Supplementary Material
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

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 1 lists the splits from VQA, Image-Text Retrieval, COCO Captioning, RefCOCO/RefCOCO+/RefCOCOg, and the bottom-up top-down (BUTD) detection [1], 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 1 bottom row. We apply the same rules to Visual Genome, Conceptual Captions, and SBU Captions.

* Equal contribution.

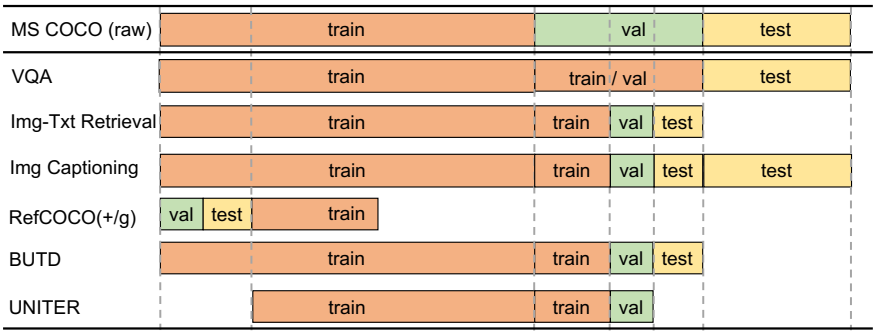


Fig. 1: Different data splits from downstream tasks based on COCO images. Our UNITER pre-training avoids seeing any downstream evaluation images

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 1: Statistics on the datasets of downstream tasks

A.2 Implementation Details

Our models are implemented based on PyTorch[1] [12]. To speed up training, we use Nvidia Apex[2] 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[3] + NCCL[4] for multi-node communications (on TCP connections through ethernet) with up to 4 nodes of 4x V100 server. Gradient accumulation [11] is also applied to reduce multi-GPU communication overheads.

Visual Question Answering (VQA) We follow [21] 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

[1] https://pytorch.org/
[2] https://github.com/NVIDIA/apex
[3] https://github.com/horovod/horovod
[4] https://github.com/NVIDIA/nccl

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 [9] 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 [21].

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 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 [19] 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 [6] 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 [7]. Flickr30K dataset contains 31K images collected from the Flickr website, with five textual descriptions per image. We follow [6] 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 [1] 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.

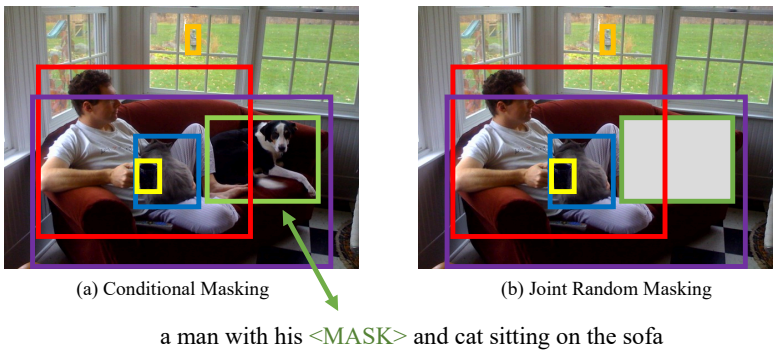
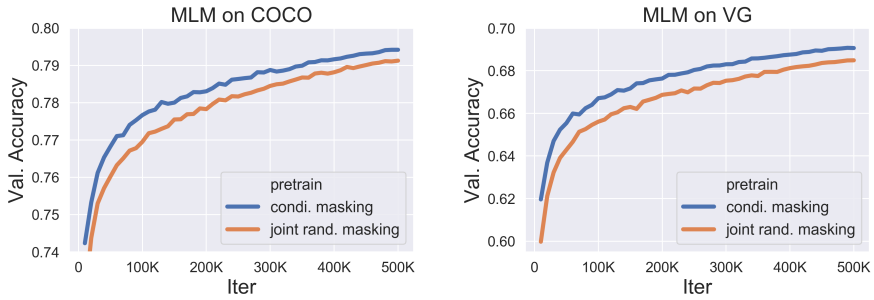


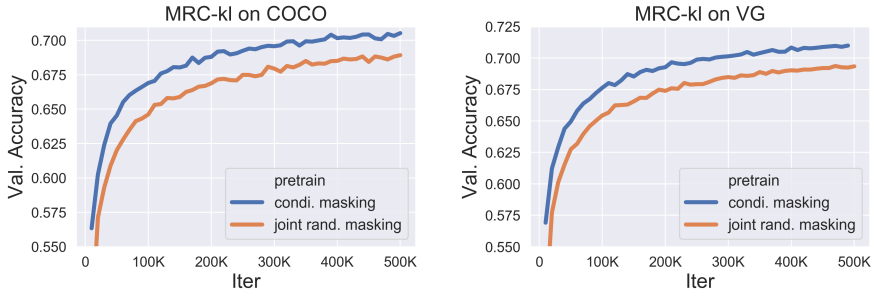
Fig. 2: Example showing difference between conditional masking and joint random masking

A.3 Conditional Masking vs. Joint Random Masking

We further discuss the advantage of our proposed conditional masking over joint random masking used in [18,10]. Intuitively, our conditional masking learns better latent alignment of entities (regions and words) across two modalities. Fig. 2 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. 2(a)), and vice versa. However, for the joint



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



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

Fig. 3: Comparison of MLM and MRC-kl validation accuracy using joint masking and our proposed conditional masking

masking implementation, it could happen when both the region of "dog" and the word "dog" are masked (Fig. 2(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. 3. 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[5] measures how well UNITER can classify the masked regions. In both cases, as shown in Fig. 3, 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.

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

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

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 2: 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 3: 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.

only on in-domain data. Table 2 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 3. We observe large performance gains in zero-shot setup for image/text retrieval and consistent gains across all other tasks.

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 4 shows the comparison between VLBERT, ViLBERT and UNITER on VCR. The $Q \rightarrow AR$ accuracy of our ensemble model outperforms ViLBERT [10] 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 [18] and VisualBERT [8] on an additional testing split of NLVR^2 in Table 5. Our results

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 4: VCR results from VLBERT [16], ViLBERT [10], 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 5: NLVR^2 results on test-U split from VisualBERT [8], LXMERT [18], and UNITER

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 6: A direct comparison between ViLBERT [10], VLBERT [16], and our UNITER, all trained on Conceptual Captions [15] only

consistently outperform the previous SOTA on all metrics[6] by a large margin of ∼4.0%.

A.6 Direct Comparison to VLBERT and ViLBERT

To further demonstrate our idea, we conduct a direct comparison to ViLBERT [10] and VLBERT [16], trained on Conceptual Captions [15]. We pre-train UNITER on Conceptual Captions only using our proposed conditional masking and the best pre-training tasks. Table 6 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 [13]:

$$\mathcal{D}_c(\mu,\nu) = \inf_{\gamma\in\Pi(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}\left[c(\mathbf{x},\mathbf{y})\right], \tag{1}$$

[6] The balanced and unbalanced evaluations were introduced in [17].

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 $\langle \mathbf{T}, \mathbf{C} \rangle$

where $\Pi(\mu, \nu)$ denotes the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals $\mu(\mathbf{x})$ and $\nu(\mathbf{y})$; $c(\mathbf{x}, \mathbf{y}) : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ 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 [2,5,14]. 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 [20]. Specifically, IPOT iteratively solves the following optimization problem using the proximal point method [3]:

$$\mathbf{T}^{(t+1)} = \underset{\mathbf{T} \in \Pi(\mathbf{a}, \mathbf{b})}{\arg\min} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle + \beta \cdot \mathcal{B}(\mathbf{T}, \mathbf{T}^{(t)}) \right\}, \tag{2}$$

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 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 [4] 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

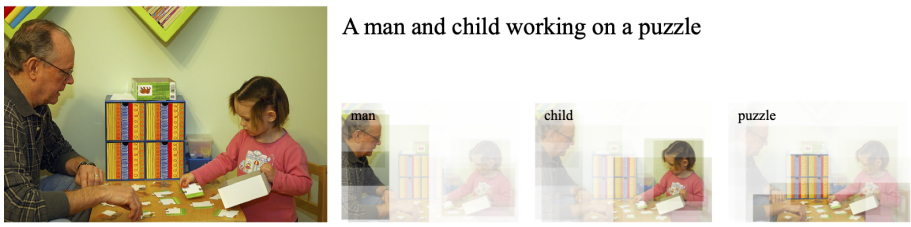


Fig. 4: Additional text-to-image attention visualization example

References

1. 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) 1, 5
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017) 9
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004) 9
4. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NIPS (2013) 9
5. Genevay, A., Peyré, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: AISTATS (2018) 9
6. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015) 4
7. Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: ECCV (2018) 4
8. 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) 7, 8
9. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 3
10. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: NeurIPS (2019) 5, 7, 8
11. Ott, M., Edunov, S., Grangier, D., Auli, M.: Scaling neural machine translation. WMT (2018) 2
12. 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) 2

13. Peyré, G., Cuturi, M., et al.: Computational optimal transport. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019) 8
14. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving GANs using optimal transport. In: ICLR (2018) 9
15. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018) 8
16. 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) 8
17. Suhr, A., Artzi, Y.: Nlvr2 visual bias analysis. arXiv preprint arXiv:1909.10411 (2019) 8
18. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP (2019) 5, 7, 8
19. 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) 4
20. Xie, Y., Wang, X., Wang, R., Zha, H.: A fast proximal point method for Wasserstein distance. In: arXiv:1802.04307 (2018) 9
21. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: CVPR (2019) 2, 3