https://sites.google.com/view/showlab

Position-guided Text Prompt for Vision-Language Pre-training

Presented by Alex Jinpeng Wang

May 30^{th}, 2023

Stage 1: Motivation

Architectures in vision-language pre-training

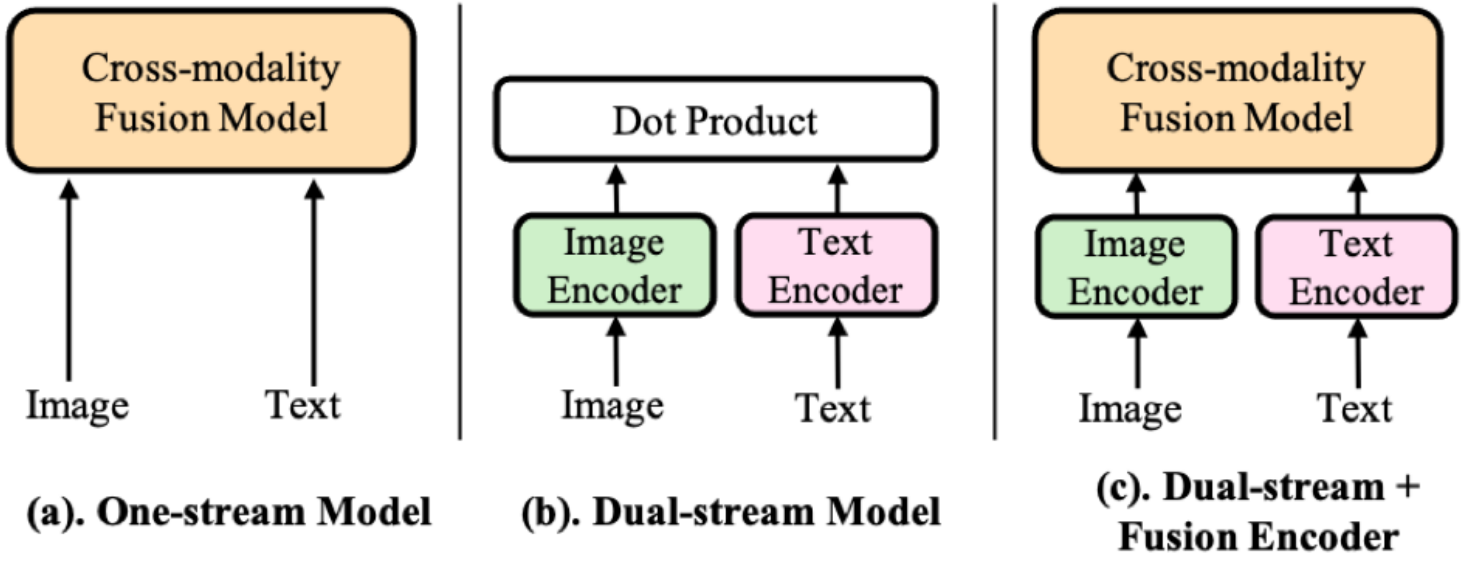


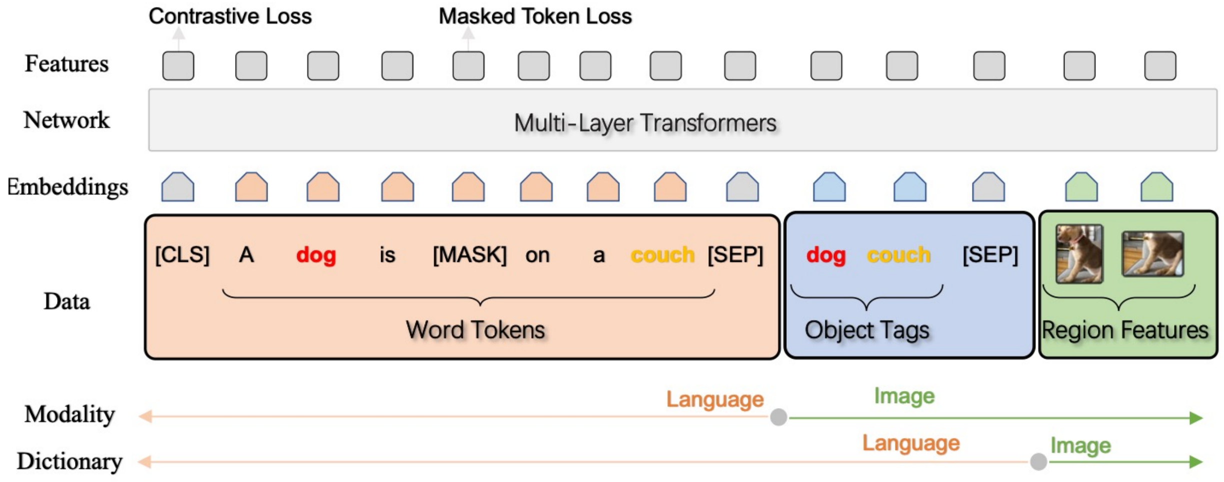
Figure 2. **Three widely-used categories of vision-and-language models.** The main difference is where to perform cross-modality information fusion. One-stream fuse at early stage and dual-stream fuse at late stage, while the last type fuse at middle stage.

Architectures in vision-language pre-training

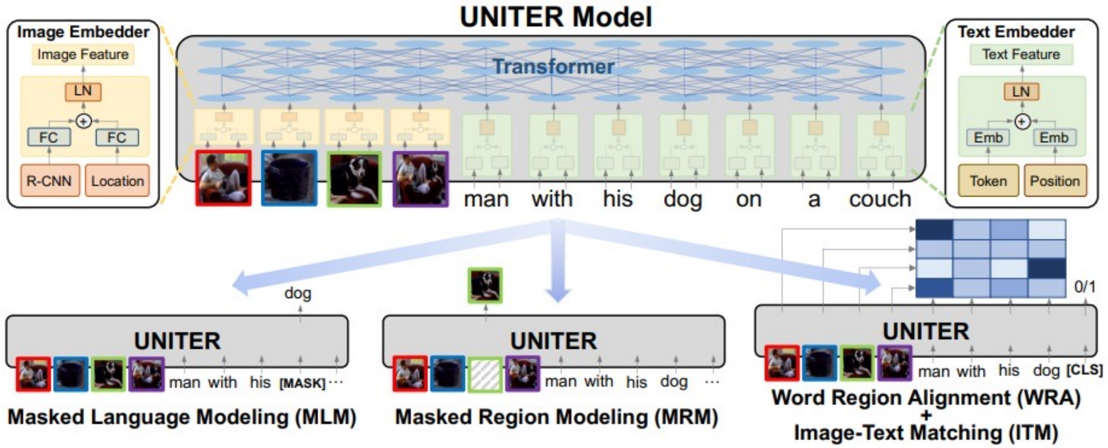
Related Works in Vision-Language Pre-training: (2 years ago)

Transform Image Into **Region Features** with Faster-RCNN

Input **Region Features** and **Bounding Box** (position) together as visual signals



OSCAR, ECCV' 20

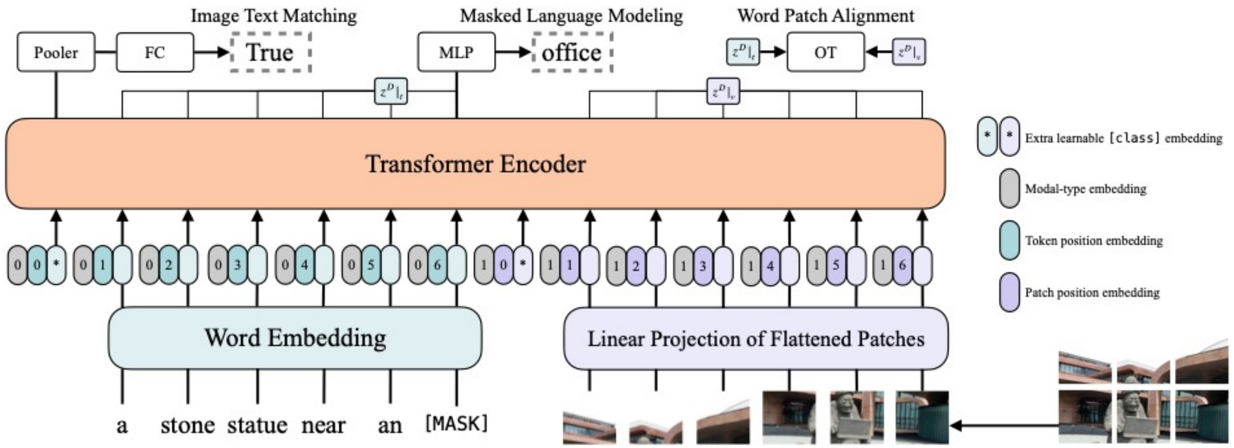


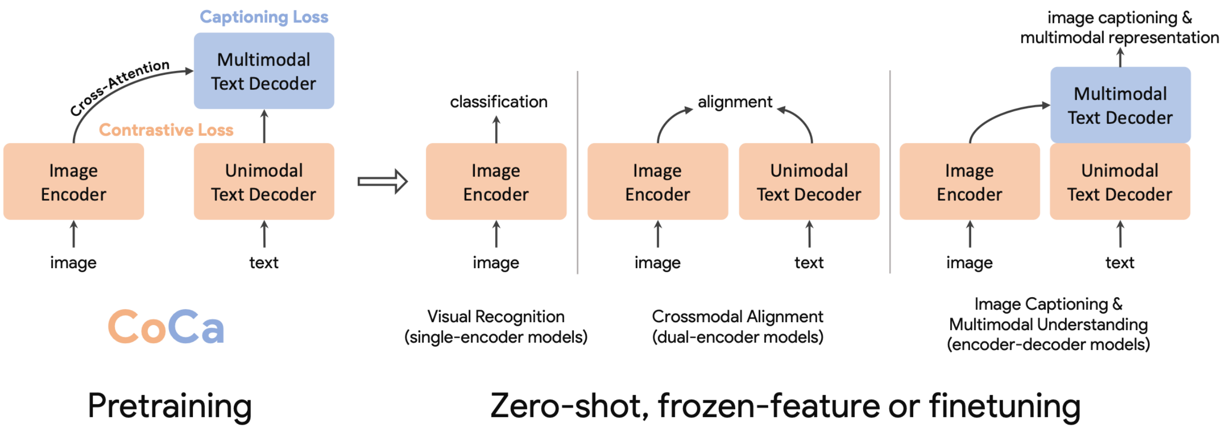
UNITER, ECCV' 20

Architectures in vision-language pre-training

Related Works **in Vision-Language Pre-training: (in 2 years)**

Input Raw-Pixel Image without Position Information Directly





Anderson et al. Bottom-up and top-down attention for image captioning and visual question answering[C, CVPR, 2018.

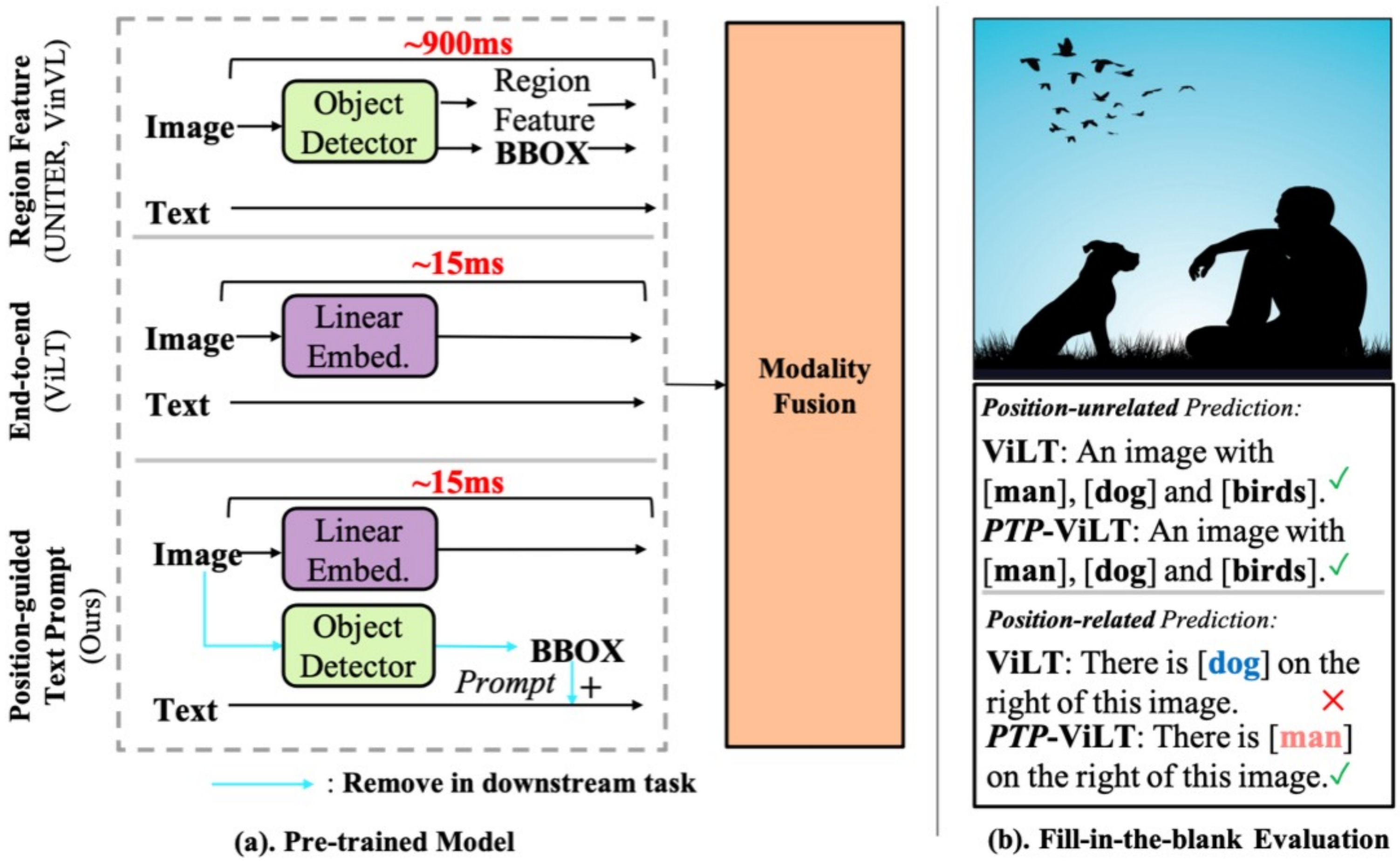
Motivation

Bring Position-information into these end-to-end models, and keep fast inference time for downstream tasks at the same time.

Slow In Inference
Good at Visual Grounding

Fast In Inference
Bad at Visual Grounding

Fast In Inference
Good at Visual Grounding



Stage 2: Methodology

Position-guided Text Prompt

“The block [P] has a [O].”

Block Tag

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

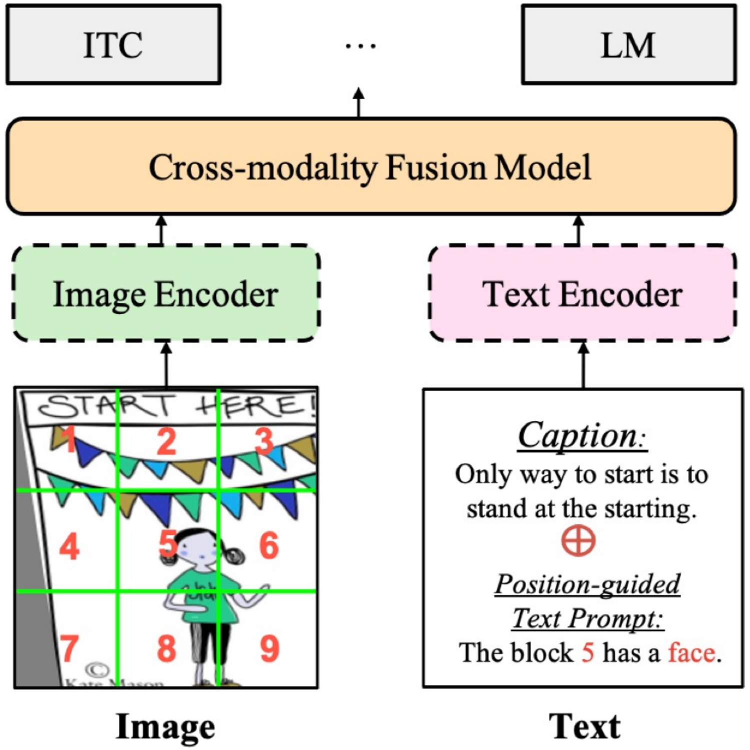


Figure 3. **Overall framework.** Any pre-training framework (one-stream, dual-stream, dual-stream+fusion encoder in Fig. 2) and most objectives can be integrated with our *PTP*. Dashed line indicates that the model may not exist. We remove the text prompt for the downstream task and evaluate the model as usual.

Stage 3: Experiments

Ablation & More Analysis

Architecture Variations

Table 6. **The ablation on different architectures under 4M setting.** 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.

Method	Time	MSCOCO (5K test set)						
		Image → Text			Text → Image			
		R@1	R@5	R@10	R@1	R@5	R@10	Avg
		One-stream Models						
ViLT [16]	∼15	61.8	86.2	92.6	41.3	72.0	82.5	72.7
***PTP*-ViLT**	∼15	**67.1**	**90.5**	**94.3**	**45.3**	**79.1**	**88.4**	**77.5**$_{+4.8}$
		Dual-stream Models						
CLIP† [32]	∼27	64.9	83.2	90.1	50.4	76.3	84.7	74.9
***PTP*-CLIP**	∼27	**68.3**	**86.4**	**92.7**	**54.1**	**80.1**	**86.8**	**78.1**$_{+3.2}$
		Dual-stream + Fusion encoder Models						
BLIP † [19]	∼33	75.2	93.3	96.3	57.4	82.1	89.5	82.3
***PTP*-BLIP**	∼33	**77.6**	**94.2**	**97.0**	**59.4**	**83.4**	**90.4**	**83.7**$_{+1.5}$
		Object-feature Based Models						
VinVL [46]	∼650	74.9	92.6	96.3	58.1	83.2	90.1	82.5

Pretext task or prompt?

Table 7. **Text prompt vs. additional pretext head.** The last column is COCO captioning task.

Method	COCO TR@1	F30K TR@1	NLVR Acc(%)	Captioning 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↑)**

Experiment

For Retrieval Task

Zero-shot results (trained on 4M data) even comparable with CoCA (1.8B data)

Table 1. **Results of zero-shot image-text retrieval on Flickr30K and MSCOCO datasets.** We gray out the methods that train on much larger corpus or use much larger models. † means the model implemented by ourself and trained on same dataset since the original datasets is not accessible or not trained on these splits. The Avg is the mean of all image-to-text recalls and text-to-image recalls.

Method	**#Images**	**Parameters**	**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 [18]	4M	170M	–	–	–	–	–	–		64.3	85.8	92.3	48.4	76.0	85.2	75.3
ImageBERT [31]	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 [16]	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 (ours)**	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	$83.5_{+3.6}$
BLIP † [19]	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 (ours)**	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**$_{+6.7}$
***PTP*-BLIP (ours)**	14M	220M	73.2	92.4	96.1	53.6	79.2	87.1	78.6	87.1	98.4	99.3	73.1	91.0	94.8	90.3
CLIP [32]	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 [14]	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 [41]	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 [24]	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

Experiment

Table 3. **Comparison with state-of-the-art image captioning methods on NoCaps and COCO Caption.** C: CIDEr, S: SPICE, B@4: BLEU@4. Notice that VinVL‡ and LEMON‡ require high resolution (800×1333) input images.

Method	#Images	Parameters	NoCaps validation								COCO Caption			
			in-domain		near-domain		out-domain		Overall		Karpathy test			
			CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	B@4	METEOR	SPICE	CIDEr
OSCAR [23]	4M	155M	79.6	12.3	66.1	11.5	45.3	9.7	80.9	11.3	37.4	30.7	23.5	127.8
VinVL‡ [46]	5.7M	347M	103.1	14.2	96.1	13.8	88.3	12.1	95.5	13.5	38.5	30.4	23.4	130.8
BLIP † [19]	4M	220M	106.5	14.4	99.3	13.6	95.6	13.0	98.8	14.2	37.0	–	–	122.6
***PTP*-BLIP (ours)**	4M	220M	**108.3**	**14.9**	**105.0**	**14.2**	**105.6**	**14.2**	**106.0**	**14.7**	**42.5**	**32.3**	**25.4**	**145.2**
Enc-Dec [6]	15M	–	92.6	12.5	88.3	12.1	94.5	11.9	90.2	12.1	–	–	–	110.9
BLIP [19]	14M	220M	111.3	15.1	104.5	14.4	102.4	13.7	105.1	14.4	38.6	–	–	129.7
***PTP*-BLIP (ours)**	14M	220M	**112.8**	**15.2**	**107.3**	**14.9**	**108.1**	**14.3**	**106.3**	**14.7**	**42.7**	**32.4**	**25.4**	**145.3**
$\text{SimVLM}_{\text{huge}}$ [40]	1.8B	1.2B	113.7	–	110.9	–	115.2	–	112.2	–	40.6	33.7	25.4	143.3
$\text{LEMON}_{\text{huge}}$‡ [12]	200M	675M	118.0	15.4	116.3	15.1	120.2	14.5	117.3	15.0	42.6	–	–	145.5
Beit-3 [39]	35M+	1.9B	–	–	–	–	–	–	–	–	44.1	32.4	25.4	147.6

Table 4. **Comparison with state-of-the-art methods on VQA and NLVR^2.** Para. is short for parameters. Notice that VinVL [46] uses larger vision backbone and object feature from faster-rcnn. ALBEF [20] performs an extra pre-training step for NLVR^2.

Method	#Images	Para.	VQA		NLVR^2	
			test-dev	test-std	dev	test-P
UNITER [8]	4M	155M	72.70	72.91	77.18	77.85
OSCAR [23]	4M	155M	73.16	73.44	78.07	78.36
UNIMO [22]	5.6M	307M	75.06	75.27	-	-
VinVL_L [46]	5.6M	347M	**76.52**	**76.60**	**82.67**	**83.98**
ViLT [16]	4M	87M	70.33	-	74.41	74.57
***PTP*-ViLT**	4M	87M	$72.13_{+1.8}$	74.36	$76.52_{+2.1}$	$77.83_{+3.3}$
BLIP † [19]	4M	220M	73.92	74.13	77.52	77.63
***PTP*-BLIP**	4M	220M	$76.02_{+2.1}$	$76.18_{+2.0}$	$80.73_{+3.2}$	$81.24_{+3.8}$
ALBEF [20]	14M	210M	75.84	76.04	82.55	83.14
BLIP [19]	14M	220M	77.54	77.62	82.67	82.30
***PTP*-BLIP**	14M	220M	$\mathbf{78.44}_{+2.9}$	$\mathbf{78.33}_{+1.7}$	$\mathbf{84.55}_{+1.9}$	$83.17_{+0.9}$
SimVLM [40]	1.8B	1.2B	77.87	78.14	81.72	81.77
GIT [38]	0.8B	0.7B	-	78.81	-	-
CoCa [24]	3B	2.1B	84.2	84.0	86.1	87.0

Table 5. **Comparisons with state-of-the-art methods for text-to-video retrieval on the 1k test split of the MSRVTT dataset.**

Method	R@1↑	R@5↑	R@10↑	MdR↓
ActBERT [49]	8.6	23.4	33.1	36.0
MIL-NCE [28]	9.9	24.0	32.4	29.5
Frozen-in-time [5]	18.7	39.5	51.6	10.0
OA-Trans [37]	23.4	47.5	55.6	8.0
PTP-ViLT	**27.9**	**52.5**	**56.3**	**7.0**

Visualization

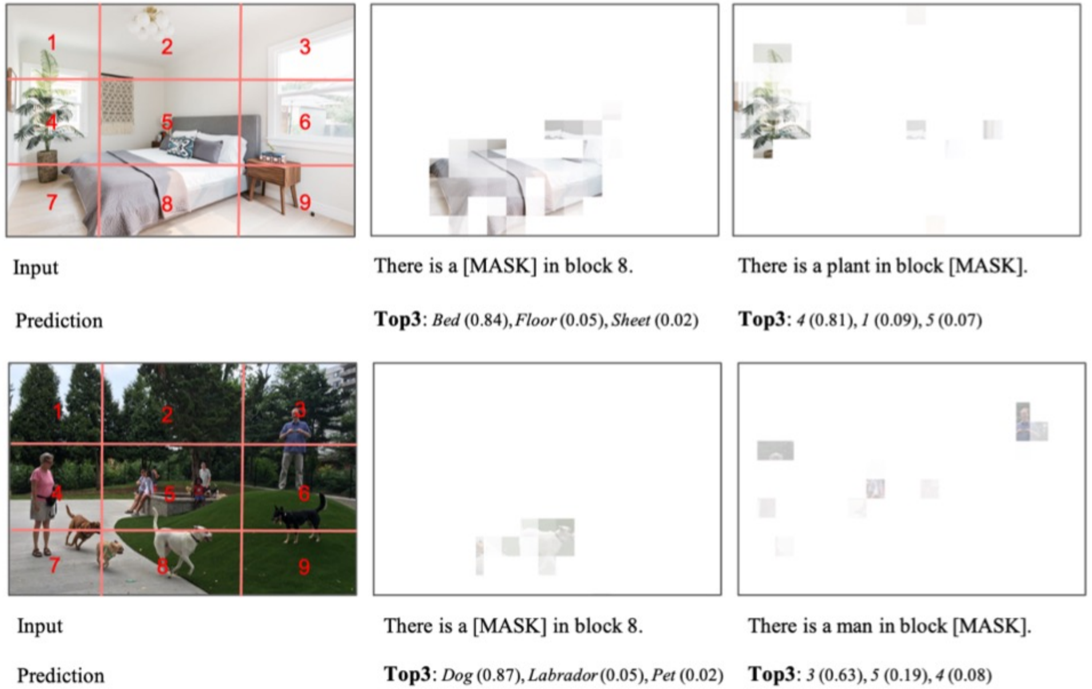


Figure 5. **The full-in-the-blank task evaluation.** We ask the model to predict *what objects are contained in given block* and *predict which blocks contain specific object*.

Project Website