DOI: 10.1109/TPAMI.2023.3343736

Enhancing Visual Grounding in Vision-Language Pre-Training With Position-Guided Text Prompts

Alex Jinpeng Wang, Pan Zhou, Mike Zheng Shou, *Member, IEEE*, and Shuicheng Yan, *Fellow, IEEE*

***Abstract*—Vision-Language Pre-Training (VLP) has demonstrated remarkable potential in aligning image and text pairs, paving the way for a wide range of cross-modal learning tasks. Nevertheless, we have observed that VLP models often fall short in terms of visual grounding and localization capabilities, which are crucial for many downstream tasks, such as visual reasoning. In response, we introduce a novel Position-guided Text Prompt (*PTP*) paradigm to bolster the visual grounding abilities of cross-modal models trained with VLP. In the VLP phase, *PTP* divides an image into N x N blocks and employs a widely-used object detector to identify objects within each block. *PTP* then reframes the visual grounding task as a fill-in-the-blank problem, encouraging the model to predict objects in given blocks or regress the blocks of a given object, exemplified by filling "*[P]*" or "*[O]*" in a PTP sentence such as "*The block [P] has a [O].*" This strategy enhances the visual grounding capabilities of VLP models, enabling them to better tackle various downstream tasks. Additionally, we integrate the seconda-order relationships between objects to further enhance the visual grounding capabilities of our proposed PTP paradigm. Incorporating *PTP* into several state-of-the-art VLP frameworks leads to consistently significant improvements across representative cross-modal learning model architectures and multiple benchmarks, such as zero-shot Flickr30 k Retrieval (+5.6 in average recall@1) for ViLT baseline, and COCO Captioning (+5.5 in CIDEr) for the state-of-the-art BLIP baseline. Furthermore, *PTP* attains comparable results with object-detector-based methods and a faster inference speed, as it discards its object detector during inference, unlike other approaches.**

***Index Terms*—Fill-in-the-blank, position-guided text prompt, vision-language pre-training, visual grounding.**

I. Introduction

VISION-AND-LANGUAGE pre-training (VLP) models, such as CLIP [40], ALIGN [20], and CoCa [58], have significantly improved cross-modal tasks like visual question answering [4], natural language visual reasoning [46], and image captioning [1], [9]. The success of these models can be attributed to a two-stage learning process: pre-training on large image-caption data to enhance generalization capabilities, followed by fine-tuning on downstream tasks for seamless adaptation. This efficient pre-training and fine-tuning paradigm has established VLP models as a dominant force in the multi-modal research domain, highlighting their potential for further development and continued performance improvements across various cross-modal applications.

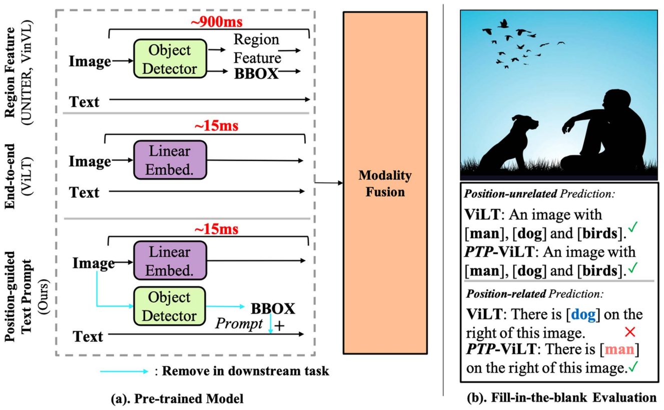

Region Feature (UNITER, VinVL)
~900ms
Image
Object Detector
Region Feature
BBOX
Text
End-to-end (ViLT)
~15ms
Image
Linear Embed.
Text
Position-guided Text Prompt (Ours)
~15ms
Image
Linear Embed.
Object Detector
BBOX
Prompt
+
Text
: Remove in downstream task
Modality Fusion
(a). Pre-trained Model
Position-unrelated Prediction:
ViLT: An image with [man], [dog] and [birds].✓
PTP-ViLT: An image with [man], [dog] and [birds].✓
Position-related Prediction:
ViLT: There is [dog] on the right of this image. ✗
PTP-ViLT: There is [man] on the right of this image.✓
(b). Fill-in-the-blank Evaluation

Fig. 1. Comparison of three VLP learning frameworks and their performance. (a) compares region feature-based VLP (RF-VLP), end-to-end VLP (E2E-VLP), and our position-guided text prompt-based VLP (PTP-VLP). Our PTP-VLP requires only about 15 ms for inference, which is on par with E2E-VLP and significantly faster than RF-VLP. (b) On position-aware questions that are commonly encountered in many downstream tasks, both RF-VLP and PTP-VLP can accurately predict objects with masked text and image input. In contrast, E2E-VLP struggles to pinpoint the position information of the object in the image.

In VLP, visual grounding plays a crucial role in various tasks, as evidenced by previous research [3], [56]. Traditional VLP models [3], [31], [62], depicted at the top of Fig. 1(a), utilize a Faster R-CNN [42] pre-trained on the 1600-class Visual Genome [23] to extract salient region features and bounding boxes. These models then take both the bounding box and object feature as input, allowing them to identify objects within the salient region and determine their locations. However, by using region features as input, the model selectively attends to information within bounding boxes while neglecting contextual data beyond their boundaries [17]. This limitation can result in suboptimal performance on downstream tasks, necessitating the use of additional object detectors to extract objects and consequently causing significantly slower inference speeds.

Manuscript received 4 April 2023; revised 12 November 2023; accepted 7 December 2023. Date of publication 18 December 2023; date of current version 3 April 2024. This work was supported in part by National Research Foun-dation, Singapore under Grant NRF-NRFF13-2021-0008, and in part by Mike Zheng Shou's Start-Up Grant from NUS. The work of Pan Zhou was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. Recommended for acceptance by G. Hua. *(Corresponding author: Mike Zheng Shou.)*

Alex Jinpeng Wang and Mike Zheng Shou are with the Show Lab, National University of Singapore, Singapore 119077 (e-mail: jinpengwang@u.nus.edu; mike.zheng.shou@gmail.com).

Pan Zhou is with the School of Computing and Information Systems, Singapore Management University, Singapore 188065 (e-mail: zhoupan@sea.com).

Shuicheng Yan is with Sea AI Lab, Singapore 138522 (e-mail: yansc@sea.com).

Our code and pre-trained models are available at https://github.com/sail-sg/ptp.

Digital Object Identifier 10.1109/TPAMI.2023.3343736

To get rid of region feature for higher efficiency, recent works [17], [22] (the middle of Fig. 1(a)) adopt raw image as input instead of region features, and train the model with Image Text Matching [10] and Masked Language Modeling [12] loss end-to-end. Despite their faster speed, these models cannot well learn the object positions and also their relations. As demonstrated in Fig. 1(b), a well-trained ViLT model [22] can successfully identify objects in an image. However, it fails to precisely learn object positions. For instance, it incorrectly predicts "*the dog is on the right of this image*." During fine-tuning evaluation, downstream tasks necessitate object position information for a comprehensive understanding of the image. This gap significantly hinders performance on downstream tasks, emphasizing the need for improved object position and relation learning.

In this work, our goal is to address the position learning issue in end-to-end (e2e) models while maintaining fast inference times for downstream tasks. Drawing inspiration from recent prompt learning methods [21], [33], [41], [57], we introduce a novel and effective *Position-guided Text Prompt (PTP)* paradigm (depicted at the bottom of Fig. 1(a)) for VLP. The core insight is that by incorporating position-based co-referential markers into both image and text, visual grounding can be transformed into a fill-in-the-blank problem, significantly simplifying the learning of object information. To establish a connection between language expressions and image, *PTP* comprises two components: (1) block tag generation, which divides the image into $N \times N$ blocks and identifies objects within each block, and (2) text prompt generation, which embeds the query text into a position-based text query template. This innovative approach facilitates more accurate position learning while retaining the efficiency advantages of e2e models.

Integrating position information into the pre-training phase, our *PTP* significantly enhances the visual grounding capabilities of VLP models. We also investigate second-order relations between objects to improve reasoning ability. Importantly, our approach maintains fast inference times, as we do not rely on object detectors for downstream tasks. Experimental results reveal that our method substantially outperforms its counterparts, particularly in the zero-shot setting. Our proposed model, *PTP*-BLIP, demonstrates exceptional performance on the zero-shot image-to-text retrieval Recall@1 task on the COCO dataset, achieving a 6.9% absolute accuracy gain over CoCa [58] while using considerably less training data (4 M versus 3B) and a smaller model size (220 M versus 2.1B). Moreover, *PTP*'s effectiveness extends beyond retrieval, as evidenced by its success in other visual language tasks such as visual grounding and image captioning. These results underline the potential of our position-guided approach for advancing the state of the art in cross-modal learning.

Our contributions can be summarized as follows: 1). We introduce a novel pre-training paradigm for vision-language models, called cross-modal prompt-based pre-training, which explicitly incorporates position information into the prompt. To the best of our knowledge, this represents the first attempt at employing such an approach for pre-training vision-language models. 2). We design and assess multiple configurations of high-quality cross-modal prompts for our proposed model, *PTP*, demonstrating the versatility and adaptability of our approach. 3). We carry out comprehensive experiments using four backbone models, showcasing the effectiveness of *PTP* across a range of vision-language tasks. We expand our approach to encompass data at the billion-level scale and demonstrate its efficacy with a potent Large Language Model, further highlighting its potential to advance the state of the art in cross-modal learning.

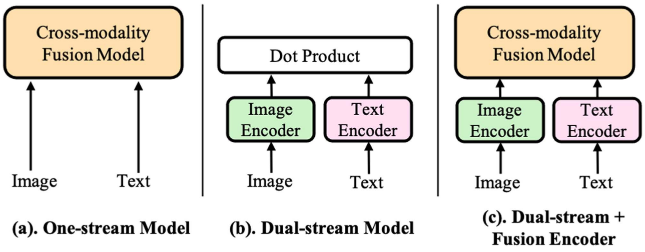


Fig. 2. Three widely-used categories of vision-and-language models. The primary distinction lies in the stage at which cross-modality information fusion occurs. One-stream models perform fusion at an early stage, while dual-stream models fuse information at a late stage. Lastly, the third type of models integrates information at a middle stage, striking a balance between the other two approaches.

This journal paper extends our previous work [50] in several ways: *First*, we propose a novel second-order prompt to further enhance the understanding of object relationships. This innovation leads to improved state-of-the-art results in various downstream tasks, such as visual-text retrieval, visual question answering, and image captioning. *Second*, we expand our evaluation by including more downstream tasks, specifically visual grounding on the RefCOCO [59] and RefCOCO+[59] datasets. Additionally, we explore video question answering on MSVD [55], TVQA [25], and TGIF [19], providing a more comprehensive comparison with VLP-related works. *Third*, we provide an in-depth analysis of the visual grounding ability, enriched visualizations of masked blocks, and a thorough ablation study. These elements together showcase the effectiveness and robustness of our second-order PTP approach in advancing the state of the art in cross-modal learning. *Finally*, we have trained the *PTP* on the extensive DataComp-1B [14] dataset at the billion-level and integrated it with popular Large Language Models, thereby showcasing the versatility and broad applicability of *PTP*.

II. Related Work

A. *Vision-Language Pre-Training Architectures*

Existing VLP models can be roughly grouped into three categories based on their architectures: one-stream models, dual-stream models, and dual-stream with fusion encoder models. We provide an overview of all three architectures:

1) One-stream: (e.g., UNITER [10], ViLT [22]) as shown in Fig. 2(a), operates on a concatenation of image and text inputs. *2) Dual-stream:* (e.g., CLIP [40]) depicted in Fig. 2(b), employs separate and equally expensive transformer encoders for each modality. The two modalities are not concatenated at the input level, with interaction between the pooled image and text vectors occurring at a shallow layer. *3) Dual-stream with*

Fusion Encoder: (e.g., BLIP [27]) illustrated in Fig. 2(c), is a combination of one-stream and dual-stream models that allows for intermediate interaction.

In this work, without loss of generality, we focus on prompting all three types of VLP models due to their prevalence and adaptability to various downstream tasks.

B. Prompt Learning for Computer Vision

Prompt learning was initially designed for probing knowledge in pre-trained language models and adapting them to specific downstream tasks [33], [41]. In recent years, there has been a growing interest in studying prompt tuning for vision tasks, including multi-modal learning and image understanding. The pioneering work Color Prompt [57] introduces color prompts on images and text color descriptions for visual grounding. Most related to our work is Multi-modality Prompt [21], which presents multi-modal prompt tuning for vision-language pre-training models, achieving promising results on various vision-language tasks.

However, these efforts, akin to early NLP research, focus on prompt engineering during the fine-tuning stage, leaving the pre-training phase unaffected. In contrast, the goal of using prompt design in our work is to equip the model with the ability to understand semantic concepts at a finer level during the pre-training stage, laying a stronger foundation for downstream tasks.

C. Learning Position Information in VLP

The grounding ability has proven to be essential for multiple cross-modal tasks [29], [34]. To introduce this ability into VLP models, bottom-up and top-down [3] approaches and their follow-up works [10], [31] concatenate region features and bounding box vectors together as input signals. However, object extraction is time-consuming during inference for downstream tasks. Recently, some works [29], [34], [61] propose training VLP models with additional object localization loss or word patch alignment loss. These methods, however, are difficult to extend as they are specifically designed for particular frameworks. In contrast, we aim to propose a general framework for learning position information. To this end, we introduce a simple text prompt that can be easily integrated into existing frameworks, providing a versatile solution for capturing position information in VLP.

III. Position-Guided Text Prompt

In this section, we first provide a detailed explanation of our proposed Position-guided Text Prompt paradigm (*PTP* for short). Following this, we demonstrate how to incorporate it into existing vision-language pre-training (VLP) frameworks to enhance their visual grounding capabilities. We use the classical and popular models VILT [22], CLIP [40], and BLIP [27] as examples to showcase the integration.

A. PTP Paradigm

To enhance the visual grounding ability of cross-modal models trained using VLP, we propose a novel and effective Position-guided Text Prompt (*PTP*) that helps a cross-modal model perceive objects and align them with the relevant text. *PTP* differs from conventional vision-language alignment methods, such as [3], [10], [31], [62], which concatenate object features and bounding boxes as input to learn the alignment between objects and relevant text. This alternative approach offers several advantages, as discussed in Section III-B. As illustrated in Fig. 3, *PTP* consists of two steps: 1) block tag generation, which divides an input image into several blocks and identifies objects in each block; and 2) text prompt generation, which reformulates the visual grounding task into a fill-in-the-blank problem based on the object position information from step 1). By solving the fill-in-the-blank problem in *PTP*, one can easily integrate PTP into a VLP model. We provide details on these steps below.

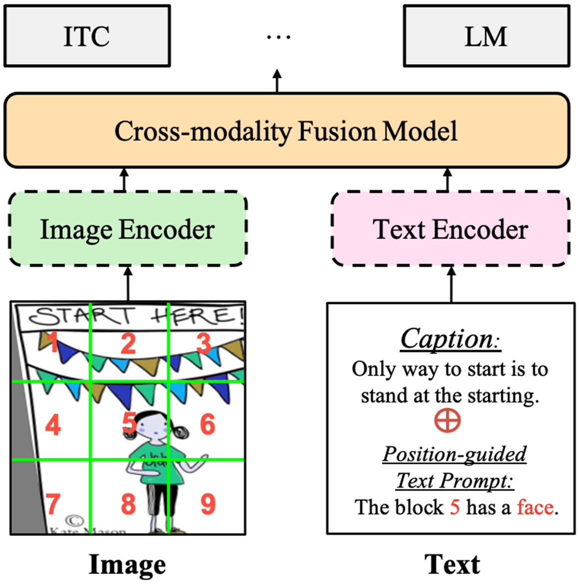


Fig. 3. Overview of our pipeline. Our *PTP* can be seamlessly integrated with any pre-training framework, including one-stream, dual-stream, and dual-stream with fusion encoder models, as well as most pre-training objectives. The dashed line in the figure indicates that certain models may not be present. For downstream tasks, we remove the text prompt and evaluate the model using standard procedures, ensuring that the integration of *PTP* does not interfere with task-specific evaluation metrics.

1) Block Tag Generation: As shown in Fig. 3, for each image-text pair in the training phase, we evenly divide the input image into $N \times N$ blocks. Then we identify the object in each block using one of the following two methods:

(1) Object Detector: First, we adopt a strong Faster-RCNN [42] used in VinVL [62] to extract all objects for each image. This Faster-RCNN version is based on ResNeXt152 and trained on the 1600-class Visual Genome [23]. Then we select the top-K objects, denoted by $\mathcal{O} = o_{i}{}_{i=1}^{K}$, with the highest prediction confidence, where $o_i = (z_i, q_i)$ denotes an object with a 4-dimensional region position vector z and object category q. For each block, we select the objects whose region centers are in that block. Finally, the block tag for this block is q of the selected objects. In this work, we generate object tags using an object detector by default.

(2) CLIP Model: As an alternative to using a heavy object detector, recent works [64], [65] have attempted to generate region supervision based on CLIP [40] due to its efficiency and

effectiveness. Inspired by these works, *PTP* can also generate block-wise object supervision using the CLIP (ViT-B) model.[1] First, we extract M (3000 by default) keywords/phrases that are most frequent in the text corpus.[2] These keywords/phrases form our vocabulary V. Then, we extract the text feature $e_i, i \in [1, \ldots, M]$ of all these M text embedding using the CLIP text encoder.

Furthermore, we take the image embedding h from each block and compute the similarity across every text feature. The keyword/phrase with the highest similarity score is selected as the object tag for this particular block. Formally, the index of the object tag per block is computed as:

$$I = \text{argmax} y \in [1, \ldots, M] \left(\frac{\exp(h^T e_y)}{\sum w \in V \exp(h^T e_w)} \right), \quad (1)$$

where h is the visual feature embedding of the selected block. Compared to the object detector, the CLIP model has two advantages. First, instead of pre-defined object categories, more diverse object tags are produced. Second, generating block tags is much faster with the CLIP model than with an object detector, e.g., it is 40× faster than the Faster-RCNN (ResNeXt152) model.

2) First-Order Text Prompt Generation: For the input image of each training pair, Section III-A1 already generate the object tags and positions which allows us to design a simple text prompt as follows:

$$\text{"}The\ block\ [P]\ has\ a\ [O].\text{"}$$

Here, $P \in 1, \ldots, N^2$ denotes the index of the selected block, and O denotes the object tag generated for that block. If there are multiple objects in a block, we randomly select one object tag for each prompt. This way, each prompt contains object position and text, which can help the model better understand the relationships between objects and text. More prompt design is explored in Section IV-D.

3) Second-Order Text Prompt Generation: The first-order relations primarily focus on identifying the position of individual objects within an image. While this approach provides a foundation for understanding object placement, it falls short in capturing the more complex, challenging relationships that exist between objects. For instance, in the example shown in Fig. 3, a *flag* is positioned on top of a *girl*, illustrating a higher-order relationship that a first-order prompt would not capture.

To address this limitation and further enhance the learning process, we propose exploring second-order relations by incorporating more sophisticated prompts. As below:

$$\text{"}The\ block\ [P]\ has\ a\ [O]\ and\ a\ [O_2]\ in\ [R]\ of\ this\ block.\text{"}$$

In this format, O_2 is another randomly selected object, and R represents the relative position, with $R \in \{top, bottom, left, right\}$. By employing this structure, the model is challenged to not only recognize individual objects but also understand their interrelationships and relative positions within the image.

We have named this method *PTP2R*, with *2R* representing second-order relations. This approach requires the model to conduct reasoning over all objects in the image, allowing it to develop a more comprehensive understanding of the visual scene. By delving into these higher-order relationships, we aim to improve the model's ability to recognize and interpret complex object interactions, ultimately enhancing its performance in VLP tasks.

B. *Pre-Training With* PTP

In this work, we integrate our *PTP* into mainstream VLP frameworks, leading to *PTP*-ViLT [22], *PTP*-CLIP [40] and *PTP*-BLIP [27]. Following receipt of the *PTP*, we have two options for training these models:

Integrate Into Existing Tasks: The simplest method for using text prompt is to change the text input. As shown in Fig. 3, the prompted text and original caption were simply padded together. Formally, the input caption x of our method is represented as:

$$x = [w, q], \quad (2)$$

where w is text and q is our generated text prompt. Then we train the VLP models end-to-end with conventional objectives. Following [22], [27], [40], we employ Language Modeling (LM) loss, Image-text Matching (ITM), and Image-text Contrastive (ITC) loss for our *PTP*-BLIP; we use ITM and Masked Language Modeling (MLM) loss to train our *PTP*-ViLT; we only use ITC loss to train our *PTP*-CLIP. We use this method as default for all experiments because of its good performance.

As a New Pretext Task: Alternatively, we explore the position prediction as an additional language modeling task. Formally, if D is the pretraining data and $y_1, \ldots, y_T$ is a training token sequence of our generated text prompt q, then at the timestep t, we devise our model to predict a probability distribution $p(t) = p(*|y_1, \ldots, y_{t-1})$. Then we regressively try to maximize the probability of being the correct token. The object prediction loss is computed as:

$$\mathcal{L}_{\text{PTP}}(\theta) = -\mathbb{E}_{\mathbf{y} \sim D} \left[\sum\nolimits_{t=1}^{T} \log P_\theta \left(\mathbf{y}_t \mid \mathbf{y}_{<t} \right) \right], \quad (3)$$

where θ is the trainable parameters of the model. In this way, the model is asked to predict *which block P has objects* and *what object O is in this block.*

Discussion: Notably, our method does not need to modify the base network and can be applied to any VLP models without bells and whistles. The model is designed to learn position information from raw-pixel image. Note that only during the pre-training stage, we would require the object's position information; yet on downstream tasks, we evaluate model in normal end-to-end ways without object information to get rid of the heavy object feature extraction.

IV. Experiments

In this section, we empirically evaluate *PTP* on multiple downstream tasks and present a comprehensive study.

[1] [Online]. Available: https://huggingface.co/openai/clip-vit-base-patch16
[2] NLTK: ([Online]. Available: https://github.com/nltk/nltk)

TABLE I
STATISTICS OF THE PRE-TRAINING DATASETS

	Dataset	# Images	# Captions	# BBox
Small	COCO	0.11M	0.55M	-
	Visual Genome	0.10M	-	-
	SBU	0.86M	0.86M	-
	CC-3M	2.8M	2.8M	2.69M
Base	4M	4.0M	5.1M	2.69M
	CC-12M	10.2M	10.2M	7M
Large	DataComp-1B	1.17B	1.17B	-
	MMC4	324M	324M	-
	LAION400M	375.3M	375.3M	-

A. Experimental Settings

We first describe the pre-training experimental conditions, including the datasets, training configurations, evaluation procedures, and baseline models used in our studies.

Statistics of the Pre-Training Datasets: In this work, we explore beginning from both 4 M (Small) and 14 M (Base) setting. As in earlier studies [31], [62], we begin by using a 4 M setup made up of four popular pre-training datasets (COCO [32], VG [23], SBU [37] and CC3M [44]). The 14 M setting is a combination of 4 M setting and CC-12 M. Following recent work [27], we also explore 14 M setting, which includes additional CC12M [8] (actually only 10 M image urls available) dataset besides 4 M datasets. We report the data statistics in Table I. As the URLs for the CC3M and CC12 M datasets are derived from the Internet, and given that a portion of them are no longer valid, we have downloaded a total of 2.8 million data instances for CC3M and 10.2 million data instances for CC12 M, respectively. It should be noted that the BLIP baseline employs 3 million data instances for CC3M, slightly more than our model. In terms of the number of images that contain bounding boxes, there are 2.69 million such images for CC3M and 7 million for CC12 M. These bounding boxs are used in our *PTP*. For quick evaluation, we pre-train the BLIP model for 50 K steps rather than the 200 K in [22], [61].

Additionally, to assess the effectiveness of our method at a Large scale, we conducted experiments on extensive datasets, encompassing billions of data points, which include DataComp-1B [14], MMC4 [67], and LAION400M [43].

Training Settings: Our models are implemented in PyTorch [38] and pre-trained on 8 NVIDIA A100 GPUs. To ensure a fair comparison, we adopt the optimizer and training hyperparameters from the original implementation in the baseline works. Additionally, we investigate the use of RandAugment [11] for data augmentation and utilize all of the original policies, except for color inversion, as color information is crucial for our tasks. Furthermore, we apply affine transformations to augment the bounding boxes in a manner similar to that used for image rotation. We take random image crops of 224 × 224 during pre-training resolution, and increase the image resolution to 384 × 384 for downstream task finetuning.

Hyper-Parameters for Downstream Tasks: Table III. The final decoder outputs of the encoder-decoder model BLIP can be utilized for both multimodal understanding and generation. Therefore, we evaluate its performance on popular vision-language benchmarks, employing the same setup as introduced in the BLIP paper [27]. Specifically, we use the AdamW optimizer for all tasks and train the retrieval task for only 6 epochs to increase efficiency. We believe that increasing the number of epochs would yield better results.

In the case of the ViLT baseline, we primarily focus on three tasks: vision-question answering, image-text retrieval, and natural language visual reasoning. The hyper-parameters for ViLT on these downstream tasks are reported in Table IV. Finally, for the CLIP baseline, we use the same hyper-parameter settings as those employed in BLIP.

Baselines: We evaluate three variants of pre-training frameworks, including one-stream ViLT [22], dual-encoder CLIP [40], and fusion-encoder BLIP [27], for their superior performance. For fair comparisons, we adopt the ViT-B/16 [13] as base vision encoder and use same dataset.

B. Main Results

In this section, we integrated our *PTP* into existing networks and compare to existing VLP methods on a wide range of vision-language downstream tasks. Include five image-text tasks and two video-text tasks.

Image Captioning: This task asks the model to describe the input image. We consider two datasets for image captioning: No-Caps [1] and COCO Captioning [32], [48], both evaluated using the model finetuned on COCO with the LM loss. Similar to BLIP, we start each caption with the phrase “a picture of,” which yields marginally better results. Since image captioning needs a decoding head, dual stream CLIP and one-stream ViLT architectures cannot test this task directly due to missing decoding head. We do not pre-train with COCO to avoid information leakage. For No-Caps dataset, following BLIP, we adopts a zero-shot setting.

As shown in Table II, related works utilizing a comparable quantity of pre-training data perform significantly worse than *PTP*-BLIP. The results of our method are closed to the VinVL [62] with fewer training samples and smaller image. Finally, with 14 M setting, our method leads to close result with LEMON, which trained on billions data and requires two times higher resolution image.

Image-Text Retrieval: We evaluate *PTP* for both image-to-text retrieval (TR) and text-to-image retrieval (IR) on COCO and Flickr30 K benchmarks. For *PTP*-BLIP, following original implementation, we adopt an additional re-ranking strategy. Specifically, we first select k candidates based on the image-text feature similarity, and then rerank the selected candidates based on their pairwise ITM scores. We set $k = 256$ for COCO and $k = 128$ for Flickr30 K.

We first report zero-shot retrieval result on both image-to-text and text-to-image setting in Table V. Mainstream methods in VLP, e.g., BUTD [3], OSCAR [31] and UNITER [10], often use object detector, e.g., Faster-RCNN that is also pretrained on Visual Genome. Moreover, the compared methods in Table 1∼4 mostly use object detector. E.g., we use Faster-RCNN adopted by VinVL, but our model (220 M) has better performance

TABLE II
Comparison With State-of-the-Art Image Captioning Methods on NoCaps and COCO Caption

Method	#Images	Param.	NoCaps validation								COCO Caption	
			in-domain		near-domain		out-domain		Overall		Karpathy test	
			CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	B@4	CIDEr
OSCAR [31]	4M	155M	79.6	12.3	66.1	11.5	45.3	9.7	80.9	11.3	37.4	127.8
VinVL‡ [62]	5.7M	347M	103.1	14.2	96.1	13.8	88.3	12.1	95.5	13.5	38.2	**129.3**
BLIP † [27]	4M	220M	106.5	14.4	99.3	13.6	95.6	13.0	98.8	14.2	37.0	122.6
PTP-BLIP	4M	220M	108.3$_{+1.8}$	14.9$_{+0.5}$	105.0$_{+5.7}$	14.2$_{+0.6}$	105.6$_{+10.0}$	14.2$_{+1.2}$	106.0$_{+8.3}$	14.7$_{+0.5}$	38.6$_{+1.6}$	128.9$_{+6.3}$
PTP2R-BLIP	4M	220M	**109.4**$_{+2.9}$	**14.9**$_{+0.5}$	**105.2**$_{+5.9}$	**14.4**$_{+0.8}$	**105.6**$_{+10.0}$	**14.3**$_{+1.3}$	**106.6**$_{+8.9}$	**14.9**$_{+0.7}$	**38.8**$_{+1.8}$	129.1$_{+6.5}$
Enc-Dec [8]	15M	–	92.6	12.5	88.3	12.1	94.5	11.9	90.2	12.1	-	110.9
BLIP [27]	14M	220M	111.3	15.1	104.5	14.4	102.4	13.7	105.1	14.4	38.6	129.7
PTP-BLIP	14M	220M	112.8$_{+1.5}$	15.2$_{+0.1}$	107.3$_{+2.8}$	14.9$_{+0.5}$	108.1$_{+6.7}$	14.3$_{+0.6}$	106.3$_{+0.8}$	14.7$_{+0.3}$	40.1$_{+1.5}$	135.0$_{+5.3}$
PTP2R-BLIP	14M	220M	**113.1**$_{+1.8}$	**15.2**$_{+0.1}$	**107.5**$_{+3.0}$	**14.9**$_{+0.5}$	**108.6**$_{+7.2}$	**14.4**$_{+0.7}$	**106.5**$_{+1.0}$	**14.8**$_{+0.4}$	**40.3**$_{+1.7}$	**135.2**$_{+5.5}$
SimVLM$_{\mathrm{H}}$ [54]	1.8B	1.2B	113.7	-	110.9	-	115.2	-	112.2	-	40.6	143.3
LEMON$_{\mathrm{H}}$‡ [16]	200M	675M	118.0	15.4	116.3	15.1	120.2	14.5	117.3	15.0	42.6	145.5

C: CIDEr, S: SPICE, B@4: BLEU@4. Notice that VinVL‡ and LEMON‡ require high resolution (800×1333) input images.

TABLE III
Hyper-Parameters for BLIP Baseline

Task	VQA	Retrieval	NLVR2	Captioning
Optimizer	AdamW with Weight Decay			
Gradient clip	1.0			
LR decay schedule	Cosine Schedule Decaying to Zero			
Weight decay rate	0.05			
RandAugment	2,5	2,5	2,5	2,5
Train epochs	10	6	5	5
Train batch size	64	24	128	16
LR	2e-5	1e-5	3e-5	1e-5

TABLE IV
Hyper-Parameters for ViLT Baseline

Task	VQA	Retrieval		NLVR2
Dataset	VQAV2	COCO	F30K	NLVR2
Optimizer	AdamW with Weight Decay			
Gradient clip	1.0			
LR decay schedule	Cosine Schedule Decaying to Zero			
RandAugment	2,9			
Weight decay rate	0.05			
Train epochs	10	10	5	10
Train batch size	256	256	256	128
LR	1e-4	3e-4	1e-4	1e-4
Warm-up steps	1500	2500	1000	500

than VinVL (347 M). So the comparison is fair. We find *PTP* significantly improves baselines on all metrics. For example, for ViLT [22] baseline, *PTP* leads to 13.8 % absolute improvement (from 41.3 % to 55.1 %) over Recall@1 of image to text retrieval on MSCOCO. In addition, our *PTP*-BLIP even outperforms CoCa [58] on most recalls of MSCOCO with much less data.

A summary comparison of the fine-tuned settings between different models is presented in Table VI. It is observed that: (1) *PTP* outperforms the BLIP and ViLT baselines by a large margin in both datasets. For instance, *PTP*-ViLT achieves an impressive 5.3% improvement on R@1 of TR in MSCOCO. (2) With the strong BLIP baseline, *PTP*-BLIP achieves state-of-the-art performance at the same scale. Notably, the training cost of *PTP* remains the same as that of the BLIP baseline, since we train *PTP* with the same settings as the baseline without increasing the maximum input text token. Moreover, we can even reduce the gap between the 4 M setting and ALBEF [28] (14 M setting) with a similar dual stream with fusion encoder architecture.

From all these results above, we point out UNITER [10], OSCAR [31], VinVL [62], ImageBERT [39] all use faster-rcnn as we used. However, our *PTP* leads to much better results than these related works. Besides, we only use object detector in pretraining stage. This indicates *object detector is not the secret for success and how to leverage the position information is essential important for VLP models.*

Visual Question Answering: In the context of visual question answering, VQA [4] requires a model to predict an answer based on an image and a corresponding question. For *PTP*-ViLT, we approach VQA as a multi-answer classification task. On the other hand, for *PTP*-BLIP, we follow the approach used in [27], [28] and consider VQA as an answer generation task to facilitate open-vocabulary VQA for improved performance.

The performance results are reported in Table VII. Our proposed model, *PTP*, shows a significant improvement over the ViLT baseline, with a gain of 1.8% on both splits. Furthermore, with the 14 M setting, *PTP*-BLIP outperforms SimVLM [54], which utilizes a ViT-Large based vision backbone and 1.8 billion training samples.

Visual Reasoning: Natural Language Visual Reasoning (NLVR^2) [46] task is a binary classification task given triplets of two images and a question in natural language. This task relies on position information heavily. As shown in Table VII, SimVLM [54] is outperformed by *PTP*-BLIP, which has a reasonable model size and was pretrained on fewer instances. Meanwhile, our method is also closed to VinVL$_{large}$ model that adopt larger model and use object feature from strong object detector instead of raw-pixel image as input.

Visual Grounding: We follow the approach used in ViL-BERT [35] to evaluate our model's visual grounding capabilities in the Referring Expression task, which involves using text to locate image regions. Table IX presents the results obtained for the 4 M setting, which demonstrate the strong grounding ability of our proposed PTP models. Notably, we observe that *PTP* outperforms several related works that rely on object features extracted using the heavy Faster R-CNN architecture. Moreover, *PTP* achieves significant improvements over BLIP.

Video-Text Retrieval: In this experiment, we test the generalization ability of our method to video-language tasks. Specifically, we perform zero-shot transfer to text-to-video retrieval in Table VIII, where we directly evaluate the models trained

TABLE V
RESULTS OF ZERO-SHOT IMAGE-TEXT RETRIEVAL ON FLICKR30 K AND MSCOCO DATASETS

Method	#Images	Param.	MSCOCO (5K test set)							Flickr30K (1K test set)						
			Image → Text			Text → Image				Image → Text			Text → Image			
			R@1	R@5	R@10	R@1	R@5	R@10	Avg	R@1	R@5	R@10	R@1	R@5	R@10	Avg
Unicoder-VL [26]	4M	170M	–	–	–	–	–	–		64.3	85.8	92.3	48.4	76.0	85.2	75.3
ImageBERT [39]	4M	170M	44.0	71.2	80.4	32.3	59.0	70.2	59.5	70.7	90.2	94.0	54.3	79.6	87.5	79.4
ViLT [22]	4M	87M	41.3	79.9	87.9	37.3	67.4	79.0	65.5	69.7	91.0	96.0	53.4	80.7	88.8	79.9
PTP-ViLT	4M	87M	55.1	82.3	89.1	43.5	70.2	81.2	$70.2_{+4.7}$	74.5	93.7	96.5	60.3	85.5	90.4	$83.5_{+3.6}$
PTP2R-ViLT	4M	87M	55.8	82.7	89.6	44.2	71.3	81.8	$70.9_{+5.4}$	75.3	94.1	96.8	60.6	85.8	90.5	$83.9_{+4.0}$
BLIP † [27]	4M	220M	57.4	81.1	88.7	41.4	66.0	75.3	68.3	76.0	92.8	96.1	58.4	80.0	86.7	81.7
PTP-BLIP	4M	220M	72.3	91.8	95.7	49.5	75.9	84.2	$77.3_{+9.0}$	86.4	97.6	98.9	67.0	87.6	92.6	$88.4_{+6.7}$
PTP2R-BLIP	4M	220M	**72.9**	**92.3**	**96.1**	**49.9**	**76.2**	**84.4**	**77.5**$_{+9.2}$	**86.9**	**97.8**	**99.0**	**67.3**	**87.8**	**92.7**	**88.6**$_{+6.9}$
BLIP † [27]	14M	220M	65.5	86.4	92.3	48.4	73.3	83.5	74.9	83.3	95.8	98.0	70.4	88.3	93.1	88.2
PTP-BLIP	14M	220M	73.2	92.4	96.1	53.6	79.2	87.1	$78.6_{+3.7}$	87.1	98.4	99.3	73.1	91.0	94.8	$90.3_{+2.1}$
PTP2R-BLIP	14M	220M	73.6	92.6	96.5	53.5	79.0	87.1	$78.8_{+3.9}$	87.4	98.5	99.3	73.2	91.1	94.8	$90.4_{+2.2}$
CLIP [40]	300M	173M	58.4	81.5	88.1	37.8	62.4	72.2	66.7	88.0	98.7	99.4	68.7	90.6	95.2	90.1
ALIGN [20]	1.8B	820M	58.6	83.0	89.7	45.6	69.8	78.6	70.9	88.6	98.7	99.7	75.7	93.8	96.8	92.2
FILIP [56]	340M	787M	61.3	84.3	90.4	45.9	70.6	79.3	72.0	89.8	99.2	99.8	75.0	93.4	96.3	92.3
Flamingo [2]	2.1B	80B	65.9	87.3	92.9	48.0	73.3	82.1	74.9	89.3	98.8	99.7	79.5	95.3	97.9	93.4
CoCa [32]	3B	2.1B	66.3	86.2	91.8	51.2	74.2	82.0	75.3	92.5	99.5	99.9	80.4	95.7	97.7	94.3

The methods that utilize significantly larger models or train on larger corpora have been grayed out. The symbol † denotes models that were implemented and trained on the same dataset, as the original datasets were either inaccessible or not trained on these splits. The Avg metric represents the mean of all image-to-text and text-to-image recalls.

TABLE VI
FINETUNING RESULTS OF IMAGE-TO-TEXT RETRIEVAL AND TEXT-TO-IMAGE RETRIEVAL ON COCO AND FLICKR30K

Method	#Images	Param.	MSCOCO (5K test set)							Flickr30K (1K test set)						
			Image → Text			Text → Image				Image → Text			Text → Image			
			R@1	R@5	R@10	R@1	R@5	R@10	Avg	R@1	R@5	R@10	R@1	R@5	R@10	Avg
UNITER [10]	4M	155M	65.7	88.6	93.8	52.9	79.9	88.0	78.2	87.3	98.0	99.2	75.6	94.1	96.8	91.8
OSCAR [31]	4M	155M	70.0	91.1	95.5	54.0	80.8	88.5	–	–	–	–	–	–	–	
VinVL [62]	4M	157M	74.6	92.6	96.3	58.1	83.2	90.1	82.5	–	–	–	–	–	–	–
ViLT [22]	4M	87M	61.8	86.2	92.6	41.3	72.0	82.5	72.7	81.4	95.6	97.6	61.9	86.8	92.8	86.0
PTP-ViLT	4M	87M	67.1	90.5	94.3	45.3	79.1	88.4	$77.5_{+4.8}$	85.2	96.9	98.5	68.8	91.4	95.3	$89.4_{+3.4}$
PTP2R-ViLT	4M	87M	68.3	91.2	94.7	45.6	80.6	89.2	$78.3_{+5.5}$	85.7	97.0	98.2	69.3	91.7	95.6	$89.6_{+3.6}$
BLIP † [27]	4M	220M	75.2	93.3	96.3	57.4	82.1	89.5	82.3	94.0	99.1	99.7	82.5	96.4	98.2	95.0
PTP-BLIP	4M	220M	83.7	97.0	98.7	68.1	89.4	94.2	$88.5_{+6.2}$	96.1	99.8	100.0	84.2	96.6	98.6	$95.9_{+0.9}$
PTP2R-BLIP	4M	220M	**84.1**	**97.2**	**98.8**	**69.2**	**89.9**	**94.5**	**89.0**$_{+6.7}$	**96.3**	**99.9**	**100.0**	**84.4**	**96.8**	**98.9**	$96.1_{+1.1}$
ALBEF [28]	14M	210M	77.6	94.3	97.2	60.7	84.3	90.5	84.1	95.9	99.8	100.0	85.6	97.5	98.9	96.3
BLIP [27]	14M	220M	80.6	95.2	97.6	63.1	85.3	91.1	85.5	96.6	99.8	100.0	87.2	97.5	98.8	96.7
PTP-BLIP	14M	220M	84.2	97.3	98.8	68.8	89.5	94.2	$88.8_{+3.3}$	97.0	99.9	100.0	87.7	98.2	99.3	$97.0_{+0.3}$
PTP2R-BLIP	14M	220M	**84.6**	**97.5**	**98.8**	**69.5**	**90.1**	**94.8**	**89.3**$_{+3.8}$	**97.1**	**99.9**	**100.0**	**87.9**	**98.3**	**99.4**	**97.1**$_{+0.4}$
ALIGN [20]	1.8B	820M	77.0	93.5	96.9	59.9	83.3	89.8	83.4	95.3	99.8	100.0	84.9	97.4	98.6	96.0
FILIP [56]	340M	787M	78.9	94.4	97.4	61.2	84.3	90.6	84.5	96.6	100.0	100.0	87.1	97.7	99.1	96.8
Florence [60]	900M	893M	81.8	95.2	–	63.2	85.7	–	–	97.2	99.9	–	87.9	98.1	–	–

Notice that UNITER [10], OSCAR [31] and VinVL [62] all use bounding box and object feature.

on COCO-retrieval. To process video input, we simply sample 8 frames uniformly from each video and average the frame features into a single sequence.

Our method outperforms OA-Trans [52], which is a retrieval-focused method, demonstrating the generality capability of *PTP*. Also, note that this simple approach not well explored temporal information.

Video Question Answering: We report the video question answering results in Table XI. Following All-in-one [49], we explore three widely used benchmarks: MSVD-QA [55], TVQA [25] and TGIF [19]. TGIF FrameQA and MSVD-AQ are open-ended VQA tasks and TVQA is a multiple-choice VQA task. Similar to video-text retrieval, we sample 8 frames for each video. We observe that *PTP*-BLIP performs well in both multiple-choice and open-ended settings. For example, out *PTP2R*-BLIP outperform All-in-one [49] by 4.3% on MSVD-QA and 8.3% on TGIF-FrameQA.

C. Scale-Up Experiments

1) Data Scale Up: In this section, we expand our research methodology to encompass the extensive DataComp-1B dataset [14], which is recognized for its exceptional data quality when compared to the LAION dataset [43]. The DataComp-1B dataset originally comprised 1.4 billion samples; however, it is worth noting that certain URLs have since become unavailable, resulting in the download of 1.17 billion data samples.

Given the substantial computational resources demanded by the BLIP and VILT architectures, our primary evaluation is centered around the CLIP (ViT B-16) model, chosen for its efficiency in training. Leveraging the inherent simplicity of PTP, which obviates the need for intricate hyperparameter selection and operates seamlessly at the data level, we enable a comprehensive comparison at this considerable scale. Moreover, in alignment with the Datacomp evaluation framework, we subject

TABLE VII
Comparison With State-of-the-Art Methods on VQA and NLVR^2: It is Worth Noting That VinVL [62] Employs a Larger Vision Backbone and Object Features Extracted Using Faster R-CNN

Method	**#Images**	**Para.**	**VQA**		**NLVR^2**	
			test-dev	test-std	dev	test-P
UNITER [10]	4M	155M	72.70	72.91	77.18	77.85
OSCAR [31]	4M	155M	73.16	73.44	78.07	78.36
UNIMO [30]	5.6M	307M	75.06	75.27	-	-
VinVL$_L$ [62]	5.6M	347M	**76.52**	**76.60**	**82.67**	**83.98**
ViLT [22]	4M	87M	70.33	-	74.41	74.57
PTP-ViLT	4M	87M	72.13	74.36	76.52	77.83
PTP2R-ViLT	4M	87M	73.44	76.16	77.31	78.50
BLIP † [27]	4M	220M	73.92	74.13	77.52	77.63
PTP-BLIP	4M	220M	75.47	75.88	80.73	81.24
PTP2R-BLIP	4M	220M	75.81	76.03	81.22	81.40
ALBEF [28]	14M	210M	75.84	76.04	82.55	83.14
BLIP [27]	14M	220M	77.54	77.62	82.67	82.30
PTP-BLIP	14M	220M	78.44	78.33	84.55	83.17
PTP2R-BLIP	14M	220M	**78.85**	**78.66**	**84.92**	**83.41**
SimVLM [54]	1.8B	1.2B	77.87	78.14	81.72	81.77
GIT [51]	0.8B	0.7B	-	78.81	-	-
Beit-3 [53]	35M+	1.9B	84.19	84.03	91.51	92.58

Additionally, ALBEF [28] performs an extra pre-training step for NLVR2, while BeIT-3 [53] uses an additional 160GB text corpus.

TABLE VIII
Comparisons With Related Works for Zero-Shot Text-to-**Video** Retrieval on the 1 k Test Split of the MSRVTT

Method	**R1↑**	**R5↑**	**R10↑**	**MdR↓**
ActBERT [66]	8.6	23.4	33.1	36.0
MIL-NCE [36]	9.9	24.0	32.4	29.5
Frozen-in-time [6]	18.7	39.5	51.6	10.0
OA-Trans [52]	23.4	47.5	55.6	8.0
ViLT† [22]	22.6	46.9	53.2	8.0
PTP-ViLT	27.9	52.5	56.3	7.0
PTP2R-ViLT	**28.4**	**53.1**	**56.6**	**7.0**

TABLE IX
Comparison With Related Works on Visual Grounding: It is Worth Noting That These Related Methods Were Pre-Trained on Object Feature

Method	**Ref-COCO**			**Ref-COCO+**		
	val	testA	testB	val	testA	testB
VL-BERT$_L$ [45]	-	-	-	72.59	78.57	62.30
ViLBERT [35]	-	-	-	72.34	78.52	62.61
UNITER [10]	81.41	87.04	74.17	75.90	81.45	66.70
VILLA [15]	**82.39**	**87.48**	**74.84**	76.17	81.54	66.84
BLIP† [27]	77.45	83.32	71.33	74.13	79.37	64.41
PTP-BLIP	79.95	85.33	72.46	74.49	80.13	66.34
PTP2R-BLIP	81.83	86.44	74.30	**76.65**	**82.14**	**67.38**

the model to a comprehensive array of image classification benchmarks, employing a zero-shot evaluation methodology.

Recognizing the significant time investment associated with deploying an object detector model, we expedite the training process by utilizing data generated by the CLIP model, as described in the methods section. We summarize the comparative performance between the original CLIP model and PTP-CLIP in Table XII. Notably, upon closer scrutiny of the enhanced version, we find that the experimental results remain promising even when dealing with data at the billion-level scale.

2) Language Model Scale up: In this experiment, we varying the language model size from BERT-base(200 M) [12] to LLAMA(7B) model [47] and analyze the impact of Large Language Models (LLMs) in multi-modality learning. Specifically, we implement our method within the open-flamingo architecture [5]. Staying true to the original implementation, we integrate our approach with OPT1.7B [63] and LLaMA-7B [47] as language models, while utilizing the Open-CLIP [18] ViT-L model as the vision encoder. The total parameter is 3B and 9B.

It's noteworthy that due to unforeseen issues with some image URLs from LAION400M [43] and MMC4 [67] datasets, preventing access for reevaluation, the sample size is marginally smaller than open-flamingo and we report results with our implementation. Given the document nature of MMC4, we exclusively introduce text prompts.

The model undergoes rigorous training over a span of 6.3 days for 3B model and 17 days for 9B models, leveraging the computational power of 64 Nvidia V100 GPUs. The comprehensive results are presented in Table X. Our focus primarily centers around *few-shot* tasks, providing a nuanced understanding of the model's representation capabilities, as reflected in the results showcased in the aforementioned table. It's essential to highlight the importance of data prompt style even for large models, as evidenced by the downstream tasks.

D. Ablation & Design Choices

In this section, we first evaluate our method on retrieval task over three well-known baselines under 4 M setting for comparison. Then we train a BLIP model on CC3M as baseline and perform various ablations.

Exploration of Diverse Architectures: In this research, we conduct experiments utilizing three distinct baseline models, specifically ViLT, CLIP, and BLIP, to examine the influence of *PTP* on a range of performance indicators. The outcomes of these experiments are detailed in Table XIII, demonstrating the performance on both the COCO 5 K test set and the Flickr30 K 1 K test set. The data acquired from the baseline models indicate that our proposed approach, *PTP*, substantially improves image-to-text (i2t) and text-to-image (t2i) performance, underscoring its adaptability and suitability for a variety of visual-language tasks.

Furthermore, we assess the execution time of our model relative to the baseline models. As we do not employ an object detector or prompts for downstream tasks, the computational overhead aligns with that of the baseline models. Impressively, *PTP* is determined to be 20 times more rapid than VinVL [62], which depends on object features. Importantly, in spite of the considerable decrease in execution time, our model produces results of a quality comparable to VinVL, highlighting its efficiency and efficacy.

Comparing Text Prompts and Additional Pretext Tasks: This research examines the effects of incorporating *PTP* as a supplementary pretext task during the pre-training phase of vision-language models. By implementing this strategy, the pretext task does not conflict with other pre-training objectives, such

TABLE X
Comparing With the Open-Flamingo Baseline Across Various Language Model Scales

Method	Shots	Captioning (CIDER)		VQA				Classification	Average
		COCO	FLICKR	ok-vqa	textvqa	vizwiz	vqav2	HatefulMemes	
Open-flamingo(3B) [5]	0	74.9	52.3	28.2	24.2	23.7	44.6	51.2	42.7
	4	77.3	57.2	30.3	27.0	27.0	45.8	50.6	45.0
	32	93.0	61.1	31.0	28.3	**44.1**	47.0	50.2	50.7
PTP-Open-flamingo(3B)	0	78.9	53.3	27.4	25.1	29.3	44.2	51.5	44.2
	4	88.4	61.2	28.4	25.7	31.2	46.3	54.0	47.9
	32	94.4	63.5	30.8	25.6	42.3	43.9	52.2	50.3
Open-flamingo(9B) [5]	0	79.5	59.5	28.2	24.2	23.7	44.6	51.6	44.5
	4	89.0	65.8	40.1	28.2	34.1	54.8	**54.0**	52.3
	32	**99.5**	61.3	**42.4**	23.8	44.0	53.3	53.8	54.1
PTP-Open-flamingo(9B)	0	80.2	60.4	27.9	22.5	25.3	46.6	51.5	44.9
	4	91.7	67.2	39.3	**28.4**	39.0	55.5	53.2	**53.4**
	32	95.4	**68.8**	41.8	27.6	42.3	**54.9**	53.7	**55.0**

TABLE XI
Comparisons With State-of-the-Art Methods for Video Question Answering on MSVD, TVQA, and TGIF-Frame Datasets

Method	MSVD Test	TVQA Val	TGIF-FrameQA
ClipBERT [24]	-	-	59.4
AllInOne [49]	46.5	69.8	62.5
ViLT [22]	45.7	70.4	65.4
PTP-ViLT	48.8	73.4	68.7
PTP2R-ViLT	**49.1**	**73.9**	**68.7**
BLIP† [27]	47.1	71.3	66.4
PTP-BLIP	50.3	72.4	70.2
PTP2R-BLIP	**50.8**	**72.7**	**70.6**

Notably, the training samples used in AllInOne [49] are 20 times larger than those used in our method.

TABLE XII
PTP Pre-Training With One Billion Data

Method	Dataset	MSCOCO R@1	Flickr30K R@1	ImageNet Acc(%)	VTAB Acc(%)
CLIP† [40]	DataComp-1B	68.9	78.2	66.2	53.4
PTP-CLIP	DataComp-1B	71.4	81.2	67.1	54.6

We report the image to text Recall1 results on MSCOCO and Flickr30K dataset.

as ITM and ITC, even though it may increase computational expenses. Conversely, the prompt design modifies the textual input, influencing all pre-training goals.

The results are presented in Table XIV. We notice that both Pretext and Prompt approaches enhance the baseline performance across all four tasks. Nonetheless, the prompt method is distinctly more advantageous than the pretext approach, particularly for COCO captioning CIDER scores (127.2 vs 123.5). Consequently, we employ the prompt design by default in this study, owing to its superior efficiency.

Exploring Various Text Prompts: In this experiment, we investigate six distinct types of prompts: *i*. The [O] is in block [P]. *ii*. The block [P] resembles [O]. *iii*. In which block is the [O]? In [P]. *iv*. The [O] is situated in block [P]. *v*. (X_1, Y_1, W, H) contains a [O]. (X_1, Y_1) represents the top-left point, while W, H denote the width and height of the bounding box. *vi*. The block [P] features a [O]. *vii*. The block [NP] includes a [O]. NP refers to the usage of nouns to describe block positions, e.g., from upper left to bottom right. The results are reported in Table XV and we observe:

A precise position does not yield superior results compared to a block, possibly because precise positioning is challenging to learn. Furthermore, we find that utilizing block IDs (e.g., 0) or nouns (e.g., upper left) produces similar outcomes. Ultimately, we discover that the hybrid version does not generate the best results. We also note that a single-word change can significantly impact performance, a common issue in prompt learning, as observed in GPT-3 [7]. Our work does not primarily focus on addressing this problem. Additionally, Table XV demonstrates that using prompts to predict the exact position (four coordinates) of an object's bounding box results in inferior performance compared to predicting the block.

Investigating Second-Order Prompts: Table XVII delves into the examination of object relation prompts, where R encompasses terms such as "left," "right," "top," and "bottom." We explore three distinct variations of *PTP2R*, including the simple repetition of first-order relations, the relative position of the object, and the relative position of the selected block. O_2/P_2 means different objects/positions. In addition, we also explore the second-order text prompts solely based on objects and positional relationships.

Incorporating relation prompts leads to a noticeable improvement in COCO text-to-image (t2i) retrieval performance. This enhancement may be ascribed to the heightened complexity linked to learning relation prompts, as indicated by the increase in language mask loss from 1.29 to 1.47. The elevated language mask loss implies that the model encounters greater difficulty in grasping the subtleties of object relations, thus pushing it to develop a more refined comprehension of the relationships between objects within the visual domain. Consequently, the model becomes more adept at handling intricate tasks involving object relations, ultimately resulting in its enhanced performance in the COCO t2i retrieval task. We have noted a challenge encountered by the model when attempting to determine the precise location of the first object in cases where positional information is absent from the initial second-order prompt. This issue primarily arises due to the presence of recurring object classes among the top K largest objects.

TABLE XIII
Ablation on Different Architectures Under 4 M Setting

Method	Time	MSCOCO (5K test set)							Flickr30K (1K test set)						
		Image → Text			Text → Image				Image → Text			Text → Image			
		R@1	R@5	R@10	R@1	R@5	R@10	Avg	R@1	R@5	R@10	R@1	R@5	R@10	Avg
		One-stream Models													
ViLT [22]	~15	61.8	86.2	92.6	41.3	72.0	82.5	72.7	81.4	95.6	97.6	61.9	86.8	92.8	86.0
PTP-ViLT	~15	**67.1**	**90.5**	**94.3**	**45.3**	**79.1**	**88.4**	**77.5**$_{+4.8}$	**85.2**	**96.9**	**98.5**	**68.8**	**91.4**	**95.3**	**89.4**$_{+3.4}$
		Dual-stream Models													
CLIP† [40]	~27	64.9	83.2	90.1	50.4	76.3	84.7	74.9	77.5	92.1	94.5	58.6	81.4	88.9	82.2
PTP-CLIP	~27	**68.3**	**86.4**	**92.7**	**54.1**	**80.1**	**86.8**	**78.1**$_{+3.2}$	**83.5**	**94.1**	**96.3**	**63.4**	**88.5**	**91.7**	**86.3**$_{+4.1}$
		Dual-stream + Fusion encoder Models													
BLIP † [27]	~33	75.2	93.3	96.3	57.4	82.1	89.5	82.3	94.0	99.1	99.7	82.5	96.4	98.2	95.0
PTP-BLIP	~33	**83.7**	**97.0**	**98.7**	**68.1**	**89.4**	**94.2**	**88.5**$_{+6.2}$	**96.1**	**99.8**	**100.0**	**84.2**	**96.6**	**98.6**	**95.9**$_{+0.9}$
		Object-feature Based Models													
VinVL [62]	~650	74.9	92.6	96.3	58.1	83.2	90.1	82.5	-	-	-	-	-	-	-

We report the i2t and t2i results on MSCOCO (5K test set). As we do not used object detector in downstream tasks, *PTP* is 20 times faster than object-feature based model.

TABLE XIV
Text Prompt versus Additional Pretext Head

Method	COCO	F30K	NLVR	Captioning
	TR@1	TR@1	Acc(%)	CIDER
Baseline	70.6	53.4	76.1	121.2
Pretext	72.3 (1.7↑)	54.7 (2.3↑)	76.9 (0.8↑)	123.5 (2.3↑)
Prompt	**73.2 (2.6↑)**	**55.4 (2.0↑)**	**77.9 (1.8↑)**	**127.2 (6.0↑)**

The last column is COCO captioning task.

TABLE XV
Case Study of Text Prompt on Image-Text Retrieval

Prompt	TR@1	IR@1
Baseline	70.6	53.4
The [O] is in the block [P].	72.7 (2.1↑)	54.1 (0.7↑)
The block [P] looks like [O].	73.3 (2.7↑)	53.9 (0.5↑)
The [O] is in which block? In [P].	72.3 (1.7↑)	54.9 (1.5↑)
The [O] is located in block [P].	72.3 (1.7↑)	54.2 (0.8↑)
(X1, Y1, W, H) has a [O].	72.5 (1.9↑)	54.3 (0.9↑)
The block in [NP] has a [O].	73.0 (2.4↑)	55.1 (1.7↑)
θ_1 θ_2 [P] θ_3 θ_4 [O].	73.1 (2.5↑)	55.2 (1.8↑)
The block [P] has a [O].	**73.2 (2.6↑)**	**55.4 (2.0↑)**
Mixed	72.3 (1.7↑)	54.7 (1.2↑)

We use the O to represent objects and P to represent positions. The symbol θ represents learnable parameter.

Exploring Additional Prompt Designs: In our approach, an object may span multiple blocks, and each category may encompass multiple objects. To address this, we also predicts all blocks or objects and investigate several other prompt. The model is trained on CC3M and evaluated on three downstream tasks. Specifically, we explore the following approaches: *i*. *Multiple Tags.* We note that a block may contain multiple objects in many cases. We attempt to refine the text prompt as *The block [P] has objects [O_1], [O_2], and [O_3]*. It is important to remember that each block contains a varying number of objects. *ii*. *Multiple Positions.* We create a multiple position setup, considering that one object could appear in several blocks. We refine the prompt using question-answer pairs. *iii*. *Synonymous Substitution.* We substitute “block” with “region” and “is” with “looks like.” *iv*. *CPT* [57] Following this work, we color the detected region for each tag, assigning a unique color to each region.

The results are reported in Table XVI. We observe that incorporating multiple objects or positions does not significantly improve the model’s performance on downstream tasks, and the language modeling loss is higher than the baseline. This suggests that the assignment is too difficult for the model to learn. We also find that the outcome of simple synonymous substitution remains consistent with the original text prompt’s outcome. Modeling location information only requires a straightforward prompt. CPT is designed for downstream visual grounding by coloring region proposals for identification, while PTP is for pretraining, which only pretrains a VLP model for numerous downstream tasks. Since most downstream tasks do not have region proposals available (e.g., VQA), CPT cannot generate color prompts to boost grounding, while PTP can, as it improves grounding during the pretraining phase. In fact, we attempted CPT for pretraining and observed worse performance, e.g., 75.1 for CPT versus 77.8 for PTP on NLVR. For CPT, downstream tasks like NLVR do not have color prompts (CP), while its pretraining uses CP, leading to inconsistent phases.

We also find that selecting the top-1 predicted object and using its corresponding bounding box provides the best performance. It should be noted that the bounding box is rectangular, while the actual object may have various shapes. One possible explanation for this observation is that other regions or blocks may contain excessive background noise, making them challenging to identify.

Assessing the Role of Position in Text Prompts: In this experiment, we investigate the effectiveness of prompting our *PTP* for information at various granularities, such as without Positional. We simply use *[P] has [O]* when removing the prompt. The results are listed in Table XVIII. From these results we observe: *i*. It is interesting to see that each component is crucial. Without any one component, the downstream performance progressively deteriorates. *ii*. Although OSCAR [31] found that using object tags as supplementary input improved results when area

TABLE XVI
OTHER VARIATIONS OF TEXT PROMPT

Prompt	Multipy Position	Multipy Tags	Prompt	COCO Retrieval TR@1	COCO Retrieval IR@1	NLVR Acc	COCO Captioning CiDER
Baseline	-	-	-	70.6	53.4	76.0	122.6
The object in region [P] looks like [O].			✓	72.5 (1.9↑)	54.3 (0.9↑)	77.8 (1.8↑)	127.4 (4.8↑)
The block [P] has objects [O_1], [O_2], [O_3].		✓	✓	71.9 (0.9↑)	54.7 (0.9↑)	76.8 (0.9↑)	124.5 (1.9↑)
The [O] is located in which region? In [P_1], [P_2] and [P_3].	✓		✓	70.7 (0.1↑)	53.6 (0.2↑)	77.1 (1.1↑)	125.2 (2.6↑)
ColorPrompt [57]	✓	✓	✓	70.4 (0.2↓)	53.6 (0.2↑)	75.1 (0.9↓)	120.3 (2.3↓)

[O] is short for object and [P] is short for position.

TABLE XVII
COCO TEXT-TO-IMAGE RETRIEVAL RESULTS WITH THE SECOND-ORDER RELATION OVER OBJECTS

Prompt	R@1	R@5	R@10
The block [P] has a [O]	73.1	91.9	96.0
+, and the block [P_2] has a [O_2].	71.5	91.5	95.9
+, and a [O_2] on the [P_2].	72.2	91.5	95.8
+, and a [O_2] on the [R] of this block.	**73.6**	**92.3**	**96.3**
The image has a [O] and a [O_2] on the [R].	71.4	91.5	95.6

TABLE XVIII
THE POSITION INFORMATION IS ESSENTIAL FOR PROMPT DESIGN

Object Tags	Prompt	Position	TR@1	IR@1
-	-	-	70.6	53.4
✓			70.2 (0.4↓)	52.7 (0.7↓)
✓	✓		70.3 (0.3↓)	52.9 (0.5↓)
✓		✓	70.8 (0.3↓)	52.4 (1.0↓)
✓	✓	✓	**73.3 (2.7↑)**	**55.4 (2.0↑)**

Different variations of object prediction prompt design and evaluate on coco retrieval.

TABLE XIX
DIFFERENT WAYS TO GET GRID PSEUDO LABEL AND ITS CORRESPONDING RUNNING TIME

Method	Time	R1	R5	R10
baseline	-	70.6	91.3	95.4
Faster-RCNN (ResNet101)	10d	72.7	91.8	95.7
Faster-RCNN (ResNeXt152)	14d	**73.3**	**92.0**	96.1
CLIP Similarity	8h	72.9	**92.0**	**96.6**

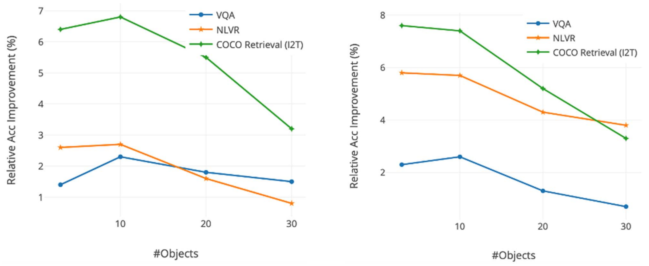
We report the imageto- text retrieval result on the COCO dataset for reference.

TABLE XX
PART SAMPLES WITH POSITION INFORMATION

Method	COCO Retrieval TR@1	COCO Retrieval IR@1	NLVR Acc	COCO Captioning CiDER
0%	79.5	62.4	80.5	129.5
19.3%	82.1	66.3	81.4	133.1
68.6%	84.6	69.1	83.1	134.6

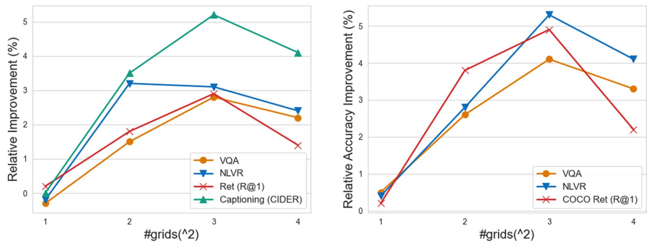
Under 14M setting, we test the result with different amount of pretraining samples with objects.

features were used as input, we demonstrate that object tags are ineffective when raw pixel images are used. This highlights the importance of devising a functional prompt for learning alignment between object tags and image regions.



(a). *PTP*-BLIP **(b). *PTP*-ViLT**

Fig. 4. We varying the number of selecting objects from 3 to 30. We report the result on downstream tasks over BLIP and ViLT baselines.



(a). *PTP*-BLIP **(b). *PTP*-ViLT**

Fig. 5. Relation between the number of blocks and the relative accuracy improvement. We explore two baselines and show the improvements over four different tasks.

Exploring the Number of Blocks: We investigate whether more fine-grained position information benefits our *PTP*. In Fig. 5, we vary the number of blocks from 1 × 1 (removing position information in *PTP*) to 4 × 4 and report the relative performance over BLIP/ViLT models. As can be seen, the results for both backbones improve when the number of blocks is more than 1. However, when there are 16 blocks, all downstream tasks experience a relative decline in performance. The reason may be that the predicted bounding box deviates from the localization of the actual object, resulting in a mesh that is too small and may not contain the selected object. Consequently, we opt for using 3 × 3 blocks, as this configuration offers better accuracy.

Determining the Optimal Number of Objects: To generate object tags, one approach is to use Faster-RCNN and detect at least 10 objects from an image. We varied the number of objects from 5 to 30, exploring the effects of different object counts. The results are shown in Fig. 4.

We observe that the BLIP baseline exhibits a slight increase in performance at the beginning, emphasizing the significance of data diversity for subsequent tasks. However, the results are less promising with a larger number of objects due to the high likelihood of false predictions resulting from a substantial number of objects with low confidence scores. PTP selects objects based on the confidence scores predicted by Faster-RCNN. A higher number of selected objects leads to lower confidence scores and increased noise in the tags, thus requiring a trade-off between the number of selected objects and the associated tag noise. In this study, we set the default number of selected objects to 10.

Partial Bounding Box Annotation: Since some URLs for the CC3M dataset are no longer valid, we have opted to extract objects from 2.7 million data points in the CC3M dataset and 7 million data points in the CC12 M dataset. Consequently, only 9.7 million of the pre-training samples have available objects. We also report results for the 14 million setting, wherein we utilize the original text without text prompts in cases where objects are not available.

The outcomes for different sampling subsets are presented in Table XX. Our analysis reveals that the 68.6% object availability results in a CiDER value of 134.6 for COCO Captioning and an accuracy of 83.2 for the NLVR test-P. These findings demonstrate that an increased number of annotated samples contributes to better overall performance. Additionally, this observation supports the notion that the *PTP* approach is well-suited for large-scale pre-training, further validating its applicability and potential in the development of more advanced models.

V. Discussion

Evaluating the Necessity of an Object Detector: In this work, part of the predicted bounding box information comes from Faster-RCNN [42]. To verify the expressive power of objects, we also consider two variations: *i*. Pure CLIP similarity. This design choice is adapted mainly for efficiency reasons, as utilizing an object detector is time-consuming and not always easy to access. *ii*. In addition to the powerful ResNext152-based object detector, we also use a smaller ResNet101 based Faster-RCNN.

The results are reported in Table XIX. We also report the overall feature extracting time on 8 NVIDIA V100 GPUs. Mainstream works in VLP use object detectors, and slow preprocessing is a common problem. The community tolerates this because objects only need to be extracted once and saved on disk (our features are downloaded from VinVL). To reduce costs further, PTP uses CLIP as a feature extractor, which is 42 times faster than Faster-RCNN. On 8 A100 GPUs, under the 4 M setting, BLIP needs 19.4 hours, while PTP with CLIP requires 8 + 19.4 = 27.4 hours to train from scratch, which is affordable.

As can be seen from the table, we find that using a stronger detector leads to better results but incurs a substantial computational cost. Moreover, we observe that the result of CLIP embedding is very close to Faster-RCNN (ResNeXt152). Additionally, it takes only around 2.3% of the time of the Faster-RCNN version to extract pseudo labels for each grid. We conclude that a CLIP model is a suitable alternative to an object detector in *PTP*.

Position Information Exploration: To explore whether model training with the *PTP* framework indeed learns position information, we design a fill-in-the-blank evaluation experiment in this section. Following ViLT [22], we mask some key words and ask the model to predict the masked words and show its corresponding heatmap. We design two text prompts: one given the noun to predict the localization and the other given the localization to predict the missing noun. We show the top-3 predictions for reference.

TABLE XXI
Comparison With BLIP Baseline on Two Splits of VQA Dataset. We Report accuracy(%)

Method	Position-related	Position-unrelated	All
BLIP	72.5	74.2	73.8
PTP-BLIP	78.4	74.6	75.6

We report accuracy(%).

The results are shown in Fig. 7. Our findings reveal that *PTP*-ViLT is capable of making accurate predictions by utilizing both the block position information and its corresponding visual concepts. Moreover, when we mask only the position noun, we still observe a high probability of correct block prediction. For instance, as depicted in the bottom part of Fig. 7, our model accurately identifies all image patches resembling the object of *"man"*. Based on these experiments and the insights presented in Fig. 1, we conclude that *PTP* is an effective tool for facilitating the learning of position information in a vision-language model, using a simple yet powerful text prompt.

Furthermore, we cluster the token-level features with the K-Means algorithm for ViLT and *PTP*-ViLT. Intuitively, tokens with similar semantics should be clustered together. We show the visualization results in Fig. 6. Comparing with the ViLT baseline, we observe that our method can cluster similar patches more accurately. This illustrates that our *PTP* has fairly accurately learned semantic information.

What Kind of Samples Does PTP Help? We undertake a comprehensive analysis of the visual-question answering task. Our investigation reveals that a significant portion of the VQA dataset samples contain position-related words, such as "top" and "sitting in." To further examine this observation, we construct a vocabulary comprising 30 commonly occurring position words. Subsequently, we categorize the VQA dataset into two subsets: the position-related subset (approximately 27%) and the position-unrelated subset (roughly 73%), based on whether the text contains words from the aforementioned vocabulary.

Table XXI demonstrates the effectiveness of the proposed categorization scheme and its corresponding performance on the test-dev set. Our analysis reveals that *PTP* achieves an accuracy of 78.4% on the position-related subset, which is 5.9% higher than the BLIP baseline. This result highlights the significant contribution of *PTP* towards enhancing the model's ability to efficiently learn position information and underscores its robust visual grounding capability. Thus, our proposed model can serve as a valuable asset in addressing visual question-answering tasks.

Comparison With Direct Object Regression: Learning position knowledge from an object detector is indeed challenging. In this section, we consider the coordinates of predicted object bounding boxes as ground truth labels and regress them during pre-training. Specifically, we add a lightweight detection head

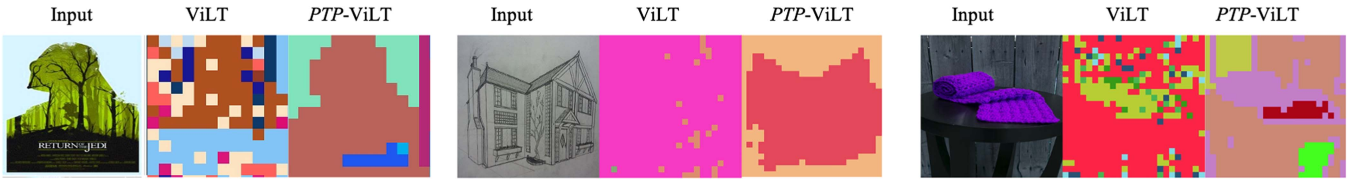


Fig. 6. Token cluster visualization. We train ViLT and *PTP*-ViLT with ViT-B/32 model on CC3M train set. We show the token cluster result with KMeans algorithm from CC3M test set [44]. *PTP*-ViLT shows preferable clusters.

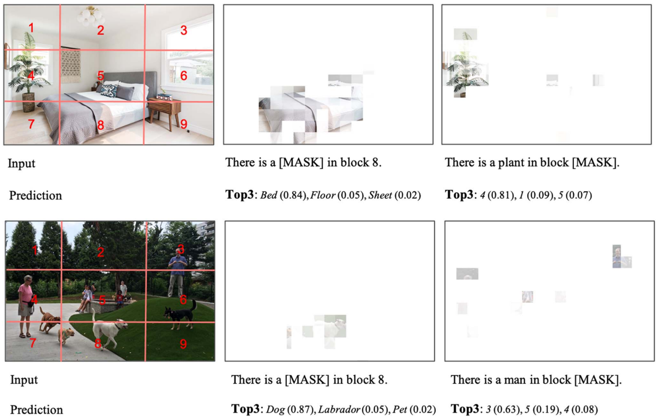

1
2
3
4
5
6
7
8
9
Input
There is a [MASK] in block 8.
There is a plant in block [MASK].
Prediction
Top3: *Bed* (0.84), *Floor* (0.05), *Sheet* (0.02)
Top3: *4* (0.81), *1* (0.09), *5* (0.07)
1
2
3
4
5
6
7
8
9
Input
There is a [MASK] in block 8.
There is a man in block [MASK].
Prediction
Top3: *Dog* (0.87), *Labrador* (0.05), *Pet* (0.02)
Top3: *3* (0.63), *5* (0.19), *4* (0.08)

Fig. 7. Evaluation of the full-in-the-blank task, where the model is required to predict the objects contained within a given block and identify the blocks having a specific object.

TABLE XXII
COCO Text-to-Image Retrieval. The Second-Order Relation Over Objects

Method	ITC	ITM	MLM	TR@1	IR@1	NLVR
Baseline	2.098	0.106	1.731	70.6	53.4	76.1
w GT Label	2.044	0.105	1.722	71.8	54.6	77.4
w *PTP*	1.883	0.093	1.4290	73.2	55.4	77.9

The second-order relation over objects.

after the BLIP image encoder. Table XXII shows that PTP performs better than the ground truth label approach in terms of pre-training loss (ITC, ITM, MLM) and downstream task performance (TR, IR, NLVR).

We find that the bounding boxes provided by Faster-RCNN are not very precise, and enforcing the model to regress coordinates could bias its grounding ability. Regressing objects directly requires predicting coordinates, which are not very precise. Such an implementation focuses only on the image encoder and does not improve the text decoder model. Moreover, this implementation is complex and involves many tricks to explore, making it difficult to extend to other frameworks easily. In contrast, our block position representation for objects is more accurate, ensuring that the model learns correct position information. With the position-guided text prompt (e.g., giving position/block to predict object), the model learns which blocks contain objects and what objects are in each block. This way, the model implicitly learns visual grounding, as experimentally demonstrated in the previous sections.

Exploring the Feasibility of Higher-Order PTP Integration: The consideration of incorporating higher-order relations into image captions presents a significant point of discussion. This deliberation is rooted in the potential extension of caption lengths, which necessitates scrutiny. Notably, conventional Visual-Language Pretraining (VLP) models adhere to specific maximum caption length constraints, typically set at 32 tokens, and this constraint bears relevance in our examination.

TABLE XXIII
Existing Object-Centric Datasets Have a Limited Number of Significant Objects

Dataset	¡=2	¡=3	¡=5
COCO [32]	34%	49%	73%
CC12M [8]	41%	58%	84%
DataComp1B(subset) [14]	39%	56%	81%

It is pivotal to recognize that the majority of images within the corpus exhibit a rather limited diversity of distinct objects. To elucidate this context, we undertook an extensive analysis to quantify the unique object classes within each image, eliminating frequent, non-discriminatory labels such as “sky,” “road,” and “tree,” alongside objects exceeding 1/10th of the image’s width. The results of this rigorous analysis are documented in Table XXIII for reference.

Our findings unveil a noteworthy statistic; approximately 40% of samples sourced from prominent pre-training datasets, such as DataComp1B [14], encompass two or fewer unique objects. Given this empirical insight, the endeavor to generate third-order relations within these object-centric datasets emerges as a formidable challenge. Therefore, the judicious selection of second-order relations assumes a pragmatic stance within the scope of our inquiry.

Erasing Misalignment Through PTP Implementation: A primary challenge observed in existing large-scale vision-language pre-training datasets is the significant misalignment between image and text pairs [14]. In other words, there is often a substantial lack of alignment between the provided images and their corresponding textual descriptions, making it difficult for the model to learn effectively.

In this experiment, we set out to tackle this issue by assessing dot product similarity scores between LAION400M [43] data and Datacomp1B [14]. Specifically, we randomly selected 10% subsets from both datasets for analysis. Utilizing the pre-trained CLIP model, available at the following link,[3] we calculated the dot product similarity scores for image-text pairs. The comparison results are visually presented in Fig. 10, demonstrating the

[3] [Online]. Available: https://github.com/mlfoundations/open_clip

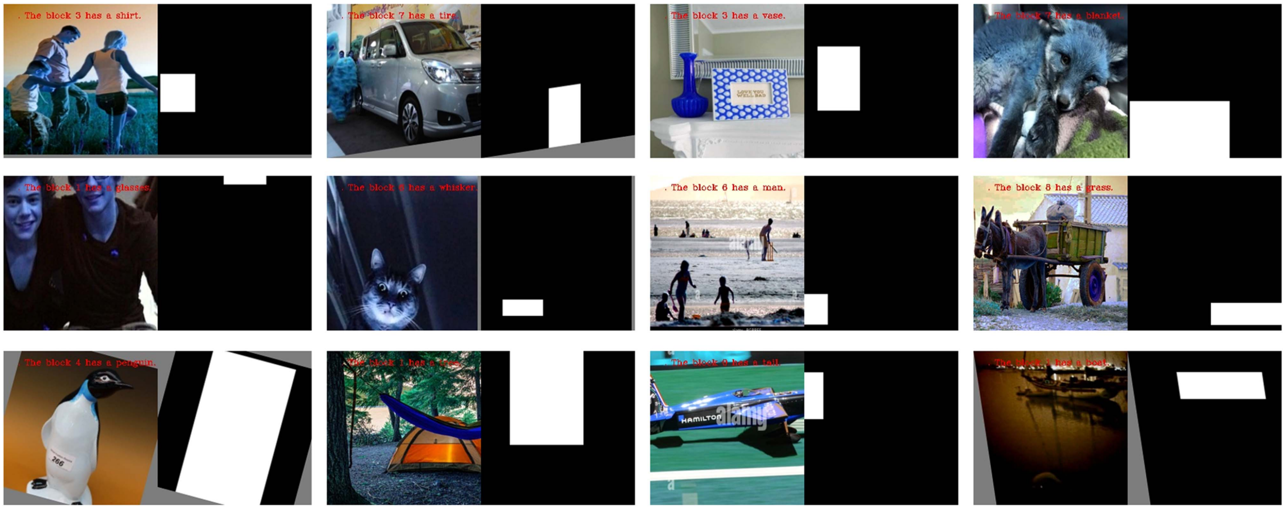

. The block 3 has a shirt.
. The block 7 has a tire.
The block 3 has a vase.
The block 7 has a blanket.
The block 1 has a glasses.
. The block has a whisker.
. The block 6 has a man.
. The block 8 has a grass.
The block 4 has a penguin.
The block 1 has a tree.
The block 0 has a tail.
The block has a boat.

Fig. 8. Our text prompt (in red color) and its corresponding bounding box's mask. The block index spans a range from 0 to 8. Furthermore, we employ data augmentation techniques on the bounding box to ensure its alignment with the transformations applied to the input image.

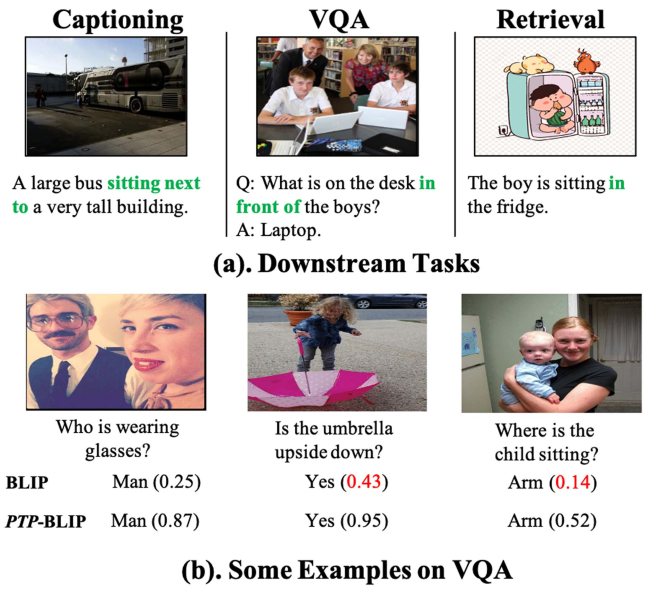

Captioning
VQA
Retrieval
A large bus sitting next to a very tall building.
Q: What is on the desk in front of the boys?
A: Laptop.
The boy is sitting in the fridge.
(a). Downstream Tasks
Who is wearing glasses?
Is the umbrella upside down?
Where is the child sitting?
BLIP
Man (0.25)
Yes (0.43)
Arm (0.14)
PTP-BLIP
Man (0.87)
Yes (0.95)
Arm (0.52)
(b). Some Examples on VQA

Fig. 9. (a) Position information is essential for mainstream downstream vision-language tasks. (b) In the context of the VQA [4] dataset, our *PTP* model provides improved predictions for position-related examples.

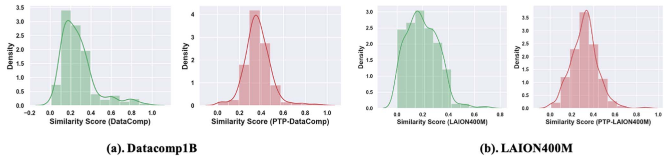


Fig. 10. Image-text similarity score distribution for original and revised captions using *PTP*.

efficacy of our method in erasing the impact of noisy image-text pairs.

It is worth noting that in each epoch, our approach generates distinct captions due to the random selection of positions and objects. This not only erases the misalignment issue but also contributes to the enhancement of the overall representation from the data aspect.

VI. Visualization

A. *Case Analysis*

In this experiment, we showcase several cases involving position information in Fig. 9. We observe that position information is crucial for various downstream tasks, including captioning, VQA, and retrieval. To comprehend these tasks, the trained model needs to learn position information.

Since a large number of samples in VQA tasks typically include position information, we evaluate our model on VQA tasks and select some representative samples. Specifically, we display the prediction probability and predicted nouns at the bottom of this figure. We observe that *PTP* provides accurate predictions in most cases, illustrating that our *PTP* learns position information more effectively.

B. *Bounding Box Visualization*

In this section, we present the object detection results obtained using our generated text prompts. Specifically, we randomly select an object from the set V and then visualize the original image along with the corresponding bounding box mask. It is important to note that we apply the same affine transformation to these bounding boxes as we do to the original image, ensuring consistency between the image and the corresponding bounding box mask.

We randomly select some samples from the overall dataset, and the results are reported in Fig. 8. We also observe that the bounding box may be very large and span multiple blocks in some examples (e.g., the first case in the third row). Since we use RandAugment [11], some objects may be outside the border of the input image. For such situations, we simply replace the specific position with [X], and the final *PTP* is *The block [X]* *has a [O]*. We also find that some masks may not be square, as seen in the third row.

VII. Limitations and Conclusion

Initially, we attempted to leverage position information from existing object detectors or trained models to enhance Visual-Language Pre-training models using simple prompts. To aid in prompt engineering, we developed a successful practice of cross-modal prompt settings. In addition to the first-order relation between objects and position, we also explored more complicated second-order relations between objects. Through rigorous experiments, we demonstrated that *PTP* serves as a general-purpose pipeline and improves the learning of position information without incurring significant extra computational costs.

Although the current version of *PTP* has demonstrated significant progress in its ability to process and interpret various input data, it is essential to acknowledge the limitations that still persist. One primary concern is that, at this time, *PTP* does not possess the capability to effectively handle instances wherein an incorrect object tag is presented. Furthermore, the present scope of this research has not delved deeply into the intricacies associated with more complex prompts. An in-depth exploration of such prompts would facilitate a better understanding of the model's strengths and weaknesses, paving the way for further refinements and enhancements in future iterations. Looking ahead, it is crucial to broaden the research horizons to evaluate *PTP*'s performance across a diverse range of vision-language tasks.

References

[1] H. Agrawal et al., "Nocaps: Novel object captioning at scale," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8948–8957.

[2] J. B. Alayrac et al., "Flamingo: A visual language model for few-shot learning," 2022, *arXiv:2204.14198*.

[3] P. Anderson et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6077–6086.

[4] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2425–2433.

[5] A. Awadalla et al., "OpenFlamingo: An open-source framework for training large autoregressive vision-language models," 2023, *arXiv: 2308.01390*.

[6] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1728–1738.

[7] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[8] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3558–3568.

[9] X. Chen et al., "Microsoft COCO captions: Data collection and evaluation server," 2015, *arXiv:1504.00325*.

[10] Y.C. Chen et al., "Uniter: Learning universal image-text representations," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 104–120.

[11] E.D. Cubuk, B. Zoph, J. Shlens, and Q.V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 702–703.

[12] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding. 2018, *arXiv:1810.04805*.

[13] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv: 2010.11929*.

[14] S.Y. Gadre et al., "DataComp: In search of the next generation of multimodal datasets," 2023, *arXiv:2304.14108*.

[15] Z. Gan, Y. C. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu, "Large-scale adversarial training for vision-and-language representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6616–6628.

[16] X. Hu et al., "Scaling up vision-language pre-training for image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17980–17989.

[17] Z. Huang, Z. Zeng, Y. Huang, B. Liu, D. Fu, and J. Fu, "Seeing out of the box: End-to-end pre-training for vision-language representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12976–12985.

[18] G. Ilharco et al., "Openclip," Zenodo, 0.1, Jun. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5143773

[19] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "TGIF-QA: Toward spatio-temporal reasoning in visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2758–2766.

[20] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4904–4916.

[21] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: Multi-modal prompt learning," 2022, *arXiv:2210.03117*.

[22] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5583–5594.

[23] R. Krishna et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, pp. 32–73, 2017.

[24] J. Lei et al., "Less is more: Clipbert for video-and-language learning via sparse sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7331–7341.

[25] J. Lei, L. Yu, T. L. Berg, and M. Bansal, "TVQA: Spatio-temporal grounding for video question answering," 2019, *arXiv: 1904.11574*.

[26] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, "Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11336–11344.

[27] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 12888–12900.

[28] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 9694–9705.

[29] L.H. Li et al., "Grounded language-image pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10965–10975.

[30] W. Li et al., "UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning," 2020, *arXiv: 2012.15409*.

[31] X. Li et al., "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 121–137.

[32] T. Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[33] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," 2021, *arXiv:2107.13586*.

[34] Z. Liu, S. Stent, J. Li, J. Gideon, and S. Han, "LocTex: Learning data-efficient visual representations from localized textual supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2167–2176.

[35] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13–23.

[36] A. Miech, J. B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9879–9889.

[37] V. Ordonez, G. Kulkarni, and T. Berg, "Im2Text: Describing images using 1 million captioned photographs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1143–1151.

[38] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[39] D. Qi, L. Su, J. Song, E. Cui, T. Bharti, and A. Sacheti, "ImageBERT: Cross-modal pre-training with large-scale weak-supervised image-text data," 2020, *arXiv: 2001.07966*.

[40] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[41] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1137–1149.

[43] C. Schuhmann et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 25278–25294.

[44] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2556–2565.

[45] W. Su et al., "VL-BERT: Pre-training of generic visual-linguistic representations," 2019, *arXiv: 1908.08530*.

[46] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," 2018, *arXiv: 1811.00491*.

[47] H. Touvron et al., "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*," 2016, pp. 652–663.

[49] A. J. Wang et al., "All in one: Exploring unified video-language pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6598–6608.

[50] A. J. Wang, P. Zhou, M. Z. Shou, and S. C. Yan, "Position-guided text prompt for vision language pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23242–23251.

[51] J. Wang et al., "GIT: A generative image-to-text transformer for vision and language," 2022, *arXiv:2205.14100*.

[52] J. Wang et al., "Object-aware video-language pre-training for retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3313–3322.

[53] W. Wang et al., "Image as a foreign language: Beit pretraining for all vision and vision-language tasks," 2022, *arXiv:2208.10442*.

[54] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "SimVLM: Simple visual language model pretraining with weak supervision," 2021, *arXiv:2108.10904*.

[55] D. Xu et al., "Video question answering via gradually refined attention over appearance and motion," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1645–1653.

[56] L. Yao et al., "FILIP: Fine-grained interactive language-image pre-training," 2021, *arXiv:2111.07783*.

[57] Y. Yao, A. Zhang, Z. Zhang, Z. Liu, T. S. Chua, and M. Sun, "CPT: Colorful prompt tuning for pre-trained vision-language models," 2021, *arXiv:2109.11797*.

[58] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "COCA: Contrastive captioners are image-text foundation models," 2022, *arXiv:2205.01917*.

[59] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 69–85.

[60] L. Yuan et al., "Florence: A new foundation model for computer vision," 2021, *arXiv:2111.11432*.

[61] Y. Zeng, X. Zhang, and H. Li, "Multi-grained vision language pre-training: Aligning texts with visual concepts," 2021, *arXiv:2111.08276*.

[62] P. Zhang et al., "VinVL: Making visual representations matter in vision-language models," 2021, *arXiv:2101.00529*.

[63] S. Zhang et al., "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.

[64] Y. Zhong et al., "RegionCLIP: Region-based language-image pretraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16793–16803.

[65] C. Zhou, C. C. Loy, and B. Dai, "DenseCLIP: Extract free dense labels from clip," 2021, *arXiv:2112.01071*.

[66] L. Zhu and Y. Yang, "ActBERT: Learning global-local video-text representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8746–8755.

[67] W. Zhu et al., "Multimodal C4: An open, billion-scale corpus of images interleaved with text," 2023, *arXiv:2304.06939*.

Alex Jinpeng Wang received the bachelor's and master's degree in Sun Yat-Sen University (SYSU). He is currently working toward the 3rd-year PhD degree in National University of Singapore. His research focuses on Large-scale Visual Language Pre-training and Data-centric AI. He has published relevant papers in the top-tier conferences: CVPR, ICCV, AAAI, NeurIPS, and ECCV. He also serves as reviewer for: CVPR, ICCV, ECCV, ICLR, and NeurIPS.

Pan Zhou received the master's degree from Peking University and the PhD degree from National University of Singapore (NUS). He is currently a senior research scientist with Sea AI Lab (SAIL) of Sea group. Before that, he worked in Salesforce as a research scientist. His research interests include deep learning theory and applications, noncovex/convex optimization. He has published papers in ICLR, ICML, NeurIPS, CVPR, ICCV, ECCV, AAAI, IJCAI, and journals: TPAMI, TIP. He serves as reviewer for top conferences: ICML, NeurIPS, CVPR, ICCV, AAAI and journals: TPAMI, IJCV, TIP, TNNLS, and TCSVT. He is awarded the Microsoft Research Asia Fellowship.

Mike Zheng Shou (Member, IEEE) received the PhD degree from Columbia University in the City of New York. He is a tenure-track assistant professor with National University of Singapore. He received the best paper finalist with CVPR'22 and the best student paper nomination with CVPR'17. His team won 1st place in multiple international challenges including ActivityNet 2017, EPIC-Kitchens 2022, Ego4D 2022 & 2023. He is a fellow of the National Research Foundation (NRF) Singapore and has been named on the Forbes 30 Under 30 Asia list.

Shuicheng Yan (Fellow, IEEE) is currently a visiting professor with BAAI, Beijing, China. Previously, he was the director of Sea AI Lab (SAIL) and group chief scientist of Sea. He is an Fellow of Academy of Engineering, Singapore, IEEE Fellow, ACM Fellow, AAAI Fellow, and IAPR Fellow. His research areas include computer vision, machine learning and multimedia analysis. Till now, he has published more than 600 papers in top international journals and conferences, with Google Scholar Citation more than 90,000 times and H-index 135. He had been among "Thomson Reuters Highly Cited Researchers" in 2014, 2015, 2016, 2018, and 2019. He is team has received winner or honorable-mention prizes for 10 times of two core competitions, Pascal VOC and ImageNet (ILSVRC), which are deemed as "World Cup" in the computer vision community. Also his team won over 10 best paper or best student paper prizes and especially, a grand slam in ACM MM, the top conference in multimedia, including Best Paper Award, Best Student Paper Award and Best Demo Award.