Probing the Role of Positional Information in Vision-Language Models

Philipp J. Rösch[1] and **Jindřich Libovický**[2]
[1]Institute for Distributed Intelligent Systems, Bundeswehr University Munich, Germany
[2]Faculty of Mathematics and Physics, Charles University, Czech Republic
philipp.roesch@unibw.de libovicky@ufal.mff.cuni.cz

Abstract

In most Vision-Language models (VL), the understanding of the image structure is enabled by injecting the position information (PI) about objects in the image. In our case study of LXMERT, a state-of-the-art VL model, we probe the use of the PI in the representation and study its effect on Visual Question Answering. We show that the model is not capable of leveraging the PI for the image-text matching task on a challenge set where only position differs. Yet, our experiments with probing confirm that the PI is indeed present in the representation. We introduce two strategies to tackle this: (*i*) Positional Information Pre-training and (*ii*) Contrastive Learning on PI using Cross-Modality Matching. Doing so, the model can correctly classify if images with detailed PI statements match. Additionally to the 2D information from bounding boxes, we introduce the object's depth as new feature for a better object localization in the space. Even though we were able to improve the model properties as defined by our probes, it only has a negligible effect on the downstream performance. Our results thus highlight an important issue of multimodal modeling: the mere presence of information detectable by a probing classifier is not a guarantee that the information is available in a cross-modal setup.

1 Introduction

Pre-trained Vision-Language models (Tan and Bansal, 2019; Lu et al., 2019; Yu et al., 2021; Chen et al., 2020) reached strong performance in many multimodal tasks such as Visual Question Answering (Antol et al., 2015; Hudson and Manning, 2019; Bigham et al., 2010) or Visual Inference (Johnson et al., 2017; Suhr et al., 2019). All these models use the Transformer architecture (Vaswani et al., 2017) and make use of several pre-training strategies like *Masked Cross-Modality Language Modeling* (MM) and *Cross-Modality Matching* (CMM) similar to masked language modeling and next sentence prediction (Devlin et al., 2019) in NLP.

Because the attention mechanism treats its inputs as unordered sets, Transformer-based NLP models need to use position encodings to represent the mutual position of the tokens, so that the models can grasp the sentence structure. The mutual position of objects is equally important for understanding the structure of an image. VL models differ in how they represent objects in the image, typically represented as sets of object features and PI. Therefore, object detectors are used to obtain bounding box information for all objects. In many models, the upper left and lower right corners of the object's bounding box are used as 2D information to create a learnable positional encoding. In addition to the spatial but flat 2D values, we determine the depth of the objects in the image and make it available as an additional feature. Until now, VL models recognize the objects on a flat map but not in the real three-dimensional context.

We found that the current LXMERT model is capable of forwarding PI through the model but is not capable to use it to solve image-text matching tasks where positional keywords are replaced by their counterparts. Introducing two new pre-training strategies, we target these unimodal and multimodal evaluation schemes and improve probing results. Yet, we did not get any perfomance increase on the downstream tasks. This is most likely due to the small fraction of position-related text in the pre-training corpus and suboptimal results of the object detector. Regarding PI type, it seems to be sufficient to input object center values which is far less than most VL model input today.

2 Positional Information in VL Models

In NLP, the importance of word order is given great attention (Ke et al., 2021; Wang and Chen, 2020). Different methods exist, including analytical position encodings (Vaswani et al., 2017), learnable

PI Type	Models
$\emptyset$	CLIP
$x1, y1, x2, y2$	LXMERT, M4C
$x1, y1, x2, y2, \frac{a}{wh}$	ViLBERT, Unicoder-VL, ERNIE-ViL
$x1, y1, x2, y2, w, h$	OSCAR
$x1, y1, x2, y2, w, h, a$	UNITER

Table 1: Positional information in Vision-Language models. Most models use the upper left and lower right of the object's bounding box $(x1, y1, x2, y2)$. Some models add the absolute (a) or relative object area $(\frac{a}{wh})$ in combination with the image width (w) and height (h). The object depth (d) is not used and $\emptyset$ denotes no PI.

additive embeddings (Devlin et al., 2019) or the relative the attention query (Shaw et al., 2018). There is no equivalent research that would specifically approach PI in VL models. However, the position of the objects is considered in almost all common Transformer-based approaches.

In LXMERT (Tan and Bansal, 2019) the upper left and lower right corners of the object are used to encode its position. The same is true for M4C (Hu et al., 2020). Other models also use the relative area fraction of the objects as an additional feature. Although the network should be able to determine this feature, it is explicitly added, as in case of ViLBERT (Lu et al., 2019), Unicoder-VL (Li et al., 2020a), and ERNIE-ViL (Yu et al., 2021). UNITER (Chen et al., 2020) uses – in addition to the objects' corners – the absolute object area and the image width and height. OSCAR (Li et al., 2020b) uses bounding box and image height and width. Only CLIP (Radford et al., 2021) does not use PI, although they use another pre-training concept. See Table 1 for an overview. To our knowledge, there is no structured analysis of PI in VL models.

Current models use only 2D object information. By introducing depth as a new feature, we represent objects in the 3D space. This is not only important to be able to define the distances between objects but also to have a more meaningful understanding of the object sizes. Using the area of the bounding box without depth information does not add the actual object size information since the sizes depend on the depth localization of the object.

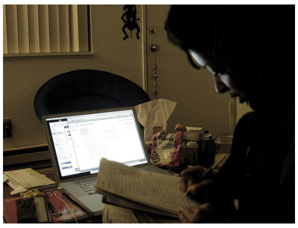
3 Evaluation of Positional Information

To determine the capability of current models with regard to PI, we experiment with three evaluation methods. Firstly, we perform an intrinsic evaluation to determine whether the PI passes through the model. Secondly, we test if the models are capable of utilizing PI using the CMM task. Lastly, we report extrinsic results for GQA downstream task (Hudson and Manning, 2019) on different data subsets. We report the results of the probing experiment in Section 5.

For our experiments, we use four types of PI. An empty set $(\emptyset)$ acts as a baseline. Object center values (x, y) act as a coarse identification of where the object is located. Moreover, we evaluate $x1, y1, x2, y2$, which is the standard representation of bounding boxes and is also often used in VL models. This PI description contains information about object width, height, and area. Therefore, we ignore further settings that add these types to the input in our evaluation. Since we are also interested in analyzing depth, we investigate the setting $x1, y1, x2, y2, d$ as well.

Mutual Position Evaluation. In the intrinsic evaluation task, we test if PI is forwarded through the whole model. We use nine different pairwise classifiers for different mutual positions, which are applied to all detected objects. LXMERT uses a fixed number of 36 objects as its input. This leads to a total number of $9 \times 36 \times 36 = 11,664$ classifications for each input image.

We use six classifiers for 2D spatial relations (operate on X and Y coordinates) and three for depth information (Z coordinate). The tasks are (1) whether the center of one object is more to the left than that of another object, (2) the same if the center is closer to the bottom, (3) whether one object is completely left of the other object (without an overlap), (4) and the same for being completely below the other object, (5) whether one object is completely inside the other bounding box, (6) and if there is no overlap in the X and Y dimension. Regarding depth, the model needs to correctly classify (7) if one object is more in the foreground regarding the median value, (8) if one object is in between the inner 50% of the other object using all pixel values, and (9) if all depth values of one object are significantly smaller than the values of the other object at a significance level of 0.05 using a t-test.



Original caption: "A student works on an academic paper at her desk, computer screen glowing in the background."

Figure 1: Pre-training data with image and description with a PI keyword. For contrastive evaluation the keyword is replaced by its counterpart (i.e. "foreground").

These classification tasks have the same inputs as the *Masked Object Prediction* task (see Section 4.1.1) and are also constructed in the same manner (see Appendix A.1). An overview of the visual pre-training tasks is provided in Figure 3. Because this is a probing task, the classification head (PI head) is not updated during pre-training. After the training process has finished, all model parameters are frozen and only the weights in the PI heads are updated for 1 epoch. The average accuracy of all 11,664 classification tasks is reported on the MS COCO validation dataset. In doing so, we evaluate the unimodal capabilities of the model to forward information through the whole Transformer. The detailed results are presented in Appendix A.6.

Contrastive Evaluation on PI using CMM. The CMM classifier can successfully match images and captions (91% accuracy on the balanced pre-training validation data). However, this says little about the type of information considered during the classification. To better assess if PI is used by the model, we build a challenge set consisting of pairs of contrastive examples. We filter the validation data for samples with keywords indicating spatial relation between objects and only keep those which are replaceable by antonyms (see Appendix A.2).

We run two evaluation setups: (1) We replace all image descriptions with a random caption of a different image (following the LXMERT pre-training strategy). (2) We take the image and for all captions we replace the PI keyword with its antonym, e.g., substitute *background* with *foreground* and vice versa. See Figure 1 for an example. This task determines if the model is able to understand PI in a multimodal fashion. In both cases, we only have samples with "no match" ground truth values (which is our positive class)[1], and consequently we report recall only.

Downstream Task Evaluation. Finally, we determine the performance of the model on a downstream task. We use GQA, since it is a carefully balanced image question answering dataset, where PI plays a role. We report the 1- and 5-best accuracy. Moreover, we evaluate (top 1) accuracy of data subsets where X, Y, and Z coordinates are important. We do this by selecting questions where specific PI keywords are present (see Appendix A.3).

Since keyword search does not work perfectly (e.g., *Which color is the bag on the back of the woman?*), we employ zero-shot text classification using a BART model[2] (Lewis et al., 2020). For zero-shot classification we need a candidate label, which is used as input to determine if both texts (i.e. caption and candidate label) fit together. We experimented with different labels and found that the simple keyword "position" works best for our use case.

Downstream evaluation is done on the GQA *test-dev* split, which has 12,578 samples, hence, an change of 0.1% is equivalent with approximately 13 more correctly classified samples. For the subsets where X, Y, Z keywords are present, the dataset size is 2,050, 1,203 and 1,349 respectively. For the zero-shot subset (indicated with P) the sample size is 1,349.

4 Model and Data

4.1 Model

Our experiments are built upon LXMERT – a Transformer-based model with two separate encoders for image and text modality and one cross-encoder to join both. LXMERT was the only model in the top-3 leaderboard in both the VQA v2.0 2019 and GQA 2019 challenge, which is why we use this model as the basis for our work. Details are provided in Section 4.1.1. A detailed description of how the object's depth feature is determined is provided in Section 4.1.2.

4.1.1 Base Model

LXMERT uses Faster R-CNN with ResNet-101 for the object detection task, originally introduced by Anderson et al. (2018). The object detector is trained on Visual Genome (Krishna et al., 2017) predicting 1,600 objects with 400 different attributes (mostly adjectives). For LXMERT the model extracts the 36 most confident objects with

[1] Hence, we have $FP = TN = 0$.

[2] https://huggingface.co/facebook/bart-large-mnli

the region-of-interest features f_j, the object class c_j, attribute a_j and the positional information p_j, where j indicates the object indexes $j = 1, \ldots, 36$. The feature map ($\mathbb{R}^{36\times 2048}$) and the bounding box coordinates ($\mathbb{R}^{36\times 4}$) are passed to two separate linear models with weight matrix W and bias b. The output is further processed by two layer normalizations (LN) and finally both results are averaged:

$$\hat{f}_j = \mathrm{LN}(W_F f_j + b_F) \qquad \hat{p}_j = \mathrm{LN}(W_P p_j + b_P)$$

$$v_j = (\hat{f}_j + \hat{p}_j)/2$$

This leads to a unified embedding $v_j \in \mathbb{R}^{36\times 768}$ representing the content of the objects and the positions at the same time. The image data is further processed in a BERT-style encoder.

On the language side, the text input is processed in a BERT-style encoder as well. Both outputs are merged in a cross-modality encoder (X-Enc) and passed to the output heads, where the losses for each pre-training strategy are calculated. The LXMERT architecture can be investigated in Figure 2.

The same pre-training strategies are used, namely *Masked Cross-Modality Language Modeling (MM)*, *Cross-Modality Matching* (CMM), *Visual Question Answering* (VQA), and *Masked Object Prediction*. The last one is composed of three tasks: two classification tasks to predict the object classes and attributes (*ObjClassif*, *AttrClassif*), and a regression task to predict the feature vector (*FeatRegr*). See Tan and Bansal (2019) for all details. Note that all pre-training strategies explicitly focus on the object features f_j, c_j, and a_j and not on the PI. The same is true for other VL models listed in Table 1. See Figure 3 for an illustration of all visual pre-training tasks.

We used the original implementation of LXMERT[3] and only made minor changes. We introduced dropout with $p = 0.1$ in the IQA head. Further, we tested different training hyperparameters to find a good ratio between model performance and training time. Our final pre-training model setup has a batch size of 2048 with a learning rate of 10^{-4} (with the same learning rate scheduler), the fine-tuning model has a batch size of 32 and a learning rate of 10^{-5}. Introducing PyTorch's *DistributedDataParallel* in the code and using 8 instead of 4 GPUs reduced the pre-training time from approximately 8.5 days to 41 hours. We used the pre-training weights reported in the paper and not in the corresponding repository (see Appendix A.4).

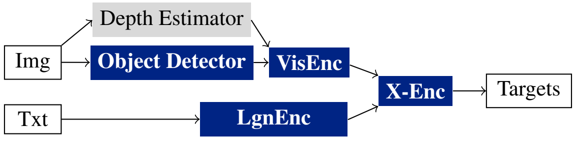


Figure 2: Architecture of the LXMERT model (blue) with depth information extension (gray). LXMERT uses object detection from Anderson et al. (2018) and has 5 visual, 9 language and 5 cross-modality (X-Enc) layers.

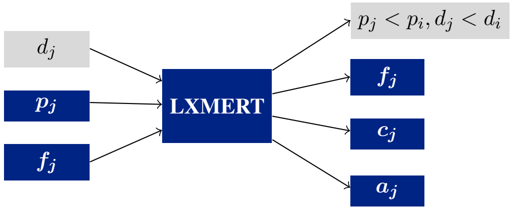


Figure 3: Visual components for the pre-training phase (text components omitted). Input data (f_j, p_j) to the visual encoder and training targets (f_j, c_j, a_j) for LXMERT's pre-training strategies are indicated in blue. Our additional depth data d_j and PI pre-training labels (PIP) are colored gray.

4.1.2 Depth Information

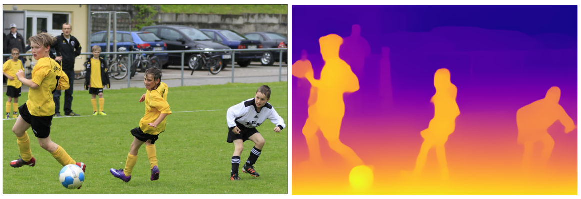
The datasets used for training LXMERT do not provide any depth information. To obtain depth values for each pixel in the image, we used MiDaS v2.1[4] (Ranftl et al., 2020) – a state-of-the-art algorithm for monocular depth estimations. It is trained on diverse datasets from indoor and outdoor environments, containing static and dynamic images and images of different quality. Hence, it fits the various picture types in our datasets. See Figure 4a for an original COCO image and Figure 4b for the depth information provided by the MiDaS model.

The depth predictions from MiDaS can be any real number. Large numbers indicate close objects, and small numbers refers to distant objects. We linearly normalized each pixel x_i with $1 - \frac{x_i - min(x)}{max(x)}$ to obtain 0 for the closest pixel and 1 for the most distant one for each individual image.

Since the rectangular bounding boxes do not surround the objects perfectly, we experimented with the object's center value, the mean and median as heuristic. We finally used the median, due to its robustness. Furthermore, it would be conceivable

[3] https://github.com/airsplay/lxmert/

[4] https://pytorch.org/hub/intelisl_midas_v2/



(a) Original image (b) Depth estimation

Figure 4: We use a monocular depth estimator to obtain a pixel-level depth prediction. We normalize the output that 0 (yellow) indicates the value that is at the very front and 1 for the furthest pixel (violet).

to additionally take the standard deviation as a measure for uncertainty if the object is on specific depth plane or spans over a larger distance. This issue can be avoided with panoptic segmentation (Kirillov et al., 2019), which we leave to the future work.

4.2 Data

Following the original LXMERT setup, our models are pre-trained using the MS COCO (Lin et al., 2014) and Visual Genome (VG; Krishna et al., 2017) data in conjunction with some Visual Question Answering task (VQA). There are in total 9.18M image-caption pairs with 180K unique images. The average sentence length per caption is 10.6 words for MS COCO and 6.2 words for VG. The sentences are short and do not provide many details. Using 10 words, only the main occurrence of the image can be described. See examples in Appendix A.5.

In Table 2, we show the relative occurrence of PI keywords (see Appendix A.3). Pre-training data do not have a lot of PI in the captions or questions. Only Y keywords appear more often (11.2%) in MS COCO and X keywords in VQA (10.7%). This is different in GQA, which we use for downstream evaluation. In the *train* part, there are many X keywords, but only a few Y and Z keywords. The distribution in the *testdev* set is different. Here, the number of X, Y, and Z questions is high.

LXMERT was also evaluated on VQA v2.0 (Goyal et al., 2017) and NLVR2 (Suhr et al., 2019). VQA v2.0 has PI relations (under 3%), so we do not run an analysis on this dataset. NLVR2 has positional relations, but PI keywords are often part of the left/right image assignment and do not indicate objects within an image according to our definition of PI. Moreover, the presence of multiple images rules out a clean analysis of PI.

Dataset	**X**	**Y**	**Z**
MS COCO	2.9	11.2	6.5
VG	3.4	3.8	4.6
VQA	10.7	3.3	4.0
GQA *train*	28.4	5.3	4.9
GQA *testdev*	16.3	9.6	10.7

Table 2: Occurrence of positional keywords in percent in pre-training (top lines) and downstream datasets (bottom lines).

5 Probing Results

This section shows the results of the experiments described in Section 3.

Mutual Position Evaluation. We determine whether PI can be passed through the model using the classifications of the PI head. Results are shown in Table 3 (top lines). The accuracy is only 80.0% for no PI, but over 88% for the remaining types. This confirms that the model is able to forward PI through the whole Transformer layer stack.

Interestingly, the model is often capable of correctly classifying the mutual position of objects, although PI is not used as the model input. This is most likely due to the high correlation between the object categories and their positions. For example, “shoes” are usually at the bottom and in the foreground. The object detector is not powerful enough to detect small objects in the background in general. “Sky” and “clouds” are usually at the top and in the background of the image. Detected objects such as “kitchen” or “office” often span the whole image width and therefore have their center in the middle of the X axis. The latent image representation f_j can be used as a proxy for object types.

In addition to that, we can see that with more PI the accuracy of this task increases by more than eight percent points and has a peak at 89.7% for the input setting $x1, y1, x2, y2, d$. Switching from object centers to bounding boxes only has a minor impact. Yet, adding depth improves accuracy on the three Z related tasks (see Appendix A.6), which boost the overall performance.

Contrastive Evaluation on PI using CMM. To further evaluate the use of PI in VL models, we test if the model can utilize the information using the CMM task. Table 4 (top lines) shows that the original setting with dissimilar image-text pairs can be predicted almost perfectly – the recall is always above 96%. Hence, this pre-training strategy be-

	PI Input	XYZ Acc	XY Acc	Z Acc
Probing	$\emptyset$	80.0	81.5	77.1
	x, y	88.5	92.1	81.1
	$x1, y1, x2, y2$	88.7	92.4	81.3
	$x1, y1, x2, y2, d$	89.7	92.2	84.7
Pre-training	$\emptyset$	88.2	88.9	82.1
	x, y	91.6	94.4	86.0
	$x1, y1, x2, y2$	92.1	94.9	86.5
	$x1, y1, x2, y2, d$	93.9	94.8	92.2

Table 3: Mutual Position Classification Evaluation: Mean accuracy of all 9 mutual classification tasks (XYZ), 6 XY tasks, and 3 Z tasks for pre-trained models for different PI inputs. Upper lines for plain LXMERT and lower lines with our version (PIP, CL; see Section 6).

haves as expected for the normal data provided. Yet, the model cannot apply fine-grained details from textual PI. It is not capable of correctly rejecting that, for example, *"A student works on an academic paper at her desk, computer screen glowing in the foreground."* does not fit to the image from Figure 1. The recall is steadily below 2%.

The model is able to pass through PI in the visual Transformer part but is not able to use it in a cross-modal fashion for solving problems. This is probably due to the fact that fine-grained matching does not play a role during pre-training. CMM is not constructed as indicated above (i.e., *background* vs. *foreground*) but to select completely dissimilar statements like *"A man sits before a light meal served on the table of a travel trailer"* to the image in Figure 1. To overcome this problem, we need more advanced negative sampling, i.e. captions that are closer to the original image-text pairs.

Downstream Task Evaluation. We evaluate downstream performance on GQA *testdev* with four different subsets targeting X, Y, Z keywords and general positional (P) samples. The results (in Table 5 top lines) reveal that using any type of PI is better or equally good than not using it (except for Y in $x1, y1, x2, y2$). Although the improvements are small, they indicate that PI is indeed helpful in this downstream task.

The best top 1 and X subset results are achieved by x, y input type. This might be due to the fact that most object relations are distinct and center values are sufficient to track this relationship. For example, the question "*Is the boy in white left or right of the ball?*" is more common than asking ambiguous questions, for example, where bounding boxes intersect ("*Is the left boy in yellow left or right of the ball?*", see Figure 5).

	PI Input	Permuted caption	Permuted PI words
Probing	$\emptyset$	97.4	1.4
	x, y	96.5	0.3
	$x1, y1, x2, y2$	96.8	1.7
	$x1, y1, x2, y2, d$	97.1	1.2
Pre-training	$\emptyset$	96.8	78.1
	x, y	97.7	79.5
	$x1, y1, x2, y2$	97.7	79.3
	$x1, y1, x2, y2, d$	97.1	79.5

Table 4: Contrastive Evaluation: Recall of the original CMM tasks with random captions (left) and text-image pairs with substituted PI antonyms (right). Upper lines for plain LXMERT and lower lines with our version (PIP, CL; see Section 6).

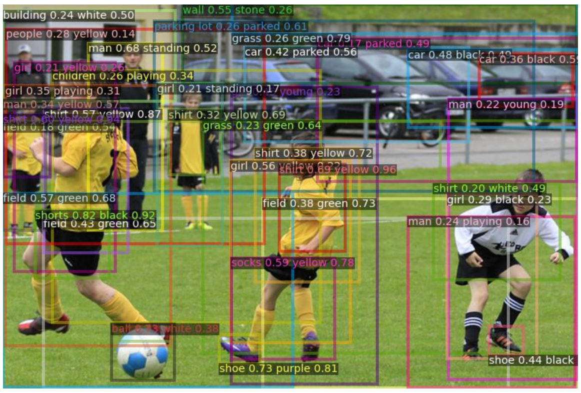


Figure 5: Bounding box predictions for the 36 objects used in LXMERT. Descriptions contain predicted label and attribute with confidence scores.

The PI input $x1, y1, x2, y2, d$ received the best results for the Y and Z subsets. Although the improvements are small, it shows that our new depth feature can help solve the Z task. But also, the improvement on Y can be attributed to the depth input. Due to the graphical perspective, objects at the top correlate with the background and objects at the bottom with the foreground (see Figure 4b). Here, object depth can act as a top/down proxy.

For the downstream evaluation, we need to keep in mind that the underlining object detector is not perfect. Therefore, we face the issue that objects asked for in the questions are not always a part of LXMERT's visual input. Moreover, our contrastive evaluation scheme shows that LXMERT has difficulties to properly matching image and text representation in a multimodal fashion. This can explain the small margin of improvements. The increase of top-1 accuracy is not reflected in the top-5 accuracy.

6 From Probing to Pre-training

In the previous section, we evaluated the role of PI in pre-trained LXMERT. In this section, we use the probing tasks as a part of model pre-training to improve weaknesses that we identified in the previous section. Alongside the established strategies, we add two tasks to learn mutual positions and fine-grained PI details in captions utilizing the CMM task. These strategies are elaborated below.

Positional Information Pre-training (PIP). Currently, all pre-training strategies rely on the visual features (f_j, c_j, a_j) rather than on the PI. Only in a small fraction of the pre-training captions and questions positional keywords are present, as Table 2 shows. Hence, we add a new pre-training strategy which exclusively focuses on PI.

We take the PI head used in Mutual Position Evaluation and add it as a new classification task which is updated during pre-training. We weight PIP by 10, since the initial loss is noticeably lower than the losses of the other strategies. Until now only visual representation of the object features, labels and attributes was part of pre-training. Using PIP, we introduce an explicit unimodal connection between the PI input and the PI output, which was not previously available (see Figure 3).

Contrastive Learning using CMM (CL). During pre-training in classical CMM in 50% of all cases the caption is replaced with another random image description. This is similar to the main pre-training concept of CLIP. Yet, doing so, the model only learns to distinguish dissimilar text and images. There are no small differences in the captions that the model needs to be aware.

In line with Contrastive Evaluation on PI using CMM, we make CMM more complex. In 50% of all captions with PI keywords the word is replaced by its counterpart, so that it has to learn fine-grained PI differences during pre-training. Dissimilar to PIP, this pre-training strategy only affects a small portion of the pre-training samples, since PI keywords are rare. Yet, it operates on both modalities and hence is able to connect both data types. This idea can also be extended to other attributes (such as color, material, shape using VG's Scene Graph).

Results. Using both pre-training strategies, we train new models for all four PI input types. We assess the models using the same three evaluation schemes as the plain LXMERT model before.

Results of Mutual Position Evaluation are shown in Table 3 (bottom lines). We observe an increase in accuracy for all input types. The largest is for the empty input type with an accuracy of 88.2%, indicating the high correlation between feature f_j and position p_j. For the other versions improvements are smaller. In Table 9 in the Appendix, the accuracies for each of the nine classification tasks are displayed. The largest increase can be seen for the empty input type with up to 23.1 percent points for task (1) of the 9 mutual position classification tasks. For classifications based on depth, the best improvements are 9.7 percent points for task (7) and 8.0 percent point for task (9) utilizing $x1, y1, x2, y2, d$. This shows that the presence of depth is useful as expected.

In the original LXMERT version, the probe on Contrastive Evaluation on PI using CMM showed that the model is not able to solve this task successfully. Recall was steadily below 2 percent. Introducing the CL pre-training strategy increases matching accuracy to over 78 percent, as shown in Table 4 (bottom lines). In CMM, we are now able to perform matching between visual and textual representations regarding PI. As a consequence, we successfully force the model to connect both types in a multimodal manner.

Downstream Task Evaluation. The third evaluation is the downstream task and results are shown in Table 5 (bottom lines). In the two former probes, our extended pre-training helped the model to solve these tasks. However, interestingly, this is not the case for GQA evaluation. The best results for the top 1 and subset tasks are obtained by plain LXMERT. Only in the (not official) best 5 accuracy, evaluation our version achieves better results. One reason for this may be that our PIP weight is too high and needs to be tuned in further studies.

We found that PI has much less impact on downstream results as previously thought. Simple object centers are often sufficient. Bounding box data, which add object width, height and area, do not add the desired information that the models can utilize. Adding depth is marginally useful on the Z task, which suggests that this feature is useful.

	PI Input	Top 1	X	Y	Z	P	Top 5
Probing	$\emptyset$	58.1	65.7	62.0	46.4	58.0	85.0
	x, y	**59.4**	**69.6**	62.0	49.6	**60.2**	85.0
	$x1, y1, x2, y2$	59.0	66.2	61.8	49.4	58.9	**85.3**
	$x1, y1, x2, y2, d$	58.6	66.0	**62.4**	**50.0**	58.4	85.1
Pre-training	$\emptyset$	**58.8**	**68.7**	60.4	48.5	**59.0**	85.1
	$x1, y1$	**58.8**	**68.7**	60.4	48.5	**59.0**	85.1
	$x1, y1, x2, y2,$	58.7	67.6	61.5	48.3	58.6	85.4
	$x1, y1, x2, y2, d$	58.7	67.8	**62.0**	**49.1**	**59.0**	**85.8**

Table 5: Evaluation on GQA *testdev*: Model comparison of plain LXMERT models (top lines) and our version with PIP and CL pre-training (bottom lines) for different PI Input types. Evaluation on Top 1 and Top 5 accuracy, moreover on subsets focusing on X, Y, and, Z keywords only and questions that focus on position (P) using zero-shot classification. Underlined numbers indicate the best overall model per column and bold numbers indicate the best model per block and column.

7 Conclusions

Current VL models make use of different PI inputs without evaluating their impact. In our work, we inspect the effect of such PI input types and investigate depth as a new input extension. In the original setting, the model is able to forward the positional information through the whole Transformer layer stack, but it cannot utilize it in the contrastive evaluation and only marginally in the downstream task. Overall, having any type of PI is helpful, though object-center values are often sufficient. However, object features f_j are already good proxies for where objects are located. Because this can be based on spurious correlations, we propose pre-training methods that should make the model rely on PI directly.

We introduced two new pre-training strategies. Firstly, Positional Information Pre-training to ensure that data is passed through the model properly and does not need to rely on feature correlations. This operates on visual component only and increases performance on the corresponding intrinsic evaluation task. Moreover, we introduce Contrastive Learning on PI using CMM. In doing so, we connect PI in the textual and visual modality. As a result, the model is now able to succeed in the contrastive evaluation task. However, these improvements do not affect the downstream performance on GQA.

It is not enough to add different features unchecked, trusting they are properly utilized by the Transformer. In line with BERTology (Rogers et al., 2020; Clark et al., 2019; Tenney et al., 2019), studies are important to understand better what a model is capable. The same is true for pre-training strategies. It is not sufficient to add new pre-training strategies, although they look promising. With our probing experiments, we tried to receive a better understanding of the inner workings of LXMERT. We see the importance to investigate differences between general concepts and impact on a downstream task.

We see two major issues for PI in VL models. Firstly, the pre-training data contains too little fraction of sentences with PI content. Hence, especially the CL pre-training strategy has not enough samples to learn from. Secondly, the used object detector is not very powerful (see predictions in Figure 5). Newer detection models like VinVL (Zhang et al., 2021) might help to have a better image representation, which consequently leverages performance regarding PI context.

Acknowledgements

The authors gratefully acknowledge the computing time granted by the Institute for Distributed Intelligent Systems and provided on the GPU cluster Monacum One at the Bundeswehr University Munich. The work at CUNI was funded by the Czech Science Foundation, grant no. 19-26934X.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002.

Drew A Hudson and Christopher D Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking Positional Encoding in Language Pre-training. In *International Conference on Learning Representations*.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. 2019. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 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.

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. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

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

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

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2020. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yu-An Wang and Yun-Nung Chen. 2020. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849, Online. Association for Computational Linguistics.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 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.

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

A Appendix

A.1 PI Classification Head

The PI head is build up in the same manner as the other visual heads, i.e. `Dense` → `Activation` → `Layer Normalization` → `Dropout` → `Dense`.

A.2 PI Antonyms

For Contrastive Evaluation, we replace some PI keywords with its antonyms.

We substitute *left* with *right*, *above* with *below*, *under* with *over*, *foreground* with *background*, *before* with *behind* and vice versa.

A.3 PI Keywords

In Table 6 we list all PI keywords used in our evaluations.

Dim.	**Keywords**
X	left, right, beside, besides, alongside, side
Y	top, down, above, below, under, beneath, underneath, over, beyond, overhead
Z	behind, front, rear, back, ahead, before, foreground, background, before, forepart, far end, hindquarters

Table 6: Overview of positional keywords regarding dimension.

A.4 Pre-training Weights

In Table 7 we compare pre-training weights from LXMERT paper (Tan and Bansal, 2019) and the repository version (`https://github.com/airsplay/lxmert/`).

Version	**MLM**	**CMM**	**ObjClassif**	**AttrClassif**	**FeatRegr**	**VQA**
Paper	1	1	1	1	1	1
Repository	1	1	$6.\overline{6}$	$6.\overline{6}$	$6.\overline{6}$	1

Table 7: Overview of pre-training weights in publication and GitHub version.

A.5 Text Examples

In Table 8 we provide examples from pre-training and downstream tasks with highlighted keywords.

Dataset	Example	Length
MS COCO	A very clean and well decorated empty bathroom	8
	A panoramic view of a kitchen and all of its appliances.	11
	Surfers waiting for the *right* wave to ride.	8
	Two dogs are laying *down* **next** to each other.	9
	A red stop sign with a Bush bumper sticker **under** the word stop.	13
VG	separate kitchen areas in a home	6
	older red Volkswagen Beetle car	5
	a woman walking *down* the sidewalk	6
	A bag in the woman's **left** hand	7
	stones **under** wood bench	4
GQA	Are there both a television and a chair in the picture?	11
	That car is what color?	5
	On which **side** of the picture is the lamp?	9
	Is the table to the **left** or to the **right** of the appliance in the center?	16
	Is there a bookcase **behind** the yellow flowers?	8

Table 8: Text examples from different datasets with word counts. Italic stands for PI keywords that are wrongly selected and bold words are correctly detected.

PI Input	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
$\emptyset$	65.0	84.1	82.1	89.9	95.6	72.3	77.7	75.3	78.4
x, y	95.1	95.6	96.2	96.1	95.8	74.1	83.3	75.7	84.4
$x1, y1, x, 2, y2$	94.3	95.2	96.8	97.0	96.0	75.0	83.5	75.8	84.6
$x1, y1, x, 2, y2, d$	94.0	95.0	96.6	96.8	96.0	74.9	88.7	76.3	89.1
$\emptyset$	88.1	89.4	92.6	93.5	95.9	74.1	83.9	77.7	84.8
x, y	98.7	98.8	98.3	98.3	96.1	75.9	89.3	78.4	90.4
$x1, y1, x, 2, y2$	98.8	98.9	98.7	99.5	96.3	77.0	89.7	78.9	90.9
$x1, y1, x, 2, y2, d$	98.9	98.9	98.6	99.0	96.3	77.2	98.4	81.0	97.1

Table 9: Average accuracy per classification task (1-9) in Mutual Positional Evaluation for plain LXMERT (top lines) and our version (bottom lines).

A.6 Mutual Positional Evaluation Details

In Table 9 we provide detailed results for all 9 mutual PI tasks. Tasks (1)-(6) relate to X and Y coordinates and tasks (7)-(9) to Z coordinates. The numbering is explained in Section 3.