A LEARNABLE LINGUISTIC POSITION EMBEDDING

In this section, we describe the learnable linguisic position embedding in detail (introduced in Sec. 3.1). To inject positional information of each tokens in multi-modal sequence, position encoding is an important procedure of transformer encoder. Existing works Deng et al. (2021); Li & Sigal (2021) project linguistic token indices to high dimensional space by linear layers. We design a new position embedding method which unifies the position encoding of multi-modal sequence into 2D-sinusoidal embedding space. Specifically, for visual tokens, we follow Carion et al. (2020) using sinusoidal positional encoding PE_{sin} to generate position embedding $\mathcal{P}_v$, while for each linguistic token, we leverage Multilayer Perceptrons to project the BERT embedding $\mathcal{F}_l$ and indices of tokens $X = \{0, 1, 2, ..., T\}$ into 2D coordinate:

$$X_1, X_2 = \sigma(\text{MLP}(\mathcal{F}_l) + \mathbf{W}X), \tag{6}$$

where MLP learns 2D coordinates from BERT embedding, and $\mathbf{W} \in \mathbb{R}^{2\times1}$ projects 1D indices into 2D coordinates, σ denotes sigmoid activate function, X_1 and X_2 is the 2D position of linguistic tokens. Then we encode the 2D indices X_1, X_2 by the same PE_{sin}, that is:

$$\mathcal{P}_l = [PE_{sin}(X_1); PE_{sin}(X_2)], \tag{7}$$

where '[;]' denotes concatenation operation, $\mathcal{P}_l$ denotes the final language position embedding.

This simple implementation of unified 2D sinusoidal position embedding benefits our model from various aspect. For example, as described in Sec. 3.2, we leverage a proto-decoder to exploit the position prior and generate the first object query and anchor query. In an other word, this module decomposes a language query into two different type of queries, where the inner mechanism is: the cross attention layer decoupled semantic and positional attention calculation. The semantic similarity from query to key is conducted by dot products between visual and linguistic encoded features $\mathcal{F}_v$ and $\mathcal{F}_l$, while the positional similarity is the dot products between 2D sinusoidal position embedding $\mathcal{P}_v$ and $\mathcal{P}_l$. Benefited from our learnable language position embedding, the positional part can be naturally seen as calculate the correlation between language feature and visual patch positions, resulting a position prior over the image of the language query.

B POSITION PRIOR FROM LANGUAGE

In this section, we visualize the position prior learned from linguistic information. It is sensible that a good REC model should be capable of estimating the location of object directly from language when the given language provides a strong position prior. As shown in Fig. 5, our model can predict the box precisely while RefTR Li & Sigal (2021) fails to handle these preposition words (*e.g.* under, above *etc.*) appeared in language.

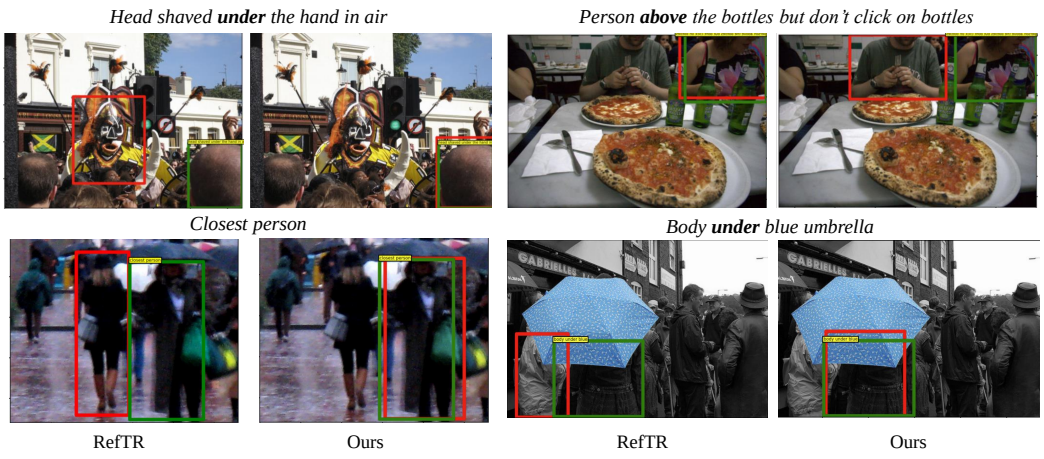

Figure 5: The comparison between our model and RefTR Li & Sigal (2021) when language contains directional words. Examples are from the validation set of RefCOCO+.

To have a deep understanding of exploiting position prior directly from language, we eliminate the impact of visual information during evaluation: we manually generate several image-sentence pairs, where each image is random Gaussian noise and each sentence contains strong position information, such as “left object”, “right object”. The results are shown in Fig. 6.

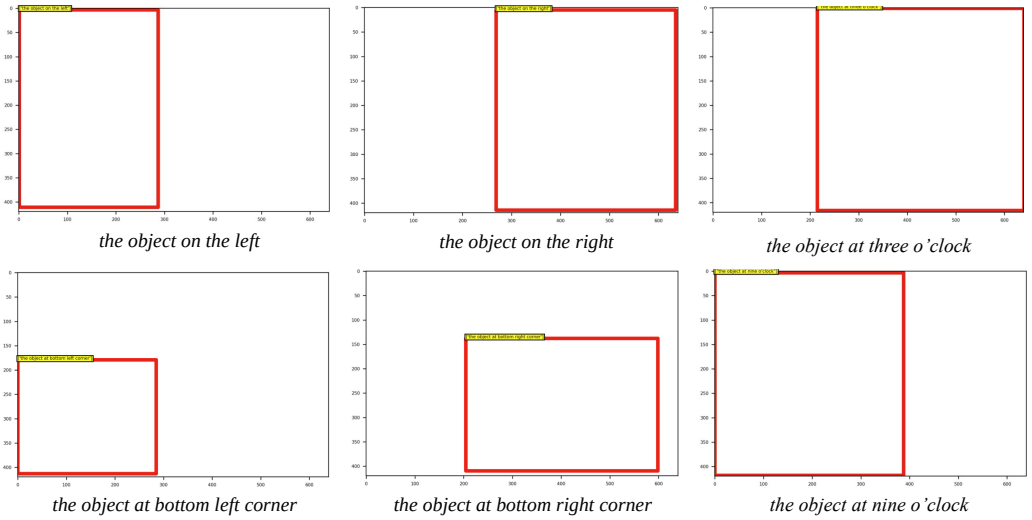

Figure 6: Visualization of our method when only language information provided and the image input are Gaussian noises. Examples of first and second columns are generated by model trained on RefCOCO dataset, and the last column examples are generated by model trained on RefCOCO+.

The first and second columns in Fig. 6 shows the prediction of our model trained on RefCOCO Yu et al. (2016) dataset. We observe that when the visual information is unavailable, the model can still learn the position prior from language directly, locating the “object” under the guidance of preposition words appeared in sentence. The third column shows prediction of our model trained on RefCOCO+ Yu et al. (2016). Notice that in this dataset, some location word like “left” or “right” are taboo words. Instead, using “on nine o’clock” or “on three o’clock” to represent the direction of the object. As shown in Fig. 6, our model can identify these directional words and estimate the position of the object.

Table 7: Comparison with state-of-the-art pretrained-based methods on four mainstream REC datasets in terms of Acc@0.5.

Method	Language Backbone	RefCOCO val	RefCOCO test A	RefCOCO test B	RefCOCO+ val	RefCOCO+ test A	RefCOCO+ test B	RefCOCOg val	RefCOCOg test	ReferIt test
Pre-trained:										
ViLBERT Lu et al. (2019)	WordPiece	-	-	-	72.34	78.52	62.61	-	-	-
ERNIE-ViL-large Yu et al. (2021)	WordPiece	-	-	-	75.89	82.37	66.91	-	-	-
UNITER-large Chen et al. (2020)	WordPiece	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77	-
VILLA-large Gan et al. (2020)	WordPiece	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71	-
RefTR-pre-trained Li & Sigal (2021)	BERT	85.65	88.73	81.16	77.55	82.26	68.99	79.25	80.01	76.18
MDETR Kamath et al. (2021)	RoBERTa	87.51	90.40	82.67	81.13	85.52	72.96	**83.35**	**83.31**	-
LUNA-pre-trained (Ours)	BERT	**88.85**	**91.31**	**84.75**	**81.26**	**86.05**	**74.85**	82.78	82.00	**79.52**

C MORE VISUALIZATION OF ATTENTION MAP

In Figure 7, we visualize more attention patterns and predictions of our model and RefTR Li & Sigal (2021). Same as aforementioned in Sec. 4.5, thanks to the position prior, the attention of our model better focus on the region language referred.

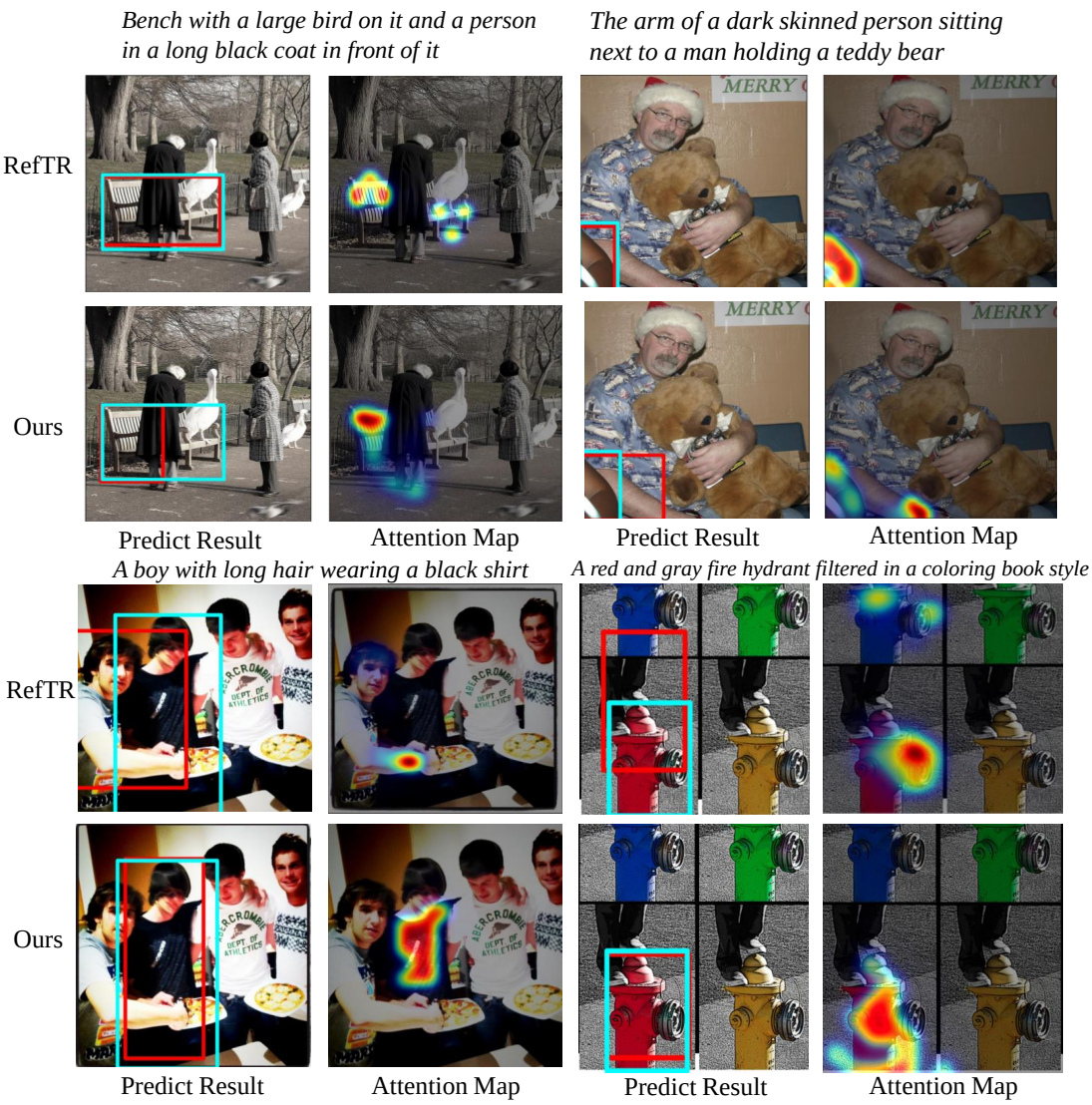

Figure 7: More visualization of attention map from the last decoder layer and the final predictions of our method and RefTR Li & Sigal (2021). Cyan boxes are the ground truth and red boxes are predictions. Examples are from the validation set of RefCOCOg Nagaraja et al. (2016).

D PRETRAINING RESULT

Notably, LUNA outperforms RefTR Li & Sigal (2021) and MDETR Kamath et al. (2021) (which also employ bounding box-level supervision during pre-training similarly to LUNA) on all subsets of RefCOCO and RefCOCO+. Our method comes to the second only on the RefCOCOg subsets after MDETR, which uses twice as much pre-training data Kamath et al. (2021).

E ALTERNATIVE STUDIES OF PROTO-DECODER

In this section, we studied multiple alternative options of proto-decoder, the variant structures and experiment results are shown in Table 8.

The several replacable cases are: (1) A one-layer Transformer encoder ("Self-attention" in Table 8), where the output is a vision-language sequence which we average pool for obtaining Q.

(2) a variant of the proto-decoder ("reversed proto-decoder" in Table 8) in which the key, value are word embeddings and language position embeddings and the query is image region embeddings and visual position embeddings, where the output is a sequence of image region embeddings which we average pool for obtaining Q.

(3) a pair of parallel cross attention layers to calculate content embeddings and positional embeddings correlations respectively between image and sentence, instead of one cross attention layer.

(4) The cross attention layer in proto-decoder only calculate similarity of content embeddings then pooled without splitting. The first anchor query is randomly initialized.

(5) Calculate the similarity between language and visual positions via language content embeddings and visual position embeddings (instead of learnable language position embedding and visual position embeddings in Figure 3)

Methods	val	test
Self-attention	70.42	70.36
Reversed proto-decoder	72.18	71.56
Parallel Cross Attention	71.82	71.47
Content only Cross Attention	70.50	70.88
2 $\times$ Lang. Content Query	72.01	71.63
Ours Proto-decoder	**74.06**	**72.75**

Table 8: Alternatives to the proto-decoder.

F Anchor Query Formulation

In this section, we supplement a key difference between our CA-guided decoder and DAB-DETR Liu et al. (2022) that is important to the effectiveness of our model.

We formulated the anchor query Q_{ai} in the CA-guided decoder as generating from the concatenated embeddings of the center point coordinates, the height, and the width of the previous anchor box (Eq. 3 in the main paper), while DAB-DETR only directly encodes the center point coordinates but uses the height and width as coefficients in modulated cross attention (Eq. 6 in Liu et al. (2022)). We present this comparison on anchor query formulation in Table 9.

Methods	val	test
Modulated attention (DAB-DETR)	70.54	69.77
Attention based on our query	**74.06**	**72.75**

Table 9: Comparison between the modulated attention formulation of DAB-DETR and ours attention for encoding the anchor query.

As shown in Table 9, our formulation of anchor query out-performs DAB-DETR's by a large margin. Observing the comparison result, we can see that it is crucial to embed wight and height information of the anchor queries during decoding process in REC task, so that the scale of anchor box could attend to language information in cross-attention calculation of each layer and the model is more sensible to the relevant language description (*e.g.* large, small *etc.*).