An Anchor-based Relative Position Embedding Method for Cross-Modal Tasks

Ya Wang¹, Xingwu Sun¹,², Fengzong Lian¹, Zhanhui Kang¹, Chengzhong Xu²

Machine Learning Platform Department, Tencent¹
State Key Lab of IOTSC, Department of Computer Science, University of Macau²

connorywang@tencent.com, sammsun@tencent.com
faxonlian@tencent.com, kegokang@tencent.com, czxu@um.edu.mo

Abstract

Position Embedding (PE) is essential for transformer to capture the sequence ordering of input tokens. Despite its general effectiveness verified in Natural Language Processing (NLP) and Computer Vision (CV), its application in cross-modal tasks remains unexplored and suffers from two challenges: 1) the input text tokens and image patches are not aligned; 2) the encoding space of each modality is different, making it unavailable for feature comparison. In this paper, we propose a unified position embedding method for these problems, called AnChor-basEd Relative Position Embedding (ACE-RPE), in which we first introduce an anchor locating mechanism to bridge the semantic gap and locate anchors from different modalities. Then we conduct the distance calculation of each text token and image patch by computing their shortest paths from the located anchors. Last, we embed the anchor-based distance to guide the computation of cross-attention. In this way, it calculates cross-modal relative position embeddings for cross-modal transformer. Benefiting from ACE-RPE, our method obtains new SOTA results on a wide range of benchmarks, such as Image-Text Retrieval on MS-COCO and Flickr30K, Visual Entailment on SNLI-VE, Visual Reasoning on NLVR2 and Weakly-supervised Visual Grounding on RefCOCO+.

1 Introduction

Transformer (Vaswani et al., 2017) has shown excellent performance in Natural Language Processing (NLP), Computer Vision (CV) as well as cross-modal tasks, including natural language inference (Devlin et al., 2018), image classification (Wu et al., 2021), visual question answering (Wu et al., 2017) and visual entailment (Xie et al., 2019), etc. Nevertheless, transformer module lacks the capability to capture the ordering information of the input tokens because of the limitation of its self-attention mechanism. Therefore, incorporating explicit position representations is crucial to improve the performance of transformer-based models (Devlin et al., 2018; Dosovitskiy et al., 2020).

Generally, there are two mainstream position encoding methods in transformer-based NLP and CV models, i.e., absolute position embedding (APE) and relative position embedding (RPE). APE methods (Vaswani et al., 2017; Devlin et al., 2018; Dosovitskiy et al., 2020) encode absolute positions of the input tokens with either trainable (Devlin et al., 2018) or fixed embedding (Vaswani et al., 2017). These position embeddings are added with the token embeddings, which are then passed to the self-attention layer to calculate the token relationship considering their positional information. It has been verified effective in a variety of NLP (Wang et al., 2020; Devlin et al., 2018) and CV (Wu et al., 2021) tasks. On the other hand, RPE methods (Chu et al., 2021; Shaw et al., 2018) encode the pairwise distances of every two tokens. Commonly, it directly interacts with the calculation of attention mechanism in different ways (Wu et al., 2021; Chu et al., 2021). Compared with APE, RPE methods are superior to modeling the positional information of extremely long or variant-length sequences. As a result, in some span prediction tasks of NLP, RPE methods are shown to achieve more performance gains than APE ones (Wang et al., 2020).

Despite the success of the position embedding methods in unimodal tasks, its exploration in the field of cross-modal modeling is still limited. Recent works on cross-modal tasks (Cho et al., 2021; Li et al., 2021) could be classified into two frameworks, 1) One-stage methods (Fig. 1(a)) which extract the cross-modal representation with a unified cross-modal encoder; 2) Two-stage methods (Fig. 1(b)), which have additional text encoder and image encoder. Both of them adopt the position embeddings in a separate way, where the text and image
position representations are embedded individually. In this way, the models can only learn position embedding in each modality separately while ignoring positional information between two tokens from different modalities. However, it is challenging to raise a unified method for cross-modal position embedding. Firstly, the inputs from two modalities are embedded into different spaces, making the input embedding not comparable. Secondly, since the text tokens and image patches are not aligned, the relative positions between two units from different modalities are meaningless.

In this paper, we advocate a new perspective for effective cross-modal position encoding (shown in Fig. 1(c)), called AnChor-basEd Relative Position Embedding (ACE-RPE). It first computes alignment between text and image tokens to locate aligned pieces, which are called anchors in this paper. Subsequently, the token-to-token (t2t) and patch-to-patch (p2p) relative position is calculated for unimodal ordering information. The relative position searching of arbitrary text token and image patch is then considered as a shortest path problem, containing three steps: 1) routing from given token and its nearby anchors; 2) routing from anchors and their located image patches, and 3) routing from the located patches to the given image patch. As illustrated in Fig. 2, the relative position of “A” and the image patch of the man is derived from three terms: the t2t relative position between “A” and the anchor “man”, the relative position from anchor “man” to the image patch matching “man”, and the relative position from the located image patch to the patch of human (obviously, 0 in this case). Finally, we embed the anchor-based relative position to the self-attention calculation. Further, we conduct extensive experiments to verify the effectiveness of the proposed ACE-RPE compared to many strong baselines. The results demonstrate that our method can boost the performance of cross-modal transformers with a large margin.

The main contributions of this work can be summarized as follows,

- We propose the ACE-RPE method to incorporate positional information into cross-modal transformers and bridge the gap of different modalities. As we know, it is the first work to model relative position in cross-modal tasks.

- We give an anchor-based RPE method to get relative positions according to the located anchors between two modalities. Extensive experiments compared with strong baselines reveals the effectiveness of this method.

- Our method achieves new SOTA in 5 cross-modal benchmarks, including Flickr30K (Plummer et al., 2015), MS-COCO (Lin et al., 2014), SNLI-VE (Xie et al., 2019), NLVR2 (Suhr et al., 2018) and RefCOCO+ (Yu et al., 2016). In addition, it also surpasses baseline methods significantly on VQA (Goyal et al., 2017).

2 Related Work

2.1 Position Embedding for NLP

Currently, Transformer (Vaswani et al., 2017) plays a major role in the field of NLP. It shows superiority in many real-world tasks, such as natural language inference (Devlin et al., 2018) and question answering (Devlin et al., 2018; Rajpurkar et al., 2016). However, the self-attention of transformer lacks the ability to capture ordering information of input tokens in a sequence. Such that, additional explicit representations for token positions are crucial to the performance of the transformer.

The position embedding in NLP could be categorized into two classes: APE and RPE. APE encodes the absolute position of tokens in a sequence. Each position has its individual embedding, which are generated with specific functions, like sinusoidal operator (Vaswani et al., 2017) or learnable encoding (Devlin et al., 2018). Usually, the generated APE is added with the input text tokens for an explicit perspective view of token positions. Therefore, the same token in different positions will have different embedding. Currently, various works on APE are proposed to further boost the performance of transformer-based methods.

RPE (Dai et al., 2019; Devlin et al., 2018; Raffel et al., 2019) encodes the pairwise relative token position via interacting with the query, key or value in self-attention modules (Shaw et al., 2018). Compared to APE, RPE is translation-invariant and could encode variable lengths of input sequences. Therefore, it is shown to surpass APE on some long-sequence tasks (Wang et al., 2020).

2.2 Position Embedding for CV

With the great success of Visual Transformer (ViT) (Dosovitskiy et al., 2020) on large-scale dataset, the transformer-based methods have also become
In summary, position embedding has been proved to have a significant effect on the performance of transformer-based models in both NLP and CV. However, the exploration on cross-modal tasks is still vacant. One of the most important reasons is that it is challenging to find a meaningful “position” between different modalities. For example, it is not available for us to define the position of the word “are” in a text and the corresponding patches in an image. To this end, we propose an anchor-based method, which bridges the gap between the text and image modalities and makes it possible to calculate position embeddings of different modalities.

3 Methods

The overview of our backbone network is presented in Fig. 3, which contains a 6-layer visual transformer (Dosovitskiy et al., 2020) as the image encoder, a 6-layer linguistic transformer (Devlin et al., 2018) as the text encoder and a 6-layer cross-modal transformer. The AnChor-basEd Position Embedding (ACE-RPE) is proposed to leverage the cross-modal encoder with cross-modal positional information. It involves two key procedures: 1) learning the locating of cross-modal anchors; 2) ACE-RPE calculation by incorporating anchor locating and t2t/i2i relative position. In this section, we first present the above procedures in detail (Sec. 3.1 and Sec. 3.2). Then, we present the overall pre-training objectives of our method.
3.1 Cross-modal Locating of Anchors

Considering an image \( x \) and its corresponding text \( y \), the “anchor” in this paper refers to the prominent tokens of \( y \), which can be located to some patches of \( x \). An illustration of cross-modal anchors is depicted in Fig. 2. Naturally, the word “man” is associated with the image patch containing the human, and “blue” can be located to the blue patches. Then, the words “tie” and “cat” are called anchors in this paper.

In this part, we propose an unsupervised method to figure out the cross-modal anchors effectively. It uses a token-wise loss to search for anchors without any additional annotations. Formally, the raw image \( x \) is segmented into \( M + 1 \) image patches (Dosovitskiy et al., 2020), i.e., \( x = \{c_x, x_1, x_2, \ldots, x_m, \ldots, x_M\} \), where each of them is embedded with a normalized \( D \)-dimensional vectors, \( c_x \) is an image [CLS] token. Similarly, the text \( y \) is tokenized to \( N + 1 \) text tokens, \( y = \{c_y, y_1, y_2, \ldots, y_n, \ldots, y_N\} \), where \( c_y \) is a text [CLS] token. The token-wise similarity between the image patch \( x_m \) and text token \( y_n \) is computed by a specific similarity function (cosine similarity in this paper) \( f \). We then introduce an anchor loss to maximize the similarity of the anchors and their matching image patches, without changing the similarity of unmatched pairs, e.g., “blue” and patches of the “horse” in Fig. 2. Accordingly, the proposed anchor loss is formulated based on contrastive learning and log-sum-exp trick:\(^1\)

\[
\mathcal{L}_{\text{ace}} = \frac{1}{2} \mathbb{E}_{(x,y)} \left[ H_{12t}(x,O_y) + H_{12t}(y,O_x) \right] - \frac{1}{\lambda} \log \sum_{m,n} e^{\lambda f(x_m,y_n)}
\]

where \( \lambda \) is a scale parameter. \( O_y \) and \( O_x \) indicate the dynamic dictionaries (He et al., 2020), containing one positive sample \( y \) and \( K - 1 \) negative samples, that is only text \( y \) in \( O_y \) matches image \( x \). \( K \) is 65, 536 in this paper, following (Li et al., 2021).

\( f \) presents the similarity function (cosine similarity in this paper). \( H_{12t}(X,O_y) \) and \( H_{12t}(Y,O_x) \) denote the image-to-text and text-to-image con-

---

\(^1\)Inspired by (Nielsen and Sun, 2016), log-sum-exp is a soft-smoothing version of maximum operation. It is used to output some maximum values (the number is adjusted by a scale parameter), while small values tend to zero.
trastive losses based on K-pairs, respectively
\[
H_{t2t}(x, O_y) = -\min \{ 0, f(x, y) - \delta - \frac{1}{\lambda} \log \sum_{z \in O_y, z \neq y} e^{\lambda f(x, z)} \} \quad (2)
\]
here \( \delta \) is the margin between positive and negative samples, which is empirically set to 0.05 in our experiments. \( H_{t2t}(y, \hat{O}_x) \) is defined accordingly.

3.2 Calculation of ACE-RPE

The calculation of ACE-RPE refers to three major components: 1) the locating of anchors with multi-group relative position; 2) the computation of anchor-based cross-modal relative position between text tokens and image patches; 3) cross-modal relative position embedding. Each step is elaborated as follows.

3.2.1 Locating of Anchors

The relative position between anchors and their relative image patches is dynamically generated with a proposed multi-group cross-modal similarity,

\[
S_G(x_m, y_n) = \left[ f(\hat{x}_m, \hat{y}_n), f(\tilde{x}_m, \tilde{y}_n), \ldots, f(\bar{x}_m, \bar{y}_n) \right] \quad (3)
\]
where \( G \) is the number of groups, \( \hat{x}_m \in \mathbb{R}^{G \times p} \), \( \hat{y}_n \in \mathbb{R}^{G \times p} \) are the reshaped versions of \( x_m \) and \( \hat{y}_n \), \( \tilde{x}_m \in \mathbb{R}^{p \times p} \), \( \tilde{y}_n \in \mathbb{R}^{p \times p} \). Note that our proposed multi-group cross-modal similarity is not a scalar but a vector of length \( G \).

Shown in Eqn. 3, the multi-group cross-modal similarity functions on all text tokens and image patches. We then introduce a post-locating for anchors with a soft shrinking operator,

\[
\tilde{S}_G(x_m, y_n) = \begin{cases} 
S_G(x_m, y_n), & S_G(x_m, y_n) \geq \delta \\
\delta \land (S_G(x_m, y_n) - \tau), & S_G(x_m, y_n) < \delta 
\end{cases} \quad (4)
\]
where \( \delta \) is a hyper-parameter. \( \tau \) is a large enough scalar, set to 10^4 in this paper.

The set of anchors is then defined as
\[
A_G(x, y) = \{ x_m \mid \exists y_n, s.t. \tilde{S}_G(x_m, y_n) \geq \delta \} \quad (5)
\]
where “\( \geq \)” is calculated element-wisely by each group of \( \tilde{S}_G \). Hence, the \( A_G(x, y) \) is a collection of \( G \) anchor sets, which may be different in different groups. As indicated in Eqn. 8 and analyzed in Sec. A.2, the multi-group anchor sets instead of a single one can enhance the flexibility of position embeddings.

Finally, the distance between anchors and their relative image patches is,

\[
D_G(x_m, y_n) = \frac{1}{\tilde{S}_G(x_m, y_n)} \quad (6)
\]

3.2.2 Anchor-based Cross-modal Relative Position Calculation

Given an arbitrary text token and an image patch, we consider the calculation of their relative position as a shortest path problem, where the path is split into three steps: 1) route from the given text token to nearby anchors; 2) route from anchors to their located image patches; 3) route from the located image patches to the given image patch. Formally, the anchor-based relative distance is,

\[
P_{ace}(x_m, y_n) = \min_{i,j} \left\{ D_{p2p}(x_m, x_i) \oplus D_{G}(x_i, y_j) \oplus D_{t2t}(y_j, y_n) \right\} \quad (7)
\]
where “\( \oplus \)” is the broadcasting addition of scalars and vectors. “\( \min(\cdot) \)” is executed in an inner-group manner, i.e., the values are compared in each group. Therefore, the output \( P_{ace}(x_m, y_n) \) keeps a vector of length \( G \). Here \( D_{p2p} \) and \( D_{t2t} \) are the common image patch-to-patch and text token-to-token physical distance, respectively. For efficiency, we only consider neighborhood of \( B_p \) tokens in \( D_{t2t} \) and a square neighborhood of \( B_t \) image paths in \( D_{p2p} \). It should be noted that, the matrix of all text tokens and image patches \( P_{ace}(x, y) \) can be implemented efficiently by Pointwise Convolution (Howard et al., 2017), reducing the computation complexity to \( O(MNB_pB_tG) \), which can be omitted since \( B_p, B_t \) and \( G \) are small enough.

3.2.3 Cross-modal Relative Position Embedding

Sec. 3.2.2 provides the multi-group relative position of each text token and image patch. The pairwise anchor-based relative position is then embedded with a learnable matrix \( W \in \mathbb{R}^{G \times D} \),

\[
E_{ace}(x_m, y_n) = P_{ace}(x_m, y_n)W \quad (8)
\]
Which is called ACE-RPE in this paper. Obviously, the proposed ACE-RPE is a specific case of RPE, where the distance of the images and texts is calculated with an anchor strategy and represented by a \( G \)-dimensional vector. Then, the distances are projected to learnable position embedding and the same distance enforces the same position embedding. Consequently, the t2t RPE in NLP, p2p RPE
in CV and t2p/p2t RPE in cross-modal tasks are united in a unified form, as formulated in Eqn. 7.

Detailedly presented in Sec. A.3, we propose two different cross-attention modes interacting with ACE-RPE, i.e., the bias mode and the contextual mode. By default, we use the contextual mode in this paper.

3.3 Pre-training Objectives

The pre-training of our models involves optimizing four objectives jointly, i.e., the proposed anchor loss for anchor locating, Masked Language Modeling (MLM) for text embedding, Masked Image Modeling (MIM) for image embedding, Image-Text Matching (ITM) for cross-modal matching, as shown in Fig. 3.

**Anchor Loss** is optimized during pre-training for better anchor locating. Noted in Eqn. 1, it enhances the similarity of anchors and their matching image patches by token-wise contrastive learning, exclusively ignores unmatched pairs through log-sum-exp trick.

**Masked Language Modeling (MLM)** predicts the masked words with both contextual text tokens and image patches. It aims to learn better text embedding by injecting extra contextual information in image patches. In this part, we conduct the MLM with a masking probability of 15% and take the output text embedding of cross-encoder to predict the masked tokens.

**Masked Image Modeling (MIM)** predicts raw pixel values of the randomly masked image patches by a lightweight one-layer head. Following (Xie et al., 2021), we implement this task by optimizing the $\ell_1$ loss between raw pixel values and the output of the prediction head.

**Image-Text Matching (ITM)** is to predict whether an image-text pair is positive (matched) or negative (unmatched), and further capture the contextual correlation between vision and language. It is a binary classification task while taking the embedding of the [CLS] token as a joint representation of the image-text pair.

4 Experiments

In this section, we first provide numerical analyses of the proposed ACE-RPE method compared with widely used baselines on 5 cross-modal tasks, including 6 benchmarks. Then, we make a detailed ablation study to analyze the contribution of each component of the proposed ACE-RPE method.

4.1 Pre-training Setup

**Pre-training Datasets** Following ALBEF (Li et al., 2021), the pre-training datasets are constructed with four public-released datasets, including two web datasets (Conceptual Captions (Sharma et al., 2018), SBU Captions (Ordonez et al., 2011)), and two in-domain datasets (MS-COCO (Lin et al., 2014) and Visual Genome (Krishna et al., 2017)). The entire pre-training dataset contains about 4.0M unique images and 5.1M image-text pairs.

**Implementation Details** Our ACE-RPE method contains 163.7M parameters, including a text encoder of 66.6M linguistic transformer (Devlin et al., 2018), an image encoder of 43.8M ViT-B/16 (Dosovitskiy et al., 2020) and a cross-modal encoder of 53.3M transformer (Devlin et al., 2018). It is notable that, the text encoder is constructed with the first 6 layers of the original BERTbase. Presented in Fig. 3, the pre-trained objectives are composed of three tasks: Masked Language Modeling (MLM) (Li et al., 2021) for text embedding, Masked Image Modeling (MIM) (Xie et al., 2021) for image embedding (Li et al., 2021), and Image-Text Matching (ITM) for cross-modal modeling. Our model is pre-trained for 30 epochs with a batch size of 512 on 8 NVIDIA A100 GPUs. We use AdamW (Loshchilov and Hutter, 2017) setting the weight decay as 0.02. The initial learning rate is $10^{-4}$ and decayed to $10^{-6}$, using a cosine schedule (Loshchilov and Hutter, 2016). We use RandAugment (Cubuk et al., 2020) as the image augmentation strategy, and then scale the augmented image to the resolution of $256 \times 256$. We also utilize the momentum distillation proposed in ALBEF (Li et al., 2021) and the queue size is 65, 536. By default, the hyper-parameters are set as $B_i = 5$, $B_p = 9$, $\lambda = 2$, $\delta = 0.05$ and $G = 8$, respectively.

4.2 Downstream Cross-modal Tasks

We conduct comprehensive experimental comparison on 5 cross-modal tasks, including: 1) Image-Text Retrieval on MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015); 2) Visual Entailment on SNLI-VE (Xie et al., 2019); 3) Visual Reasoning on NLVR2 (Suhr et al., 2018); 4) Visual Question Answering on VQA (Goyal et al., 2017) and 5) Weakly-supervised Visual Grounding on RefCOCO+ (Yu et al., 2016).

**Image-Text Retrieval** Image-Text Retrieval refers to retrieving the most relative images given a query text, and vice versa. We evaluate our methods on
**Cross-modal Position Embedding**

<table>
<thead>
<tr>
<th>Pre-trained Images</th>
<th>Flickr30K (1K test set)</th>
<th>MS-COCO (5K test set)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TR R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>None</td>
<td>94.3</td>
<td>99.5</td>
</tr>
<tr>
<td>APE</td>
<td>94.5</td>
<td>99.6</td>
</tr>
<tr>
<td>RPE</td>
<td>94.4</td>
<td>99.5</td>
</tr>
<tr>
<td>APE + RPE</td>
<td>94.5</td>
<td>99.6</td>
</tr>
<tr>
<td>Uniform†</td>
<td>94.6</td>
<td>99.6</td>
</tr>
<tr>
<td>ACE-RPE</td>
<td>95.2</td>
<td>99.6</td>
</tr>
<tr>
<td>ACE-RPE+Lace</td>
<td>95.4</td>
<td>99.7</td>
</tr>
<tr>
<td>ACE-RPE+lace*</td>
<td>96.7</td>
<td>99.9</td>
</tr>
</tbody>
</table>

†: calculates the distance of all words and patches by a uniform distance without the guidance of “anchor”. *
: extended with extra pre-training dataset CC12M (Changpinyo et al., 2021).

Table 1: Comparison in the Image-Text Retrieval task on Flickr30K and MS-COCO. For text retrieval (TR) and image retrieval (IR), we report the Top-1 Recall (R@1), Top-5 Recall (R@5) and Top-10 Recall (R@10). The FLOPs of our ACE-RPE model is 122G, which has just 6.1% computational overhead compared with “None” version (115G FLOPs).

**4.3 Comparison with Baseline Methods**

In this part, we conduct 4 downstream cross-modal tasks (except for RefCCOCO+) to compare the proposed ACE-RPE with the baseline methods, including 1) APE method (Dosovitskiy et al., 2020); 2) RPE method (Dosovitskiy et al., 2020); 3) a unified method combining APE and RPE (Wu et al., 2021). It is remarkable that among all methods, our ACE-RPE is the only cross-modal position embedding. The mentioned APE, RPE and their combined version are all conducted for each modality separately. They are simply concatenated together, and then injected into the cross-modal encoder. Furthermore, we also conduct a uniformed version of our ACE-RPE, where the distances of all words and patches are naively calculated by a uniform distance without the guidance of “anchor”.

<table>
<thead>
<tr>
<th>Cross-modal Position Embedding</th>
<th>VQA dev test</th>
<th>SNLIE-V dev test</th>
<th>NLVR dev test</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>73.2</td>
<td>73.6</td>
<td>79.2</td>
</tr>
<tr>
<td>APE</td>
<td>73.9</td>
<td>74.1</td>
<td>80.2</td>
</tr>
<tr>
<td>RPE</td>
<td>73.8</td>
<td>73.9</td>
<td>79.4</td>
</tr>
<tr>
<td>APE+RPE</td>
<td>73.9</td>
<td>74.1</td>
<td>80.1</td>
</tr>
<tr>
<td>Uniform†</td>
<td>74.1</td>
<td>74.2</td>
<td>80.3</td>
</tr>
<tr>
<td>ACE-RPE</td>
<td>74.9</td>
<td>75.1</td>
<td>81.1</td>
</tr>
<tr>
<td>ACE-RPE+Lace</td>
<td>75.4</td>
<td>75.7</td>
<td>81.4</td>
</tr>
<tr>
<td>ACE-RPE+lace*</td>
<td>76.8</td>
<td>76.9</td>
<td>82.0</td>
</tr>
</tbody>
</table>

†: calculates the distance of all words and patches by a uniform distance without the guidance of “anchor”. *
: pre-trained on CC12M (Changpinyo et al., 2021).

Table 2: Evaluation of the proposed methods on VQA (Goyal et al., 2017), Visual Entailment (SNLI-VE (Xie et al., 2019)) and Visual Reasoning (NLVR (Suhr et al., 2018)) tasks. “dev” and “std” in VQA are the test-dev and test-std datasets.

Numerical results are presented in Table 1 and
Table 3: Experimental results of Image-Text Retrieval on Flickr30K and MS-COCO.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pre-trained Images</th>
<th>Flickr30K (1K test set)</th>
<th>MS-COCO (5K test set)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>TR</td>
<td>IR</td>
</tr>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>UNITER</td>
<td>4M</td>
<td>87.5</td>
<td>98.0</td>
</tr>
<tr>
<td>VILLA</td>
<td>4M</td>
<td>89.0</td>
<td>97.5</td>
</tr>
<tr>
<td>OSCAR</td>
<td>4M</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALIGN</td>
<td>1.2B</td>
<td>95.3</td>
<td>99.8</td>
</tr>
<tr>
<td>ALBEF</td>
<td>4M</td>
<td>94.3</td>
<td>99.9</td>
</tr>
<tr>
<td>ALBEF</td>
<td>14M</td>
<td>95.9</td>
<td>99.8</td>
</tr>
<tr>
<td>Ours</td>
<td>4M</td>
<td>95.4</td>
<td>99.7</td>
</tr>
<tr>
<td>Ours</td>
<td>14M</td>
<td>96.7</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Table 3. It is shown that, in the task of Image-Text Retrieval (Table 1), our proposed ACE-RPE could enhance the performance of backbones by large margins. Specifically, compared with baseline cross-modal position embedding, i.e., None position embedding counterparts, our methods improve the performance over 1.1% and 1.0% R@1 in the “TR” and “IR” on Flickr30K. Similar gains in “TR” and “IR” on MS-COCO are up to 1.6% and 1.4%. It is worth noting that, these gains are achieved with the same backbone networks and same pre-training dataset. Meanwhile, while trained on a larger dataset with 14M samples, our model achieves two new SOTA performances on both Flickr30K and MS-COCO.

Table 4: Comparison with SOTA works on VQA, SNLI-VE and NLVR benchmarks. “dev” and “std” in VQA are the test-dev and test-std datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA dev</th>
<th>SNLI-VE dev</th>
<th>NLVR dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisualBERT (Li et al., 2019)</td>
<td>70.8</td>
<td>71.0</td>
<td>-</td>
</tr>
<tr>
<td>VL-BERT (Su et al., 2020)</td>
<td>71.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LXMERT (Tan and Bansal, 2019)</td>
<td>72.4</td>
<td>72.5</td>
<td>-</td>
</tr>
<tr>
<td>12-in-1 (Li et al., 2020)</td>
<td>73.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UNITER (Chen et al., 2020b)</td>
<td>72.7</td>
<td>72.9</td>
<td>-</td>
</tr>
<tr>
<td>VL-BART/T5 (Cho et al., 2021)</td>
<td>71.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ViLT (Kim et al., 2021)</td>
<td>70.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OSCAR (Li et al., 2020)</td>
<td>73.2</td>
<td>73.4</td>
<td>-</td>
</tr>
<tr>
<td>VILLA (Gan et al., 2020)</td>
<td>73.6</td>
<td>73.7</td>
<td>79.4</td>
</tr>
<tr>
<td>ALBEF (Li et al., 2021) (4M)</td>
<td>74.5</td>
<td>74.7</td>
<td>80.1</td>
</tr>
<tr>
<td>ALBEF (Li et al., 2021) (14M)</td>
<td>75.8</td>
<td>76.0</td>
<td>80.8</td>
</tr>
<tr>
<td>ACE-RPE(4M)</td>
<td>74.9</td>
<td>75.1</td>
<td>81.1</td>
</tr>
<tr>
<td>ACE-RPE + L_{ace} (4M)</td>
<td>75.4</td>
<td>75.7</td>
<td>81.4</td>
</tr>
<tr>
<td>ACE-RPE + L_{ace} (14M)</td>
<td>76.8</td>
<td>76.9</td>
<td>82.0</td>
</tr>
</tbody>
</table>

Table 4. Comparison with SOTA works on VQA, SNLI-VE and NLVR benchmarks. “dev” and “std” in VQA are the test-dev and test-std datasets.

For the tasks of Visual Question Answering on VQA, Visual Entailment on SNLI-VE and Visual Reasoning on NLVR, the proposed ACE-RPE also outperforms baseline methods robustly, as shown in Table 2. Furthermore, the comparison between “ACE-RPE” and “ACE-RPE + L_{ace}” reveals that the proposed L_{ace} is key for the performance improvement of ACE-RPE.

4.4 Comparison with SOTA Methods

Table 3, Table 4 and Table 5 report the results of the proposed ACE-RPE and previous SOTA methods. Pretrained on the dataset with 4M images, our methods achieve absolute improvements over ALBEF of 1.1% R@1 in “TR” and 1.2% R@1 in “IR” on Flickr30K. Similar gains in R@1 “TR” and “IR” on MS-COCO are up to 1.1% and 1.1%. For Visual Entailment, Visual Reasoning and Weakly-supervised Visual Grounding tasks, ACE-RPE also outperforms existing methods by substantial margins. With the 14M pre-trained dataset, which is also used in ALBEF, our method achieves 5 new SOTA results on all benchmarks 1, which presents the superiority and robustness of our ACE-RPE.

Table 5: Weakly-supervised visual grounding on RefCOCO+ benchmark.

<table>
<thead>
<tr>
<th>Method</th>
<th>Val</th>
<th>TestA</th>
<th>TestB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARN (Liu et al., 2019)</td>
<td>32.8</td>
<td>34.4</td>
<td>32.1</td>
</tr>
<tr>
<td>CCL (Zhang et al., 2020)</td>
<td>34.3</td>
<td>36.9</td>
<td>33.6</td>
</tr>
<tr>
<td>ALBEF (Li et al., 2021)</td>
<td>58.5</td>
<td>65.9</td>
<td>46.3</td>
</tr>
<tr>
<td>ACE-RPE(4M)</td>
<td>59.4</td>
<td>66.6</td>
<td>47.1</td>
</tr>
<tr>
<td>ACE-RPE + L_{ace} (4M)</td>
<td>60.1</td>
<td>67.5</td>
<td>47.9</td>
</tr>
<tr>
<td>ACE-RPE + L_{ace} (14M)</td>
<td>60.5</td>
<td>67.9</td>
<td>48.2</td>
</tr>
</tbody>
</table>

Table 5. Weakly-supervised visual grounding on RefCOCO+ benchmark.

4.5 Visualization of ACE-RPE

In order to reveal the inherent ability of the proposed ACE-RPE to model the cross-modal positional information, we provide Grad-CAM visualization (Selvaraju et al., 2017; Li et al., 2021) of the anchor-based relative position in the last cross-modal transformer. Fig. 4 shows some examples in MS-COCO. The visualization of cross-modal locating is highly correlated with human priors, which indicates the correctness of our ACE-RPE.

5 Conclusion

In this paper, we present a cross-modal position embedding method, called ACE-RPE, in which we first utilize an anchor locating method to learn to match the text words and the image patches.

1Except for VQA, where the champion achieved the best score of 82.78 according to https://eval.ai/web/challenges/challenge-page/830/leaderboard/2278. But we think it is not fair to compare the methods in Table 4 with the champion because of different pretrained datasets and great finetuning gap.

5408
A man in blue is sitting on a fake horse.

The man is reading a newspaper while carrying an umbrella.

Figure 4: The Grad-CAM (Selvaraju et al., 2017) visualization of cross-modal distance on the last cross-attention layer. The words in red are the anchors.

Then, we compute physical distances between anchors and tokens from different modalities, which are applied for cross-modal fusion. We conduct comprehensive experiments to analyze the effectiveness of different components of ACE-RPE as well as the performance under different modes and hyper-parameter settings. As we know, this work is the first to present position embeddings for cross-modal tasks, and the experimental results also demonstrate the superiority of our method.

Limitations

Though the proposed ACE-RPE method achieves significant and substantial performance on 6 benchmarks. However, it has two major limitations: 1) the ACE-RPE is injected into backbone model during both pretraining and finetuning procedures. As we know, pretraining is much more time-consuming than finetuning. It will be more efficient to be implemented if it can maintain comparable results by simply initializing our models with a public released pretrained model, and only finetuning our models in downstream tasks. That is to say, the ACE-RPE is only employed in the finetuning model. We think it is worthy of more experimental results to study this kind of implementation. 2) The experiments in this paper are conducted on 8 NVIDIA A100 GPUs, which is expensive for personal researchers.

References


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


A Appendices

A.1 Shared V.S. Unshared

ACE-RPE could also be used in a shared mode for fewer parameters. In this part, we conduct experiments with shared ACE-RPE and compared the results with the unshared version. Table 6 shows that shared ACE-RPE would result in a slight performance drop on Image-Text Retrieval and Visual Reasoning task.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Flickr30K TR</th>
<th>IR</th>
<th>MS-COCO TR</th>
<th>IR</th>
<th>NLVR dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared</td>
<td>98.1</td>
<td>93.2</td>
<td>87.4</td>
<td>76.7</td>
<td>81.2</td>
<td>81.5</td>
</tr>
<tr>
<td>Unshared</td>
<td>98.3</td>
<td>93.5</td>
<td>87.7</td>
<td>76.8</td>
<td>81.7</td>
<td>81.9</td>
</tr>
</tbody>
</table>

Table 6: Ablation study on Image-Text Retrieval and Visual Reasoning task. The average recall on the test set is reported on Flickr30K and MS-COCO.

A.2 Robustness on Hyper-parameters

The default hyper-parameters of the proposed method are: $\lambda = 2$, $\delta = 0.05$ and $G = 8$. Table 7 presents the performance comparison of different choice of these hyper-parameters. Anchor loss with larger $\lambda$ (Eqn. 1) forces the model to learn more about the most similar anchor, while smaller ones reduce to predict more possible anchors. $\delta$ serves as the threshold parameter to select the anchors, and $G$ is the number of groups in the proposed multi-head distance. It is shown that, $\lambda$ and $G$ influence the performance more significantly compared with $\delta$. It is also indicated that as $G$ is greater than 8, the performance of ACE-RPE maintains almost unchanged.

<table>
<thead>
<tr>
<th>MS-COCO</th>
<th>$\lambda$</th>
<th>$\delta$</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>IR</td>
<td>85</td>
<td>9</td>
<td>87.3</td>
</tr>
<tr>
<td></td>
<td>75.0</td>
<td>76.8</td>
<td>76.4</td>
</tr>
</tbody>
</table>

Table 7: Ablation study on Image-Text Retrieval task on MS-COCO. The average recall on the test set is reported.

A.3 Bias V.S. Contextual Modes

ACE-RPE presents the position embedding of each text word and image patch. In this part, we propose two different cross-attention modes interacting with ACE-RPE, i.e., the bias mode and the contextual mode.

Bias Mode In this mode, ACE-RPE has no explicit interaction with the query, key or value in the transformer block. Instead, it functions as the bias of the cross-attention block. Formally,

\[
F_{i2i}(x, y) = \frac{(\pi W^Q)(y W^K)^T + E_{ace}(x, y) W_E}{\sqrt{D}}, \quad F_{i2i}(y, x) = \frac{(y W^Q)(x W^K)^T + E_{ace}(x, y) W_E}{\sqrt{D}}
\]  

(9)

where $F_{i2i}$ and $F_{i2i}$ are the image-to-text and text-to-image cross-attention, respectively. $E_{ace}(x, y) \in \mathbb{R}^{M \times N \times D}$ is a 3-dimensional tensor, denoting the ACE-RPE between all text tokens and image patches. $W^Q$ and $W^K$ are learnable matrices. $W_E \in \mathbb{R}^D$ is a learnable vector, which maps $E_{ace}(x, y)$ into a 2-dimensional matrix.

Contextual Mode ACE-RPE in contextual mode is first flatten into 2-dimension by average pooling, then added with the token/patch embedding.

\[
\begin{align*}
\pi_i &= x_i + \mathbb{E}_{j=1}^N E_{ace}(x_i, y_j) \\
\gamma_i &= y_i + \mathbb{E}_{i=1}^M E_{ace}(x_i, y_j)
\end{align*}
\]

(10)

The cross-attention is then,

\[
\begin{align*}
F_{i2i}(\pi, \gamma) &= (\pi W^Q)(\gamma W^K)^T \\
F_{i2i}(\gamma, \pi) &= (\gamma W^Q)(\pi W^K)^T
\end{align*}
\]

(11)

In this case, ACE-RPE interacts with the queries, keys in a cross-attention block. Besides, it can also be applied to value embeddings,

\[
\begin{align*}
Z_{i2i}(\pi, \gamma) &= \sigma(F_{i2i}(\pi, \gamma))(\gamma W^V + E_{ace})^T \\
Z_{i2i}(\gamma, \pi) &= \sigma(F_{i2i}(\gamma, \pi))(\pi W^V + E_{ace})^T
\end{align*}
\]

(12)

Here, $\sigma(\cdot)$ presents the softmax function, and $W^V$ is a learnable matrix. $E_{ace}$ is $E_{ace}(x, y)$ for short.

Experimental Result In this part, we compare the performances of two cross-modal modes, i.e., “Bias” and “Contextual” modes. Table 8 illustrates the numerical results in Image-Text Retrieval and Visual Reasoning task. Using the proposed ACE-RPE in contextual mode is demonstrated to be a better way.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Flickr30K TR</th>
<th>IR</th>
<th>MS-COCO TR</th>
<th>IR</th>
<th>NLVR dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias</td>
<td>98.1</td>
<td>93.4</td>
<td>87.4</td>
<td>76.6</td>
<td>81.5</td>
<td>81.6</td>
</tr>
<tr>
<td>Contextual</td>
<td>98.3</td>
<td>93.5</td>
<td>87.7</td>
<td>76.8</td>
<td>81.7</td>
<td>81.9</td>
</tr>
</tbody>
</table>

Table 8: Ablation study on Image-Text Retrieval and Visual Reasoning task. The average recall on the test set is reported on Flickr30K and MS-COCO.

A.4 Component-wise Analysis

Inspired by (Wu et al., 2021), in the field of image processing, the position embedding interacts with the calculation of the query, key and value in the self-attention layer. Accordingly, we analyze the recall of each choice in cross-modal modeling, and the results are shown in Table 9. It is shown...
that ACE-RPE calculated on values could only get slight gains over the version without ACE-RPE, but the ones embedded on queries and values would result in significant performance gains.

Table 9: Ablation study on Image-Text Retrieval and Visual Reasoning. The average recall on the test set is reported on Flickr30K and MS-COCO.

<table>
<thead>
<tr>
<th>Position</th>
<th>Flickr30K</th>
<th>MS-COCO</th>
<th>NLVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TR</td>
<td>IR</td>
<td>TR</td>
</tr>
<tr>
<td>×  ×  ×</td>
<td>97.8</td>
<td>92.7</td>
<td>86.5</td>
</tr>
<tr>
<td>✓  ×  ×</td>
<td>98.1</td>
<td>93.3</td>
<td>87.4</td>
</tr>
<tr>
<td>×  ✓  ×</td>
<td>98.1</td>
<td>93.2</td>
<td>87.5</td>
</tr>
<tr>
<td>×  ×  ✓</td>
<td>97.8</td>
<td>92.8</td>
<td>86.7</td>
</tr>
<tr>
<td>✓  ✓  ×</td>
<td>98.2</td>
<td>93.3</td>
<td>87.5</td>
</tr>
<tr>
<td>✓  ✓  ✓</td>
<td>98.3</td>
<td>93.5</td>
<td>87.7</td>
</tr>
</tbody>
</table>