Applying Positional Encoding to Enhance Vision-Language Transformers

Xuehao Liu[a], Sarah Jane Delany[b] and Susan McKeever[c]
School of Computer Science, Technological University Dublin, Ireland

Keywords: Image Captioning, Positional Encoding, Vision-Language Transformer.

Abstract: Positional encoding is used in both natural language and computer vision transformers. It provides information on sequence order and relative position of input tokens (such as of words in a sentence) for higher performance. Unlike the pure language and vision transformers, vision-language transformers do not currently exploit positional encoding schemes to enrich input information. We show that capturing location information of visual features can help vision-language transformers improve their performance. We take Oscar, one of the state-of-the-art (SOTA) vision-language transformers as an example transformer for implanting positional encoding. We use image captioning as a downstream task to test performance. We added two types of positional encoding into Oscar: DETR as an absolute positional encoding approach and iRPE, for relative positional encoding. With the same training protocol and data, both positional encodings improved the image captioning performance of Oscar by between 6.8% to 24.1% across five image captioning evaluation criteria used.

1 INTRODUCTION

Transformer-based models have been widely adopted in the fields of language and vision over the past five years. There are two essential parts of a Transformer-based model: the self-attention block and the positional encoding. The self-attention mechanism of the transformer method captures the long distance relationship between tokens more effectively than traditional Recurrent Neural Networks (RNN). However, it is invariant to sequence ordering of input tokens (Shaw et al., 2018). The same token (e.g. a word) in different positions of the input sequence (e.g. a sentence) is the same to the self-attention mechanism. The consequence of this is that valuable relative positional information is not used. For example, there are different meanings associated with "he genuinely needs to do that" versus "he needs to do that genuinely". Positional encoding is added to the input tokens as additional information, as it is a critical part for building the sequence order for the transformer. The vanilla transformer (Vaswani et al., 2017) added a sinusoidal signal in different frequencies on tokens in different location. Similarly, for visual input transformers, DETR (Carion et al., 2020) proposed 2d absolute positional encoding, which is two sinusoidal signals in two dimensions, in order to provide location information for object region features. Relative positional encoding has recently been introduced in other works (Shaw et al., 2018; Dai et al., 2019; Wu et al., 2021; Chu et al., 2021) as an improvement to the original absolute positional encoding.

In addition to vision-only and language-only tasks, transformers are now used in tasks that involve both modalities. Cross-modal transformers (Li et al., 2021; Chen et al., 2020b; Zhou et al., 2020; Yu et al., 2021; Li et al., 2020a) have received huge success in a variety of downstream tasks such as image captioning, by combining vision features and language token embeddings. The vision features are extracted from either Convolutional Neural Networks (CNNs) or object detectors. The transformer can have two self-attention blocks taking two modalities (Zhou et al., 2020) separately, or a single transformer encoder (Li et al., 2020b) for two kinds of input. Most research works in this domain have focused on the challenge of aligning vision representation and word embeddings. As a multi-stream transformer, Meter (Dou et al., 2022) shares the attention between two modality attention blocks. mPLUG (Li et al., 2022) proposes the asymmetric co-attention block, which allowed text encoder to take visual attention from any attention layer. Another simple improvement is to have a larger pretrain dataset (Li et al., 2020b; Chen et al., 2020b; Zhang et al., 2021; Wang et al., 2022).

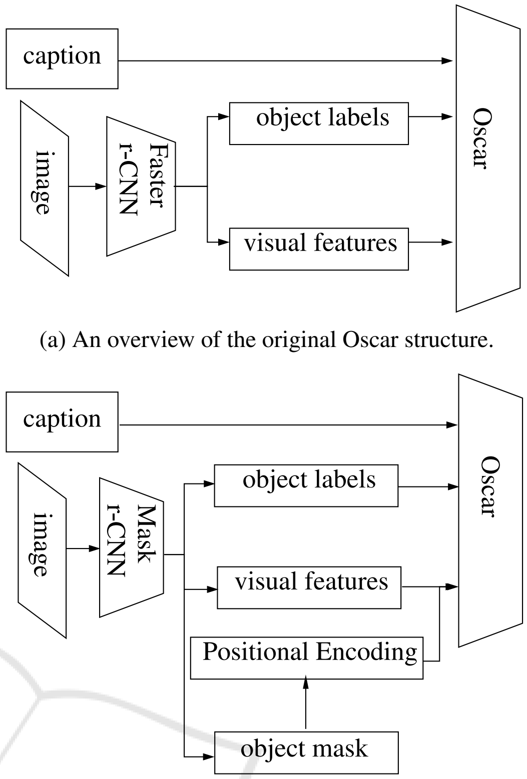
[a] https://orcid.org/0000-0001-9815-489X
[b] https://orcid.org/0000-0002-2062-7439
[c] https://orcid.org/0000-0003-1766-2441

Liu, X., Delany, S. and McKeever, S.
Applying Positional Encoding to Enhance Vision-Language Transformers.
DOI: 10.5220/0011796100003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 838-845
ISBN: 978-989-758-634-7; ISSN: 2184-4321

The visual representation input consists of the output vector from an object detector or a CNN classifier. The feature vector is concatenated with the height and width of the object bounding box (Li et al., 2020b; Yu et al., 2021). However, we did not find any previous study that refined the visual features with positional encoding for vision-language transformers. Our hypothesis is that the location of objects will contribute information, so including object location information using positional encoding could result in better models. To verify this, we implanted two typical positional encodings on a leading cross model transformer, Oscar (Li et al., 2020b); DETR a 2d absolute positional encoding approach, and iRPE the SOTA relative positional encoding approach. We found that simply adding DETR positional encoding with a Mask r-CNN (He et al., 2017) feature improved the performance of Oscar. Applying positional encoding on query, value and key in the self-attention head gave further improvement.

We summarise our contribution as follows:

- To the best of our knowledge, we are the first work that introduced positional encoding to vision-language pre-training transformers. We built the visual feature vectors with two kinds of positional encoding.
- With positional encoding, we demonstrate that with the same amount of training data, Oscar reaches a better image captioning performance compared to the original model. The Bleu4 score increased by 24.1%. The CIDEr score increased by 14.6%.
- The improvement of Oscar indicates that adding positional encoding into the vision-language transformers can enhance the performance of vision-language downstream tasks.



(a) An overview of the original Oscar structure.

(b) An overview of Oscar with positional encoding.

Figure 1: A comparison between the original Oscar (a) and Oscar with positional encoding (b).

2 RELATED WORK

The relatively recent success of transformer models is evident in their use as pre-trained models for vision and language tasks. While positional encoding has shown good success in object detection and image classification tasks, it has not been widely used in vision-language pre-trained models. This section describes the main vision-language models and outlines how positional encoding has been successfully used to date.

Since the transformer structure was first introduced (Vaswani et al., 2017), attention-based models have been the model of choice in both language and vision area. Pretrained language models such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2018) leveraged the advantage of the attention mechanism, with a better capability to model long term relationships compared to RNN methods. Moreover, off-the-shelf attention models are able to process visual inputs. Image GPT(iGPT) (Chen et al., 2020a), Pyramid ViT (Wang et al., 2021), Swin Transformer (Liu et al., 2021), and DETR (Carion et al., 2020) take visual features as input and use attention models to do object detection and image classification tasks.

Both multi-stream and single-stream transformers (Khan et al., 2022) have been applied to image captioning. Following the intuition of taking visual features as transformer input, ViLBERT (Zhou et al., 2020) was designed with two parallel transformer blocks as a co-attention framework, which takes visual feature vectors and language token embeddings separately (Vaswani et al., 2017). The output of different modality attention heads is then multiplied across the different modalities. ViLBERT is considered a multi-stream transformer.

Similarly, single-stream transformers have one transformer block for both visual and language inputs simultaneously. They take image region features and captions as input, and multiply them across the modalities at the first layer of the transformer encoder. Unicoder-VL (Li et al., 2020a), UNITER (Chen et al., 2020b), Oscar (Li et al., 2020b), and OFA (Wang et al., 2022) are all classified as single-stream transformers and have used a number of other techniques to improve the performance of transformer architectures. Unicoder-VL used three objectives in the pre-train process, Masked Language Modeling (MLM) which predicts a token based on the surrounding word and image features, Masked Object Classification (MOC) which included zero-padding in the input region feature and Visual Linguistic Matching which considers whether the vision-language inputs are semantically similar.

UNITER, however, included the objective of Masked Region Feature Regression (MRFR) in pre-training which includes a fully-connected layer on top of the transformer output. It learns L2 regression between the input region of interest features and the predicted vector from the transformer. Another general strategy used is to pre-train on multiple datasets to generalize the transformer further. UNITER is pre-trained on COCO (Lin et al., 2014), Visual Genome (VG) (Krishna et al., 2017), Conceptual Captions (CC) (Sharma et al., 2018), and SBU Captions (Ordonez et al., 2011). The large amount of image-caption pairs across different datasets provides extra generalization for the model to reach a better performance on downstream tasks such as image captioning.

Oscar innovatively changed the image-caption input pair to a caption-tag-image pair. The tags are English words obtained from the Faster r-CNN (Ren et al., 2015) object detector. Oscar also pre-trained the transformer with a larger group of datasets, including Open Images (Kuznetsova et al., 2020) and Object365 (Shao et al., 2019). More recently, OFA used multi and uni-modal data combined across more than 10 vision and language datasets, and trained across a wider range of downstream tasks. OFA achieved a higher performance in image captioning compared to other pre-trained transformers.

Positional encoding was first introduced for language transformers as a sinusoidal signal added between token embeddings and multi-head attention blocks (Vaswani et al., 2017). However, in vision transformers, the location cannot be encoded into a 1d sinusoidal signal. The original positional encoding is improved to 2d encoding to cater for image features.

All of these vision-language transformers are using the original 1-d positional encoding. None of the transformers examined exploiting positional encoding as an extra visual input. In this paper we explore adding positional encoding to the visual features.

Considering the absolute and relative position of a visual object, there are two kinds of positional encoding: Absolute PE and Relative PE:

Absolute PE adds the 2-d sinusoidal encoding directly to the image feature vector. ViT (Dosovitskiy et al., 2020) firstly applied both 1-d and 2-d positional encoding to the image visual input in a CNN. It demonstrated that even the image patches that are encoded in the raster order can significantly improve performance. DETR (Carion et al., 2020) then innovatively introduced 2d absolute positional encoding into a vision transformer. For positional encoding with length d DETR uses $d/2$ sine and cosine functions computed in different frequencies. Following the structure of DETR, Deformable-DETR (Zhu et al., 2020) added a sparse prior to the attention head. For a query element, Deformable-DETR will only focus on several elements based on the sparse prior, which reduces the training epochs for a better performance.

Relative PE adds the weighted sum of the sinusoidal encoding between attention layers. Relative positional encoding was first proposed in (Shaw et al., 2018). It is a weighted vector computed using the query and key based on a clipped relative distance. Transformer-XL (Dai et al., 2019) further improved Shaw's positional encoding by introducing a trainable offset for the query and key weight. In a simpler design, Huang et al. (Huang et al., 2020) proposed to subtract the relative position from the original absolute positional encoding. Image Relative Positional Encoding (iRPE) (Wu et al., 2021) showed that positional encoding can be added into the self-attention module with a bias or contextual mode. It also introduced the concept of adding positional encoding to any of the query, key, and value. All the works focus on adding positional encoding to word tokens, for language inputs. For visual input, (Ramachandran et al., 2019) proposed to replace convolutions with a fully attentional layer. The positional encoding added to the input is the 2-d relative distance to the central query pixel. CPVT (Chu et al., 2021) innovatively generates the positional encoding by doing a convolution operation on the original image feature.

3 APPROACH

To determine whether the performance of vision-language transformers can benefit from positional encoding, we applied positional encoding on a SOTA transformer, and applied multiple positional encoding approaches for comparison. We chose Oscar (Li et al., 2020b) as the example transformer. Although Oscar is a relatively simple typical single stream transformer architecture, it has competitive performance. In this section, we will firstly review the input structure and training objectives of Oscar. We will then explain our approach to including positional encoding into Oscar.

3.1 Example Transformer: Oscar

Oscar is a pre-trained transformer for vision-language downstream tasks such as text retrieval, image retrieval, image captioning, and visual question answering. We take image captioning as the target task for our work. Similar to other vision-language transformers, Oscar can take two types of modalities: token embeddings and vision features - noting that training Oscar for different tasks the input structure could be different. We took the off-the-shelf Oscar model with the same BERT self-attention backbone. The part changed is that the positional encoding was added to the input for training, as an additional visual feature.

3.1.1 Input Structure for Training

The input to Oscar is a triple representing three aspects of an image: the caption, the tags, and the object features. The caption is the word embedding sequence of the image caption. The tags are the English words for the object labels. In the original Oscar the object features are extracted from Faster r-CNN (Ren et al., 2015). The three parts of the input are separated by the special token [SEP], and the entire input sequence is started with the class token [CLS]. In our approach we use the tags for object labels and object features extracted from Mask r-CNN (He et al., 2017) given that we have to use the object mask to generate positional encoding rather than the location of object bounding box.

3.1.2 Pre-Training Objective

We follow the same loss objective as Oscar (Li et al., 2020b), and BERT (Devlin et al., 2018). The losses are computed on (i) Contrastive Loss: verifying two modalities of the input, the visual part(tags and object features) and the language part (caption); (2) Masked Token Loss (MTL): predicting the masked tokens.

- **Contrastive Loss:** The contrastive loss is from the perspective of the modalities. The model should be able to recognize whether the visual modality is pairing with language modality. In the transformer, the special token [CLS] is the representation of the vision-language input. Similar to Oscar, we generated 50% false input triples by replacing the visual part randomly across the dataset. Then we fully-connected the [CLS] embedding to predict if it is a triple from a real image or a false input triple.
- **Masked Token Loss (MTL):** In the language-only environment, given the surrounding tokens, the model should be able to retrieve the missing token where the context is a combination of language and vision. For each input sequence, we randomly masked 15% of the English word tokens with the special token [MASK], and predict this masked token.

3.1.3 Image Captioning Finetuning

After the pre-train process, Oscar has built the object-semantic mapping between objects and English tokens. The next step is to fine-tune Oscar to adapt it to the downstream task which is image captioning. There are two steps to image captioning finetuning: captioning pre-training and caption generation training. The loss objectives used are the seq2seq objectives of image captioning used in the original Oscar.

- **Captioning Pre-training:** The input of captioning pre-training is the same as the input structure of section 3.1.1. The loss objective is MTL loss in section 3.1.2. 15% of the input tokens are masked. The model predicts the corresponding missing token. The tokens in the caption part will be able to access the attention of both object labels and features, but it cannot reach the attention of the tokens behind the current token.
- **Caption Generation Training:** The input of image captioning are the object labels and feature vectors, rather than the triples as in pre-training. The goal is to infer the first part of the triple which is the caption. First the model takes the special token [CLS], the object labels and the object feature vectors. Second the generation starts with the model predicting a sampled token based on the input. This sampled token and a [MASK] token are the input for next round for the next word prediction. The whole inference process stops when the [EOS] token is predicted. Following Oscar, the objective of caption generation is SCST (Self-Critical Sequence Training) (Rennie et al., 2017) where the inference process is treated as a Reinforcement Learning process. The reward is based on the CIDEr (Vedantam et al., 2015) score against a random baseline.

Having examined Oscar's structure and downstream task training process, we move next to the positional encoding we have selected to apply to Oscar. We implemented two positional encoding approaches: The first approach we consider is DETR

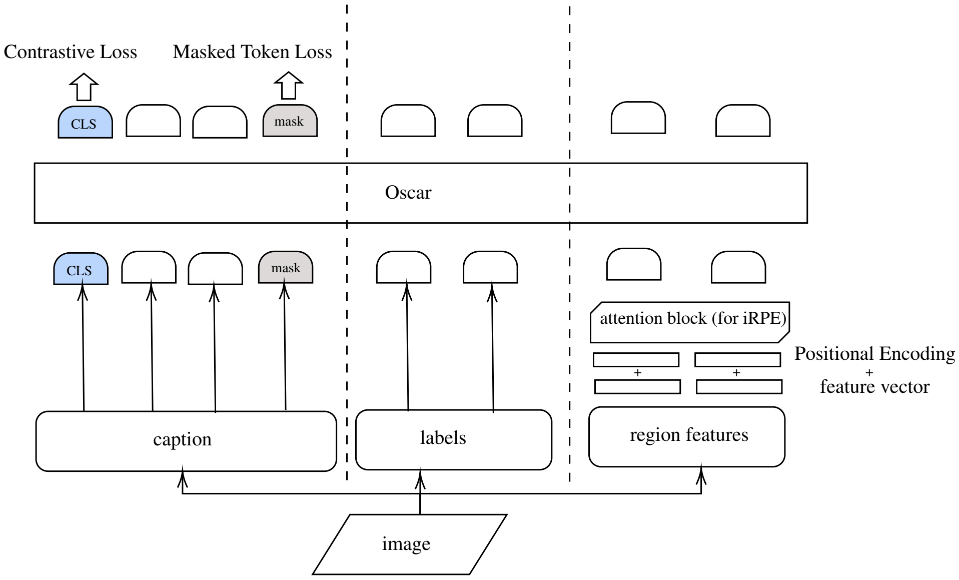


Figure 2: Illustration of Oscar with DETR/iRPE positional encoding: For both DETR and iRPE, the positional encoding is added to the feature vector before pushing into Oscar. The only difference is iRPE has one more multihead attention block than DETR, where the iRPE positional encoding can be added in a bias or contextual mode.

(Carion et al., 2020) a 2d absolute positional encoding that uses the original sinusoidal encoding (Vaswani et al., 2017). The performance of DETR positional encoding will determine if a 2d positional encoding applied to visual features will help the model. We then consider iRPE, a relative positional encoding that is more complex and achieves better performance improvements on transformer tasks than DETR. iRPE (Wu et al., 2021) was proposed as positional encoding for vision tasks only, for image classification and for object detection. In this paper we propose using it for image captioning, a vision-language task for which it hasn't been used before.

3.2 Oscar with DETR Positional Encoding

The original DETR is a vision-only transformer for object detection. The input is the feature vector extracted by ResNet50 (He et al., 2016). The positional encoding is calculated directly from the image feature. In our approach we use the 2-d mask generated from Mask r-CNN(He et al., 2017). For the 2d coordinates, $d/2$ sine and cosine functions, in different frequencies, are applied to the mask, and then they are concatenated together to a d dimension positional encoding. It will be added to the object feature vector directly before building the input triples. Figure 2 illustrates the structure of Oscar adding in the positional encoding.

3.3 Oscar with iRPE Positional Encoding

iRPE positional encoding is calculated based on the relative distance between objects. The positional encoding is a piecewise mapping function between the actual distance to a clipping distance to save computational cost. iRPE can be added to either the query, key or value in a contextual or bias mode. Before the first layer of the encoder, the positional encoding will be added to the object feature vector as the input of the iRPE self-attention block. This attention head is shown in Figure 2. The input to Oscar is the English token embeddings including both captions and object tags combined with the output of iRPE self-attention block.

4 EVALUATION & RESULTS

Our aim was to evaluate the impact of each of the two positional encoding schemes DETR and iRPE on the performance of Oscar, a vision-language transformer, for the task of image captioning.

Due to the limitation of training GPU resources, our aim is to simply establish an implementation of Oscar to work as a suitable baseline. We then add in the approaches for positional encoding to show that this improves on baseline performance.

Similar to the original work that proposed Oscar (Li et al., 2020b), both pre-training and image captioning training are on the COCO dataset (Chen et al., 2015). All approaches are evaluated on the COCO validation set of 5K images

All of the models are pre-trained for 5 epochs with a batch size of 256. Both the image captioning pre-training and finetuning are then conducted for 10 epochs. The original Oscar weight is downloaded from the original model zoo (Li et al., 2020b). For iRPE positional encoding, we used the product method and contextual mode, which was shown to be the best choice from the original iRPE paper (Wu et al., 2021).

We measured performance using the same metrics as originally used to evaluate Oscar. These include the following:

- **Bleu:** Bleu (Papineni et al., 2002) is a common metric for machine translation. It calculates the coexistence of n-grams between the ground truth and the predicted sentence. The Bleu4 that we used is comparing the 4-gram precision between the caption generated by the model and the ground truth.
- **METEOR:** METEOR (Denkowski and Lavie, 2014) is also a score focusing on the co-occurrence of word chunks where a word chunk is a sequence of n-grams. The length of word chunks is not limited by the length of n-grams This measure punishes small fraction chunks.
- **CIDEr:** For each n-gram in both reference sentence and predicted sentence, the term frequency inverse document frequency (TF-IDF) is calculated. The cosine similarity between the sentences is the final CIDEr (Vedantam et al., 2015) score.
- **Spice:** Spice (Anderson et al., 2016) parses a sentence to a direct graph, which is further deconstructed as tuples of words. The score is the F1 score on tuple hits between predicted and ground truth sentences.
- **Rouge_L:** Rouge_L (Lin, 2004) is also a widely used metric for text summarisation. Given the Longest Common Subsequence (LCS) between two sentences, Rouge_L is calculated as the F-measure between the sentences.

5 RESULTS AND DISCUSSION

Table 1 reports the image captioning performance for our scenarios: baseline Oscar and the addition of the two positional encoding approaches, DETR and iRPE. The original Oscar implanted with positional encoding from DETR is labelled *Oscar+DETR*. With iRPE relative positional encoding, we explored adding it in a number of ways as iRPE positional encoding can be added to any of the query, key or value. *Oscar+iRPE (Q)* means the Oscar is using positional encoding applied only to the query. Similarly, *Oscar+iRPE (QK)* and *Oscar+iRPE (QKV)* means the positional encoding is applied to the query and key and the query, key and value respectively.

Adding positional encoding improves on the baseline Oscar in all cases, across all five metrics. While better performance than the baseline is achieved with DETR, Oscar+iRPE (QKV) has the highest score in all 4 criteria except Bleu4. Generally iRPE significantly outperforms DETR. Oscar+iRPE (Q), iRPE applied to the query only, is the only iRPE implementation that does not outperform the less complex DETR positional encoding.

The improvement of adding positional encoding to the Oscar baseline is significant across all 5 evaluation metrics. The simpler absolute positioning approach of DETR (2d sinusoidal signals at different frequencies) is outperformed by the relation positioning approach in iRPE. Image captioning using iRPE improves by up to 24.1% when measured with Bleu4 and up to 14.6% when measured with CIDEr. Whilst our results are demonstrated for image captioning, our results suggest that improved positional encoding enriches the knowledge available to the model. This holds promise for improvements in other vision-language application areas such as visual question answering and image retrieval.

Our results are shown against a baseline implementation of Oscar. Future work can examine whether increasing to the level of epoch training (hundreds) and enlarged training sets used in Oscar's original implementation (Li et al., 2020b) impacts on the positional encoding results. Other visual language tasks can be examined with positional encoding to investigate the impact. We implemented two common positional encoding schemes, but there are abundant choices for further examination of performance impact.

6 CONCLUSION

In this paper we added two types of positional encoding into a SOTA vision-language transformer, Oscar. Positional encoding provides additional location information that has been shown to improve performance in vision-only transformers on vision-only tasks such as object detection and image classification. In this paper we have shown that positional encoding can significantly improve perfor-

Table 1: Image captioning performance on the COCO validation set. The percentage in parentheses is the improvement compared to the Oscar (baseline) performance.

	Bleu4	Metor	CIDEr	Spice	Rouge_L
Oscar (baseline)	0.277	0.249	99.6	0.184	0.528
Oscar+DETR	0.318 (14.8%)	0.257 (3%)	109.6 (10%)	0.189 (2.7%)	0.546 (3.4%)
Oscar+iRPE (Q)	0.316 (14.0%)	0.252 (1.2%)	103.9 (4.3%)	0.188 (2.1%)	0.547 (3.5%)
Oscar+iRPE (QK)	**0.344** (24.1%)	0.265 (6.4%)	112.4 (12.8%)	0.197 (7%)	0.560 (6%)
Oscar+iRPE (QKV)	0.342 (23.4%)	**0.269** (8%)	**114.2** (14.6%)	**0.200** (8.6%)	**0.564** (6.8%)

mance in vision-language transformers, on the task of image captioning. While absolute positional encoding (implemented using the DETR approach) improved on performance, relative positional encoding (using iRPE) had a significantly higher benefit.

We compared image captioning performance of Oscar with different kinds of positional encoding. Using the same set of data for training, the experiment results show that positional encoding improved image captioning performance. More work on training Oscar with more data and epochs could further validate the experiment. In addition, the implantation of other positional encoding approaches in different vision-language transformers is also promising future work.

REFERENCES

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020a). Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2020b). Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., and Shen, C. (2021). Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dou, Z.-Y., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., et al. (2022). An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, Z., Liang, D., Xu, P., and Xiang, B. (2020). Improve transformer models with better relative position embeddings. *arXiv preprint arXiv:2009.13658*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. (2020). The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981.

Li, C., Xu, H., Tian, J., Wang, W., Yan, M., Bi, B., Ye, J., Chen, H., Xu, G., Cao, Z., et al. (2022). mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*.

Li, G., Duan, N., Fang, Y., Gong, M., and Jiang, D. (2020a).

Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. (2021). Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020b). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.

Ordonez, V., Kulkarni, G., and Berg, T. (2011). Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., and Sun, J. (2019). Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. (2022). Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578.

Wu, K., Peng, H., Chen, M., Fu, J., and Chao, H. (2021). Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041.

Yu, F., Tang, J., Yin, W., Sun, Y., Tian, H., Wu, H., and Wang, H. (2021). Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3208–3216.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. (2021). Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., and Gao, J. (2020). Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.