Inflate and Shrink: Enriching and Reducing Interactions for Fast Text-Image Retrieval

Haoliang Liu, Tan Yu, Ping Li
Cogntive Computing Lab
Baidu Research
No.10 Xibeiwang East Road, Beijing 100193, China
10900 NE 8th St. Bellevue, Washington 98004, USA
{haoliangliu.leo,tanyuuynat,pingli98}@gmail.com

Abstract

By exploiting the cross-modal attention, cross-BERT methods have achieved state-of-the-art accuracy in cross-modal retrieval. Nevertheless, the heavy text-image interactions in the cross-BERT model are prohibitively slow for large-scale retrieval. Late-interaction methods trade off retrieval accuracy and efficiency by exploiting cross-modal interaction only in the late stage, attaining a satisfactory retrieval speed. In this work, we propose an inflating and shrinking approach to further boost the efficiency and accuracy of late-interaction methods. The inflating operation plugs several codes in the input of the encoder to exploit the text-image interactions more thoroughly for higher retrieval accuracy. Then the shrinking operation gradually reduces the text-image interactions through knowledge distilling for higher efficiency. Through an inflating operation followed by a shrinking operation, both efficiency and accuracy of a late-interaction model are boosted. Systematic experiments on public benchmarks demonstrate the effectiveness of our inflating and shrinking approach.

1 Introduction

Efficiency and accuracy are two key factors of a retrieval system. In many cases, designing a retrieval system is striving to balance efficiency and accuracy. Embedding-based methods (Ordonez et al., 2011; Gong et al., 2014; Faghri et al., 2018) are early works for tackling the cross-modal retrieval. They encode each image or text into a global embedding. Then the text-image similarity is measured by the distance between their embeddings in the learned feature space. Since there are no interactions between text and image, embedding-based methods only need $\mathcal{O}(N+M)$ computational complexity to encode N images and M texts. The linear computational complexity of embedding-based methods makes them scalable to large-scale cross-modal retrieval. They hence have been widely deployed in real-world cross-modal retrieval tasks.

Recently, inspired by the great success of Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) in natural language processing, some methods (Li et al., 2019, 2020a; Chen et al., 2020; Li et al., 2020b; Fei et al., 2021) investigate cross-BERT model to exploit the cross-modal attention and devise several pre-training tasks. Benefiting from cross-modal attention and pre-training, they have achieved significantly higher retrieval accuracy than their embedding-based counterparts. Nevertheless, the heavy text-image interaction from utilizing cross-modal attention leads to an $\mathcal{O}(NM)$ computational complexity in encoding when calculating the similarities between N images and M texts. The quadratic computational complexity of cross-BERT methods makes them not suitable for large-scale cross-modal retrieval applications.

Several methods attempt to gain satisfactory efficiency and maintain high accuracy through trading off efficiency and accuracy. These methods can be coarsely grouped into two categories: two-stage methods (Sun et al., 2021; Geigle et al., 2021; Miech et al., 2021) and late-interaction methods (Lee et al., 2018; Khattab and Zaharia, 2020; Lu et al., 2021). The two-stage methods apply a retrieve-and-rerank strategy. Given a query, in the first stage, they conduct a coarse-level retrieval through an embedding-based method to obtain an initial top-t list of potentially relevant items. Then in the second stage, the items in the top-t list are re-ranked through a powerful cross-BERT model. Since the heavy interactions are only deployed in the second stage, and t is smaller than the number of total items in the corpus, the efficiency is boosted. Nevertheless, to obtain a satisfying retrieval accuracy, t should be large enough. In consequence, two-stage methods cannot achieve high efficiency as embedding-based methods.

In parallel, the late-interaction methods improve the retrieval accuracy of the embedding-based methods through lightweight text-image interac-

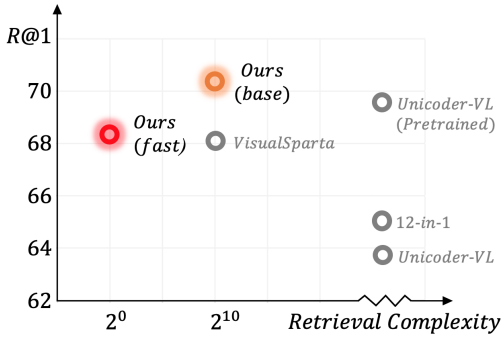


Figure 1: Text-to-Image Recall@1 on MSCOCO1K benchmark versus retrieval complexity. Our models are not pretrained on other multi-modal datasets.

tions. To be specific, SCAN (Lee et al., 2018) and VisualSparta (Lu et al., 2021) only conduct word-region interactions in the late stage when the word/region features have already been extracted by the image encoder and the text encoder. Therefore, the text-image interactions in late-interaction methods are cheap. Empirically, due to only using light-weight interactions, late-interaction methods normally attain faster inference but lower accuracy than their two-stage counterparts (Sun et al., 2021; Geigle et al., 2021; Miech et al., 2021). In fact, late-interaction methods can be deployed into the first stage of a two-stage method as the alternative to the embedding-based method to further improve the performance of the two-stage methods.

In this work, we propose an inflating and shrinking approach to enhance the effectiveness and efficiency of the existing late-interaction methods. We observe that the representing capability of the late-interaction methods is limited by the text length and the region count. For instance, given a sentence of n words and an image with m regions, SCAN and VisualSparta only have access to nm times region-word interactions between n words and m regions. To thoroughly exploit region-word interactions, we propose an inflating operation. We plug additional k codes in the input of the text/image encoder besides the word/region features. It generates $m + k$ image vectors in the output of the image encoder and $n + k$ text vectors in that of the text encoder. The number of region-word interactions increases from mn to $(n + k)(m + k)$. Nevertheless, incorporating additional codes inevitably brings more computational cost and makes the retrieval slower. To boost efficiency, we propose a shrinking operation based on distilling to reduce interactions.

Through inflating followed by shrinking, we obtain two models, the base model and the fast model. Our base model obtains a considerably higher retrieval accuracy than VisualSparta. Meanwhile, our fast model achieves a comparable retrieval accuracy as VisualSparta but takes much less latency. We visualize the efficiency and accuracy comparisons with VisualSparta and cross-BERT model including Unicoder-VL (Li et al., 2020a) and 12-in-1 (Lu et al., 2020) in Figure1. Systematic experiments on public benchmarks, including MSCOCO1K and Flickr30K, demonstrate the effectiveness of the proposed inflating and shrinking approach.

2 Related Works

Embedding-based methods. Early embedding-based methods (Ordonez et al., 2011; Gong et al., 2014) depend on Canonical Component Analysis (CCA) (Hardoon et al., 2004) to project texts and images into a joint feature space. With the progress of deep learning, the architecture of mainstream embedding-based methods has evolved into a dual-encoder structure (Klein et al., 2015; Wang et al., 2016; Faghri et al., 2018; Dong et al., 2019; Wang et al., 2019a) consisting of an image encoder and a text encoder. In the retrieval phase, the text-image similarity is determined by the distance between the image embedding and the text embedding generated from image and text encoders, separately.

Attention-based methods. By paying attention to key visual cues in the image and key words in the text, the attention-based methods (Huang et al., 2017; Lee et al., 2018; Li et al., 2019; Wei et al., 2020; Zhu et al., 2020; Yu et al., 2020, 2021b) achieve considerably better performance than the embedding-based methods in cross-modal retrieval. sm-LSTM (Huang et al., 2017) takes a multi-modal context-modulated attention scheme to selectively attend to a pair of instances of image and sentence. SCAN (Lee et al., 2018) computes the similarities between regions and words, and only counts the region-word pairs of high relevance. VSRN (Li et al., 2019) adopts graph convolution to attend the region features based on the textual context. MMCA (Wei et al., 2020) incorporates self-attention, which is originally used in Transformer, to enhance the region and word features. Drawn inspiration from the success achieved by BERT through pre-training, several cross-BERT methods (Li et al., 2020a; Lu et al., 2019; Li et al., 2020b; Zhang et al., 2020; Fei et al., 2021) are proposed for cross-modal retrieval. They stack a few Transformer blocks and devise several

pre-training tasks for facilitating multi-modal understanding such as masked language modeling, masked region modeling and text-image matching. After being pre-trained on large-scale datasets, they have achieved state-of-the-art performance in cross-modal retrieval. Nevertheless, cross-BERT methods need quadratic computational complexity, and they are slow and not scalable.

Trade-off methods. To alleviate the heavy computational burden while maintaining the high retrieval accuracy, several trade-off methods are proposed. They can be coarsely grouped into two-stage methods and late-interaction methods. Two-stage methods (Geigle et al., 2021; Sun et al., 2021; Miech et al., 2021) take a retrieve and re-rank strategy. In the first stage, they adopt an embedding-based method to conduct coarse-level retrieval and obtain a top-t list. After that, in the second stage, they re-rank the top-t list using a cross-BERT method. Since the number of items for re-ranking, t, is smaller than the total number of reference items, two-stage methods achieve higher efficiency than cross-BERT methods. In parallel, the late-interaction trade-off methods (Khattab and Zaharia, 2020; Lu et al., 2021) apply a light-weight interaction in the late stage after feature encoding, and also achieve higher efficiency than cross-BERT methods with heavy interaction in encoding.

3 Background

Embedding-based methods (Wang et al., 2016; Faghri et al., 2018) and cross-BERT methods (Li et al., 2020a; Chen et al., 2020; Li et al., 2020b) are two mainstream approaches for measuring the similarity between an image I and a text T. Embedding-based methods encode each image as well as each text into a global embedding. The text-image similarity is determined by the similarity between their global embeddings. Consequently, given N images and M texts, the embedding-based methods only take $\mathcal{O}(N+M)$ complexity in encoding.

In contrast, cross-BERT methods take each text-image pair as input. Making use of self-attention operation, cross-BERT methods achieve significantly higher retrieval accuracy than embedding-based methods. But self-attention brings significantly more computational cost. To give an example, given N images and M texts, cross-BERT methods take $\mathcal{O}(NM)$ complexity to encode NM text-pairs. Thus, cross-BERT methods are prohibitively slow in large-scale retrieval. To trade off the efficiency and accuracy, researchers proposed late-interaction methods (Khattab and Zaharia, 2020; Lu et al., 2021). Similar to embedding-based methods, they only need $\mathcal{O}(N+M)$ complexity in encoding. But they exploit light-weight attention in the scoring phase based on extracted local embeddings. Taking advantage of attention, they obtain higher accuracy than embedding-based methods and they are efficient since the attention is lightweight. Below we introduce embedding-based methods and late-interaction methods in detail.

3.1 Embedding-based methods

To bridge the domain gap between texts and images, embedding-based methods map texts and images to the same feature space. They normally adopt a dual-encoder structure (Geigle et al., 2021; Sun et al., 2021) to encode texts and images respectively.

Image encoder. Given an image I, following previous works (Lee et al., 2018; Chen et al., 2020; Lu et al., 2021), each image is represented by a set of m image region features $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_m\}$. They are extracted by a Faster R-CNN (Ren et al., 2015) object detector pre-trained on Visual Genome dataset (Krishna et al., 2017). The image region features $\mathcal{R}$ are the input ofa Transformer encoder. The attended region features, $\bar{\mathcal{R}}$, are the output of the Transformer encoder:

$$\bar{\mathcal{R}} = \text{Transformer}(\mathcal{R}) = [\bar{\mathbf{r}}_1, \cdots, \bar{\mathbf{r}}_m] \in \mathbb{R}^{m \times d}. \tag{1}$$

We term $\bar{\mathbf{r}}_i$ ($i \in [1, m]$) as an image fragment.

Text encoder. Following BERT (Devlin et al., 2019), each text T is converted into n words, which are further embedded into n word embeddings $\mathcal{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$. The attended word embeddings $\bar{\mathcal{W}}$ are the output of the Transformer encoder:

$$\bar{\mathcal{W}} = \text{Transformer}(\mathcal{W}) = [\bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_n] \in \mathbb{R}^{n \times d}, \tag{2}$$

where $\bar{\mathbf{w}}_i$ ($i \in [1, n]$) is termed as a text fragment.

Scoring. To measure the text-image distance based on their attended word and region feature sets, $\bar{\mathcal{W}}$ and $\bar{\mathcal{R}}$, common practices are taking the first token, *i.e.*, [CLS], to summarize $\bar{\mathcal{W}}$/$\bar{\mathcal{R}}$ into a global embedding. The text-image similarity is determined by the cosine similarity between their embeddings:

$$s(\bar{\mathbf{r}}_1, \bar{\mathbf{w}}_1) = \cos(\bar{\mathbf{r}}_1, \bar{\mathbf{w}}_1). \tag{3}$$

Since the matching is conducted in the global embedding level, we term it as global-level matching.

Training. Given B text-image pairs $\{(T_i, I_i)\}_{i=1}^{B}$ in a mini-batch, the text T_i is only relevant with the image I_i and is irelevant with other images in the batch, I_j ($j \neq i$). Triplet loss aims to make the similarity between the positive text-image pair (T_i, I_i) larger than that between the negative pair (T_i, I_j) where $j \neq i$ by a margin m:

$$\mathcal{L} = -\sum_{i=1}^{B}\sum_{j\neq i}^{B}\Big\{[m - s(I_i, T_i) + s(I_i, T_j)]_+ + [m - s(I_i, T_i) + s(I_j, T_i)]_+\Big\}, \quad (4)$$

where $s(T_i, I_j)$ is the similarity between T_i and I_j computed as Eq. (3). m is the margin predefined, and $[x]_+ = \max(x, 0)$ is a clip function. Some approaches (Faghri et al., 2018; Lee et al., 2018; Geigle et al., 2021) conduct hard negative mining to enhance the effectiveness of the triplet loss.

3.2 Late-interaction methods

Recent works (Lee et al., 2018; Khattab and Zaharia, 2020; Humeau et al., 2020; Zhao et al., 2021; Lu et al., 2021) boost accuracy of embedding-based methods and maintain high efficiency through text-image interaction in the late stage. They utilize the same dual-encoder structure as embedding-based methods for encoding, but they utilize attention in the scoring phase. Given a text with fragments $\bar{\mathcal{W}}$ from the output of the encoder and an image with fragments $\bar{\mathcal{R}}$, the text-image similarity is calculated by interactions between these two bags of fragments, denoted as $s(\bar{\mathcal{W}}, \bar{\mathcal{R}})$. Specifically, Visualsparta (Lu et al., 2021) implements $s(\bar{\mathcal{W}}, \bar{\mathcal{R}})$ by

$$s(\bar{\mathcal{W}}, \bar{\mathcal{R}}) = \sum\nolimits_{i=1}^{n} \max_{j\in[1,m]} \left(cos(\bar{\mathbf{w}}_i, \bar{\mathbf{r}}_j)\right), \quad (5)$$

where $\bar{\mathcal{W}} = [\bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_n]$, $\bar{\mathcal{R}} = [\bar{\mathbf{r}}_1, \cdots, \bar{\mathbf{r}}_m]$, and $cos(\mathbf{u}, \mathbf{v})$ measures the cosine similarity between the vectors $\mathbf{u}$ and $\mathbf{v}$. As illustrated in Eq. (5), every text fragment interacts with m image fragments through computing maximum cosine similarity, and the scores from n text segments are summed up to generate the text-image similarity. To calculate the similarity between a text-image pair, it takes $m \times n$ fragment interactions in total.

Retrieval latency. The retrieval latency consists of the encoding latency for extracting the text/image fragments $\bar{\mathcal{R}}/\bar{\mathcal{W}}$ and the scoring latency for computing $s(\bar{\mathcal{W}}, \bar{\mathcal{R}})$ in Eq. (5). In practice, in the text-to-image retrieval application, the image fragments $\bar{\mathcal{R}}$ have been extracted in the offline phase before the query comes. Given a query, the encoder only needs to extract the query's fragments, taking a constant computation complexity, $\mathcal{O}(1)$. In contrast, the scoring is conducted between the query's fragments and N images' fragments in the corpus, taking a linear computation complexity, $\mathcal{O}(N)$. Thus, in the large-scale retrieval scenario, the inference speed is mainly determined by the scoring latency.

4 Method

Section 4.1 introduces our inflating operation to thoroughly exploit the fragment interactions for higher retrieval accuracy. In section 4.2, we present the proposed shrinking operation to reduce the fragment interactions for higher efficiency.

4.1 Inflating

Benefiting from fragment interactions, late-interaction methods achieve higher retrieval accuracy than embedding-based methods. Nevertheless, as shown in Eq. (5), the scale of interactions of late-interaction methods, mn, is limited by the number of word features of the sentence (m) and that of region features from the image (n).

To exploit more informative interactions, we devise a set of synthetic tokens $\mathcal{C}_T = \{\mathbf{c}_1^T, \cdots, \mathbf{c}_k^T\}$ as the additional input for the text encoder and a set of synthetic tokens $\mathcal{C}_I = \{\mathbf{c}_1^I, \cdots, \mathbf{c}_k^I\}$ as the additional input for the image encoder. The synthetic tokens are similar to the [CLS] token used in BERT. But a single [CLS] token has a limited representation capability, and the devised synthetic tokens $\mathcal{C}_I$ and $\mathcal{C}_T$ have a much more powerful capability by using multiple codes. In the implementation, $\mathcal{C}_T$ and $\mathcal{C}_I$ are parameters of the model, which are randomly initialized and updated by back-propagating the gradients in the training phase. In this case, the input of the image encoder is a concatenation of image region features and the inflating codes, $[\mathcal{R}; \mathcal{C}_I]$. In the same way, the input of the text encoder is $[\mathcal{W}; \mathcal{C}_T]$. Then they are encoded in the same manner as Eq. (1) and Eq. (2), respectively:

$$\begin{aligned}\bar{\mathcal{R}}_{\text{inflate}} &= \text{Transformer}([\mathcal{R}; \mathcal{C}_I]) \\ &= [\bar{\mathbf{r}}_1, \cdots, \bar{\mathbf{r}}_m, \bar{\mathbf{c}}_1^T, \cdots, \bar{\mathbf{c}}_k^T] \in \mathbb{R}^{(m+k)\times d}, \\ \bar{\mathcal{W}}_{\text{inflate}} &= \text{Transformer}([\mathcal{W}; \mathcal{C}_T]) \\ &= [\bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_n, \bar{\mathbf{c}}_1^I, \cdots, \bar{\mathbf{c}}_k^I] \in \mathbb{R}^{(n+k)\times d}.\end{aligned}$$

Then the text-image similarity is determined by fragment-level matching between $\bar{\mathcal{W}}_{\text{inflate}}$ and $\bar{\mathcal{R}}_{\text{inflate}}$ in the same manner as Eq. (5).

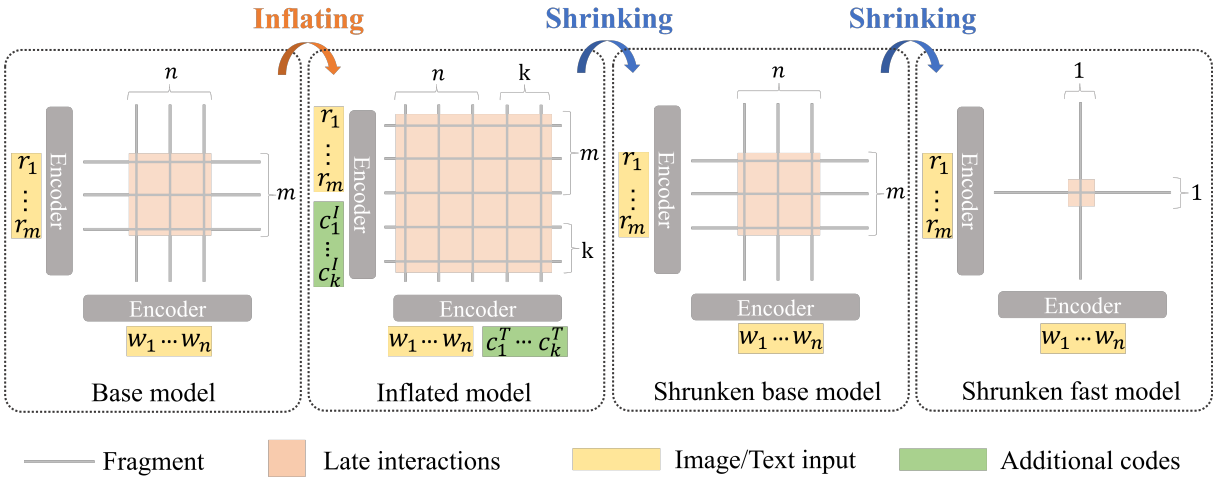


Figure 2: The pipeline of the proposed inflating and shrinking strategy. The base model adopts the late interaction (Khattab and Zaharia, 2020; Lu et al., 2021). The proposed inflating operation plugs several codes in the input of image/text encoders to thoroughly exploit fragment-level interactions. Then the shrinking operation distills the knowledge of the inflated model and generates the shrunken base model, which is further shrunken to a fast model.

Through plugging additional codes in the text encoder and the image encoder, our inflated model more thoroughly exploit the fragment interactions. More precisely, when computing $\mathrm{s}(\bar{\mathcal{W}}, \bar{\mathcal{R}})$ in Eq. (5), only nm word-region interactions are available. In contrast, when computing $\mathrm{s}(\bar{\mathcal{W}}_{\text{inflate}}, \bar{\mathcal{R}}_{\text{inflate}})$, $(n+k)(m+k)$ interactions are conducted. Our experiments show that more fragment interactions through inflating boost retrieval accuracy. Nevertheless, more interactions brought by inflating inevitably increases computational cost and makes the retrieval slower.

4.2 Shrinking

Different from inflating which expands the scale of interactions to enhance effectiveness, shrinking aims to reduce the interactions to boost efficiency. The idea behind shrinking is knowledge distillation, which is originally developed for classification task (Hinton et al., 2015). By exploiting the contrastive learning (Gutmann and Hyvärinen, 2010), knowledge distillation can be naturally extended to the retrieval task. Assume that, in a batch of text-image pairs $\{(T_i, I_i)\}_{i=1}^{B}$, T_i is only relevant with I_i and is irrelevant with the rest. Let s_{ij}^t denote the similarity between T_i and I_j from the teacher model and s_{ij}^s denote that from the student model. The distillation loss is devised as

$$\mathcal{L}_{\mathrm{D}} = -\sum_{i=1}^{B} \frac{e^{\frac{s_{ii}^t}{\tau_t}}}{\sum_{k=1}^{B} e^{\frac{s_{ik}^t}{\tau_t}}} \log\left(\frac{e^{\frac{s_{ii}^s}{\tau_s}}}{\sum_{l=1}^{B} e^{\frac{s_{il}^s}{\tau_s}}}\right), \quad (6)$$

where τ_t and τ_s are pre-defined temperature factors controlling the softness.

Our shrinking is to distill the knowledge from the inflated model with intensive interactions to the student model with fewer interactions to boost the efficiency. To be exact, we conduct the shrinking in two steps. As shown in Figure 2, in the first step, we distill the text-image similarity $\mathrm{s}(\bar{\mathcal{W}}_{\text{inflate}}, \bar{\mathcal{R}}_{\text{inflate}})$ from the teacher model to the text-image similarity $\mathrm{s}(\bar{\mathcal{W}}, \bar{\mathcal{R}})$ of the first student model. We term the student model from the first-step shrinking as shrunken base model. In the second step, we distill $\mathrm{s}(\bar{\mathcal{W}}, \bar{\mathcal{R}})$ from our shrunken base model to $\mathrm{s}(\bar{\mathbf{w}}_1, \bar{\mathbf{r}}_1)$ computed in the manner as Eq. (3) from the second student model. The second student model has degenerated to the embedded-based method, and thus it only needs once global interaction and is extremely fast. We term the second student model as shrunken fast model.

Relation with existing distilling methods. Existing knowledge distilling methods (Jiao et al., 2020) normally distill the knowledge from a large-scale teacher model to a small-scale student model for faster inference. Nevertheless, the architecture gap between the student model and the teacher model will inevitably lead to considerable losses. In contrast, in the proposed shrinking operation, the encoder used in the teacher model adopts the same architecture as that in the student model used in our shrinking operation. The only difference between the teacher model and the student model lies in the

Settings	Codes	Frag.	Inter.	Flickr30K						MSCOCO1K					
				T-to-I R@			I-to-T R@			T-to-I R@			I-to-T R@		
				1	5	10	1	5	10	1	5	10	1	5	10
Fast w/o	0	1	1^2	47.7	77.0	85.5	59.9	86.8	93.2	56.1	85.9	91.8	69.4	92.7	97.1
Base w/o	0	32	32^2	60.6	85.6	91.6	75.8	93.5	96.3	68.1	91.9	96.4	82.0	97.6	99.2
Inflate	16	32+16	48^2	62.9	86.4	92.3	76.9	94.0	97.5	68.8	92.1	96.6	82.6	97.2	99.1
	32	32+32	64^2	63.5	86.9	92.2	77.6	94.4	97.6	69.3	92.4	96.7	82.6	97.6	99.0
	64	32+64	96^2	63.7	86.5	92.4	77.9	94.3	97.8	69.2	92.6	96.4	82.5	97.6	99.3
	96	32+96	128^2	63.2	86.6	92.6	77.7	94.2	97.5	69.2	92.5	96.5	82.9	97.7	99.5

Table 1: Comparisons between the fast/base model without inflating and the model after inflating.

scale of interactions in the scoring phase. Since there is no architecture gap between the encoder in the teacher model and that in the student model, the teacher can effectively transfer its knowledge to its student in our shrinking operation.

5 Experiments

Datasets. MSCOCO consists of $123,287$ images, and each image contains 5 ground-truth captions. We adopt Karpathy split (Karpathy and Li, 2015) with $113,287$ images for training and $1,000$ images for testing. Flickr30K contains $31,783$ images, and each one has 5 annotated textual descriptions. Following Karpathy and Li (2015), we use 1000 images for testing.

Settings. We conduct experiments on an NVIDIA V100 GPU with float16 operations. The input sequence length is set as 32 for the text and image encoders. The weights of text and image encoders are shared. For each image, we detect 100 bounding boxes using Faster-RCNN pre-trained on Visual Genome (Krishna et al., 2017) by Anderson et al. (2018). We cluster 100 bounding boxes into 32 clusters, and 32 cluster centers are the input of the image encoder. We train all models using the same batch size as that in the baseline experiments. For training, to save the memory and computation cost, we apply a tiny BERT model with only 3 layers of Transformer blocks. This is due to that our experiments show that 3-layer model achieves a comparable performance with 12-layer model. We train the inflated model using the triplet loss in Eq. (4). We set the margin m as 0.2 for fast model and 0.6 for base and inflated models. In Eq. (6), the temperature τ_t and τ_s are both set as 12 for shrinking to a base model and set as 2 and 12 for shrinking to a fast model. The implementation was based on the PaddlePaddle deep learning framework.

5.1 Inflating

In the upper part of Table 1, we show the performance of the base model and the fast model without inflating or shrinking. The fast model is the embedding-based method relying on the global-level matching to compute the text-image similarity as Eq. (3). The base model exploits the fragment-level interactions and obtains the similarity based on Eq. (5). It is straightforward to observe from Table 1 that, the base model exploiting fragment-level interactions attains higher retrieval accuracy than the fast model with only global-level matching.

In the lower part of Table 1, we show the performance improvement by inflating the base model by a various number of codes. The codes we plug in the input of encoders will enrich the fragment interactions in the scoring, which is beneficial to retrieval. We vary the number of codes, k, from 16 to 96, leading to 48^2 to 128^2 times interactions. As shown in Table 1 that, the inflated model with more fragment interactions outperforms the base model. Generally, more codes plugged in the input of the encoder tend to yield larger improvement.

5.2 Shrinking

We first evaluate the straightforward shrinking without inflating. The base model (B) without inflating is the teacher for distilling, and the fast model (F) is the student. The experiments are shown in the first part of Table 2. As shown in the table, after shrinking (B$\rightarrow$ F), the fast model obtains significantly higher retrieval accuracy than the fast model without shrinking. Meanwhile, after shrinking, the student model achieves a comparable accuracy as the teacher model (B).

Then we evaluate the proposed method, inflating followed by shrinking. We evaluate the performance on two different settings: (i) inflating the

Settings	Codes	Frag.	Inter.	Flickr30K						MSCOCO1K					
				T-to-I R@			I-to-T R@			T-to-I R@			I-to-T R@		
				1	5	10	1	5	10	1	5	10	1	5	10
B	0	32	32^2	60.6	85.6	91.6	75.8	93.5	96.3	68.1	91.9	96.4	82.0	97.6	99.2
F	0	1	1^2	47.7	77.0	85.5	59.9	86.8	93.2	56.1	85.9	91.8	69.4	92.7	97.1
B→F	0	1	1^2	58.2	84.6	90.9	73.0	92.6	96.1	66.0	91.7	96.1	79.1	96.8	98.6
I	32	64	64^2	63.5	86.9	92.2	77.6	94.4	97.6	69.3	92.4	96.7	82.6	97.6	99.0
I→F	0	1	1^2	60.9	85.5	91.9	76.3	93.0	96.1	67.7	92.3	96.8	81.7	97.7	99.2
I→B	0	32	32^2	64.5	88.2	92.4	79.4	95.6	97.7	70.5	93.0	97.3	83.7	97.5	99.6
I→B→F	0	1	1^2	62.2	87.0	92.5	77.1	93.8	96.9	68.5	92.5	97.1	82.4	97.3	99.1

Table 2: Effectiveness of shrinking. "B" denotes the base model, "F" denotes the fast model, "I" denotes the inflated model, and the symbol → denotes the knowledge distilling from the left model to the right model.

base model by adding 32 codes and then shrink it to a fast model; (ii) inflating the base model by adding 32 codes, shrink it to a base model, and further shrink the base model to a fast model. The second part of Table 2 presents the results of these two settings. First, the fast model from an inflated teacher (I →F) gains better performance than that from a base model teacher (B→F). For instance, the Recall@1 of text-to-image retrieval gets improved from 58.2 to 60.9 on Flickr30K. Second, the multi-step shrinking (I→B→F) further boosts the fast model to a higher recall@1, 62.5. And the intermediate base model (I→B) also benefits from the inflated model, which gains a 64.5 recall@1.

5.3 Efficiency

We evaluate the text-to-image retrieval latency. For embedding-based methods and late-interaction methods, the image features for retrieval have been encoded before the text query comes. Hereafter, in the retrieval phase, the whole latency consists of the encoding latency only for the query text and the scoring latency to compute the similarity between the query and all images for retrieval.

Table 3 shows the encoding latency for our fast model, base model and the model after inflating. Meanwhile, we compare them with the encoding time of the cross-BERT method, Unicoder-VL (Li et al., 2020a). Since our fast, base and inflated model only need to encode the query text, its encoding time is invariant to the number of candidate items. In contrast, the cross-BERT method taking text-image pairs as input, which needs to encode all images (candidates) in the retrieval phase. Accordingly, as shown in Table 3, the encoding latency of our fast, base and inflated model are much less than that of the cross-BERT method, Unicoder-VL.

model	# of candidates			
	1K	10K	100K	1000K
Fast	2ms	2ms	2ms	2ms
Base	2ms	2ms	2ms	2ms
Inflated	3ms	3ms	3ms	3ms
Unicoder-VL	5s	50s	500s	5000s

Table 3: GPU time in encoding per query.

model	Inter.	FLOPs	GPU Time
Fast	1^2	×1	0.4ms
Base	32^2	$\times 32^2$	82ms
VisualSparta	32^2	$\times 32^2$	82ms
Inflated	64^2	$\times 64^2$	325ms

Table 4: Time in scoring per query on 100K candidates.

Table 4 shows the scoring latency. In theory, the scoring latency is in linear with the number of fragment interactions. As shown in the table, our fast model takes only 0.4ms latency in the scoring, which is much less than that of our base model, inflated model and VisualSparta (Lu et al., 2021). It demonstrates the significant efficiency boost brought by shrinking. Later, we will show that our fast model attains a comparable cross-modal retrieval accuracy as VisualSparta.

5.4 Comparisons with existing methods

We compare with existing methods in Table 5, which are grouped into three categories. The first category of methods, cross-BERT methods, achieve high retrieval accuracy through pre-training. But they are prohibitively slow due to quadratic complexity. Compared with them, our shrunken base model can surpass some of them, such as ViLT (Kim et al., 2021), ViLBERT (Lu

Method	Pre.	MSCOCO1K						Flickr30K					
		T-to-I R@			I-to-T R@			T-to-I R@			I-to-T R@		
		1	5	10	1	5	10	1	5	10	1	5	10
Cross-BERT													
ViLBERT (Lu et al., 2019)	√	-	-	-	-	-	-	58.2	84.9	91.5	-	-	-
Uni-VL (Li et al., 2020a)		63.9	91.6	96.5	75.1	94.3	97.8	57.8	82.2	88.9	73.0	89.0	94.1
Uni-VL (Li et al., 2020a)	√	69.7	93.5	97.2	84.3	97.3	99.3	71.5	90.9	94.9	86.2	96.3	99.0
12-in-1 (Lu et al., 2020)	√	65.2	91.0	96.2	-	-	-	65.0	88.7	93.5	-	-	-
ERNIE-ViL (Yu et al., 2021a)	√	-	-	-	-	-	-	74.4	92.7	95.9	86.7	97.8	99.0
VILLA (Gan et al., 2020)	√	-	-	-	-	-	-	74.7	92.8	95.8	86.6	97.9	99.2
OSCAR (Li et al., 2020b)	√	75.7	95.2	98.3	88.4	99.1	96.2	-	-	-	-	-	-
UNITER (Chen et al., 2020)	√	-	-	-	-	-	-	72.5	92.3	96.0	85.9	97.1	98.8
UNIMO (Li et al., 2021)	√	-	-	-	-	-	-	74.6	93.4	96.0	89.7	98.4	99.1
ViLT-B/32 (Kim et al., 2021)	√	-	-	-	-	-	-	61.9	86.8	92.8	81.4	95.6	97.6
Cross-BERT + Embedding-based													
LightDOT (Sun et al., 2021)	√	-	-	-	-	-	-	75.6	94.0	96.5	87.2	98.3	99.0
RFRS (Geigle et al., 2021)	√	75.4	95.4	98.3	88.2	98.4	99.4	76.5	93.5	96.5	89.1	98.0	98.9
Embedding-based and Late-interaction													
SMLSTM (Huang et al., 2017)		40.7	75.8	87.4	-	-	-	30.2	60.4	72.3	-	-	-
DAN (Nam et al., 2017)		-	-	-	-	-	-	39.4	69.2	79.1	55.0	81.8	89.0
VSE++ (Faghri et al., 2018)		52.0	84.3	92.0	64.6	90.0	95.7	39.6	70.1	79.5	52.9	80.5	87.2
CAMP (Wang et al., 2019c)		58.5	87.9	95.0	72.3	94.8	98.3	51.5	77.1	85.3	68.1	89.7	95.2
SCAN (Lee et al., 2018)		58.8	88.4	94.8	72.7	94.8	98.4	48.6	77.7	85.2	67.4	90.3	95.8
PFAN (Wang et al., 2019b)		61.6	89.6	95.2	76.5	96.3	99.0	50.4	78.7	86.1	70.0	91.8	95.0
RDAN (Hu et al., 2019)		61.6	89.2	94.7	74.6	96.2	98.7	54.1	80.9	87.2	68.1	91.0	95.9
CVSE (Wang et al., 2020)		66.3	91.8	96.3	78.6	95.0	97.5	56.1	83.2	90.0	73.6	90.4	94.4
VisualSparta (Lu et al., 2021)		68.2	91.8	96.3	-	-	-	57.4	82.0	88.1	-	-	-
Our base model (32×32)		**70.5**	**93.0**	**97.3**	**83.7**	**97.5**	**99.6**	**64.5**	**88.2**	**92.4**	**79.4**	**95.6**	**97.7**
Our fast model (1×1)		**68.5**	**92.5**	**97.1**	**82.4**	**97.3**	**99.1**	**62.2**	**87.0**	**92.5**	**77.1**	**93.8**	**96.9**

Table 5: Comparisons with existing methods.

et al., 2019), 12-in-1 (Lu et al., 2020). Note that, ViLT (Kim et al., 2021), ViLBERT (Lu et al., 2019) and 12-in-1 (Lu et al., 2020) are pre-trained on large-scale multimodal datasets, whereas ours is not pre-trained on these datasets. Pre-training might further improve our model, but our limited computing resources cannot afford the huge computational cost of pre-training on huge-scale datasets. We further compare with the second category of methods, two-stage methods using an embedding-based method in the first stage and a cross-BERT method in the second stage. As mentioned, they are more efficient than cross-BERT methods. But to maintain a high accuracy, they have to re-rank a number of candidates and thus are still slow.

At last, we compare the third category of methods including embedding-based methods and late-interaction methods. Due to a lack of text-image interactions, embedding-based methods cannot achieve competitive accuracy. In contrast, late-interaction methods such as SCAN (Lee et al., 2018) and VisualSparta (Lu et al., 2021) achieve a good trade-off between accuracy and efficiency. Our base and fast models also fall into this category. Compared with the existing late-interaction methods, both our base model and fast model from inflating and shrinking achieve considerably better trade-off between efficiency and accuracy. To be specific, our base model obtains higher accuracy than VisualSparta with the comparable scale of computation cost. Meanwhile, our fast model obtains a comparable retrieval accuracy as VisualSparta but takes much less latency.

5.5 Comprehensive comparisons

Figure 3 visualizes the comparisons among models without inflating and shrinking, models with only inflating, models with only shrinking, and

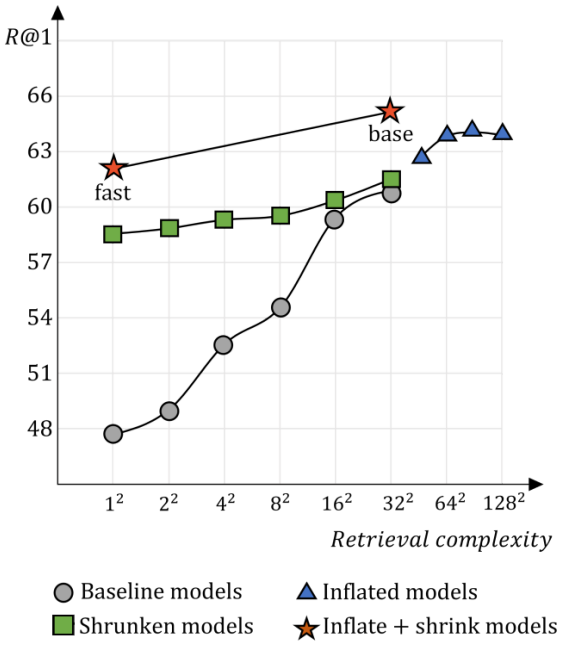
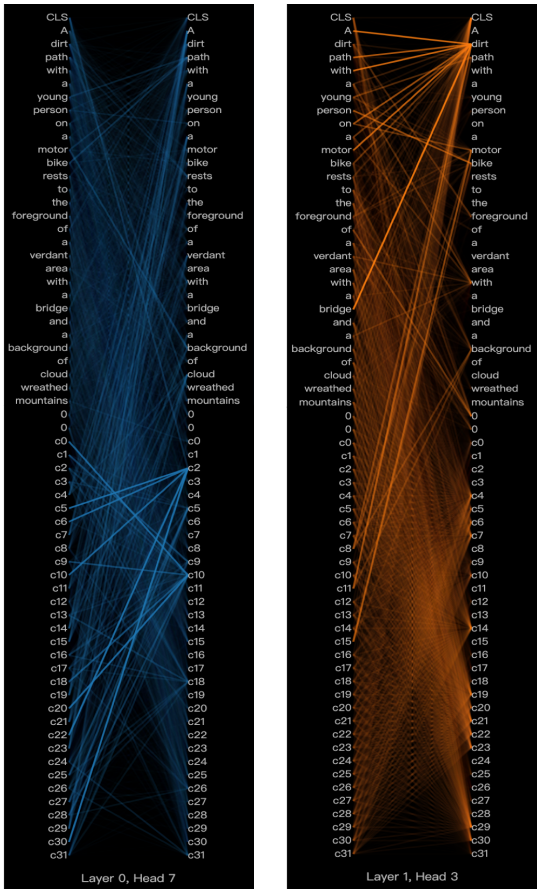


Figure 3: Comparison between baselines and proposed methods on MSCOCO(T-to-I R@1).

the models with inflating and shrinking. The x-axis represents the retrieval complexity determined by the number of fragments in the scoring phase. The curve with **gray circles** in Figure 3 denotes the performance of baseline models with a various number of interactions. In the implementation, we take the first l fragments from the image encoder and the first l fragments from the text encoder into consideration when computing the text-image score. We vary l among $\{1, 2, 4, 8, 16, 32\}$ and it takes $\{1^2, 2^2, 4^2, 8^2, 16^2, 32^2\}$ text-image interactions. When $l = 1$, it is equivalent to our fast model. When $l = 32$, it is equivalent to our base model. In parallel, the curve with **green squares** in Figure 3 shows the performance of models shrunken from the base model. The curve with **blue triangles** in Figure 3 demonstrates the performance of models through inflating the base model by $\{16, 32, 64, 96\}$ codes. Moreover, the curve with **red stars** shows that of our base and fast models from inflating and shrinking. Comparing the orange curve with the blue curve, it is straightforward to infer that inflating effectively boosts the performance by enhancing the interactions. Besides, comparing the green curve with the blue curve, we observe that a shrunken model achieves considerably higher accuracy than its counterpart with the same complexity. At last, as shown in Figure 3, the fast and base models from inflating and shrinking achieve the best trade-off between efficiency and accuracy.

5.6 Visualization of codes in inflating

In Figure 4, we use BertViz (Vig, 2019) to visualize the attention weights of a transformer layer from an inflated model. Each connection represents the relevance of the two tokens on the two sides and the brightness of this connection represents the attention strength. For an inflated model, the first 32 tokens are text words and the other 32 tokens are plugged codes when inflating. The figure shows that the codes pay nontrivial attention to text words.



(a) layer0,head7 (b) layer1,head3

Figure 4: Visualization of attention patterns.

6 Conclusion

In this paper, we propose an inflating and shrinking approach to boost the accuracy and efficiency of cross-modal retrieval. The inflating operation plugs multiple codes in the input of the image encoder and the text encoder. It enriches the text-image interactions and improves the retrieval accuracy. The shrinking operation gradually reduces the text-image interactions through knowledge distilling to improve the retrieval speed. Systematic experiments on two widely-used public benchmarks demonstrate the effectiveness and efficiency of the proposed inflating and shrinking approach.

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 *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT.

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 *Proceedings of the 16th European Conference on Computer Vision (ECCV), Part XXX*, pages 104–120, Glasgow, UK.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, MN.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9346–9355, Long Beach, CA.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*, Newcastle, UK.

Hongliang Fei, Tan Yu, and Ping Li. 2021. Cross-lingual cross-modal pretraining for multimodal retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3644–3650, Online.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, virtual.

Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2021. Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval. *arXiv preprint arXiv:2103.11920*.

Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *Int. J. Comput. Vis.*, 106(2):210–233.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 297–304, Chia Laguna Resort, Sardinia, Italy.

David R. Hardoon, Sándor Szedmák, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Zhibin Hu, Yongsheng Luo, Jiong Lin, Yan Yan, and Jian Chen. 2019. Multi-level visual-semantic alignments with relation-wise dual attention network for image and text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 789–795, Macao, China.

Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-aware image and sentence matching with selective multimodal LSTM. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7254–7262, Honolulu, HI.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: (EMNLP Findings)*, pages 4163–4174, Online Event.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, Boston, MA.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR)*, pages 39–48, Virtual Event, China.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 5583–5594, Virtual Event.

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. 2015. Associating neural word embeddings with deep image representations using fisher vectors.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4437–4446, Boston, MA.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the 15th European Conference on Computer Vision (ECCV), Part IV*, pages 212–228, Munich, Germany.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 11336–11344, New York, NY.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4653–4661, Seoul, Korea.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021. UNIMO: towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, pages 2592–2607, Virtual Event.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of the 16th European Conference on Computer Vision (ECCV), Part XXX*, pages 121–137, Glasgow, UK.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13–23, Vancouver, Canada.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR)*, pages 10434–10443, Seattle, WA.

Xiaopeng Lu, Tiancheng Zhao, and Kyusong Lee. 2021. Visualsparta: Sparse transformer fragment-level matching for large-scale text-to-image search. *arXiv preprint arXiv:2101.00265*.

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2021. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9826–9836.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2156–2164, Honolulu, HI.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1143–1151, Granada, Spain.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, Montreal, Canada.

Siqi Sun, Yen-Chun Chen, Linjie Li, Shuohang Wang, Yuwei Fang, and Jingjing Liu. 2021. Lightning-dot: Pre-training visual-semantic embeddings for real-time image-text retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Online.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, Long Beach, CA.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 37–42, Florence, Italy.

Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. 2020. Consensus-aware visual-semantic embedding for image-text matching. In *Proceedings of the 16th European Conference on Computer Vision (ECCV), Part XXIV*, pages 18–34, Glasgow, UK.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. 2019a. Learning two-branch neural networks for image-text matching tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):394–407.

Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5005–5013, Las Vegas, NV.

Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. 2019b. Position focused attention network for image-text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3792–3798, Macao, China.

Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. 2019c. CAMP: cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5763–5772, Seoul, Korea.

Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020. Multi-modality cross attention network for image and sentence matching. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10938–10947, Seattle, WA.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021a. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 3208–3216, Virtual Event.

Tan Yu, Yi Yang, Yi Li, Xiaodong Chen, Mingming Sun, and Ping Li. 2020. Combo-attention network for baidu video advertising. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2474–2482, Virtual Event, CA.

Tan Yu, Yi Yang, Yi Li, Lin Liu, Hongliang Fei, and Ping Li. 2021b. Heterogeneous attention network for effective and efficient cross-modal retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1146–1156, Virtual Event, Canada.

Bowen Zhang, Hexiang Hu, Vihan Jain, Eugene Ie, and Fei Sha. 2020. Learning to represent image and text with denotation graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 823–839, Online.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2021. SPARTA: efficient open-domain question answering via sparse transformer matching retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 565–575, Online.

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal joint attribute prediction and value extraction for e-commerce product. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2129–2139, Online.

A Appendix

The influence of fragment interactions. To obtain the text-image similarity, our base model exploits 32^2 fragment interactions, and the fast model only uses 1 fragment interaction. We evaluate the performance of the intermediate models with $\{1^2, 2^2, 4^2, 8^2, 16^2, 32^2\}$ fragment interactions. In the implementation, we use the first l fragments from the output of the encoder when computing the text-image score and vary l among $\{1, 2, 4, 8, 16, 32\}$. When $l = 1$, it is equivalent to our fast model. When $l = 32$, it is equivalent to our base model. The results of the models with different l are presented in Table 6. Note that we do not use inflating and shrinking in these models.

Frag.	Inter.	MSCOCO1K					
		T-to-I R@			I-to-T R@		
		1	5	10	1	5	10
1	1^2	56.1	85.9	91.8	69.4	92.7	97.1
2	2^2	59.7	87.6	93.1	71.6	94.9	98.2
4	4^2	61.8	89.1	94.8	76.2	95.8	98.3
8	8^2	63.8	89.9	95.7	78.7	95.6	98.5
16	16^2	65.7	91.6	96.1	80.1	97.3	99.1
32	32^2	68.1	91.9	96.4	82.0	97.6	99.2

Table 6: Models with different scales of interactions. Frag. indicates the number of fragments. Inter. indicates the number of fragment interactions.

We can observe from Table 6 that more fragment interactions yield better performance. In Table 1 of our main manuscript, the inflating operation plugs 32 codes in the input of the encoder, which further increases the number of fragment interactions to 128×128. It yields a higher retrieval accuracy.

Alternative settings for the student in shrinking. We have discussed the default settings for the shrinking operation, "B $\rightarrow$ F", in Table 2 of the main manuscript. Additionally, we conduct experiments of shrinking a base model to several intermediate models. As shown in Table 7, the shrunken model with more interactions gets better performance after knowledge distilling. Compared with models in Table 6 without inflating and shrinking, the shrunken models get considerably improved.

Alternative settings for the teacher in shrinking. Note that, in our shrinking operation, both teacher and student adopt the dual-encoder structure for encoding. An alternative choice is using the cross-BERT as the teacher. We compare with the alternative choice in Table 8.

	Inter.	MSCOCO1K					
		T-to-I R@			I-to-T R@		
		1	5	10	1	5	10
B	32^2	68.1	91.9	96.4	82.0	97.6	99.2
B $\rightarrow$ #	32^2	68.3	92.3	96.3	81.8	97.6	99.0
	16^2	67.7	92.1	96.5	80.7	97.3	99.3
	8^2	67.4	91.7	96.3	79.5	97.5	98.9
	4^2	66.8	91.7	96.4	79.9	97.1	98.5
	2^2	66.5	91.8	96.0	79.4	97.2	98.6
	1^2	66.0	91.7	96.1	79.1	96.8	98.6

Table 7: Additional results of shrinking. B denotes the base model, and the symbol $\rightarrow$ denotes the knowledge distilling from the left model to the right model. # indicates the settings with varying interactions.

	Inter.	MSCOCO1K					
		T-to-I R@			I-to-T R@		
		1	5	10	1	5	10
F	1^2	56.1	85.9	91.8	69.4	92.7	97.1
B	32^2	68.1	91.9	96.4	82.0	97.6	99.2
C	$\gg 32^2$	68.4	92.3	97.0	81.5	97.7	99.0
B $\rightarrow$ F	1^2	66.0	91.7	96.1	79.1	96.8	98.6
C $\rightarrow$ F	1^2	60.1	89.1	94.8	73.4	93.9	97.0

Table 8: Comparison with shrinking from a cross-BERT teacher. F denotes the fast model, B denotes the base model, C denotes the cross-BERT model and the symbol $\rightarrow$ denotes the knowledge distilling.

As shown in the table, distilling from a cross-BERT teacher to a fast model does not achieve a competitive retrieval accuracy. This is due to the large gap between the architecture of the teacher and that of the student.

Alternative pipelines for inflating and shrinking. Since we can construct intermediate models with different number of fragments and codes, there are a number of choices for inflating and shrinking. We have investigated the effectiveness the "I $\rightarrow$ B" and "I $\rightarrow$ B $\rightarrow$ F" pipelines in Table 2 of the main manuscript. In this section, we explore several other options for multi-step inflating and shrinking. We mainly change three key factors: number of additional codes, number of fragments, number of shrinking steps. We conduct experiments following three main strategies: I shrinks interactions and then removes codes; II removes codes and then shrink interactions; and III remove codes and shrink interactions at the same time. Table 9 shows the results. As shown in the table, II performs marginally better than others. Finally, 4 is selected for our final strategy, which obtains an excellent performance in a simple manner.

Table 9: Various pipelines of sequentially inflating and shrinking on MSCOCO1K. B denotes the base model, F denotes the fast model, I denotes the inflated model, B^+ and F^+ donate intermediate models, and the symbol $\rightarrow$ denotes the knowledge distilling from the left model to the right model.

		Settings	Codes	Frag.	Inter.	MSCOCO1K					
						T-to-I R@			I-to-T R@		
						1	5	10	1	5	10
baseline		B	0	32	32^2	68.1	91.9	96.4	82.0	97.6	99.2
		B $\rightarrow$ F	0	1	1^2	66.0	91.7	96.1	79.1	96.8	98.6
I	1	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ F$^+$	32	1	1^2	67.5	91.4	96.2	81.3	96.9	98.6
		I $\rightarrow$ F$^+$ $\rightarrow$ F	0	1	1^2	67.3	91.0	96.0	81.0	96.3	98.6
	2	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B	32	32	32^2	70.3	92.8	97.0	83.0	97.7	99.3
		I $\rightarrow$ B $\rightarrow$ F	0	1	1^2	67.6	91.4	96.9	80.5	96.5	98.9
	3	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B	32	32	32^2	70.3	92.8	97.0	83.0	97.7	99.3
		I $\rightarrow$ B $\rightarrow$ F$^+$	32	1	1^2	67.6	91.1	96.3	81.0	96.9	98.8
		I $\rightarrow$ B $\rightarrow$ F$^+$ $\rightarrow$ F$^+$	8	1	1^2	67.6	91.0	96.4	80.4	96.7	98.8
		I $\rightarrow$ B $\rightarrow$ F$^+$ $\rightarrow$ F$^+$ $\rightarrow$ F	0	1	1^2	66.8	90.8	96.4	80.1	96.8	98.5
II	4	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B	0	32	32^2	70.5	93.0	97.3	83.7	97.5	99.6
		I $\rightarrow$ B $\rightarrow$ F	0	1	1^2	68.5	92.5	97.1	82.4	97.3	99.1
	5	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B$^+$	16	48	48^2	70.4	93.1	97.1	83.7	97.7	99.4
		I $\rightarrow$ B$^+$ $\rightarrow$ F	0	1	1^2	67.9	91.8	96.9	81.9	97.4	99.0
	6	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B$^+$	16	48	48^2	70.4	93.1	97.1	83.7	97.7	99.4
		I $\rightarrow$ B$^+$ $\rightarrow$ B	0	32	32^2	70.5	92.9	97.0	83.5	97.4	99.0
		I $\rightarrow$ B$^+$ $\rightarrow$ B $\rightarrow$ F$^+$	0	8	8^2	69.1	92.7	96.5	82.4	97.2	99.1
		I $\rightarrow$ B$^+$ $\rightarrow$ B $\rightarrow$ F$^+$ $\rightarrow$ F	0	1	1^2	68.3	92.6	96.5	82.0	96.9	99.0
III	7	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ F	0	1	1^2	67.7	92.3	96.5	81.1	96.1	98.3
	8	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ F$^+$	8	8	8^2	68.8	92.0	96.3	81.9	96.4	98.9
		I $\rightarrow$ F$^+$ $\rightarrow$ F	0	1	1^2	67.6	91.4	98.4	81.3	96.2	98.9
	9	I	32	64	64^2	69.3	92.4	96.7	82.6	97.6	99.0
		I $\rightarrow$ B$^+$	16	32	32^2	70.2	92.9	97.0	83.3	97.6	99.6
		I $\rightarrow$ B$^+$ $\rightarrow$ F$^+$	8	16	16^2	69.7	92.6	97.0	83.0	96.3	99.1
		I $\rightarrow$ B$^+$ $\rightarrow$ F$^+$ $\rightarrow$ F$^+$	4	8	8^2	69.3	92.3	96.9	82.6	96.4	99.0
		I $\rightarrow$ B$^+$ $\rightarrow$ F$^+$ $\rightarrow$ F$^+$ $\rightarrow$ F	0	1	1^2	68.0	91.9	96.4	82.2	96.3	98.7