UNITER: UNiversal Image-TExt Representation Learning

Yen-Chun Chen*, Linjie Li*, Licheng Yu*, Ahmed El Kholy
Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu

Microsoft Dynamics 365 AI Research
{yen-chun.chen,lindsey.li,licheng.yu,ahmed.elkholy,fiahmed,
zhe.gan,yu.cheng,jingjl}@microsoft.com

Abstract. Joint image-text embedding is the bedrock for most Vision-and-Language (V+L) tasks, where multimodality inputs are simultaneously processed for joint visual and textual understanding. In this paper, we introduce UNITER, a UNiversal Image-TExt Representation, learned through large-scale pre-training over four image-text datasets (COCO, Visual Genome, Conceptual Captions, and SBU Captions), which can power heterogeneous downstream V+L tasks with joint multimodal embeddings. We design four pre-training tasks: Masked Language Modeling (MLM), Masked Region Modeling (MRM, with three variants), Image-Text Matching (ITM), and Word-Region Alignment (WRA). Different from previous work that applies joint random masking to both modalities, we use conditional masking on pre-training tasks (*i.e.*, masked language/region modeling is conditioned on full observation of image/text). In addition to ITM for global image-text alignment, we also propose WRA via the use of Optimal Transport (OT) to *explicitly* encourage fine-grained alignment between words and image regions during pre-training. Comprehensive analysis shows that both conditional masking and OT-based WRA contribute to better pre-training. We also conduct a thorough ablation study to find an optimal combination of pre-training tasks. Extensive experiments show that UNITER achieves new state of the art across six V+L tasks (over nine datasets), including Visual Question Answering, Image-Text Retrieval, Referring Expression Comprehension, Visual Commonsense Reasoning, Visual Entailment, and NLVR^2.[1]

1 Introduction

Most Vision-and-Language (V+L) tasks rely on joint multimodel embeddings to bridge the semantic gap between visual and textual clues in images and text, although such representations are usually tailored for specific tasks. For example, MCB [11], BAN [19] and DFAF [13] proposed advanced multimodal fusion methods for Visual Question Answering (VQA) [3]. SCAN [23] and MAttNet [55]

* Equal contribution.
[1] Code is available at https://github.com/ChenRocks/UNITER.

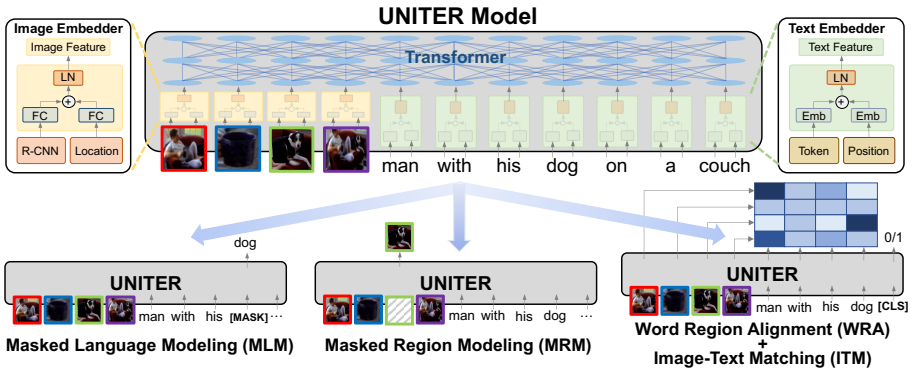


Fig. 1: Overview of the proposed UNITER model (best viewed in color), consisting of an Image Embedder, a Text Embedder and a multi-layer Transformer, learned through four pre-training tasks

studied learning latent alignment between words and image regions for Image-Text Retrieval [50] and Referring Expression Comprehension [18]. While each of these models has pushed the state of the art on respective benchmarks, their architectures are diverse and the learned representations are highly task-specific, preventing them from being generalizable to other tasks. This raises a million-dollar question: can we learn a universal image-text representation for all V+L tasks?

In this spirit, we introduce **UN**iversal **I**mage-**TE**xt **R**epresentation (**UNITER**), a large-scale pre-trained model for joint multimodal embedding. We adopt Transformer [49] as the core of our model, to leverage its elegant self-attention mechanism designed for learning contextualized representations. Inspired by BERT [9], which has successfully applied Transformer to NLP tasks through large-scale language modeling, we pre-train UNITER through four pre-training tasks: (*i*) Masked Language Modeling (MLM) *conditioned on image*; (*ii*) Masked Region Modeling (MRM) *conditioned on text*; (*iii*) Image-Text Matching (ITM); and (*iv*) Word-Region Alignment (WRA). To further investigate the effectiveness of MRM, we propose three MRM variants: (*i*) Masked Region Classification (MRC); (*ii*) Masked Region Feature Regression (MRFR); and (*iii*) Masked Region Classification with KL-divergence (MRC-kl).

As shown in Figure 1, UNITER first encodes image regions (visual features and bounding box features) and textual words (tokens and positions) into a common embedding space with Image Embedder and Text Embedder. Then, a Transformer module is applied to learn generalizable contextualized embeddings for each region and each word through well-designed pre-training tasks. Compared with previous work on multimodal pre-training [47,29,1,24,42,60,25]: (*i*) our masked language/region modeling is conditioned on full observation of image/text, rather than applying joint random masking to both modalities; (*ii*) we introduce a novel WRA pre-training task via the use of Optimal Transport

(OT) [37,7] to *explicitly* encourage fine-grained alignment between words and image regions. Intuitively, OT-based learning aims to optimize for distribution matching via minimizing the cost of transporting one distribution to another. In our context, we aim to minimize the cost of transporting the embeddings from image regions to words in a sentence (and vice versa), thus optimizing towards better cross-modal alignment. We show that both conditional masking and OT-based WRA can successfully ease the misalignment between images and text, leading to better joint embeddings for downstream tasks.

To demonstrate the generalizable power of UNITER, we evaluate on six V+L tasks across nine datasets, including: (*i*) VQA; (*ii*) Visual Commonsense Reasoning (VCR) [58]; (*iii*) NLVR^2 [44]; (*iv*) Visual Entailment [52]; (*v*) Image-Text Retrieval (including zero-shot setting) [23]; and (*vi*) Referring Expression Comprehension [56]. Our UNITER model is trained on a large-scale V+L dataset composed of four subsets: (*i*) COCO [26]; (*ii*) Visual Genome (VG) [21]; (*iii*) Conceptual Captions (CC) [41]; and (*iv*) SBU Captions [32]. Experiments show that UNITER achieves new state of the art with significant performance boost across all nine downstream datasets. Moreover, training on additional CC and SBU data (containing unseen images/text in downstream tasks) further boosts model performance over training on COCO and VG only.

Our contributions are summarized as follows: (*i*) We introduce UNITER, a powerful UNiversal Image-TExt Representation for V+L tasks. (*ii*) We present Conditional Masking for masked language/region modeling, and propose a novel Optimal-Transport-based Word-Region Alignment task for pre-training. (*iii*) We achieve new state of the art on a wide range of V+L benchmarks, outperforming existing multimodal pre-training methods by a large margin. We also present extensive experiments and analysis to provide useful insights on the effectiveness of each pre-training task/dataset for multimodal encoder training.

2 Related Work

Self-supervised learning utilizes original data as its own source of supervision, which has been applied to many Computer Vision tasks, such as image colorization [59], solving jigsaw puzzles [31,48], inpainting [35], rotation prediction [15], and relative location prediction [10]. Recently, pre-trained language models, such as ELMo [36], BERT [9], GPT2 [39], XLNet [54], RoBERTa [27] and ALBERT [22], have pushed great advances for NLP tasks. There are two keys to their success: effective pre-training tasks over large language corpus, and the use of Transformer [49] for learning contextualized text representations.

More recently, there has been a surging interest in self-supervised learning for multimodal tasks, by pre-training on large-scale image/video and text pairs, then finetuning on downstream tasks. For example, VideoBERT [46] and CBT [45] applied BERT to learn a joint distribution over video frame features and linguistic tokens from video-text pairs. ViLBERT [29] and LXMERT [47] introduced the two-stream architecture, where two Transformers are applied to images and text independently, which is fused by a third Transformer in a later stage. On the

other hand, B2T2 [1], VisualBERT [25], Unicoder-VL [24] and VL-BERT [42] proposed the single-stream architecture, where a single Transformer is applied to both images and text. VLP [60] applied pre-trained models to both image captioning and VQA. More recently, multi-task learning [30] and adversarial training [12] were used to further boost the performance. VALUE [6] developed a set of probing tasks to understand pre-trained models.

Our Contributions The key differences between our UNITER model and the other methods are two-fold: (*i*) UNITER uses conditional masking on MLM and MRM, *i.e.*, masking only one modality while keeping the other untainted; and (*ii*) a novel Word-Region Alignment pre-training task via the use of Optimal Transport, while in previous work such alignment is only implicitly enforced by task-specific losses. In addition, we examine the best combination of pre-training tasks through a thorough ablation study, and achieve new state of the art on multiple V+L datasets, often outperforming prior work by a large margin.

3 UNiversal Image-TExt Representation

In this section, we first introduce the model architecture of UNITER (Section 3.1), then describe the designed pre-training tasks and V+L datasets used for pre-training (Section 3.2 and 3.3).

3.1 Model Overview

The model architecture of UNITER is illustrated in Figure 1. Given a pair of image and sentence, UNITER takes the visual regions of the image and textual tokens of the sentence as inputs. We design an Image Embedder and a Text Embedder to extract their respective embeddings. These embeddings are then fed into a multi-layer Transformer to learn a cross-modality contextualized embedding across visual regions and textual tokens. Note that the self-attention mechanism in Transformer is order-less, thus it is necessary to explicitly encode the positions of tokens and the locations of regions as additional inputs.

Specifically, in *Image Embedder*, we first use Faster R-CNN[2] to extract the visual features (pooled ROI features) for each region. We also encode the location features for each region via a 7-dimensional vector.[3] Both visual and location features are then fed through a fully-connected (FC) layer, to be projected into the same embedding space. The final visual embedding for each region is obtained by summing up the two FC outputs and then passing through a layer normalization (LN) layer. For *Text Embedder*, we follow BERT [9] and tokenize the input sentence into WordPieces [51]. The final representation for each sub-word

[2] Our Faster R-CNN was pre-trained on Visual Genome object+attribute data [2].

[3] $[x_1, y_1, x_2, y_2, w, h, w * h]$ (normalized top/left/bottom/right coordinates, width, height, and area.)

token[4] is obtained via summing up its word embedding and position embedding, followed by another LN layer.[5]

We introduce four main tasks to pre-train our model: Masked Language Modeling *conditioned on image regions* (MLM), Masked Region Modeling *conditioned on input text* (with three variants) (MRM), Image-Text Matching (ITM), and Word-Region Alignment (WRA). As shown in Figure 1, our MRM and MLM are in analogy to BERT, where we randomly mask some words or regions from the input and learn to recover the words or regions as the output of Transformer. Specifically, word masking is realized by replacing the token with a special token `[MASK]`, and region masking is implemented by replacing the visual feature vector with all zeros. Note that each time we only mask one modality while keeping the other modality intact, instead of randomly masking both modalities as used in other pre-training methods. This prevents potential misalignment when a masked region happens to be described by a masked word (detailed in Section 4.2).

We also learn an instance-level alignment between the whole image and the sentence via ITM. During training, we sample both positive and negative image-sentence pairs and learn their matching scores. Furthermore, in order to provide a more fine-grained alignment between word tokens and image regions, we propose WRA via the use of Optimal Transport, which effectively calculates the minimum cost of transporting the contextualized image embeddings to word embeddings (and vice versa). The inferred transport plan thus serves as a propeller for better cross-modal alignment. Empirically, we show that both conditional masking and WRA contributes to performance improvement (in Section 4.2). To pre-train UNITER with these tasks, we randomly sample one task for each mini-batch, and train on only one objective per SGD update.

3.2 Pre-training Tasks

Masked Language Modeling (MLM) We denote the image regions as $\mathbf{v} = \{\mathbf{v}_1, ..., \mathbf{v}_K\}$, the input words as $\mathbf{w} = \{\mathbf{w}_1, ..., \mathbf{w}_T\}$, and the mask indices as $\mathbf{m} \in \mathbb{N}^M$.[6] In MLM, we randomly mask out the input words with probability of 15%, and replace the masked ones $\mathbf{w_m}$ with special token `[MASK]`.[7] The goal is to predict these masked words based on the observation of their surrounding words $\mathbf{w}_{\backslash\mathbf{m}}$ and all image regions $\mathbf{v}$, by minimizing the negative log-likelihood:

$$\mathcal{L}_{\mathrm{MLM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} \log P_\theta(\mathbf{w_m}|\mathbf{w}_{\backslash\mathbf{m}}, \mathbf{v}) \,, \tag{1}$$

[4] We use word/sub-word and token interchangeably throughout the rest of the paper.

[5] We also use a special modality embedding to help the model distinguish between textual and visual input, which is similar to the 'segment embedding' in BERT. This embedding is also summed before the LN layer in each embedder. For simplicity, this modality embedding is omitted in Figure 1.

[6] $\mathbb{N}$ is the natural numbers, M is the number of masked tokens, and $\mathbf{m}$ is the set of masked indices.

[7] Following BERT, we decompose this 15% into 10% random words, 10% unchanged, and 80% `[MASK]`.

where θ is the trainable parameters. Each pair $(\mathbf{w}, \mathbf{v})$ is sampled from the whole training set D.

Image-Text Matching (ITM) In ITM, an additional special token `[CLS]` is fed into our model, which indicates the fused representation of both modalities. The inputs to ITM are a sentence and a set of image regions, and the output is a binary label $y \in \{0, 1\}$, indicating if the sampled pair is a match. We extract the representation of `[CLS]` token as the joint representation of the input image-text pair, then feed it into an FC layer and a sigmoid function to predict a score between 0 and 1. We denote the output score as $s_\theta(\mathbf{w}, \mathbf{v})$. The ITM supervision is over the `[CLS]` token.[8] During training, we sample a positive or negative pair $(\mathbf{w}, \mathbf{v})$ from the dataset D at each step. The negative pair is created by replacing the image or text in a paired sample with a randomly-selected one from other samples. We apply the binary cross-entropy loss for optimization:

$$\mathcal{L}_{\text{ITM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D}[y \log s_\theta(\mathbf{w}, \mathbf{v}) + (1 - y) \log(1 - s_\theta(\mathbf{w}, \mathbf{v}))]) \,. \tag{2}$$

Word-Region Alignment (WRA) We use Optimal Transport (OT) for WRA, where a transport plan $\mathbf{T} \in \mathbb{R}^{T\times K}$ is learned to optimize the alignment between $\mathbf{w}$ and $\mathbf{v}$. OT possesses several idiosyncratic characteristics that make it a good choice for WRA: (*i*) *Self-normalization*: all the elements of $\mathbf{T}$ sum to 1 [37]. (*ii*) *Sparsity*: when solved exactly, OT yields a sparse solution $\mathbf{T}$ containing $(2r-1)$ non-zero elements at most, where $r = \max(K, T)$, leading to a more interpretable and robust alignment [37]. (*iii*) *Efficiency*: compared with conventional linear programming solvers, our solution can be readily obtained using iterative procedures that only require matrix-vector products [53], hence readily applicable to large-scale model pre-training.

Specifically, $(\mathbf{w}, \mathbf{v})$ can be considered as two discrete distributions $\boldsymbol{\mu}, \boldsymbol{\nu}$, formulated as $\boldsymbol{\mu} = \sum_{i=1}^{T} \mathbf{a}_i \delta_{\mathbf{w}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^{K} \mathbf{b}_j \delta_{\mathbf{v}_j}$, with $\delta_{\mathbf{w}_i}$ as the Dirac function centered on $\mathbf{w}_i$. The weight vectors $\mathbf{a} = \{\mathbf{a}_i\}_{i=1}^{T} \in \Delta_T$ and $\mathbf{b} = \{\mathbf{b}_j\}_{j=1}^{K} \in \Delta_K$ belong to the T- and K-dimensional simplex, respectively (*i.e.*, $\sum_{i=1}^{T} \mathbf{a}_i = \sum_{j=1}^{K} \mathbf{b}_j = 1$), as both $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are probability distributions. The OT distance between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ (thus also the alignment loss for the $(\mathbf{w}, \mathbf{v})$ pair) is defined as:

$$\mathcal{L}_{\text{WRA}}(\theta) = \mathcal{D}_{ot}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T}\in \Pi(\mathbf{a},\mathbf{b})} \sum_{i=1}^{T}\sum_{j=1}^{K} \mathbf{T}_{ij} \cdot c(\mathbf{w}_i, \mathbf{v}_j) \,, \tag{3}$$

where $\Pi(\mathbf{a}, \mathbf{b}) = \{\mathbf{T} \in \mathbb{R}_+^{T\times K} | \mathbf{T}\mathbf{1}_m = \mathbf{a}, \mathbf{T}^\top \mathbf{1}_n = \mathbf{b}\}$, $\mathbf{1}_n$ denotes an n-dimensional all-one vector, and $c(\mathbf{w}_i, \mathbf{v}_j)$ is the cost function evaluating the distance between $\mathbf{w}_i$ and $\mathbf{v}_j$. In experiments, the cosine distance $c(\mathbf{w}_i, \mathbf{v}_j) = 1 - \frac{\mathbf{w}_i^\top \mathbf{v}_j}{||\mathbf{w}_i||_2 ||\mathbf{v}_j||_2}$ is used. The matrix $\mathbf{T}$ is denoted as the transport plan, interpreting the alignment between two modalities. Unfortunately, the exact minimization over $\mathbf{T}$ is computational intractable, and we consider the IPOT algorithm [53] to approximate

[8] Performing this during pre-training also alleviates the mismatch problem between pre-training and downstream finetuning tasks, since most of the downstream tasks take the representation of the `[CLS]` token as the joint representation.

the OT distance (details are provided in the supplementary file). After solving $\mathbf{T}$, the OT distance serves as the WRA loss that can be used to update the parameters θ.

Masked Region Modeling (MRM) Similar to MLM, we also sample image regions and mask their visual features with a probability of 15%. The model is trained to reconstruct the masked regions $\mathbf{v_m}$ given the remaining regions $\mathbf{v}_{\backslash\mathbf{m}}$ and all the words $\mathbf{w}$. The visual features of the masked region are replaced by zeros. Unlike textual tokens that are represented as discrete labels, visual features are high-dimensional and continuous, thus cannot be supervised via class likelihood. Instead, we propose three variants for MRM, which share the same objective base:

$$\mathcal{L}_{\mathrm{MRM}}(\theta) = \mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{w}) \,. \tag{4}$$

1) **Masked Region Feature Regression (MRFR)** MRFR learns to regress the Transformer output of each masked region $\mathbf{v_m}^{(i)}$ to its visual features. Specifically, we apply an FC layer to convert its Transformer output into a vector $h_\theta(\mathbf{v_m}^{(i)})$ of same dimension as the input ROI pooled feature $r(\mathbf{v_m}^{(i)})$. Then we apply L2 regression between the two: $f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{w}) = \sum_{i=1}^{M} \|h_\theta(\mathbf{v_m}^{(i)}) - r(\mathbf{v_m}^{(i)})\|_2^2$.

2) **Masked Region Classification (MRC)** MRC learns to predict the object semantic class for each masked region. We first feed the Transformer output of the masked region $\mathbf{v_m}^{(i)}$ into an FC layer to predict the scores of K object classes, which further goes through a softmax function to be transformed into a normalized distribution $g_\theta(\mathbf{v_m}^{(i)}) \in \mathbb{R}^K$. Note that there is no ground-truth label, as the object categories are not provided. Thus, we use the object detection output from Faster R-CNN, and take the detected object category (with the highest confidence score) as the label of the masked region, which will be converted into a one-hot vector $c(\mathbf{v_m}^{(i)}) \in \mathbb{R}^K$. The final objective minimizes the cross-entropy (CE) loss: $f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{w}) = \sum_{i=1}^{M} \mathrm{CE}(c(\mathbf{v_m}^{(i)}), g_\theta(\mathbf{v_m}^{(i)}))$.

3) **Masked Region Classification with KL-Divergence (MRC-kl)** MRC takes the most likely object class from the object detection model as the hard label (w.p. 0 or 1), assuming the detected object class is the ground-truth label for the region. However, this may not be true, as no ground-truth label is available. Thus, in MRC-kl, we avoid this assumption by using soft label as supervision signal, which is the raw output from the detector (*i.e.*, a distribution of object classes $\tilde{c}(\mathbf{v}_m^{(i)})$). MRC-kl aims to distill such knowledge into UNITER as [16], by minimizing the KL divergence between two distributions: $f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{w}) = \sum_{i=1}^{M} D_{KL}(\tilde{c}(\mathbf{v_m}^{(i)})||g_\theta(\mathbf{v_m}^{(i)}))$.

3.3 Pre-training Datasets

We construct our pre-training dataset based on four existing V+L datasets: COCO [26], Visual Genome (VG) [21], Conceptual Captions (CC) [41], and SBU Captions [32]. Only image and sentence pairs are used for pre-training, which

	In-domain		Out-of-domain	
Split	COCO Captions	VG Dense Captions	Conceptual Captions	SBU Captions
train	533K (106K)	5.06M (101K)	3.0M (3.0M)	990K (990K)
val	25K (5K)	106K (2.1K)	14K (14K)	10K (10K)

Table 1: Statistics on the datasets used for pre-training. Each cell shows #image-text pairs (#images)

makes the model framework more scalable, as additional image-sentence pairs are easy to harvest for further pre-training.

To study the effects of different datasets on pre-training, we divide the four datasets into two categories. The first one consists of image captioning data from COCO and dense captioning data from VG. We call it "In-domain" data, as most V+L tasks are built on top of these two datasets. To obtain a "fair" data split, we merge the raw training and validation splits from COCO, and exclude all validation and test images that appear in downstream tasks. We also exclude all co-occurring Flickr30K [38] images via URL matching, as both COCO and Flickr30K images were crawled from Flickr and may have overlaps.[9] The same rule was applied to Visual Genome as well. In this way, we obtain 5.6M image-text pairs for training and 131K image-text pairs for our internal validation, which is half the size of the dataset used in LXMERT [47], due to the filtering of overlapping images and the use of image-text pairs only. We also use additional Out-of-domain data from Conceptual Captions [41] and SBU Captions [32] for model training.[10] The statistics on the cleaned splits are provided in Table 1.

4 Experiments

We evaluate UNITER on six V+L tasks[11] by transferring the pre-trained model to each target task and finetuning through end-to-end training. We report experimental results on two model sizes: UNITER-base with 12 layers and UNITER-large with 24 layers.[12]

4.1 Downstream Tasks

In VQA, VCR and NLVR^2 tasks, given an input image (or a pair of images) and a natural language question (or description), the model predicts an answer

[9] A total of 222 images were eliminated through this process.

[10] We apply the same URL matching method, excluding 109 images from training.

[11] VQA, VCR, NLVR^2, Visual Entailment, Image-Text Retrieval, and Referring Expression Comprehension. Details about the tasks are listed in the supplementary.

[12] UNITER-base: L=12, H=768, A=12, Total Parameters=86M. UNITER-large: L=24, H=1024, A=16, Total Parameters=303M (L: number of stacked Transformer blocks; H: hidden activation dimension; A: number of attention heads). 882 and 3645 V100 GPU hours were used for pre-training UNITER-base and UNITER-large.

Pre-training Data		Pre-training Tasks	Meta-Sum	VQA	IR (Flickr)	TR (Flickr)	NLVR^2	Ref-COCO+
				test-dev	val	val	dev	val^d
None	1	None	314.34	67.03	61.74	65.55	51.02	68.73
Wikipedia + BookCorpus	2	MLM (text only)	346.24	69.39	73.92	83.27	50.86	68.80
In-domain (COCO+VG)	3	MRFR	344.66	69.02	72.10	82.91	52.16	68.47
	4	ITM	385.29	70.04	78.93	89.91	74.08	72.33
	5	MLM	386.10	71.29	77.88	89.25	74.79	72.89
	6	MLM + ITM	393.04	71.55	81.64	91.12	75.98	72.75
	7	MLM + ITM + MRC	393.97	71.46	81.39	91.45	76.18	73.49
	8	MLM + ITM + MRFR	396.24	71.73	81.76	92.31	76.21	74.23
	9	MLM + ITM + MRC-kl	397.09	71.63	82.10	92.57	76.28	74.51
	10	MLM + ITM + MRC-kl + MRFR	399.97	71.92	83.73	92.87	76.93	74.52
	11	MLM + ITM + MRC-kl + MRFR + WRA	400.93	72.47	83.72	93.03	76.91	74.80
	12	MLM + ITM + MRC-kl + MRFR (w/o cond. mask)	396.51	71.68	82.31	92.08	76.15	74.29
Out-of-domain (SBU+CC)	13	MLM + ITM + MRC-kl + MRFR + WRA	396.91	71.56	84.34	92.57	75.66	72.78
In-domain + Out-of-domain	14	MLM + ITM + MRC-kl + MRFR + WRA	**405.24**	**72.70**	**85.77**	**94.28**	**77.18**	**75.31**

Table 2: Evaluation on pre-training tasks and datasets using VQA, Image-Text Retrieval on Flickr30K, NLVR^2, and RefCOCO+ as benchmarks. All results are obtained from UNITER-base. Averages of R@1, R@5 and R@10 on Flickr30K for Image Retrieval (IR) and Text Retrieval (TR) are reported. Dark and light grey colors highlight the top and second best results across all the tasks trained with In-domain data

(or judges the correctness of the description) based on the visual content in the image. For Visual Entailment, we evaluate on the SNLI-VE dataset. The goal is to predict whether a given image semantically entails an input sentence. Classification accuracy over three classes ("Entailment", "Neutral" and "Contradiction") is used to measure model performance. For Image-Text Retrieval, we consider two datasets (COCO and Flickr30K) and evaluate the model in two settings: Image Retrieval (IR) and Text Retrieval (TR). Referring Expression (RE) Comprehension requires the model to select the target from a set of image region proposals given the query description. Models are evaluated on both ground-truth objects and detected proposals[13] (MAttNet [55]).

For VQA, VCR, NLVR^2, Visual Entailment and Image-Text Retrieval, we extract the joint embedding of the input image-text pairs via a multi-layer perceptron (MLP) from the representation of the `[CLS]` token. For RE Comprehension, we use the MLP to compute the region-wise alignment scores. These MLP layers are learned during the finetuning stage. Specifically, we formulate VQA, VCR, NLVR^2, Visual Entailment and RE Comprehension as classification problems and minimize the cross-entropy over the ground-truth answers/responses. For Image-Text Retrieval, we formulate it as a ranking problem. During finetuning, we sample three pairs of image and text, one positive pair from the dataset and two negative pairs by randomly replacing its sentence/image with others. We

[13] The evaluation splits of RE comprehension using detected proposals are denoted as val^d, test^d, etc.

compute the similarity scores (based on the joint embedding) for both positive and negative pairs, and maximize the margin between them through triplet loss.

4.2 Evaluation on Pre-training Tasks

We analyze the effectiveness of different pre-training settings through ablation studies over VQA, NLVR^2, Flickr30K and RefCOCO+ as representative V+L benchmarks. In addition to standard metrics[14] for each benchmark , we also use Meta-Sum (sum of all the scores across all the benchmarks) as a global metric.

Firstly, we establish two baselines: Line 1 (L1) in Table 2 indicates no pre-training is involved, and L2 shows the results from MLM initialized with pre-trained weights from [9]. Although MLM trained on text only did not absorb any image information during pre-training, we see a gain of approximately +30 on Meta-Sum over L1. Hence, we use the pre-trained weights in L2 to initialize our model for the following experiments.

Secondly, we validate the effectiveness of each pre-training task through a thorough ablation study. Comparing L2 and L3, MRFR (L3) achieves better results than MLM (L2) only on NLVR^2. On the other hand, when pre-trained on ITM (L4) or MLM (L5) only, we observe a significant improvement across all the tasks over L1 and L2 baselines. When combining different pre-training tasks, MLM + ITM (L6) improves over single ITM (L4) or MLM (L5). When MLM, ITM and MRM are jointly trained (L7-L10), we observe consistent performance gain across all the benchmarks. Among the three variants of MRM (L7-L9), we observe that MRC-kl (L9) achieves the best performance (397.09) when combined with MLM + ITM, while MRC (L7) the worst (393.97). When combining MRC-kl and MRFR together with MLM and ITM (L10), we find that they are complimentary to each other, which leads to the second highest Meta-Sum score. The highest Meta-Sum Score is achieved by MLM + ITM + MRC-kl + MRFR + WRA (L11). We observe significant performance improvements from adding WRA, especially on VQA and RefCOCO+. It indicates the fine-grained alignment between words and regions learned through WRA during pre-training benefits the downstream tasks involving region-level recognition or reasoning. We use this optimal pre-training setting for the further experiments.

Additionally, we validate the contributions of conditional masking through a comparison study. When we perform random masking on both modalities simultaneously during pre-training, *i.e.*, w/o conditional masking (L12), we observe a decrease in Meta-Sum score (396.51) compared to that with conditional masking (399.97). This indicates that the conditional masking strategy enables the model to learn better joint image-text representations effectively.

Lastly, we study the effects of pre-training datasets. Our experiments so far have been focused on In-domain data. In this study, we pre-train our model on Out-of-domain data (Conceptual Captions + SBU Captions). A performance drop (396.91 in L13) from the model trained on In-domain data (COCO + Visual Genome) (400.93 in L11) shows that although Out-of-domain data contain more

[14] Details about the metrics are listed in the supplementary.

images, the model still benefits more from being exposed to similar downstream images during pre-training. We further pre-train our model on both In-domain and Out-of-domain data. With doubled data size, the model continues to improve (405.24 in L14).

4.3 Results on Downstream Tasks

Table 3 presents the results of UNITER on all downstream tasks. Both our base and large models are pre-trained on In-domain+Out-of-domain datasets, with the optimal pre-training setting: MLM+ITM+MRC-kl+MRFR+WRA. The implementation details of each task are provided in the supplementary file. We compare with both task-specific models and other pre-trained models on each downstream task. SOTA task-specific models include: MCAN [57] for VQA, MaxEnt [44] for NLVR^2, B2T2 [1] for VCR, SCAN [23] for Image-Text Retrieval, EVE-Image [52] for SNLI-VE, and MAttNet for RE Comprehension (RefCOCO, RefCOCO+ and RefCOCOg).[15] Other pre-trained models include: ViLBERT [29], LXMERT [47], Unicoder-VL [24], VisualBERT [25] and VL-BERT [42].

Results show that our UNITER-large model achieves new state of the art across all the benchmarks. UNITER-base model also outperforms the others by a large margin across all tasks except VQA. Specifically, our UNITER-base model outperforms SOTA by approximately +2.8% for VCR on Q→AR, +2.5% for NLVR^2, +7% for SNLI-VE, +4% on R@1 for Image-Text Retrieval (+15% for zero-shot setting), and +2% for RE Comprehension.

Note that LXMERT pre-trains with downstream VQA (+VG+GQA) data, which may help adapt the model to VQA task. However, when evaluated on unseen tasks such as NLVR^2, UNITER-base achieves 3% gain over LXMERT. In addition, among all the models pre-trained on image-text pairs only, our UNITER-base outperforms the others by >1.5% on VQA.

It is also worth mentioning that both VilBERT and LXMERT observed two-stream model outperforms single-stream model, while our results show empirically that with our pre-training setting, single-stream model can achieve new state-of-the-art results, with much fewer parameters (UNITER-base: 86M, LXMERT: 183M, VilBERT: 221M).[16]

For VCR, we propose a two-stage pre-training approach: (*i*) pre-train on standard pre-training datasets; and then (*ii*) pre-train on downstream VCR dataset. Interestingly, while VLBERT and B2T2 observed that pre-training is not very helpful on VCR, we find that the second-stage pre-training can significantly boost model performance, while the first-stage pre-training still helps but with limited effects (results shown in Table 4). This indicates that the proposed two-stage approach is highly effective in our pre-trained model over new data that are unseen in pre-training datasets.

[15] MAttNet results are updated using the same features as the others. More details are provided in the supplementary file.

[16] The word embedding layer contains excessive rare words, thus excluded from the parameter counts.

Tasks		SOTA	ViLBERT	VLBERT (Large)	Unicoder -VL	VisualBERT	LXMERT	UNITER Base	UNITER Large
VQA	test-dev	70.63	70.55	71.79	-	70.80	72.42	72.70	**73.82**
	test-std	70.90	70.92	72.22	-	71.00	72.54	72.91	**74.02**
VCR	Q→A	72.60	73.30	75.80	-	71.60	-	75.00	**77.30**
	QA→R	75.70	74.60	78.40	-	73.20	-	77.20	**80.80**
	Q→AR	55.00	54.80	59.70	-	52.40	-	58.20	**62.80**
NLVR^2	dev	54.80	-	-	-	67.40	74.90	77.18	**79.12**
	test-P	53.50	-	-	-	67.00	74.50	77.85	**79.98**
SNLI-VE	val	71.56	-	-	-	-	-	78.59	**79.39**
	test	71.16	-	-	-	-	-	78.28	**79.38**
ZS IR (Flickr)	R@1	-	31.86	-	48.40	-	-	66.16	**68.74**
	R@5	-	61.12	-	76.00	-	-	88.40	**89.20**
	R@10	-	72.80	-	85.20	-	-	92.94	**93.86**
IR (Flickr)	R@1	48.60	58.20	-	71.50	-	-	72.52	**75.56**
	R@5	77.70	84.90	-	91.20	-	-	92.36	**94.08**
	R@10	85.20	91.52	-	95.20	-	-	96.08	**96.76**
IR (COCO)	R@1	38.60	-	-	48.40	-	-	50.33	**52.93**
	R@5	69.30	-	-	76.70	-	-	78.52	**79.93**
	R@10	80.40	-	-	85.90	-	-	87.16	**87.95**
ZS TR (Flickr)	R@1	-	-	-	64.30	-	-	80.70	**83.60**
	R@5	-	-	-	85.80	-	-	**95.70**	**95.70**
	R@10	-	-	-	92.30	-	-	**98.00**	97.70
TR (Flickr)	R@1	67.90	-	-	86.20	-	-	85.90	**87.30**
	R@5	90.30	-	-	96.30	-	-	97.10	**98.00**
	R@10	95.80	-	-	99.00	-	-	98.80	**99.20**
TR (COCO)	R@1	50.40	-	-	62.30	-	-	64.40	**65.68**
	R@5	82.20	-	-	87.10	-	-	87.40	**88.56**
	R@10	90.00	-	-	92.80	-	-	93.08	**93.76**
Ref-COCO	val	87.51		-	-	-	-	91.64	**91.84**
	testA	89.02	-	-	-	-	-	92.26	**92.65**
	testB	87.05	-	-	-	-	-	90.46	**91.19**
	val^d	77.48	-	-	-	-	-	81.24	**81.41**
	testA^d	83.37	-	-	-	-	-	86.48	**87.04**
	testB^d	70.32	-	-	-	-	-	73.94	**74.17**
Ref-COCO+	val	75.38	-	80.31	-	-	-	83.66	**84.25**
	testA	80.04	-	83.62	-	-	-	86.19	**86.34**
	testB	69.30	-	75.45	-	-	-	78.89	**79.75**
	val^d	68.19	72.34	72.59	-	-	-	75.31	**75.90**
	testA^d	75.97	78.52	78.57	-	-	-	81.30	**81.45**
	testB^d	57.52	62.61	62.30	-	-	-	65.58	**66.70**
Ref-COCOg	val	81.76	-	-	-	-	-	86.52	**87.85**
	test	81.75	-	-	-	-	-	86.52	**87.73**
	val^d	68.22	-	-	-	-	-	74.31	**74.86**
	test^d	69.46	-	-	-	-	-	74.51	**75.77**

Table 3: Results on downstream V+L tasks from UNITER model, compared with task-specific state-of-the-art (SOTA) and previous pre-trained models. ZS: Zero-Shot, IR: Image Retrieval and TR: Text Retrieval

Different from other tasks, NLVR^2 takes two images as input. Thus, directly finetuning UNITER pre-trained with image-sentence pairs might not lead to op-

Stage I	Stage II	Q→A	QA→ R	Q → AR
N	N	72.44	73.71	53.52
N	Y	73.52	75.34	55.6
Y	N	72.83	75.25	54.94
Y	Y	**74.56**	**77.03**	**57.76**

Table 4: Experiments on two-stage pre-training for VCR. Results are from UNITER-base on VCR val split. Stage I and Stage II denote first-stage and second-stage pre-training

Setting	dev	test-P
Triplet	73.03	73.89
Pair	75.85	75.80
Pair-biattn	**77.18**	**77.85**

Table 5: Experiments on three modified settings for NLVR^2. All models use pre-trained UNITER-base

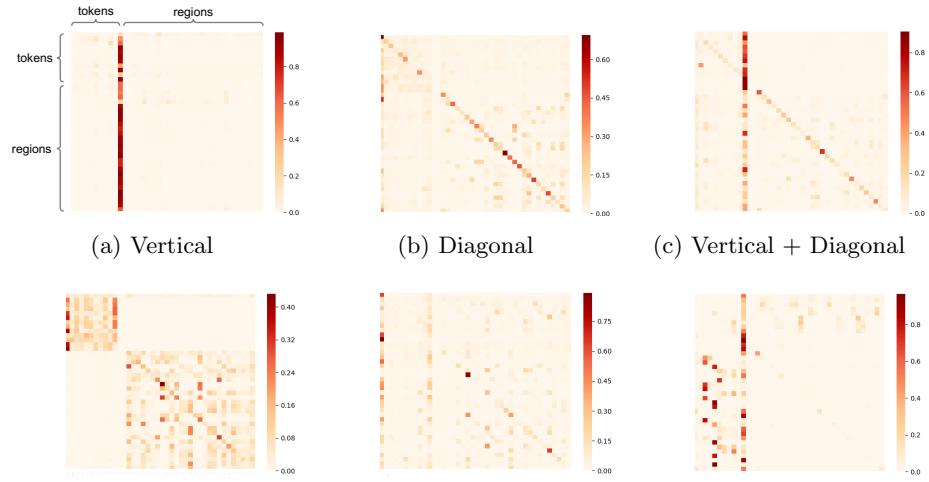
timal performance, as the interactions between paired images are not learned during the pre-training stage. Thus, we experimented with three modified settings on NLVR^2: (*i*) *Triplet*: joint embedding of images pairs and query captions; (*ii*) *Pair*: individual embedding of each image and each query caption; and (*iii*) *Pair-biattn*: a bidirectional attention is added to the *Pair* model to learn the interactions between the paired images.

Comparison results are presented in Table 5. The *Pair* setting achieves better performance than the *Triplet* setting even without cross-attention between the image pairs. We hypothesize that it is due to the fact that our UNITER is pre-trained with image-text pairs. Thus, it is difficult to finetune a pair-based pre-trained model on triplet input. The bidirectional attention mechanism in the *Pair-biattn* setting, however, compensates the lack of cross-attention between images, hence yielding the best performance with a large margin. This show that with minimal surgery on the top layer of UNITER, our pre-trained model can adapt to new tasks that are very different from pre-training tasks.

4.4 Visualization

Similar to [20], we observe several patterns in the attention maps of the UNITER model, as shown in Fig. 2. Note that different from [20], our attention mechanism operates in both inter- and intra-modalitiy manners. For completeness, we briefly discuss each pattern here:

- *Vertical*: attention to special tokens `[CLS]` or `[SEP]`;
- *Diagonal*: attention to the token/region itself or preceding/following tokens/regions;
- *Vertical + Diagonal*: mixture of vertical and diagonal;
- *Block*: intra-modality attention, *i.e.*, textual self-attention and visual self-attention;
- *Heterogeneous*: diverse attentions that cannot be categorized and is highly dependent on actual input;
- *Reversed Block*: inter-modality attention, *i.e.*, text-to-image and image-to-text attention.



(a) Vertical (b) Diagonal (c) Vertical + Diagonal

(d) Block (e) Heterogeneous (f) Reversed Block

Fig. 2: Visualization of the attention maps learned by the UNITER-base model

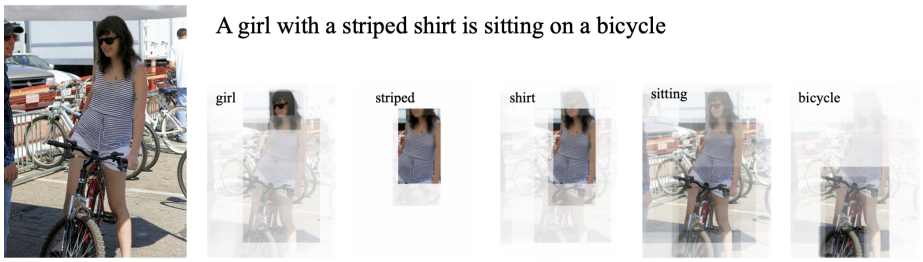


Fig. 3: Text-to-image attention visualization example

Note that *Reversed Block* (Fig. 2f) shows cross-modality alignment between tokens and regions. In Fig. 3, we visualize several examples of text-to-image attention to demonstrate the local cross-modality alignment between regions and tokens.

5 Conclusion

In this paper, we present UNITER, a large-scale pre-trained model providing UNiversal Image-TExt Representations for Vision-and-Language tasks. Four main pre-training tasks are proposed and evaluated through extensive ablation studies. Trained with both in-domain and out-of-domain datasets, UNITER outperforms state-of-the-art models over multiple V+L tasks by a significant margin. Future work includes studying early interaction between raw image pixels and sentence tokens, as well as developing more effective pre-training tasks.

References

1. Alberti, C., Ling, J., Collins, M., Reitter, D.: Fusion of detected objects in text for visual question answering. In: EMNLP (2019) 2, 4, 11
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018) 4, 18, 21
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual Question Answering. In: ICCV (2015) 1
4. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017) 25
5. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004) 25
6. Cao, J., Gan, Z., Cheng, Y., Yu, L., Chen, Y.C., Liu, J.: Behind the scene: Revealing the secrets of pre-trained vision-and-language models. arXiv preprint arXiv:2005.07310 (2020) 4
7. Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., Liu, J.: Graph optimal transport for cross-domain alignment. In: ICML (2020) 3
8. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NIPS (2013) 26
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019) 2, 3, 4, 10
10. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015) 3
11. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: EMNLP (2017) 1
12. Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. arXiv preprint arXiv:2006.06195 (2020) 4
13. Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S.C., Wang, X., Li, H.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In: CVPR (2019) 1
14. Genevay, A., Peyré, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: AISTATS (2018) 25
15. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018) 3
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 7
17. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015) 20, 21
18. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP (2014) 2
19. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. In: NeurIPS (2018) 1
20. Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the dark secrets of bert. In: EMNLP (2019) 13
21. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV (2017) 3, 7

22. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. In: ICLR (2020) 3
23. Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: ECCV (2018) 1, 3, 11, 21
24. Li, G., Duan, N., Fang, Y., Jiang, D., Zhou, M.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In: AAAI (2020) 2, 4, 11
25. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 (2019) 2, 4, 11, 24
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 3, 7
27. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 3
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 19
29. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: NeurIPS (2019) 2, 3, 11, 22, 24, 25
30. Lu, J., Goswami, V., Rohrbach, M., Parikh, D., Lee, S.: 12-in-1: Multi-task vision and language representation learning. In: CVPR (2020) 4
31. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016) 3
32. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2text: Describing images using 1 million captioned photographs. In: NeurIPS (2011) 3, 7, 8
33. Ott, M., Edunov, S., Grangier, D., Auli, M.: Scaling neural machine translation. WMT (2018) 19
34. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017) 19
35. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016) 3
36. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL (2018) 3
37. Peyré, G., Cuturi, M., et al.: Computational optimal transport. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019) 3, 6, 25
38. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV (2015) 8
39. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019) 3
40. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving GANs using optimal transport. In: ICLR (2018) 25
41. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018) 3, 7, 8, 25
42. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: Vl-bert: Pre-training of generic visual-linguistic representations. In: ICLR (2020) 2, 4, 11, 24, 25
43. Suhr, A., Artzi, Y.: Nlvr2 visual bias analysis. arXiv preprint arXiv:1909.10411 (2019) 24

44. Suhr, A., Zhou, S., Zhang, I., Bai, H., Artzi, Y.: A corpus for reasoning about natural language grounded in photographs. In: ACL (2019) 3, 11
45. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Contrastive bidirectional transformer for temporal representation learning. arXiv preprint arXiv:1906.05743 (2019) 3
46. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV (2019) 3
47. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP (2019) 2, 3, 8, 11, 22, 24
48. Trinh, T.H., Luong, M.T., Le, Q.V.: Selfie: Self-supervised pretraining for image embedding. arXiv preprint arXiv:1906.02940 (2019) 3
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 2, 3, 20
50. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: CVPR (2016) 2
51. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016) 4
52. Xie, N., Lai, F., Doran, D., Kadav, A.: Visual entailment: A novel task for fine-grained image understanding. arXiv preprint arXiv:1901.06706 (2019) 3, 11
53. Xie, Y., Wang, X., Wang, R., Zha, H.: A fast proximal point method for Wasserstein distance. In: arXiv:1802.04307 (2018) 6, 25
54. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: NeurIPS (2019) 3
55. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR (2018) 1, 9
56. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV (2016) 3
57. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: CVPR (2019) 11, 19
58. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: CVPR (2019) 3
59. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016) 3
60. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J.J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. In: AAAI (2020) 2, 4

A Appendix

This supplementary material has eight sections. Section A.1 describes the details of our dataset collection. Section A.2 describes our implementation details for each downstream task. Section A.3 provides detailed quantitative comparison between conditional masking and joint random masking. Section A.5 provides more results on VCR and NLVR^2. Section A.6 provides a direct comparison to VLBERT and ViLBERT. Section A.7 provides some background on optimal transport (OT) and the IPOT algorithm that is used to calculate the OT distance. Section A.8 provides additional visualization example.

A.1 Dataset Collection

As introduced, our full dataset is composed of four existing V+L datasets: COCO, Visual Genome, Conceptual Captions, and SBU Captions. The dataset collection is not simply combining them, as we need to make sure none of the downstream evaluation images are seen during pre-training. Among them, COCO is the most tricky one to clean, as several downstream tasks are built based on it. Figure 4 lists the splits from VQA, Image-Text Retrieval, COCO Captioning, RefCOCO/RefCOCO+/RefCOCOg, and the bottom-up top-down (BUTD) detection [2], all from COCO images.

As observed, the validation and test splits of different tasks are scattered across the raw COCO splits. Therefore, we exclude all those evaluation images that appeared in the downstream tasks. In addition, we also exclude all co-occurring Flickr30K images via URL matching, making sure the zero-shot image-text retrieval evaluation on Flickr is fair. The remaining images become the COCO subset within our full dataset, as shown in Figure 4 bottom row. We apply the same rules to Visual Genome, Conceptual Captions, and SBU Captions.

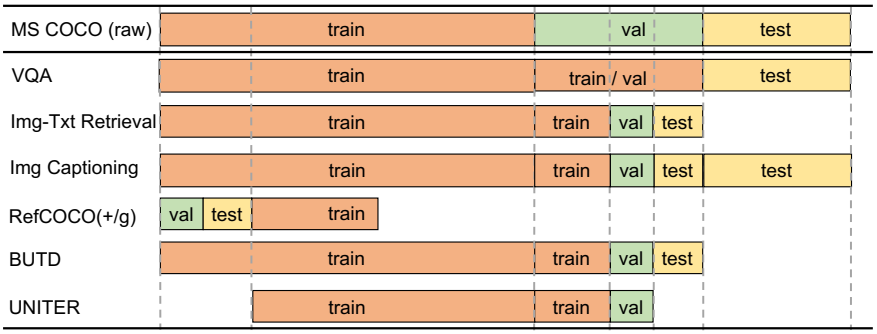


Fig. 4: Different data splits from downstream tasks based on COCO images. Our UNITER pre-training avoids seeing any downstream evaluation images

Task	Datasets	Image Src.	#Images	#Text	Metric
1 VQA	VQA	COCO	204K	1.1M	VQA-score
2 VCR	VCR	Movie Clips	110K	290K	Accuracy
3 NLVR^2	NLVR^2	Web Crawled	214K	107K	Accuracy
4 Visual Entailment	SNLI-VE	Flickr30K	31K	507K	Accuracy
5 Image-Text Retrieval	COCO	COCO	92K	460K	Recall@1,5,10
	Flickr30K	Flickr30K	32K	160K	
6 RE Comprehension	RefCOCO	COCO	20K	142K	Accuracy
	RefCOCO+		20K	142K	
	RefCOCOg		26K	95K	

Table 6: Statistics on the datasets of downstream tasks

A.2 Implementation Details

Our models are implemented based on PyTorch[17] [34]. To speed up training, we use Nvidia Apex[18] for mixed precision training. All pre-training experiments are run on Nvidia V100 GPUs (16GB VRAM; PCIe connection). Finetuning experiments are implemented on the same hardware or Titan RTX GPUs (24GB VRAM). To further speed up training, we implement dynamic sequence length to reduce padding and batch examples by number of input units (text tokens + image regions). For large pre-training experiments, we use Horovod[19] + NCCL[20] for multi-node communications (on TCP connections through ethernet) with up to 4 nodes of 4x V100 server. Gradient accumulation [33] is also applied to reduce multi-GPU communication overheads.

Visual Question Answering (VQA) We follow [57] to take 3129 most frequent answers as answer candidates, and assign a soft target score to each candidate based on its relevancy to the 10 human responses. To finetune on VQA dataset, we use a binary cross-entropy loss to train a multi-label classifier using batch size of 10240 input units over maximum 5K steps. We use AdamW optimizer [28] with a learning rate of $3e-4$ and weight decay of 0.01. At inference time, the max-probable answer is selected as the predicted answer. For results on `test-dev` and `test-std` splits, both training and validation sets are used for training, and additional question-answer pairs from Visual Genome are used for data augmentation as in [57].

Visual Commonsense Reasoning (VCR) VCR can be decomposed into two multiple-choice sub-tasks: question-answering task (Q $\rightarrow$ A) and answer-justification task (QA $\rightarrow$ R). In the holistic setting (Q $\rightarrow$ AR), a model needs to first choose an answer from the answer choices, then select a supporting rationale from rationale choices if the chosen answer is correct. We train our model in two settings simultaneously. When testing in the holistic setting, we first apply the model to predict an answer, then obtain the rationale from the same model based on the given question and the predicted answer. To finetune

[17] https://pytorch.org/
[18] https://github.com/NVIDIA/apex
[19] https://github.com/horovod/horovod
[20] https://github.com/NVIDIA/nccl

on VCR dataset, we concatenate the question (the qeustion and the ground truth answer) and each answer (rationale) choice from the four possible answer (rationale) candidates. The 'modality embedding' is extended to help distinguish question, answer and rationale. Cross-entropy loss is used to train a classifier over two classes (``right'' or ``wrong'') for each question-answer pair (question-answer-rationale triplet) with a batch size of 4096 input units over maximum 5K steps. We use AdamW optimizer with a learning rate of $1e-4$ and weight decay of 0.01.

Since the images and text in VCR dataset are very different from our pre-training dataset, we further pre-train our model on VCR, using MLM, MRFR and MRC-kl as the pre-training tasks. ITM is discarded because the text in VCR does not explicitly describe the image. The results of both pre-trainings on VCR are reported in Table 4 (in the main paper) and discussed in the main text. In conclusion, for downstream tasks that contain new data which is very different from the pre-training datasets, second-stage pre-training helps further boost the performance.

In our implementation, the second-stage pre-training is implemented with a batch size of 4096 intput units, a learning rate of $3e-4$ and a weight decay of 0.01 over maximum 60K steps. After second-stage pre-traing, we finetune our model with a learning rate of $6e-5$ over maximum 8K steps.

Natural Language for Visual Reasoning for Real (NLVR^2) NLVR2 is a new challenging task for visual reasoning. The goal is to determine whether a natural language statement is true about the given image pair. Here we discuss the three architecture variants of NLVR^2 finetuning in detail. Since UNITER only handles one image and one text input at pre-training, the 'modality embedding' is extended to help distinguish the additional image presented in the NLVR^2 task. For the *Triplet* setup, we concatenate the image regions and then feed into the UNITER model. An MLP transform is applied on the `[CLS]` output for binary classification. For the *Pair* setup, we treat one input example as two text-image pairs by repeating the text. The two `[CLS]` outputs from UNITER are then depth concatenated as the joint embedding for the example. Another MLP further transform this embedding for the final classification. For the *Pair-biattn* setup, the input format is the same as the Pair setup. As for the joint representation, instead of rely on only two `[CLS]` outputs, we apply a multi-head attention layer [49] on one sequence of joint image-text embeddings to attend to the other sequence of embeddings, and vice versa. After this 'bidirectional' attention interactions, a simple attentional pooling is applied on each output sequences and then a final concat+MLP layer transforms the cross-attended joint representation for true/false classification.

We finetune UNITER on NLVR^2 for 8K steps with a batch size of 10K input units. AdamW optimizer is used with learning rate of $1e-4$ and weight decay of 0.01.

Image-Text Retrieval Two datasets are considered for this task: COCO and Flickr30K. COCO consists of 123K images, each accompanied with five human-written captions. We follow [17] to split the data into 82K/5K/5K training/

validation/test images. Additional 30K images from MSCOCO validation set are also included to improve training as in [23]. Flickr30K dataset contains 31K images collected from the Flickr website, with five textual descriptions per image. We follow [17] to split the data into 30K/1K/1K training/validation/test splits. During finetuning, we sample two negative image-text pairs per positive sample from image and text sides, respectively. For COCO, we use batch size of 60 examples, learning rate of $2e-5$ and finetune our model for 20K steps. For Flickr30K, we finetune our model with a batch size of 120 examples and a learning rate of $5e-5$ over maximum 16K steps.

To obtain the final results in Table 3 in the main text, we further sample hard negatives to facilitate the finetuning. For every N steps, we randomly sample 128 negative images per text input and obtain a sparse scoring matrix for the whole training set. For each image, we choose the top 20 ranked negative sentences as hard negative samples. Similarly, we get 20 hard negative images for each sentence according to their scores. The hard negatives are sent to the model as additional negative samples. In the end, we have two randomly sampled negatives and two hard negative samples per positive sample. N is set to 4000 for COCO and 2500 for Flickr30K.

Visual Entailment (SNLI-VE) Visual Entailment is a task derived from Flickr30K images and Stanford Natural Language Inference (SNLI) dataset, where the goal is to determine the logical relationship between a natural language statement and an image. Similar to BERT for Natural Language Inference (NLI), we treat SNLI-VE as a three-way classification problem and apply an MLP Transform on `[CLS]` output. The UNITER model is finetuned using cross-entropy loss. The batch size is set to 10K input units and we use AdamW with learning rate of $8e-5$ to train for 3K steps.

Referring Expression Comprehension We use three referring expression datasets: RefCOCO, RefCOCO+, and RefCOCOg for the evaluation, all collected on COCO images. To finetune UNITER on this task, we add a MLP layer on top of the region outputs from Transformer, to compute the alignment score between the query phrase/sentence and each region. Since only one object is paired with the query phrase/sentence, we apply cross-entropy loss on the normalized alignment scores. The finetuning is efficient - we train the model with a batch size of 64 examples and a learning rate of $5e-5$ for only 5 epochs, and achieve state-of-the-art performance.

Note all works including ours use off-the-shelf object detectors trained on COCO (and Visual Genome) to extract the visual features. While this does not affect other downstream tasks, it raises an issue for RE comprehension, as the val/test images of RefCOCO, RefCOCO+, and RefCOCOg are a subset of COCO's training split. Strictly, our object detector is not allowed to train with these val/test images. However, just for a "fair" comparison with concurrent works, we ignore this issue and use the same features [2] as the others. We also update the results of MAttNet using this "contaminated" features, whose accuracy is 1.5% higher than the original one. As aforementioned, the interaction between sentence and image could start from tokens and pixels instead of the

extracted features. We leave this study and RE comprehension with strictly correct features to future work.

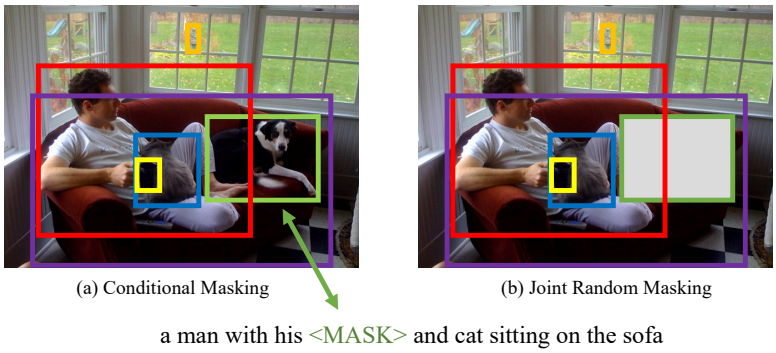


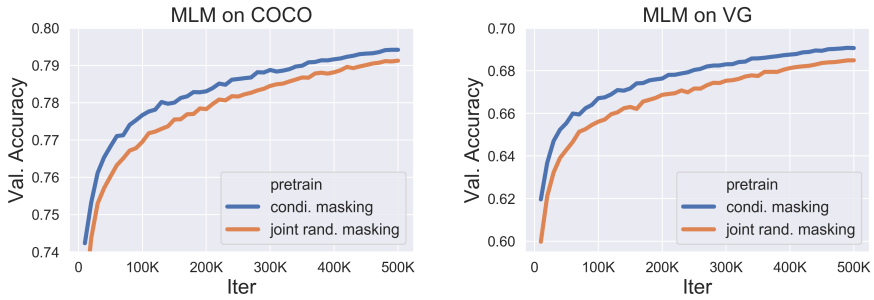
Fig. 5: Example showing difference between conditional masking and joint random masking

A.3 Conditional Masking vs. Joint Random Masking

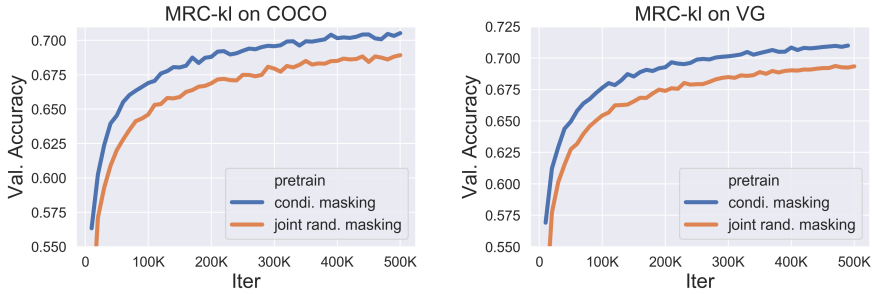
We further discuss the advantage of our proposed conditional masking over joint random masking used in [47,29]. Intuitively, our conditional masking learns better latent alignment of entities (regions and words) across two modalities. Fig. 5 shows an example image with "man with his dog and cat sitting on a sofa". With conditional masking, when the region of dog is masked, our model should be able to infer that the region is dog, based on the context of both surrounding regions and the full sentence (Fig. 5(a)), and vice versa. However, for the joint masking implementation, it could happen when both the region of dog and the word dog are masked (Fig. 5(b)). In such case, the model has to make the prediction blindly, which might lead to mis-alignment.

To verify this intuition, we show the validation curves during pre-training of MLM and MRC-kl in Fig. 6. Each sub-figure shows a comparison between applying conditional masking and joint random masking during the pre-training of UNITER. The MLM accuracy measures how well UNITER can reconstruct the masked words, and MRC-kl accuracy[21] measures how well UNITER can classify the masked regions. In both cases, as shown in Fig. 6, our conditional masking converges faster and achieves higher final accuracy than joint random masking. In addition, Table 2 (row 10 & 11) in the main paper shows our conditional masking also performs better on fine-tuned downstream tasks.

[21] When validating on MRC-kl accuracy, we simply pick the most confident category from the predicted probability and measure its correctness.



(a) Validation accuracy of MLM on COCO and VG datasets



(b) Validation accuracy of MRC-kl on COCO and VG datasets

Fig. 6: Comparison of MLM and MRC-kl validation accuracy using joint masking and our proposed conditional masking

A.4 More Ablation Studies on Pre-training Settings

MRC-only Pre-training In addition to ablations shown in Table 2 in the main paper, we include results from UNITER-base when pre-trained with MRC only on in-domain data. Table 7 shows that MRC-only pre-training leads to a similar downstream performance to MRFR-only prer-training, which is a weak baseline compared with all other pre-training settings with in-domain data (line 4 - 12 in Table 2).

Significance of WRA In Table 2 of the main paper, we show that adding WRA significantly improves model performance on VQA and RefCOCO+, while achieves comparable results on Flickr and NLVR^2. By design, WRA encourages local alignment between each image region and each word in a sentence. Therefore, WRA mostly benefits downstream tasks relying on region-level recognition and reasoning such as VQA, while Flickr and NLVR^2 focus more on global rather than local alignments. We add additional ablation results for WRA of UNITER-large when pre-trained with both In-domain and Out-of-domain data in Table 8. We observe large performance gains in zero-shot setup for image/text retrieval and consistent gains across all other tasks.

Pre-training Data	Pre-training Tasks	Meta-Sum	VQA	IR (Flickr)	TR (Flickr)	NLVR^2	Ref-COCO+
			test-dev	val	val	dev	val^d
In-domain (COCO+VG)	MRC	350.97	66.23	77.17	84.57	52.31	70.69

Table 7: Additional ablation results of MRC-only pre-training for UNITER-base with in-domain data.

WRA pre-train	VQA	NLVR^2	SNLI-VE	ZS IR (flickr)	ZS TR (flickr)	Ref-COCO	Ref-COCO+	Ref-COCOg
	test-std	test	test	val	val	testB^d	testB	test
N	73.40	79.50	78.98	65.82	77.50	74.17	78.89	87.73
Y	**74.02**	**79.98**	**79.38**	**68.74**	**83.60**	**74.98**	**79.75**	**88.47**

Table 8: A direct ablation on WRA pre-training task using UNITER-large, all pre-trained on both In-domain + Out-of-domain data, with MLM + ITM + MRC-kl + MRFR (+ WRA). For simplicity, only R@1 is reported for ZS IR and ZS TR.

Model	Q→A	QA→ R	Q → AR
VLBERT-large (single)	75.8	78.4	59.7
ViLBERT (10 ensemble)	76.4	78.0	59.8
UNITER-large (single)	77.3	80.8	62.8
UNITER-large (10 ensemble)	**79.8**	**83.4**	**66.8**

Table 9: VCR results from VLBERT [42], ViLBERT [29], and UNITER

Model	Balanced	Unbalanced	Overall	Consistency
VisualBERT	67.3	68.2	67.3	26.9
LXMERT	76.6	76.5	76.2	42.1
UNITER-large	**80.0**	**81.2**	**80.4**	**50.8**

Table 10: NLVR^2 results on test-U split from VisualBERT [25], LXMERT [47], and UNITER

A.5 More Results on VCR and NLVR2

Following the VCR setup in Table 4 of the main paper, we further construct an ensemble model using 10 UNITER-large. Table 9 shows the comparison between VLBERT, ViLBERT and UNITER on VCR. The $Q \rightarrow AR$ accuracy of our ensemble model outperforms ViLBERT [29] ensemble by a large margin of 7.0%. Note even single UNITER-large already outperforms ViLBERT ensemble and VLBERT-large by 3.0%.

Besides, we also compare our UNITER-large with LXMERT [47] and VisualBERT [25] on an additional testing split of NLVR^2 in Table 10. Our results consistently outperform the previous SOTA on all metrics[22] by a large margin of ∼4.0%.

[22] The balanced and unbalanced evaluations were introduced in [43].

Model	VQA	RefCOCO+ (det)		
	test-dev	val	testA	testB
ViLBERT	70.55	72.34	78.52	62.61
VLBERT-base	71.16	71.60	77.72	60.99
UNITER-base	**71.22**	**72.49**	**79.36**	**63.65**

Table 11: A direct comparison between ViLBERT [29], VLBERT [42], and our UNITER, all trained on Conceptual Captions [41] only

A.6 Direct Comparison to VLBERT and ViLBERT

To further demonstrate our idea, we conduct a direct comparison to ViLBERT [29] and VLBERT [42], trained on Conceptual Captions [41]. We pre-train UNITER on Conceptual Captions only using our proposed conditional masking and the best pre-training tasks. Table 11 shows that UNITER still consistently outperforms the other models by a visible margin on VQA and RefCOCO+.

A.7 Review of Optimal Transport and the IPOT Algorithm

Optimal Transport We first provide a brief review of optimal transport, which defines distances between probability measures on a domain $\mathbb{X}$ (the sequence space in our setting). The *optimal transport distance* for two probability measures μ and ν is defined as [37]:

$$\mathcal{D}_c(\mu,\nu) = \inf_{\gamma\in\Pi(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}\left[c(\mathbf{x},\mathbf{y})\right], \tag{5}$$

where $\Pi(\mu,\nu)$ denotes the set of all joint distributions $\gamma(\mathbf{x},\mathbf{y})$ with marginals $\mu(\mathbf{x})$ and $\nu(\mathbf{y})$; $c(\mathbf{x},\mathbf{y}):\mathbb{X}\times\mathbb{X}\rightarrow\mathbb{R}$ is the cost function for moving $\mathbf{x}$ to $\mathbf{y}$, *e.g.*, the Euclidean or cosine distance. Intuitively, the optimal transport distance is the minimum cost that γ induces in order to transport from μ to ν. When $c(\mathbf{x},\mathbf{y})$ is a metric on $\mathbb{X}$, $\mathcal{D}_c(\mu,\nu)$ induces a proper metric on the space of probability distributions supported on $\mathbb{X}$, commonly known as the Wasserstein distance. One of the most popular choices is the 2−Wasserstein distance $W_2^2(\mu,\nu)$ where the squared Euclidean distance $c(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$ is used as cost.

The IPOT algorithm Unfortunately, the exact minimization over $\mathbf{T}$ is in general computational intractable [4,14,40]. To overcome such intractability, we consider an efficient iterative approach to approximate the OT distance. We propose to use the recently introduced Inexact Proximal point method for Optimal Transport (IPOT) algorithm to compute the OT matrix $\mathbf{T}^*$, thus also the OT distance [53]. Specifically, IPOT iteratively solves the following optimization problem using the proximal point method [5]:

$$\mathbf{T}^{(t+1)} = \underset{\mathbf{T}\in\Pi(\mathbf{a},\mathbf{b})}{\arg\min} \left\{ \langle \mathbf{T}, \mathbf{C}\rangle + \beta\cdot\mathcal{B}(\mathbf{T},\mathbf{T}^{(t)}) \right\}, \tag{6}$$

where the proximity metric term $\mathcal{B}(\mathbf{T},\mathbf{T}^{(t)})$ penalizes solutions that are too distant from the latest approximation, and $\frac{1}{\beta}$ is understood as the generalized

Algorithm 1 IPOT algorithm

Input: Feature vectors $\mathbf{S} = \{\mathbf{w}_i\}_{i=1}^{n}$, $\mathbf{S}' = \{\mathbf{v}_j\}_{j=1}^{m}$ and generalized stepsize $1/\beta$,
$\boldsymbol{\sigma} = \frac{1}{m}\mathbf{1_m}$, $\mathbf{T}^{(1)} = \mathbf{1_n}\mathbf{1_m}^\top$
$\mathbf{C}_{ij} = c(\mathbf{w}_i, \mathbf{v}_j)$, $\mathbf{A}_{ij} = \mathrm{e}^{-\frac{\mathbf{C}_{ij}}{\beta}}$
for $t = 1, 2, 3 \ldots$ **do**
  $\mathbf{Q} = \mathbf{A} \odot \mathbf{T}^{(t)}$ // $\odot$ is Hadamard product
  for $k = 1, \ldots K$ **do** // $K = 1$ in practice
    $\boldsymbol{\delta} = \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} = \frac{1}{m\mathbf{Q}^\top\boldsymbol{\delta}}$
  end for
  $\mathbf{T}^{(t+1)} = \mathrm{diag}(\boldsymbol{\delta})\mathbf{Q}\mathrm{diag}(\boldsymbol{\sigma})$
end for
Return $\langle \mathbf{T}, \mathbf{C} \rangle$

stepsize. This renders a tractable iterative scheme towards the exact OT solution. In this work, we employ the generalized KL Bregman divergence $\mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) = \sum_{i,j} \mathbf{T}_{ij} \log \frac{\mathbf{T}_{ij}}{\mathbf{T}_{ij}^{(t)}} - \sum_{i,j} \mathbf{T}_{ij} + \sum_{i,j} \mathbf{T}_{ij}^{(t)}$ as the proximity metric. Algorithm 1 describes the implementation details for IPOT.

Note that the Sinkhorn algorithm [8] can also be used to compute the OT matrix. Specifically, the Sinkhorn algorithm tries to solve the entropy regularized optimization problem: $\hat{\mathcal{L}}_{\mathrm{ot}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{T}, \mathbf{C} \rangle - \frac{1}{\epsilon} H(\mathbf{T})$, where $H(\mathbf{T}) = -\sum_{i,j} \mathbf{T}_{ij}(\log(\mathbf{T}_{ij}) - 1)$ is the entropy regularization term and $\epsilon > 0$ is the regularization strength. However, in our experiments, we empirically found that the numerical stability and performance of the Sinkhorn algorithm is quite sensitive to the choice of the hyper-parameter ϵ, thus only IPOT is considered in our model training.

A.8 Additional Visualization

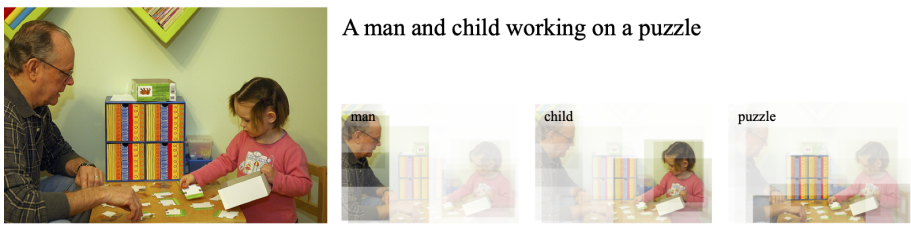


Fig. 7: Additional text-to-image attention visualization example