these dual-encoder structure does not hold a fusion layer to enable the interaction between images and texts. Then ALBEF [21] proposes to align representations of vision and language first, and then uses a fusion layer to understand different modalities. Recent works [9, 10, 42] follow the idea of alignment before fusion and achieve superior performance. Our method shares similar spirits with ALBEF, but introduces a generator-discriminator structure to enhance the image and text matching at both token level and instance level.

Pretraining Objectives Multiple vision-language pretraining objectives have been proposed [1], including Masked Language Modeling [35, 37, 18], Image-Text

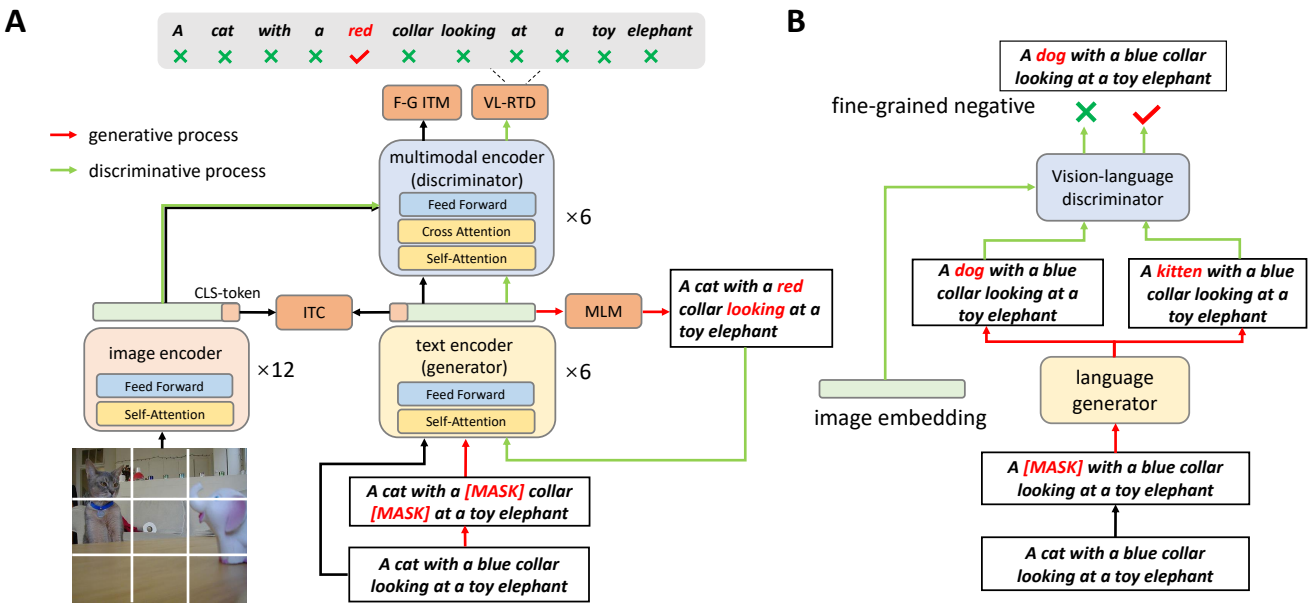

Figure 2. **(A)** Overview of VL-Match pretraining. The entire model includes an image encoder, a text encoder and a multimodal encoder. Pretraining objectives include ITC, MLM, VL-RTD, and FG-ITM. Some of the negative cases of FG-ITM are the output of (B). **(B)** Framework of NegGen. A masked text is fed into a language generator to generate a synthesized text that differs from the original text at the token level. To ensure the generated text is negative, we adopt a vision-language discriminator to predict image-text matching probabilities and to filter out potentially positive samples.

Matching [37, 18, 21, 39], Image-Text Contrastive Learning [31, 17], Prefix Language Modeling [40], Masked Region Classification [37], and Word-Patch/Region Alignment [4, 18]. SimVLM [40] proposes to train the vision-language model using Prefix Language Modeling on image-text pairs and text-only data. FLAVA [34] combines Masked Image Modeling with Masked Language Modeling, and uses Image-Text Contrast and Matching with a dual encoder + multimodal encoder structure. In this paper, we design two pretraining objectives based on our generator-discriminator structure, including Vision-Language Replaced Token Detection and Fine-Grained Image-Text Matching. Compared with Masked Language Modeling, Vision-Language Replaced Token Detection enhances token-level matching by capturing pure language priors in corrupted text input and involving more text tokens to interact with the image. And Fine-Grained Image-Text Matching supplements more fine-grained negative texts with a generate-then-filter bootstrapping method, to enhance the instance-level Image-Text Matching.

Data Augmentation Data Augmentation (DA) is widely applied in computer vision, especially in visual self-supervised pretraining [14, 2]. In recent years, DA is also helpful for visual-language pretraining: FILIP [43] uses back-translation to rewrite the original text to augment positive text samples. BLIP [20] introduces a decoder to generate synthetic positive captions for images. Different from these methods which focus on augmenting dataset with positive samples, our method demonstrates the advantage of fine-grained negative captions. We introduce NegGen that includes a language generator and a vision-language discriminator, where the generator outputs potentially negative texts that are different from the positive texts in a fine-grained manner, then the discriminator filters out the non-negatives according to the image-text matching probabilities.

3. Method

In this section, we first introduce our model structure, then elaborate on the proposed Vision-Langauge Replaced Token Detection task. Next, we briefly introduce the Image-Text Contrastive learning and Fine-Grained Image-Text Matching. Finally, we explain how the NegGen is designed.

3.1. Model Architecture

As shown in Figure 2, VL-Match contains a text encoder, an image encoder and a multimodal encoder. The text encoder and the multimodal encoder are both 6-layer transformers. We use a 12-layer vision transformer [8] as the image encoder. Given an image-text pair $(\boldsymbol{I}, \boldsymbol{T})$,

the input image $\boldsymbol{I}$ is encoded into a sequence of embeddings: $\{v_{cls}, v_1, \ldots, v_N\}$, where v_{cls} is the embedding of the `[CLS]` patch and N is the number of the image patches. The input text $\boldsymbol{T}$ is encoded into a sequence of embeddings $\{t_{cls}, t_1, \ldots, t_M\}$, where t_{cls} is the embedding of the `[CLS]` token and M is the maximum sequence length. The two different modal representations are fused by cross attention [38] of the multimodal encoder. Details of each component will be elaborated in the following sections.

3.2. Vision-Language Replaced Token Detection

Vision-Language Replaced Token Detection (VL-RTD) is divided into a generative process and a discriminative process. The two processes are described respectively as follows.

Generative Process In the generative process, we regard the text encoder as a generator to learn text representations according to the text context and generate corrupted texts (red arrow in Figure 2). Similar to Masked Language Modeling, the tokens in the selected positions are replaced with a `[MASK]` token, we denote this as $\mathrm{REPLACE}(\boldsymbol{T}, \boldsymbol{m}, \texttt{[MASK]})$, where $\boldsymbol{T}$ is the original text tokens and $\boldsymbol{m}$ is the selected positions. Given the masked text $\boldsymbol{T}^{\text{masked}}$, the text encoder learns to predict original tokens of the masked-out tokens. $p_{\mathrm{G}}(\boldsymbol{T}^{\text{masked}})$ denotes the predicted probability. Each token is sampled based on this probability to get a corrupted text $\boldsymbol{T}^{\text{corrupt}}$ without `[MASK]`. The generative process is formalized as follows. Typically $k = \lceil 0.15M \rceil$.

$$\begin{aligned} m_i &\sim \mathrm{unif}\{1, M\} \text{ for } i = 1 \text{ to } k \\ \boldsymbol{T}^{\text{masked}} &= \mathrm{REPLACE}(\boldsymbol{T}, \boldsymbol{m}, \texttt{[MASK]}) \\ \hat{T}_i &\sim p_G\left(T_i \mid \boldsymbol{T}^{\text{masked}}\right) \text{ for } i \in \boldsymbol{m} \\ \boldsymbol{T}^{\text{corrupt}} &= \mathrm{REPLACE}(\boldsymbol{T}, \boldsymbol{m}, \hat{\boldsymbol{T}}) \end{aligned} \tag{1}$$

The text encoder (which works as the generator) is trained to minimize a cross-entropy loss defined as H:

$$\mathcal{L}_{\mathrm{G}} = \mathbb{E}_{(\boldsymbol{T}^{\text{masked}})\sim D} \mathrm{H}\left(\boldsymbol{y}^{\text{masked}}, \boldsymbol{p}_{\mathrm{G}}(\boldsymbol{T}^{\text{masked}})\right) \tag{2}$$

where y_{masked} is the ground-truth of Masked Language Modeling: a one-hot vocabulary distribution where the original tokens at the masked positions are ones and the rest of the tokens are zeros.

Discriminative Process In the discriminative process, we regard the multimodal encoder as a discriminator to discriminate whether each token in the text aligns with the image and the text context (green arrow in Figure 2). Given the corrupted text $\boldsymbol{T}^{\text{corrupt}}$, the text encoder transforms the text into a corrupted text representation $\{t^c_{cls}, t^c_1, \ldots, t^c_M\}$, which is fed into the multimodal encoder to learn interactively with the image representation $\{v_{cls}, v_1, \ldots, v_N\}$ through cross attention [38]. Finally, the output of the multimodal encoder passes through a classification layer to obtain the binary probability distribution $p_{\mathrm{D}}(\boldsymbol{I}, \boldsymbol{T}^{\text{corrupt}})$ of each token.

The multimodal encoder is trained to minimize a binary cross-entropy loss defined as F:

$$\mathcal{L}_{\mathrm{D}} = \mathbb{E}_{(\boldsymbol{I}, \boldsymbol{T}^{\text{corrupt}})\sim D} \mathrm{F}\left(\boldsymbol{y}^{\text{corrupt}}, \boldsymbol{p}_{\mathrm{D}}(\boldsymbol{I}, \boldsymbol{T}^{\text{corrupt}})\right) \tag{3}$$

where $\boldsymbol{y}^{\text{corrupt}}$ is the ground-truth: a two-dimensional one-hot distribution formulated as

$$y_j^{corrupt} = 1 \quad \text{if } T_j^{\text{corrupt}} \neq T_j \text{ else } 0 \text{ for } j = 1 \text{ to } M \tag{4}$$

The training objective of VL-RTD is

$$\mathcal{L}_{\mathrm{rtd}} = \mathcal{L}_{\mathrm{G}} + \lambda \mathcal{L}_{\mathrm{D}} \tag{5}$$

where λ is the weight of the discriminator loss.

3.3. Image-Text Contrastive learning

We follow the same settings of Image-Text Contrastive learning (ITC) loss in ALBEF [21]. ITC loss aims to learn the alignment of image and text representations. Specifically, the similarity between image and text is calculated by the similarity function $s(\boldsymbol{I}, \boldsymbol{T}) = g_v(\boldsymbol{v}_{\mathrm{cls}})^\top g_t(\boldsymbol{t}_{\mathrm{cls}})$, where g_v and g_t are linear transformations that map $\boldsymbol{v}_{cls}$ and $\boldsymbol{t}_{cls}$ to normalized low-dimensional representations. Two queues are maintained to cache the most recently obtained Q image and text representations, which are calculated by a momentum text encoder and a momentum image encoder respectively [14]. The normalized features obtained from the momentum model are denoted as $g'_v(\boldsymbol{v}'_{\mathrm{cls}})$ and $g'_t(\boldsymbol{t}'_{\mathrm{cls}})$. $s(\boldsymbol{I}, \boldsymbol{T}^{\mathrm{mom}}) = g_v(\boldsymbol{v}_{\mathrm{cls}})^\top g'_t(\boldsymbol{t}'_{\mathrm{cls}})$ and $s(\boldsymbol{T}, \boldsymbol{I}^{\mathrm{mom}}) = g_t(\boldsymbol{t}_{\mathrm{cls}})^\top g'_v(\boldsymbol{v}'_{\mathrm{cls}})$ define the similarity functions between the positive representations from the pretraining encoders and the negative representations from the momentum encoders. For each image and text, we compute the softmax-normalized image-to-text and text-to-image similarities as:

$$\boldsymbol{p}^{\mathrm{i2t}}(\boldsymbol{I}) = \frac{\exp\left(s\left(\boldsymbol{I}, \boldsymbol{T}^{\mathrm{mom}}\right)/\tau\right)}{\sum_{q=1}^{Q} \exp\left(s\left(\boldsymbol{I}, \boldsymbol{T}^{\mathrm{mom}}\right)/\tau\right)} \tag{6}$$

$$\boldsymbol{p}^{\mathrm{t2i}}(\boldsymbol{T}) = \frac{\exp\left(s\left(\boldsymbol{T}, \boldsymbol{I}^{\mathrm{mom}}\right)/\tau\right)}{\sum_{q=1}^{Q} \exp\left(s\left(\boldsymbol{T}, \boldsymbol{I}^{\mathrm{mom}}\right)/\tau\right)} \tag{7}$$

where τ is a learnable temperature parameter. The ground-truths $\boldsymbol{y}^{i2t}_{\text{one-hot}}(\boldsymbol{I})$ and $\boldsymbol{y}^{t2i}_{\text{one-hot}}(\boldsymbol{T})$ are similarity matrices with the same shape as p^{i2t} and p^{t2i}, with ones on the diagonal and zeros on the rest. Momentum Distillation [21] leverages the momentum model to distill current training model, which is adopted to learn from pseudo-targets generated by the momentum model. The final targets are:

$$\boldsymbol{y}^{i2t}(\boldsymbol{I}) = (1-\alpha)\boldsymbol{y}^{i2t}_{\text{one-hot}}(\boldsymbol{I}) + \alpha \boldsymbol{p}^{i2t}(\boldsymbol{I}^{\mathrm{mom}}) \tag{8}$$

$$\boldsymbol{y}^{t2i}(\boldsymbol{T}) = (1-\alpha)y^{t2i}_{\text{one-hot}}(\boldsymbol{T}) + \alpha \boldsymbol{p}^{t2i}(\boldsymbol{T}^{\text{mom}}) \quad (9)$$

The image-text contrastive loss is defined as the cross-entropy H between $\boldsymbol{p}$ and $\boldsymbol{y}$:

$$\begin{aligned}\mathcal{L}_{\text{itc}} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{I},\boldsymbol{T})\sim D}\,[\,&\mathrm{H}\left(\boldsymbol{y}^{\text{i2t}}(\boldsymbol{I}), \boldsymbol{p}^{\text{i2t}}(\boldsymbol{I})\right) \\ &+ \mathrm{H}\left(\boldsymbol{y}^{\text{t2i}}(\boldsymbol{T}), \boldsymbol{p}^{\text{t2i}}(\boldsymbol{T})\right)]\end{aligned} \quad (10)$$

3.4. Fine-Grained Image-Text Matching

Image-Text Matching (ITM) predicts whether a given image-text pair is positive (matched) or negative (not matched), which is a binary classification task. Based on ITM, the proposed Fine-Grained Image-Text Matching (FG-ITM) aims to capture fine-grained differences of image-text pairs. For each input image-text pair, we use two types of negative samples: an in-batch hard negative sample selected according to Equation 6 or 7, and a fine-grained negative sample generated by NegGen (Section 3.5). We use the multimodal encoder's output embedding of the `[CLS]` token as the joint representation of the image-text pair, and append a classification layer to predict the image-text matching probability $\boldsymbol{p}^{\text{itm}}(\boldsymbol{I}', \boldsymbol{T}')$. The FG-ITM loss is:

$$\mathcal{L}_{\text{itm}} = \mathbb{E}_{(\boldsymbol{I},\boldsymbol{T})\sim D}\mathrm{H}(\boldsymbol{y}^{\text{itm}}, \boldsymbol{p}^{\text{itm}}(\boldsymbol{I}', \boldsymbol{T}')) \quad (11)$$

where y^{itm} is the ground-truth of ITM, which is a 2-dimensional one-hot vector. $(\boldsymbol{I}', \boldsymbol{T}')$ includes $(\boldsymbol{I}, \boldsymbol{T})$, $(\boldsymbol{I}, \boldsymbol{T}^{\text{neg}})$, $(\boldsymbol{I}^{\text{neg}}, \boldsymbol{T})$ and $(\boldsymbol{I}, \boldsymbol{T}^{\text{fine-grained}})$, where $\boldsymbol{I}^{\text{neg}}$ is the negative image, $\boldsymbol{T}^{\text{neg}}$ is the negative text, and $\boldsymbol{T}^{\text{fine-grained}}$ is the fine-grained negative text.

The overall pretraining objective for ELECREA-VL is:

$$\mathcal{L} = \mathcal{L}_{\text{rtd}} + \mathcal{L}_{\text{itm}} + \mathcal{L}_{\text{itc}} \quad (12)$$

3.5. NegGen

NegGen is a novel data augmentation strategy to generate negative texts that only have token-level differences with the positive texts, thus enabling the multimodal encoder to capture more fine-grained information for better image-text alignment.

As shown in Figure 2 (B). The original text is masked and then fed into a language model to generate fine-grained negative samples. In order to mask tokens with rich semantic information, we employ a simple part-of-speech tagger to identify nouns and adjectives in the original text, and randomly mask 50% of them. The process of generating fine-grained negative samples is formalized as follows and $q = \lceil 0.5(M_{\text{noun}} + M_{\text{adj}})\rceil$, where M_{noun} and M_{adj} are the numbers of nouns and adjectives respectively.

$$\begin{aligned}&m_i \sim \text{unif}\{1, M_{\text{noun}} + M_{\text{adj}}\} \text{ for } i = 1 \text{ to } q \\ &\boldsymbol{T}^{\text{masked}} = \text{REPLACE}(\boldsymbol{T}, \boldsymbol{m}, \texttt{[MASK]}) \\ &\hat{T}_i \sim \boldsymbol{p}_G\left(T_i \mid \boldsymbol{T}^{\text{masked}}\right) \text{ for } i \in \boldsymbol{m} \\ &\boldsymbol{T}^{\text{fine-grained}} = \text{REPLACE}(\boldsymbol{T}, \boldsymbol{m}, \hat{\boldsymbol{T}})\end{aligned} \quad (13)$$

Furthermore, to ensure the generated text sample to be negative, we adopt a vision-language model trained with ITM as a discriminator to predict image-text matching probabilities and filter out potential positive samples. For example, in Figure 2 (B), although "kitten" and "cat" are different tokens, their semantic differences are negligible. Therefore, we use the discriminator to evaluate whether the text $\boldsymbol{T}^{\text{fine-grained}}$ matches the image $\boldsymbol{I}$, and then discard the pairs that match.

4. Experiments

4.1. Pretraining Setup

Following previous benchmarks [21, 20], we use COCO [23], Visual Genome (VG) [19], Conceptual Captions (CC) [33], and SBU Captions [28] as our pretraining datasets, which have a total of 4M unique images and 5.1M image-text pairs.

For the NegGen, we adopt $\text{BERT}_{\text{BASE}}$ [7] as the language generator and ALBEF [21] as the vision-language discriminatorm, and inference all pretraining datasets. In our model, the text encoder is initialized from the first 6 layers of $\text{BERT}_{\text{BASE}}$, the multimodal encoder is initialized from the last 6 layers of $\text{BERT}_{\text{BASE}}$, and the image encoder is initialized from CLIP-ViT-224/16. During the pretraining phase, the model is trained for 30 epochs using a batch size of 512. We adopt the mini-batch AdamW optimizer with a weight decay of 0.02. In the first 1000 iterations, the learning rate is warmed-up to $1e^{-4}$, and decayed to $1e^{-5}$ following a cosine schedule. Each image is randomly cropped to 256×256 resolution as input, and RandAugment [6] is adopted (color changes are not included because text descriptions often contain color information). During the finetuning stage, the resolution of an image is up-scaled to 384×384, and the positional encoding of the image patches is interpolated. The momentum parameter for updating the momentum model is 0.995, and the queue length of cached features for image-text contrastive learning is set to $Q = 65,536$. All experiments are performed on 8 NVIDIA A100 GPUs and take around 3 days to train. The weight λ of the discriminator loss is set as 10.

4.2. Vision-Language Downstream Tasks

Image-Text Retrieval includes two subtasks: image-to-text retrieval (TR) and text-to-image retrieval (IR). The pretrained model is evaluated on Flickr30K [30] and COCO [23], including finetuning and zero-shot settings. For the finetuning setting, the pretrained model is finetuned on the training set and then evaluated on the validation/test set. For the zero-shot setting, the pretrained model is directly evaluated on the test set without finetuning. Following [21], the zero-shot retrieval is conducted on Flickr30K, and the finetuning is conducted on Flickr30K and COCO.

Method	Pretrain Images	Flickr30K (1K test set)						MSCOCO (5K test set)					
		Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER [4]	4M	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
VILLA [11]	4M	87.9	97.5	98.8	76.3	94.2	96.8	-	-	-	-	-	-
OSCAR [22]	4M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
ViLT [18]	4M	83.5	96.7	98.6	64.4	88.7	93.8	61.5	86.3	92.7	42.7	72.9	83.1
ALIGN [17]	1.2B	95.3	99.8	100.0	84.9	97.4	98.6	**77.0**	93.5	96.9	59.9	83.3	89.8
ALBEF [21]	4M	94.3	99.4	99.8	82.8	96.7	98.4	73.1	91.4	96.0	56.8	81.5	89.2
METER-CLIP [9]	4M	94.3	99.6	99.9	82.2	96.3	98.3	76.1	93.1	96.8	57.0	82.6	90.0
VL-Match	4M	**96.4**	**99.8**	**100.0**	**86.0**	**97.5**	**99.0**	76.6	**93.8**	**97.1**	**60.2**	**83.6**	**90.1**

Table 1. Finetuned image-text retrieval results on Flickr30K and COCO.

Method	Flickr30K (1K test set)					
	Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
UNITER [4]	83.6	95.7	97.7	68.7	89.2	93.9
ViLT [18]	73.2	93.6	96.5	55.0	82.5	89.8
CLIP [31]	88.0	98.7	99.4	68.7	90.6	95.2
ALIGN [17]	88.6	98.7	99.7	75.7	93.8	96.8
ALBEF [21]	90.5	98.8	99.7	76.8	93.7	96.7
METER-CLIP [9]	90.9	98.3	99.5	79.6	94.9	97.2
VL-Match	**93.3**	**99.3**	**99.8**	**82.0**	**95.1**	**97.4**

Table 2. Zero-shot image-text retrieval results on Flickr30K.

Visual Entailment (VE) [41] predicts the semantic relationship of a given image-text pair, which is a three-classification task where the categories include: entailment, neutral and contradictory. This task focuses on examining the model's fine-grained understanding of images and texts.

Visual Question Answering (VQA) [12] aims to predict the answer to a given image and query (in textual form), which requires the model to understand the image and the text, and to obtain information about their interactions. Following [21], the task is regarded as a generative task, where a decoder is added during finetuning to sample answers from 3192 candidates.

Natural Language for Visual Reasoning (NLVR2) [36] judges whether a text description matches a pair of images, which is a binary classification task. We use NLVR2 dataset to evaluate the pretrained model. Following [21], we extend the multimodal encoder to receive two images, where the cross-attention layers of the extended part and the original part share parameters.

4.3. Evaluation on Image-Text Retrieval

In the evaluation on image-text retrieval, the loss function ITC + ITM is adopted for finetuning, and a re-rank mechanism is used for inference [21]. First, images and texts are encoded separately, and their similarity matrices are calculated to obtain top-k candidates. Then, representations of the candidates are fed into the multimodal encoder for re-ranking.

Table 1 and Tabel 2 show the comparative results of our method with previous works [4, 18, 31, 17, 21, 9] on two benchmarks: zero-shot retrieval and finetuned retrieval. VL-Match achieves state-of-the-art performance on all retrieval tasks. For the zero-shot retrieval, our method outperforms METER-CLIP by 2.4% (TR) and 2.4% (IR) on R@1. For the finetuned retrieval, our method outperforms METER-CLIP by 2.1% (TR) and 3.8% (IR) on R@1 of Flickr30K, and 0.5% (TR) and 2.8% (IR) on R@1 of COCO.

4.4. Evaluation on VQA, NLVR2, and VE

Table 3 shows the performance on VQA, NLVR2, and VE, which requires joint input of image + text. METER [9] performs successfully on these tasks, but it includes two multimodal encoders and uses tricks such as hierarchical learning rates and larger image resolution in the finetuning stage. Our model includes only one multimodal encoder and achieves competitive performance with only 55% of METER parameters. For VQA, many existing methods [4, 11, 9] regard it as a classification task and use Binary CrossEntropy Loss for finetuning, while our method regards it as a generative task and adopts an encoder-decoder structure. Compared to ALBEF [21] with the same settings, we achieve an improvement of 0.78%.

4.5. Ablations on Pretraining Tasks

We perform ablation experiments in the finetuned text-retrieval tasks on Flickr30K and NLVR2, the results are shown in Table 4. Meta-sum is the sum of results on NLVR2 and Flickr30k. Compared with the first row, the second row adds MLM task, and can be observed that this addition does not provide any significant improvement, which demonstrates that the gains achieved by our method do not stem from the MLM task to the text encoder. The third row introduces the generator-discriminator structure (G-D), where the generator outputs a corrupted text without `[MASK]`, and the discriminator takes as input the represen-

Method	Params	Generative	VQA test-dev	VQA test-std	NLVR2 dev	NLVR2 test-P	VE val	VE test
Include two multimodal encoders								
METER-Swin [9]	380M	✗	76.43	76.42	82.23	82.47	80.61	80.45
METER-CLIP [9]	380M	✗	**77.68**	**77.64**	**82.33**	**83.05**	**80.86**	**81.19**
Include one multimodal encoder								
UNITER [4]	300M	✗	73.82	74.02	79.12	79.98	79.39	79.38
VILLA [11]	300M	✗	74.69	74.87	79.79	81.47	80.18	80.02
ViLT [18]	87M	✗	71.26	-	75.70	76.13	-	-
ALBEF [21]	210M	✓	74.54	74.40	80.24	80.50	80.14	80.30
VL-Match	210M	✓	**75.11**	**75.18**	**81.96**	**82.23**	**80.44**	**81.26**

Table 3. Finetuned results on vision+language tasks. For fair comparison, we divide the existing methods into two categories: including two multimodal encoders and including one multimodal encoder.

Text encoder		Multimodal encoder		G-D	NLVR2		Flickr30k		Meta-sum
instance-level	token-level	instance-level	token-level		dev	test-P	TR	IR	
ITC	✗	ITM	MLM	✗	79.95	80.20	95.4	84.7	340.25
ITC	MLM	ITM	MLM	✗	79.80	80.77	95.2	84.7	340.47
ITC	MLM	ITM	MLM	✓	80.97	80.77	95.3	85.1	342.41
ITC	MLM	ITM	VL-RTD	✓	81.59	82.08	95.6	85.1	344.37
ITC	MLM	FG-ITM	VL-RTD	✓	**81.96**	**82.23**	**96.4**	**86.0**	**346.59**

Table 4. Ablation studies of pretraining tasks. We report R@1 on Flickr30k. G-D means whether to use the generator-discriminator structure, where the generator generates a corrupted text and the discriminator takes as input the representation of the corrupted text. When using the G-D structure, the difference between VL-RTD and MLM for the multimodal encoder is that VL-RTD is a binary-classification task and MLM is a vocabulary-size-classification. Meta-sum is the sum of results on NLVR2 and Flickr30k

tation of the corrupted text, achieving +1.94% in the Meta-sum. The fourth row replaces MLM with VL-RTD in the multimodal encoder, significantly improving the Meta-sum by 1.96%. Finally, the last row introduces FG-ITM to enhance the instance-level matching, bringing significant benefits, especially on cross-modal retrieval tasks. In summary, VL-Match enhances vision-language matching at both token and instance levels, substantially improving the performance in the downstream vision-language tasks (Meta-sum increases from 340.25 to 346.59).

Method	NLVR2 dev	NLVR2 test-P	Flickr30k TR	Flickr30k IR
NegGen	**81.96**	**82.23**	**96.4**	**86.0**
w/o mask noun. and adj.	81.24	81.29	96.3	85.9
w/o filter	80.67	81.34	95.8	85.6

Table 5. Ablation studies of NegGen.

4.6. Ablations on NegGen

We also conduct an ablation study to investigate the masking and filtering mechanisms of NegGen. As shown in Table 5, "*w/o mask noun. and adj.*" uses the typical random masking mechanism and the result shows the importance of masking nouns and adjectives that have rich semantic meanings. Besides, "*w/o filter*" does not filter out the matched pairs, resulting in performance degradation. Figure 3 shows cases of the generated-then-filtered negative texts and their corresponding images, which qualitatively demonstrate the effectiveness of NegGen to generate fine-grained negatives.

Figure 3. Examples of the positive text T^p and the fine-grained negative text T^f generated by NegGen.

4.7. Analysis of Fine-grained Matching

For the qualitative analysis, in Figure 4, the cross-modal retrieval cases show the superiority of FG-ITM, which force

Figure 4. Cross-modal retrieval cases. The first, second and third rows are the inputs, results of ITM and FG-ITM respectively.

Model	Existence	Plurality	Counting			Sp.rel.	Action		Coreference		Foil-it!	Avg.
	quantifiers	number	balanced	sns.	adv.	relations	repl.	actant swap	standard	clean		
ALBEF	71.29	78.73	61.98	64.89	59.62	73.08	73.30	57.74	52.68	53.85	95.55	68.91
VL-Match*	69.30	**80.26**	62.10	62.44	56.73	82.06	72.22	60.17	52.96	50.96	96.39	69.34
VL-Match	**72.67**	78.96	**62.21**	**65.50**	**60.78**	**83.93**	**73.91**	**61.01**	**57.34**	**57.69**	**97.99**	**71.22**

Table 6. Results on the VALSE benchmark, where * means "without FG-ITM".

our model sensitive to fine-grained differences. In the first case, FG-ITM successfully captures the fine-grained match between the number "two" in the image and text, whereas ITM fails to do so. This can partly be attributed to the model being exposed to such negatives with numerical mismatches via FG-ITM (e.g., the case in Figure 3). In Figure 5, the Grad-CAM visualization is applied to a visual grounding task on RefCOCO+, and the result 68.21% outperforms ALBEF by 2.32%. The visualization and the competitive results demonstrate that our method achieves fine-grained matching between text tokens and image patches. For the quantitative comparison, we add experiments on two multimodal fine-grained understanding benchmarks: VALSE [29] and SVO-Probes [16]. They require models to predict a matching score for a given image–text pair in a zero-shot setting, and we use pairwise ranking accuracy to evaluate the models. The results are shown in Table 6 and Table 7.

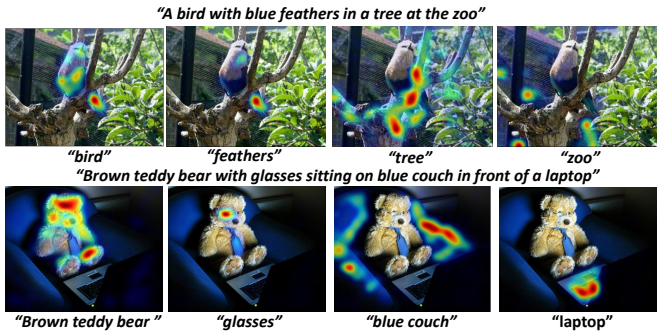

Figure 5. Grad-CAM visualization on cross-attention maps.

Model	Overall	Subj. Negative	Verb Negative	Obj. Negative
ALBEF	87.54	89.09	85.21	93.51
VL-Match*	87.96	90.97	85.30	93.84
VL-Match	**89.38**	**94.68**	**85.96**	**95.87**

Table 7. Results on SVO-Probes for sub., verb, and obj. negatives, where * means "without FG-ITM".

4.8. Analysis of Discriminator Loss Weight

Table 8 studies effect of the discriminator loss weight λ in the VL-RTD loss. We search for the best loss weight λ using Meta-sum as a general measure. Comprehensively, the loss weight λ of the discriminator should be higher than the loss weight of the generator, where $\lambda = 10$ is the best. As the discriminator is trained with the binary classification task instead of the generator's multi-classification task, the discriminator's loss was typically much lower than the generator's. Besides, we need to prevent the generator from achieving too high accuracy, resulting in very few replaced tokens [5]. Therefore, we need to assign a higher loss weight to the discriminator, thus balancing abilities of the generator and the discriminator [13, 26].

Loss weight	NLVR2		Flickr30k		Meta-sum
	dev	test-P	TR	IR	
$\lambda = 1$	79.64	79.89	96.3	84.7	340.53
$\lambda = 5$	80.47	80.71	95.5	**85.4**	342.08
$\lambda = 10$	81.59	**82.08**	95.6	85.1	**344.37**
$\lambda = 15$	81.20	80.94	96.1	84.5	342.74
$\lambda = 20$	**81.67**	81.16	**96.4**	84.4	343.63

Table 8. Comparison of discriminator loss weights λ in the VL-RTD loss

4.9. Analysis of Computation Cost

As shown in Table 9, our NegGen need little time to generate fine-grained negative samples for FG-ITM. Moreover, replacing MLM with VL-RTD increases the pre-training FLOPs by only 2.8%, due to one more forward pass of the text encoder. The FG-ITM further increases the FLOPs by 1.6% to train on the augmented negatives. Overall, our approach has a slight increase in computation cost, but delivers significant gains.

	FLOPs(1e19)	Time	Memory(MB)
NegGen	0.882	1h	4296
baseline	1.898	3d 18h	35702
+ VL-RTD	1.952	3d 19h	36062
+ FG-ITM	1.984	3d 23h	37294

Table 9. Computation cost. The input length of image/text is 256/25.

5. Conclusion

In this paper, we propose VL-Match, a generator-discriminator framework that enhances vision-language pretraining with token-level and instance-level matching. At the token level, we propose Vision-Language Replaced Token Detection task for the multimodal encoder, which introduces more language prior knowledge and involves more text tokens to match the image, improving the matching efficiency. Moreover, we propose Fine-Grained Image-Text Matching at the instance level, which adds fine-grained negatives generated by a novel bootstrapping method NegGen. This method can improve the model's ability to recognize fine-grained differences between images and texts. Experimental results on widely-used benchmarks show that VL-Match outperforms existing SOTA methods by a large margin. Theoretically, our method can be easily generalized to any VLP architectures which is trained with MLM and ITM. For future work, training the image encoder to generate a corrupted image, and then guiding the multimodal encoder to discriminate whether each patch of the corrupted image is replaced may be worth studying.

Acknowledgments

We would like to thank Wenhui Wang for his helpful discussions.

References

[1] Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. Vl-beit: Generative vision-language pretraining. *CoRR*, abs/2206.01127, 2022.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[3] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In *ECCV (30)*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer, 2020.

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[6] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 3008–3017. Computer Vision Foundation / IEEE, 2020.

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

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

[9] 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. An empirical study of training end-to-end vision-and-language transformers. In *CVPR*, pages 18145–18155. IEEE, 2022.

[10] Jiali Duan, Liqun Chen, Son Tran, Jinyu Yang, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Multi-modal alignment using representation codebook. In *CVPR*, pages 15630–15639. IEEE, 2022.

[11] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*, 2020.

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *ECCV*, 2017.

[13] Yaru Hao, Li Dong, Hangbo Bao, Ke Xu, and Furu Wei. Learning to sample replacements for ELECTRA pre-training. In *ACL/IJCNLP (Findings)*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4495–4506. Association for Computational Linguistics, 2021.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[15] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*, 2021.

[16] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. In *ACL*, 2021.

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

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

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

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

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

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

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[25] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[26] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul N. Bennett, Jiawei Han, and Xia Song. Pretraining text encoders with adversarial mixture of training signal generators. In *ICLR*. OpenReview.net, 2022.

[27] BEiT: BERT Pre-Training of Image Transformers. Bao, hangbo and dong, li and piao, songhao and wei, furu. In *ICLR*, 2022.

[28] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011.

[29] Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *ACL*, 2022.

[30] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, 2017.

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

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

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

[34] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *CVPR*, pages 15617–15629. IEEE, 2022.

[35] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[36] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *ACL*, 2019.

[37] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[39] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts, 2021.

[40] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*. OpenReview.net, 2022.

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

[42] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *CVPR*, 2022.

[43] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2022.

[44] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. In *CVPR*, 2021.