LightningDOT: Pre-training Visual-Semantic Embeddings for Real-Time Image-Text Retrieval

Siqi Sun[*], **Yen-Chun Chen**[*], **Linjie Li, Shuohang Wang, Yuwei Fang, Jingjing Liu**
Microsoft Corporation
{siqi.sun, yen-chun.chen, lindsey.li, shuohang.wang, yuwfan, jingjl}@microsoft.com

Abstract

Multimodal pre-training has propelled great advancement in vision-and-language research. These large-scale pre-trained models, although successful, fatefully suffer from slow inference speed due to enormous computation cost mainly from cross-modal attention in Transformer architecture. When applied to real-life applications, such latency and computation demand severely deter the practical use of pre-trained models. In this paper, we study Image-text retrieval (ITR), the most mature scenario of V+L application, which has been widely studied even prior to the emergence of recent pre-trained models. We propose a simple yet highly effective approach, LightningDOT that accelerates the inference time of ITR by thousands of times, without sacrificing accuracy. LightningDOT removes the time-consuming cross-modal attention by pre-training on three novel learning objectives, extracting feature indexes offline, and employing instant dot-product matching with further re-ranking, which significantly speeds up retrieval process. In fact, LightningDOT achieves new state of the art across multiple ITR benchmarks such as Flickr30k, COCO and Multi30K, outperforming existing pre-trained models that consume 1000× magnitude of computational hours.[1]

1 Introduction

Image-text retrieval (ITR) has been widely studied as a staple benchmark task in both NLP and computer vision communities. Traditional ITR search engines typically deploy ranking-based models built upon visual-semantic embedding matching (Faghri et al., 2017; Huang et al., 2018) or deep cross-modal fusion with attention mechanism (Lee et al., 2018; Li et al., 2020a,b). Earliest works (Kiros et al., 2014; Faghri et al., 2017; Wang et al., 2018) employ separate image encoder (*e.g.,* CNN) and text encoder (*e.g.,* RNN), the embeddings from which are then measured by doc product for similarity matching (Figure 1(a)). Later studies (Lee et al., 2018, 2019; Wang et al., 2019; Zhang et al., 2020) improve this paradigm by employing advanced region-level visual encoder (*e.g.,* Faster-RCNN) and applying cross-attention between word features and region features for multimodal fusion (Figure 1(b)).

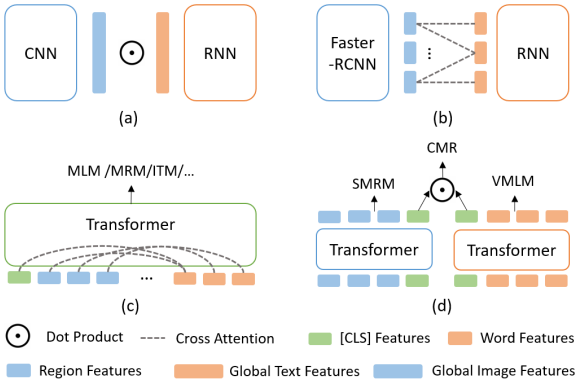

Figure 1: Evolution of Image-Text Retrieval (ITR) paradigm. (a) Early work (Faghri et al., 2017) using dot product to learn the similarity between global image features and global text features. (b) Later study (Lee et al., 2018) applying cross-attention between the features of each region and each word. (c) Pre-trained V+L models (Chen et al., 2020) with deep Transformer. (d) LightningDOT without cross-attention.. CMR, SMRM and VMLM refer to different pre-training tasks, which will be introduced later in method section.

With the advent of Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019), cross-modal retrieval tasks are more recently dominated by vision-and-language (V+L) pre-trained models, such as ViLBERT (Lu et al., 2019), UNITER (Chen et al., 2020), OSCAR (Li et al., 2020b), and VILLA (Gan et al., 2020). Large-scale pre-trained models learned from massive corpus of image-text pairs can power heterogeneous downstream tasks that take diverse modalities as inputs (*e.g.,* text, image, video, audio). These models benefit from the self-attention mechanism in Transformer ar-

[*] Equal Contribution.
[1] Code and pre-training checkpoints are available at https://github.com/intersun/LightningDOT.

chitecture, learning joint image+text embeddings through pre-training objectives such as masked language modeling (MLM) and masked region modeling (MRM) (Figure 1(c)).

However, the very ingredient that engenders the success of these pre-trained models, *cross-modal attention* between two modalities (through self-attention), also destines the inevitable latency and huge computation cost in training and deploying such massive-scale models. For example, UNITER (Chen et al., 2020) builds upon 12/24 Transformer layers, and trains over 10 million image+text pairs. The inference time of such large models with 110 million parameters is 48 seconds on average for text query from COCO dataset (Chen et al., 2015), not scalable in real-life applications serving millions of queries per second.

To make real-time ITR possible with low latency, we ask a bold question: can we go back to the beginning, reverting to simple dot product for efficient cross-modal retrieval? To make this retro experiment feasible, we rely on Transformer to pre-train high-quality image and text encoders, but use efficient dot product for multimodal fusion instead of computationally heavy self-attention. To still facilitate effective cross-modal embedding learning, we use a special `[CLS]` token on both encoders, which transfers the learned embedding from the other modality (Figure 1(d)). We name this new paradigm *LightningDOT*, for its lightening speed benefiting from dot product computation.

By removing the time-consuming cross-attention between modalities, the model can learn visual-semantic embeddings without extensive matching between each image-text pair during inference, as used in existing pre-trained models (Chen et al., 2020; Li et al., 2020b; Lu et al., 2019). Further, by eliminating the dependency on real-time computation over image-text pairs, we can compute all image and text embeddings independently offline just for once, and reuse these embeddings as cached indexes for new queries on the fly (Figure 2).

For model training, we propose three learning objectives to jointly train two Transformer blocks: Image Encoder and Language Encoder. Specifically, Visual-embedding fused MLM (namely *VMLM*) and Semantic-embedding fused MRM (namely *SMRM*) ensure cross-modal information is harnessed even without cross-modality self-attention. A cross-modal retrieval objective (namely *CMR*) encourages the model to learn multimodal fusion through pre-training. To maintain competitive model performance, we further introduce a re-ranking mechanism to bring back the benefit of cross-attention methods.

In summary, LightningDOT is designed with late fusion to learn visual-semantic embeddings. Experiments on popular ITR benchmarks show that LightningDOT is 600/1900 times faster than existing pre-trained models on Flickr30k/COCO, while achieving new state-of-the-art results. When retrieving from larger candidate pool (>120K images), LightningDOT is *23,000* times faster. To the best of our knowledge, this is the first known effort on improving V+L model efficiency.

2 Related Work

V+L Pre-training Inspired by the success of Transformer-based (Vaswani et al., 2017) language model pre-training (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Raffel et al., 2020; Lan et al., 2020; Clark et al., 2020), vision-and-language pre-training (Huang et al., 2020b; Su et al., 2020; Li et al., 2020b, 2019a) has become the prevailing paradigm in learning multimodal representations, with strong results on tasks such as image-text retrieval (Kiros et al., 2014), visual question answering (Antol et al., 2015) and referring expression comprehension (Yu et al., 2016). Exemplary works include two-stream (Tan and Bansal, 2019; Lu et al., 2019) and single-stream models (Chen et al., 2020; Li et al., 2020a; Zhou et al., 2020). Multi-task learning (Lu et al., 2020) and adversarial training (Gan et al., 2020) are also explored. This family of pre-training methods aims for general-purpose V+L without computation cost consideration. To the best of our knowledge, our work is the first known effort on pre-training visual-semantic embedding that enables low-latency real-time cross-modal retrieval. Ours is concurrent work with CLIP (Radford et al., 2021).

Image-Text Retrieval Early cross-modal embedding works (Kiros et al., 2014; Wang et al., 2018; Faghri et al., 2017) focus on using a two-stream model to learn a unified visual-semantic embedding, with progressive improvement on two popular benchmarks: Flickr30K (Plummer et al., 2015) and COCO (Chen et al., 2015). Later methods with cross-attention (Lee et al., 2018, 2019; Wang et al., 2019; Zhang et al., 2020) become more popular, with significant performance gain. Pre-trained V+L models also fall into this category.

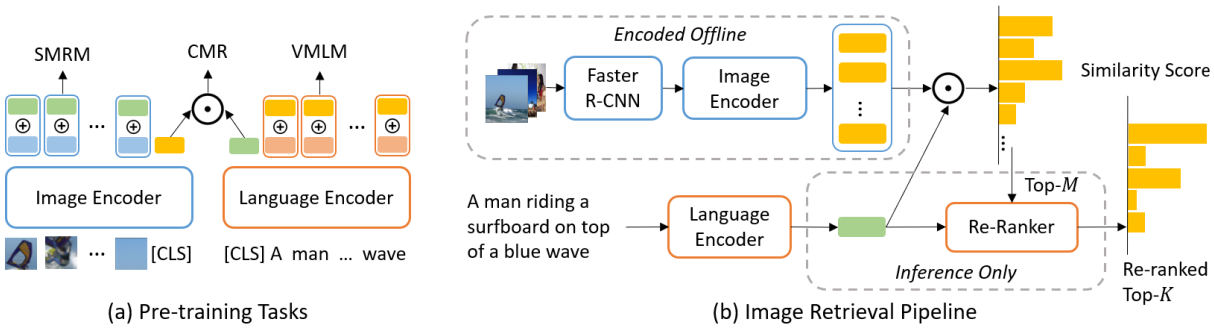

Figure 2: An overview of our proposed framework. (a) LightningDOT is pre-trained with Sementic-embedding Fused Mask Region Modeling (SMRM), Visual-embedding Fused Mask Language Modeling (VMLM) and Cross-modal Retrieval (CMR). (b) LightningDOT ITR pipeline (image retrieval as an example). Similarities between input textual query and image candidates are computed via dot product. During inference, image representations can be computed offline, and a re-ranker can be applied for better accuracy, still with significant speedup.

By exploiting large-scale image-text datasets, pre-trained V+L models further push the performance on Flickr30K and COCO. Although achieving high recall, cross-attention requires excessive computation cost during inference that cannot be overlooked.[2] In this work, inspired by dense retrieval in text retrieval domain (Guu et al., 2020; Karpukhin et al., 2020; Xiong et al., 2020; Mao et al., 2020; Lewis et al., 2020), we propose a more efficient attention-less framework. With pre-training, our model achieves better performance while being significantly faster than cross-modal attention methods. Note that the proposed approach is orthogonal to model compression techniques that reduce the number of layers/parameters (Sun et al., 2019; Jiao et al., 2020), since we do not reduce the number of parameters from the UNITER baseline. These two approaches can be combined to further boost the speed, which is an interesting future work direction.

3 LightningDOT Framework

In this section, we present the proposed LightningDOT framework, which consists of two deep Transformers as image and language encoders. We first introduce three tasks designed to pre-train the model, then present our inference pipeline from offline feature extraction to online instant retrieval.

3.1 Model Pre-training

We denote the Transformer-based (Vaswani et al., 2017) image encoder and language encoder by f_{θ_V} and f_{θ_L}, respectively (θ_V, θ_L are learnable parameters). Given a dataset of paired image and text $\{(i,t)\}$, we first extract region features $\mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_N\}$ ($\mathbf{v}_j \in \mathbb{R}^{d_v}$, N is the number of regions) for image i, along with bounding box positions of regions via a pre-trained Faster-RCNN (Ren et al., 2015; Anderson et al., 2018).[3] The image encoder f_{θ_V} encodes this sequence of image regions into a d-dimensional space $f_{\theta_V}(\mathbf{v}) = \mathbf{h} = \{\mathbf{h}_0, \ldots, \mathbf{h}_N\}$ ($\mathbf{h}_j \in \mathbb{R}^d$). The corresponding text t is tokenized into sub-word units and projected into high-dimensional feature vectors $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_T\}$ ($\mathbf{w}_j \in \mathbb{R}^{d_w}$, T is the number of tokens) following Devlin et al. (2019).[4] Similarly, the text encoding process can be written as $f_{\theta_L}(\mathbf{w}) = \mathbf{z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_T\}$ ($\mathbf{z}_j \in \mathbb{R}^d$). We regard the output `[CLS]` embedding $\mathbf{h}_0$ as global image representation, and $\mathbf{z}_0$ as global text representation. Following sections discuss how to jointly train these two encoders to learn strong visual-semantic embeddings, through three pre-training objectives.

Visual-embedding Fused Masked Language Modeling (VMLM) Masked Language Modeling (MLM) pre-training is first proposed by Devlin et al. (2019), where 15% of the words are masked[5] and the model is trained to reconstruct the masked words. Formally, we denote $\mathbf{w_m} = \{\mathbf{w}_{\mathbf{m}_1}, \ldots, \mathbf{w}_{\mathbf{m}_M}\}$ as masked tokens, where $\mathbf{m} \in \mathbb{N}^M$ is the set of masked indices of size M, randomly sampled from a natural number $\mathbb{N}$. $\mathbf{w}_{\setminus\mathbf{m}}$ are the unmasked words. MLM can be optimized by

[2] The total inference time is quadratic to the dataset size with cross-attention for image-text retrieval task.

[3] $\mathbf{v}_0$ is a special `[CLS]` embedding.

[4] A 30k BPE (Sennrich et al., 2016) vocabulary (bert-base-cased) is used to tokenize the text. A special `[CLS]` token is also prepended following the common practice ($\mathbf{w}_0$).

[5] In practice, this 15% is further decomposed into 10% random words, 10% unchanged, and 80% `[MASK]`.

minimizing the negative log-likelihood:

$$\begin{aligned}\mathcal{L}_{\text{MLM}}(t) &= -\log P_{\theta_L}(\mathbf{w}_\mathbf{m}|\mathbf{w}_{\backslash\mathbf{m}}) \\ &= -\frac{1}{M}\sum_{k=1}^{M}\log P_{\theta_{\text{mlm}}}(\mathbf{w}_{\mathbf{m}_k}|\mathbf{z}_{\mathbf{m}_k})\,,\end{aligned} \tag{1}$$

where θ_{mlm} is the additional parameters introduced to map hidden states $\mathbf{z}$ to word probabilities.

Under the V+L setting, the textual input is usually highly correlated with the image. To leverage this cross-modal relation, we propose visual-embedding fused MLM (VMLM), in which the paired image i is considered as additional input when training the model to reconstruct masked tokens in sentence t. The loss function of VMLM can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{VMLM}}(t,i) &= -\log P_\theta(\mathbf{w}_\mathbf{m}|\mathbf{w}_{\backslash\mathbf{m}},\, i) \\ &= -\frac{1}{M}\sum_{k=1}^{M}\log P_{\theta_{\text{mlm}}}(\mathbf{w}_{\mathbf{m}_k}|\mathbf{z}_{\mathbf{m}_k}+\mathbf{h}_0)\,,\end{aligned} \tag{2}$$

where $\theta = \{\theta_V, \theta_L\}$ and the word probabilities P_θ are conditioned on the corresponding image i via the global image representation $\mathbf{h}_0$. Although VMLM takes a similar mathematical form to the MLM task proposed in UNITER, they differ in two main aspects: 1) LightningDOT uses two separate encoders ($\mathbf{h}_0$ is computed by f_{θ_V}); and 2) visual dependency is explicitly injected to text representations ($\mathbf{z}_{\mathbf{m}_k} + \mathbf{h}_0$), instead of implicitly learned through cross-modal attention.

Semantic-embedding Fused Masked Region Modeling (SMRM) Recent works on V+L pre-training (Lu et al., 2019; Tan and Bansal, 2019) have shown that *mask-then-reconstruct* pre-training on image regions also helps image+text embedding learning. Similar to MLM, Masked Region Modeling (MRM) is supervised by:

$$\begin{aligned}\mathcal{L}_{\text{MRM}}(i) &= \mathcal{D}_{\theta_{\text{mrm}}}(\mathbf{v}_\mathbf{m},\, f_{\theta_V}(\mathbf{v}_{\backslash\mathbf{m}})) \\ &= \frac{1}{M}\sum_{k=1}^{M}\mathcal{D}_{\theta_{\text{mrm}}}(\mathbf{v}_{\mathbf{m}_k}, \mathbf{h}_{\mathbf{m}_k})\,,\end{aligned} \tag{3}$$

where $\mathcal{D}$ can be any differentiable distance function. Among the variants of MRM, we consider Masked Region Feature Regression (MRFR) with L2 distance and Masked Region Classification with KL-Divergence (MRC-kl), due to their proven success in learning V+L representations (Chen et al., 2020).[6] In MRFR, the L_2 distance between two feature vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$\mathcal{D}_{\theta_{\text{fr}}}(\mathbf{x},\mathbf{y}) = \sum_k \|\mathbf{x}_k - g_{\theta_{\text{fr}}}(\mathbf{y}_k)\|_2^2\,,$$

where $\|\cdot\|_2$ denotes L_2-norm, and $g_{\theta_{\text{fr}}}(\cdot)$ is a learnable Multi-layer Perceptron (MLP) with parameters θ_{fr}. The KL-divergence $\mathcal{D}_{\text{KL}}$ in MRC-kl measures distance between two probability distributions:

$$\mathcal{D}_{\theta_{\text{mrc}}}(\mathbf{x},\mathbf{y}) = \sum_k \mathcal{D}_{\text{KL}}(c(\mathbf{x}_k)\,||\,g_{\theta_{\text{mrc}}}(\mathbf{y}_k))\,,$$

where θ_{mrc} is the parameters of a trainable MLP that maps feature vector $\mathbf{x}_k$ to the object class distribution $c(\mathbf{x}_k)$ predicted by Faster R-CNN.

To incorporate language information encoded in the paired text, we extend MRM to Semantic-embedding fused MRM (SMRM), where the global text representation $\mathbf{z}_0$ is exploited when reconstructing masked regions.

$$\begin{aligned}\mathcal{L}_{\text{SMRM}}(i,t) &= \mathcal{D}_{\theta_{\text{mrm}}}(\mathbf{v}_\mathbf{m}, f_{\theta_V}(\mathbf{v}_{\backslash\mathbf{m}}), t) \\ &= \frac{1}{M}\sum_{k=1}^{M}\mathcal{D}_{\theta_{\text{mrm}}}(\mathbf{v}_{\mathbf{m}_k}, \mathbf{h}_{\mathbf{m}_k}+\mathbf{z}_0)\,.\end{aligned} \tag{4}$$

The specific variants SMRFR and SMRC-kl can be derived using the corresponding distance function, which is omitted for simplicity. Note that both the cross-modal fusion introduced in Eqn. (2) and Eqn. (4) uses simple addition without introducing extra parameters from their uni-modal counterpart. Moreover, the extra parameters θ_{mlm} and θ_{mrm} is not needed at downstream inference so will not slow down the retrieval.

Cross-modal Retrieval Objective (CMR) Beyond image or text focused reconstructive objectives, we also propose a new pre-training task, Cross-modal Retrieval (CMR), to leverage the paired information between image and text. With this learning objective, the model is optimized to promote high similarity score for a matched image-sentence pair (i,t) and vice versa. The similarity score between query t and image i is defined as:

$$S(t,i) = \langle \mathbf{z}_0, \mathbf{h}_0 \rangle\,, \tag{5}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product between two vectors, and $\mathbf{h}_0$ and $\mathbf{z}_0$ are the output `[CLS]` embeddings from image encoder f_{θ_V} and language encoder f_{θ_L}, respectively.

[6] In our implementation, no textual inputs are directly concatenated with image regions due to separate encoding of image and text.

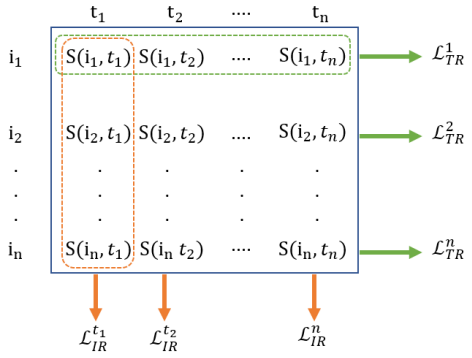

Figure 3: An illustration of the proposed CMR Loss. Note that positive pairs lie in the diagonal of the matrix.

In order to capture both image-retrieval and text-retrieval supervision signals in a single forward-backward pass, we propose a bi-directional variant of contrastive loss. Given any matched image-text pair (i, t), we treat text t as the query, sample $n-1$ negative images $\{i_2, i_3, \ldots, i_n\}$, and then compute the objective function as:

$$\mathcal{L}^{(t)}_{\text{IR}} = -\log \frac{e^{S(t,i_1)}}{\sum_{k=1}^{n} e^{S(t,i_k)}},$$

where $t_1 := t$. Similarly, we take image i as query ($i_1 := i$), sample $n-1$ negative text, and compute:

$$\mathcal{L}^{(i)}_{\text{TR}} = -\log \frac{e^{S(i,t_1)}}{\sum_{k=1}^{n} e^{S(i,t_k)}}$$

to optimize for text retrieval.

Following Henderson et al. (2017); Gillick et al. (2019); Karpukhin et al. (2020), we use in-batch negatives to avoid the actual sampling of a negative image or text: given a batch of n positive image-text pairs $B = \{(i_1, t_1), \ldots, (i_n, t_n)\}$, we use all other images from within the batch as negatives ($\{i_j\}$, where $j \in \{1, 2, \ldots, n\}$ and $j \neq k$) for every positive pair (i_k, t_k), and vice versa for negative text. The final CMR loss for batch B is:

$$\mathcal{L}_{\text{CMR}}(B) = \frac{1}{2n} \sum_{k=1}^{n} \mathcal{L}^{(i_k)}_{\text{TR}} + \mathcal{L}^{(t_k)}_{\text{IR}}. \quad (6)$$

An illustration of $\mathcal{L}_{\text{CMR}}$ is presented in Figure 3.[7] Through joint pre-training with CMR, VMLM and SMRM, the visual-semantic embeddings learned from image encoder and language encoder can be readily applied to downstream tasks. During fine-tuning stage, we directly adopt CMR loss to supervise the training process.

[7]The whole similarity matrix can be computed efficiently with one batched matrix multiplication call. This operation can take advantage of GPU hardware with Tensor Cores for faster training.

3.2 Real-time Inference

For simplicity, we take text-to-image retrieval as an example to introduce the real-time inference pipeline (Figure 2(b)): (i) Offline image feature extraction and encoding; (ii) Online retrieval with text query; and (iii) Online re-ranking with top-retrieved images. Text retrieval is conducted in a symmetric manner.

Offline Feature Extraction Image retrieval task requires the model to rank every image i in an image database I based on its similarity to a text query t. In LightningDOT, we first apply the image encoder f_{θ_V} to all images in I, and cache the resulting global image representations $\{\mathbf{h}^{(i)}_0 \in \mathbb{R}^d | i \in I\}$ into an index (Johnson et al., 2019) in memory for later use. Note that the entire image-to-index process, including Faster-RCNN feature extraction and Transformer encoding, can all be conducted offline. Therefore, for every new query t at real time, the cached index can be reused for maximum inference time saving.

Online Retrieval During inference, given a text query t, we encode it with the language encoder θ_L, and then compute its similarity score to the embedding of every image in I (stored in memory index) via Eqn (5). Finally, the images will be ranked by their similarity scores, from the highest to lowest. In practice, people are more interested in top-K retrieval, with a list of K images I_t satisfying:

$$I_t := \{i_{m_k}\}_{k=1}^{K}, \text{ where}$$
$$S(t, i_{m_1}) \geq S(t, i_{m_2}) \geq \cdots \geq S(t, i_{m_K}) \quad \text{and}$$
$$S(t, i_{m_K}) \geq S(t, i) \quad \forall i \in (I \setminus I_t). \quad (7)$$

This optimization problem has been well studied, and we use FAISS (Johnson et al., 2019) to solve it in our implementation. It is worth noting that in order to apply fast search, the similarity function has to be *decomposable*. Therefore, we choose the simple dot product as S instead of a more complicated neural network function. Similarly, for text retrieval, the same architecture can be applied by simply pre-computing the embedding for all sentences and using an image as query instead.

Re-ranking To further improve retrieval accuracy, we propose a two-stage approach by adopting an optional re-ranking model. In the first stage, we use LightningDOT to retrieve top-M images (or texts), where M is an integer much smaller

Model	COCO Test (5k images)							Flickr30K Test (1k images)						
	Text Retrieval			Image Retrieval				Text Retrieval			Image Retrieval			
	R@1	R@5	R@10	R@1	R@5	R@10	AR	R@1	R@5	R@10	R@1	R@5	R@10	AR
VSE++*	41.3	69.2	81.2	30.3	59.1	72.4	58.9	52.9	80.5	87.2	39.6	70.1	79.5	68.3
SCO*	42.8	72.3	83.0	33.1	62.9	75.5	61.6	55.5	82.0	89.3	41.1	70.5	81.1	69.9
GXN	42.0	-	84.7	31.7	-	74.6	-	56.8	-	89.6	41.5	-	80.0	-
SCAN-single	46.4	77.4	87.2	34.4	63.7	75.7	64.1	67.9	89.0	94.4	43.9	74.2	82.8	75.4
R-SCAN	45.4	77.9	87.9	36.2	65.6	76.7	65.0	66.3	90.6	96.0	51.4	77.8	84.9	77.8
CAMP	50.1	82.1	89.7	39.0	68.9	80.2	68.3	68.1	89.7	95.2	51.5	77.1	85.3	77.8
CAAN	52.5	83.3	90.9	41.2	70.3	82.9	70.2	70.1	91.6	97.2	52.8	79.0	87.9	79.8
ViLBERT	-	-	-	-	-	-	-	-	-	-	58.2	84.9	72.8	-
Unicoder-VL	62.3	87.1	92.8	46.7	76.0	85.3	75.0	86.2	86.3	99.0	71.5	90.9	94.9	88.1
UNITER-base	64.4	87.4	93.1	50.3	78.5	87.2	76.8	85.9	97.1	98.8	72.5	92.3	95.9	90.4
UNITER-large	65.7	88.6	93.8	52.9	79.9	88.0	78.1	86.9	98.1	**99.2**	75.5	**94.0**	**96.6**	91.7
OSCAR	73.5	92.2	**96.0**	**57.5**	**82.8**	89.8	82.0	-	-	-	-	-	-	-
LightningDOT*	60.1	85.1	91.8	45.8	74.6	83.8	73.5	83.9	97.2	98.6	69.9	91.1	95.2	89.3
+UNITERbase Re-Ranker	64.6	87.6	93.5	50.3	78.7	87.5	77.0	86.5	97.5	98.9	72.6	93.1	96.1	90.8
+UNITERlarge Re-Ranker	65.7	89.0	93.7	53.0	80.1	88.0	78.2	**87.2**	**98.3**	99.0	**75.6**	**94.0**	96.5	**91.8**
+OSCAR Re-Ranker	**74.2**	**92.4**	**96.0**	57.4	82.7	**89.9**	**82.1**	-	-	-	-	-	-	-

Table 1: Evaluation results on image-to-text and text-to-image retrieval over Flickr30k and COCO test sets. We compare the proposed method with both task-specific models: VSE++ (Faghri et al., 2017), GXN (Gu et al., 2018), SCO (Huang et al., 2018), SCAN (Lee et al., 2018), R-SCAN (Lee et al., 2019), CAMP (Wang et al., 2019) and CAAN (Zhang et al., 2020), and V+L pre-trained models: ViLBERT (Lu et al., 2019), Unicoder-VL (Li et al., 2020a), UNITER (Chen et al., 2020) and OSCAR (Li et al., 2020b). Models in ***bold**** are embedding-based methods without cross-attention.

than the database (index) size. Next, we apply a stronger retrieval model (usually slower due to the use of cross-attention) to re-rank the retrieved top-M pairs from the first stage. The final M similarity scores obtained from the second stage will be used to re-compute the desired top-K retrieval ($K \leq M$) in Eqn. (7). Please refer to figure 2 for a more detailed visualization. Our experiments show that this two-stage approach can benefit from the best of both worlds: maintaining a constant fast speed per query[8] while achieving state-of-the-art accuracy. Another advantage of this pipeline is that it can readily incorporate any advanced model as the re-ranker, thus future stronger image-text retrieval models can take advantage of LightningDOT for better efficiency.

4 Experiments

This section discusses our experiments on pre-training and evaluating LightningDOT on downstream ITR benchmarks.

4.1 Datasets and Metrics

For pre-training, we use pre-processed data provided by Chen et al. (2020), including 4.2 million images with 9.5 million associated captions from COCO (Chen et al., 2015), VG (Krishna et al., 2017), Conceptual Captions (Sharma et al., 2018), and SBU captions (Ordonez et al., 2011).

For evaluation, we use Flickr30k (Plummer et al., 2015) and COCO (Lin et al., 2014) datasets, which include 31K/123K images, respectively, each associated with 5 human-written captions. Following (Faghri et al., 2017), we split COCO into 114K/5K/5K and Flickr30K into 29K/1k/1k images for train, validation and test.

Downstream performance is measured by recall at K (R@K) for both image and text retrieval tasks. We also use an additional metric "AR", the average of R@K for all K across both image and sentence retrieval tasks.

4.2 Results on Flickr30K and COCO

We compare the proposed approach with state-of-the-art methods (with and without pre-training) and report the results in Table 1. Without cross-attention, our method outperforms non-pre-training approaches by large margins on all metrics. Specifically, our model improves over CAAN (Zhang et al., 2020) (SOTA method with cross-attention) by 3.3% (73.5 vs. 70.2) on COCO and 9.5% (89.3 vs. 79.8) on Flickr30K in terms of AR. When compared with methods without cross-attention (VSE++ (Faghri et al., 2017) and SCO (Huang et al., 2018)), LightningDOT achieves nearly

[8]The computation time of LightningDOT is negligible compared to that of UNITER. Therefore, the empirical speed is proportional to the number of pairs UNITER has to rank: constant M for LightningDOT + UNITER vs. the whole database (index) size for UNITER only.

Model	COCO Full (123K Images)							Flickr30K Full (31K Images)						
	Text Retrieval			Image Retrieval				Text Retrieval			Image Retrieval			
	R@5	R@10	R@20	R@5	R@10	R@20	AR	R@5	R@10	R@20	R@5	R@10	R@20	AR
LightningDOT	40.1	51.0	62.0	28.2	37.4	47.8	44.4	69.6	78.9	86.1	51.8	62.3	72.3	70.2
+ Re-Ranker-base	47.9	58.5	67.8	35.7	45.2	55.2	51.7	74.2	81.7	88.2	56.9	66.7	75.6	73.9
+ Re-Ranker-large	**48.0**	**59.0**	**68.9**	**37.3**	**46.8**	**56.4**	**52.7**	**75.1**	**83.9**	**90.5**	**60.1**	**69.5**	**78.3**	**76.2**

Table 2: Results on the extreme retrieval setting of full Flickr30k and full COCO datasets.

Method	#images	SCAN	Ours	+Re-ranker
Flickr30K-test	1,000	1.8×	639×	46×
COCO-test	5,000	1.9×	1,927×	95×
Flickr30K-full	31,014	1.8×	6,591×	1,255×
COCO-full	123,287	1.9×	23,869×	2,235×

Table 3: Speedup w.r.t. UNITER-base. We compare LightningDOT (Ours) and +Re-Ranker, plus a lightweight cross-attention method SCAN (Lee et al., 2018). LightningDOT with/without UNITER-base re-ranker is significantly faster.

20-point gain on AR. Although LightningDOT achieves slightly lower AR than UNITER (pre-training method with cross-attention), with 3.5/1.1 points drop on Flickr30K/COCO, it is 600/1900 × faster than UNITER during inference time.

We further apply second-stage re-ranking, and use UNITER to score top-M retrieved image-text pairs from LightningDOT to obtain the final top-K ranked lists. With re-ranking, LightningDOT achieves an instant performance lift, surpassing UNITER on both benchmarks, while still 46-95 times faster than UNITER. With an even stronger re-ranker OSCAR, LightningDOT achieves similar results to the state-of-the-art performance on COCO.

4.3 Speed & Space Improvement

To demonstrate the efficiency of LightningDOT, we use UNITER-base as baseline to compare inference speed. We also compare with a more lightweight cross-attention method SCAN (Lee et al., 2018), which uses GRU (Chung et al., 2014) instead of a 12-layer Transformer. All methods are tested on a single TITAN RTX GPU, with batch size of 400. As shown in Table 3, SCAN is ∼1.9× faster than UNITER-base across both benchmarks, as the computational cost of GRU is much cheaper than that of Transformer (performance drop is significant though). However, the speedup from SCAN is limited, as it computes cross-attention between each query and *all* images. On the other hand, LightningDOT is 639× faster than UNITER on Flickr30K. When tested with 5 times more images in COCO, the speedup from LightningDOT is 1927×. Even with re-ranking, LightningDOT is still much more efficient than UNITER-base (46× faster on Flickr30K and 95× faster on COCO).

To mimic a real-life scenario for image retrieval, where the candidate pool contains hundreds of thousands of images, we combine all images from training, validation and test set to form a larger candidate pool. Note that models are still trained on the training set. Although the number of text queries remain the same, the number of candidate images scales up by >20×, where cross-attention methods immediately become impractical. We refer this setting on both benchmarks as Flickr30k-full (31k) and COCO-full (123k). Our algorithm is 6,591× faster on Flickr30k-full and 23,869× faster on COCO-full, which clearly shows the advantage of LightningDOT and its potential in real-world applications. With re-ranking, LightningDOT is still more than 1,000× and 2,000× faster on Flickr30k-full and COCO-full, respectively. In general, for other re-rankers such as OSCAR, our algorithm can approximately speed up inference by N_{images}/M times, where N_{images} is the number of candidate images, and M is number of re-ranked images from top-M retrieved results by LightningDOT.

Similarly, we construct a full setting for text retrieval by combining all text queries from training, validation and test set. Results are summarized in Table 2. Considering the size of candidate pool has become more than 20× larger, we adopt recall at top 5, 10, 50 as evaluation metrics. Our method achieves reasonably good performance, with AR of 44.4 on COCO and 70.2 on Flickr30K. Re-ranking further lifts AR to 56.4 and 76.2. Results from UNITER or SCAN are not included as the computation of pairwise scores is extremely expensive, given the excessive amount of retrieval candidates. While LightningDOT only takes minutes to evaluate, UNITER-base is estimated to take about 28 days[9] to evaluate under the full setting for both

[9]This estimation is based on the inference time taken by

Method	Text Retrieval			Image Retrieval			
	R@1	R@5	R@10	R@1	R@5	R@10	AR
R-CNN only	62.2	85.9	91.1	42.0	70.9	80.3	72.1
+Image Encoder	73.4	92.5	95.6	59.5	84.5	90.3	82.6
+PT†	83.5	**96.4**	**98.7**	68.6	**90.5**	**94.8**	88.8
LightningDOT	**85.2**	**96.4**	**98.7**	**69.9**	90.4	94.5	**89.2**

Table 4: Ablation studies on model design over Flickr30K validation set. PT† indicates pre-training with MLM+MRM+CMR, while LightningDOT is pre-trained with VMLM+SMRM+CMR.

LightningDOT	Text Retrieval			Image Retrieval			
	R@1	R@5	R@10	R@1	R@5	R@10	AR
No PT	73.4	92.5	95.6	59.5	84.5	90.3	82.6
PT(CMR)	75.0	93.9	**97.3**	61.5	85.5	91.1	84.0
PT(All)	**78.1**	**94.0**	96.9	**62.6**	**85.7**	**91.8**	**84.8**

Table 5: Ablation studies on pre-training tasks over Flickr30K validation set after finetuning on the corresponding training set. All pre-training experiments are conducted on COCO dataset only. PT is short for pre-training. PT(CMR) refers to pre-training using CMR task only, and PT(All) refers to pre-training with all of the three tasks.

image retrieval and text retrieval.

In addition, We compare all models with the same setting: cache as much as possible for fastest speed, where our model outperforms others in both speed and space on image retrieval. The proposed algorithm maps each image to a 768-dimensional vector, which only consumes about 300Mb storage space for the whole COCO dataset. For cross-attention models such as SCAN, UNITER or OSCAR, they also need to cache image features, which typically requires to save a 36 x 2048 dimensional vector per image, and it consumes about 28GB storage space for COCO dataset.

4.4 Ablation Studies

We conduct ablation studies on Flickr30K (Table 4) and compare LightningDOT (L4) against 3 ablated instances: (i)"R-CNN only" (L1): image representations are extracted from Faster R-CNN directly, with no image encoder applied; (ii) "+Image Encoder" (L2): regional features are encoded with a 12-layer Transformer as the image encoder; (iii) "+PT†" (L3): our model is pre-trained with MLM+MRM+CMR, then finetuned on Flickr30K. Note that the difference between MLM vs. VMLM and MRM vs. SMRM is whether the predictions of masked tokens (regions) rely on infused embeddings from the other modality.

UNITER-base on a smaller dataset.

Method	Multi30K			COCO		
	DE	FR	CS	ZH	JA	Meta-Ave
S-LIWE	72.1	63.4	59.4	73.6	70.0	67.7
MULE	64.1	62.3	57.7	75.9	75.6	67.1
SMALR	69.8	65.9	64.8	77.5	76.7	70.9
M^{3}P	82.0	73.5	70.2	81.8	**86.8**	78.9
UNITER	85.9	**87.1**	85.7	**88.4**	85.9	86.6
LightningDOT	83.3	83.7	82.2	87.2	82.3	83.7
+Re-Ranker	**86.1**	**87.1**	**86.2**	**88.4**	86.1	**86.8**

Table 6: Evaluation on multilingual image-text retrieval over Multi30K and COCO datasets. We compare with task-specific methods: S-LIWE (Wehrmann et al., 2019), MULE (Kim et al., 2020), SMALR (Burns et al., 2020), pre-trained method M^{3}P (Huang et al., 2020a) and UNITER with *translate-test*. Numbers in blue indicate the use of different dev/test splits of COCO compared to other methods. UNITER and Re-ranker are large model size.

Results show that "R-CNN only" is not sufficient in learning good image representations for ITR task, while image encoder with Transformer architecture can effectively learn contextualized image representations, hence achieving better performance. Pre-trained models (L3-4) generally achieve better performance, compared to non-pretrained models (L1-2). Comparing "+PT†" to the full instance of LightningDOT, dependency on the other modality in VMLM and SMRM brings universal performance lift across all metrics. This indicates that these cross-modal dependencies introduced by VMLM and SMRM are effective in learning the association between image and text inputs.

In addition, we investigate the effectiveness of each pre-training task in Table 5. Comparing to baseline without pre-training, pre-training with CMR alone lifts $+1.4$ on AR. Pre-training with all three tasks achieves the best performance, indicating that the learning of contextualized word and region representations promotes better global alignment between image and text, and these three pre-training tasks work collaboratively to yield better visual-semantic embeddings.

4.5 Multilingual Image-Text Retrieval

We further report results on multilingual image-text retrieval tasks. Specially, we evaluate LightningDOT under the *translate-test* setting, which is to translate the test captions in other languages to English by leveraging Machine Translation (MT) tool.[10] Note that our method is only trained on English captions, without exploiting the original or translated captions from multilingual benchmarks.

[10]We use Microsoft Azure Translation API Service.

Figure 4: Retrieved top 10 images from the query "Sky view of a blue and yellow biplane flying near each other." The ground truth is in the red rectangle.

We consider two benchmarks: Multi30K (Elliott et al., 2016, 2017; Barrault et al., 2018) with captions in German, French and Czech; and COCO Japanese (Yoshikawa et al., 2017) and Chinese (Li et al., 2019b).

Average Recall (AR) is used as the evaluation metric. Meta-Ave, the average of AR over different languages across two benchmarks, is used as a global metric. More details on multilingual ITR benchmarks are included in Appendix.

We compare LightningDOT against 3 task-specific methods: S-LIWE (Wehrmann et al., 2019), MULE (Kim et al., 2020) and SMALR (Burns et al., 2020), which all exploit captions in different languages to learn multilingual or language-agnostic word embeddings. We also compare with a pre-trained model M^3P (Huang et al., 2020a), which is alternatively pre-trained with image-caption pairs labeled in English and cross-lingual corpus in 100 different languages. Note that all methods discussed above are trained/finetuned on captions in different languages. For fair comparison, we report performance of UNITER under the same translate-test setting, which is finetuned with English captions only and tested on translated captions.

Table 6 shows similar trends of performance improvements as on English benchmarks. Compared to both state-of-the-art task-specific methods and pre-trained models, LightningDOT under *translate-test* setting achieves new state of the art on most languages and establishes a strong baseline for future study on these multilingual benchmarks.

4.6 Qualitative Examples

We show an example of image retrieval results here at figure 4 for query as "Sky view of a blue and yellow biplane flying near each other". In addition to the ground truth image in the red rectangle, all the 10 images retrieved by our model are valid retrieval since multiple keywords ("sky", "blue", "yellow", "airplane", "near") are captured for each image. Please see the appendix A.4 for more examples.

5 Conclusion

In this paper, we propose a pre-training framework that learns joint visual-semantic embedding without any cross-attention between modalities. LightningDOT outperforms previous state of the art, while significantly speeding up inference time by 600-2000× on Flickr30K and COCO image-text retrieval benchmarks. Future work includes extending the efficient training framework to other V+L tasks.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Find-

ings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. ACL.

Andrea Burns, Donghyun Kim, Derry Wijaya, Kate Saenko, and Bryan A. Plummer. 2020. Learning to scale multilingual representations for vision-language tasks. In *ECCV*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Deep Learning and Representation Learning Workshop*. NeurIPS.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*. ACL.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*. ACL.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *CoNLL*.

Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. 2018. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Haoyang Huang, Lin Su, Di Qi, Nan Duan, Edward Cui, Taroon Bharti, Lei Zhang, Lijuan Wang, Jianfeng Gao, Bei Liu, Jianlong Fu, Dongdong Zhang, Xin Liu, and Ming Zhou. 2020a. M3p: Learning universal representations via multitask multilingual multimodal pre-training. *arXiv preprint arXiv:2006.02635*.

Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning semantic concepts and order for image and sentence matching. In *CVPR*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020b. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of EMNLP*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan A. Plummer. 2020. MULE: Multimodal Universal Language Embedding. In *AAAI*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. In *Deep Learning and Representation Learning Workshop*. NeurIPS.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *ECCV*.

Kuang-Huei Lee, Hamid Palangi, Xi Chen, Houdong Hu, and Jianfeng Gao. 2019. Learning visual relation priors for image-text matching and image captioning with neural scene graph generators. *arXiv preprint arXiv:1909.09953*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019a. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xirong Li, Chaoxi Xu, Xiaoxu Wang, Weiyu Lan, Zhengxiong Jia, Gang Yang, and Jieping Xu. 2019b. Coco-cn for cross-lingual image tagging, captioning, and retrieval. *IEEE Transactions on Multimedia*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *CVPR*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPs*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *EMNLP*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. 2018. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. 2019. Camp: Cross-modal adaptive message passing for text-image retrieval. In *ICCV*.

Jonatas Wehrmann, Douglas M. Souza, Mauricio A. Lopes, and Rodrigo C. Barros. 2019. Language-agnostic visual-semantic embeddings. In *ICCV*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Yuya Yoshikawa, Yutaro Shigeto, and Akikazu Takeuchi. 2017. STAIR captions: Constructing a large-scale Japanese image caption dataset. In *ACL*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *ECCV*.

Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. 2020. Context-aware attention network for image-text retrieval. In *CVPR*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI*.

A Appendix

A.1 Implementation Details

To further facilitate the reproductivity of our proposed method, we include more details about the choice of model size and hyper-parameters for both pre-training and fine-tuning.

The model dimensions are set to (L=12, H=768, A=12) for both image encoder and language encoder, where L is the number of stacked Transformer blocks; H stands for hidden activation dimension, and A is the number of attention heads. The total number of parameters in LightningDOT is 220M. Pre-training and finetuning learn the parameters of both encoders. During inference, with offline representation caching, only the forwarding pass with one encoder from the query modality will be performed online.

For both pre-training and finetuning, AdamW (Loshchilov and Hutter, 2019) is used to optimize the model training, with β_1=0.9, β_2=0.98. We adopt a learning rate warmup strategy, where the learning rate is linearly increased during the first 10% of training steps, followed by a linear decay to 0. We set the L2 weight decay to be 0.01.

During pre-training, we follow UNITER (Chen et al., 2020) to randomly sample 1 task per mini-batch update.[11] Our best model is pre-trained on VMLM+SMRM+CRM for 300,000 optimization steps. We set the batch size to 10240 per GPU (batch size is specified by #tokens + #regions, as in UNITER). Pre-training experiments are conducted on 8× V100 GPUs with 6-step gradient accumulation, and the learning rate is set to be 5e-5. For ablation studies presented in Table 5, the ablated instances of our model are pre-trained for 30k steps on COCO dataset (Lin et al., 2014) only, and the same choice of learning rate and batch size are applied as in the best pre-training setting.

For finetuning, we set batch size n to 96 (n is in examples, instead of the sequence length of tokens and regions), and search learning rate from {1e-5, 2e-5, 5e-5}. We select models based on their AR on the validation set. The best learning rate is 5e-5 for COCO and 1e-5 for Flickr30K. Our models are trained for 15 epochs on Flickr30k, and 20 epochs on COCO. For re-ranking, we choose k from {20, 50}.

[11] Code obtained from https://github.com/ChenRocks/UNITER.

A.2 Multilingual Image-Text Retrieval Benchmarks

When evaluating on ITR under the multilingual setting, we consider two benchmarks: Multi30K (Elliott et al., 2016, 2017; Barrault et al., 2018) and COCO Japanese (Yoshikawa et al., 2017) and Chinese (Li et al., 2019b). Multi30K is constructed by manually translating English captions in Flickr30K (Plummer et al., 2015) to German, French, and Czech. Each image in Multi30K is paired with 5 captions in German, 1 caption in French and Czech. We adopt the same train/val/test split as in Flickr30K. COCO Japanese (Yoshikawa et al., 2017) collected 820K Japanese captions for 165K COCO images (Lin et al., 2014). We use the same train/dev/test splits for COCO Japanese as in Karpathy and Fei-Fei (2015), and present results on the 1K test set. Similarly, Li et al. (2019b) collected 1-2 Chinese captions per image for 20K COCO images to build COCO Chinese. We follow the original split defined in Li et al. (2019b).

A.3 Inference Time

We present the detailed inference time of UNITER-base, SCAN the proposed LightningDOT and LightningDOT with UNITER-base re-ranker in Table 7, measured by seconds/query. UNITER clearly is the slowest, as the 12-layer Transformer model inference needs to be run between each query and *all* images. Comparing between Flickr30k-test and COCO-test, its inference time scales up linearly with the number of images. With the lightweight GRU (Chung et al., 2014), SCAN is ∼1.9× faster than UNITER. Across all settings, LightningDOT is significantly faster than both cross-attention methods (UNITER-base and SCAN). When adding UNITER-base as the re-ranker, our method slows down by ∼10, but still achieves decent speedup.

A.4 More Qualitative Examples

We show several qualitative results of image retrieval (top-10). All results are retrieved from COCO-Full dataset (123k images in total). Our model can well understand the underlying semantic meaning. For example, "romantic" only appears twice in the whole COCO dataset annotations, yet the top retrieved images are all topic-related (Figure 5). With multiple keywords, our model attempts to retrieve the combinations of them (if not all). For example, for the query "blue girl boy ball" with four keywords, our model retrieves images

Method	#images	UNITER-base	SCAN	LightningDOT	LightningDOT+Re-ranker
Flickr30K-test	1000	0.41	0.23	0.00064	0.0089
COCO-test	5000	1.95	1.04	0.00101	0.020
Flickr30K-full	31014	12.8*	7.10*	0.00193	0.010
COCO-full	123287	48.0*	25.7*	0.00201	0.021

Table 7: Image retrieval time cost measured by computation time (in seconds) for each query. The computation time for UNITER and SCAN is roughly linear to #images. Numbers with * are estimated by running time on test set.

Figure 5: Retrieved top-10 images for query "romantic".

Figure 6: Retrieved top-10 images for query "blue girl boy ball"

that capture at least three keywords (Figure 6).

We also present image retrieval results where the text query is sampled from COCO dataset. We randomly sample 3 queries and present the results as below (ground truth on the top, retrieved top-10 images at the bottom). Clearly, our model retrieves related images from the full dataset.

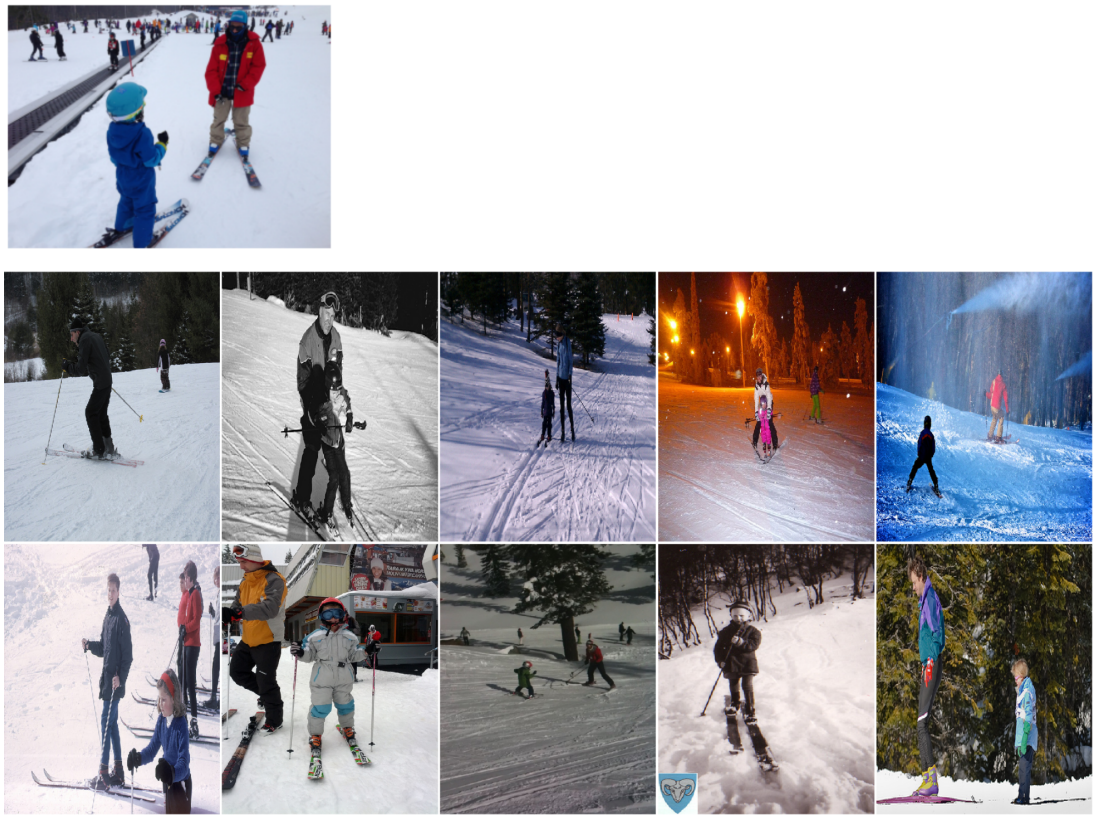

Figure 7: Retrieved top 10 images from the query "A man and a little boy on skis on a ski hill." (Top picture is the ground truth.)

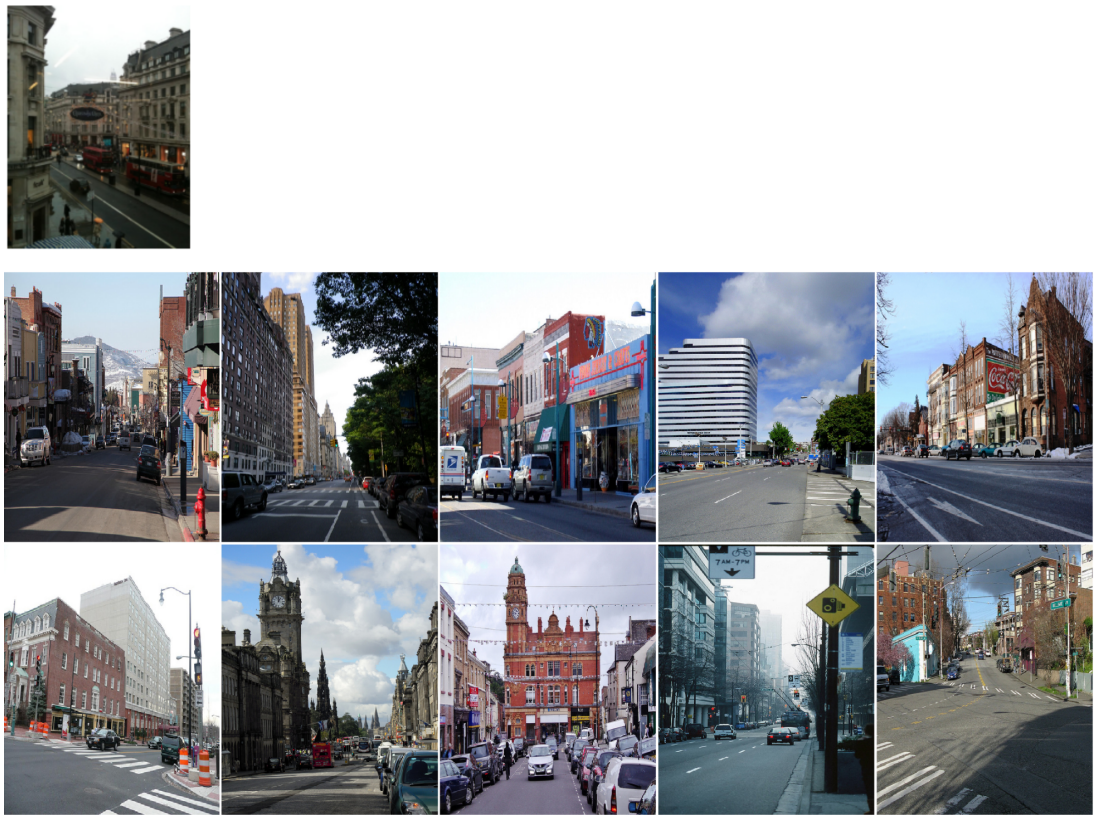

Figure 8: Retrieved top 10 images from the query "A road is lined with buildings and has cars on it." (Top picture is the ground truth.)

Figure 9: Retrieved top 10 images from the query "Two train employees stand near the open train car door." (Top picture is the ground truth.)

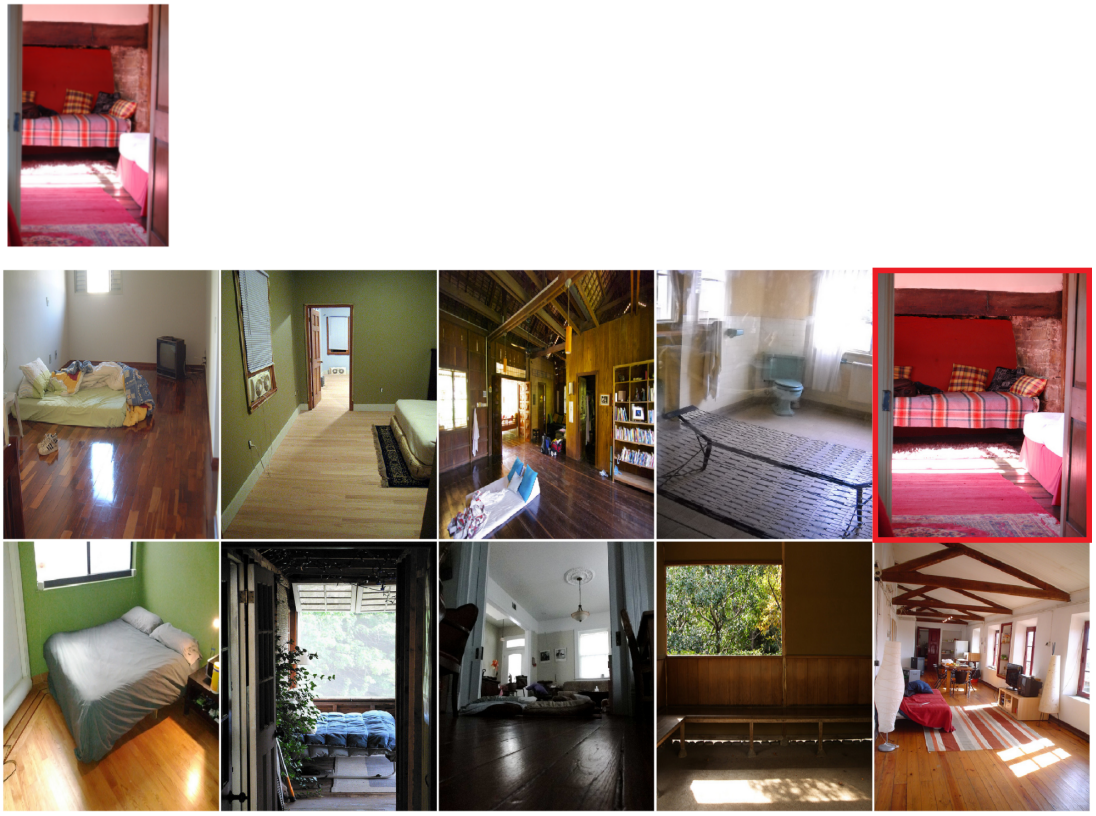

Figure 10: Retrieved top 10 images from the query "The sun hits the floor in a rustic bedroom." (Top picture is the ground truth.)