

# Thinking Fast and Slow: Efficient Text-to-Visual Retrieval with Transformers

Antoine Miech<sup>1\*</sup>      Jean-Baptiste Alayrac<sup>1\*</sup>  
 Ivan Laptev<sup>2</sup>      Josef Sivic<sup>3</sup>      Andrew Zisserman<sup>1,4</sup>  
<sup>1</sup>DeepMind    <sup>2</sup>ENS/Inria    <sup>3</sup>CIIRC CTU    <sup>4</sup>VGG Oxford  
 {miech, jalayrac}@google.com

## Abstract

*Our objective is language-based search of large-scale image and video datasets. For this task, the approach that consists of independently mapping text and vision to a joint embedding space, a.k.a. dual encoders, is attractive as retrieval scales and is efficient for billions of images using approximate nearest neighbour search. An alternative approach of using vision-text transformers with cross-attention gives considerable improvements in accuracy over the joint embeddings, but is often inapplicable in practice for large-scale retrieval given the cost of the cross-attention mechanisms required for each sample at test time. This work combines the best of both worlds. We make the following three contributions. First, we equip transformer-based models with a new fine-grained cross-attention architecture, providing significant improvements in retrieval accuracy whilst preserving scalability. Second, we introduce a generic approach for combining a Fast dual encoder model with our Slow but accurate transformer-based model via distillation and re-ranking. Finally, we validate our approach on the Flickr30K image dataset where we show an increase in inference speed by several orders of magnitude while having results competitive to the state of the art. We also extend our method to the video domain, improving the state of the art on the VATEX dataset.*

## 1. Introduction

Imagine yourself looking for an image that best matches a given textual description among thousands of other images. One effective way would be to first isolate a few promising candidates by giving a quick glance at all the images with a *fast* process, *e.g.* by eliminating images that

\*Equal contribution.

<sup>1</sup>Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.

<sup>3</sup>Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

<sup>4</sup>VGG, Dept. of Engineering Science, University of Oxford

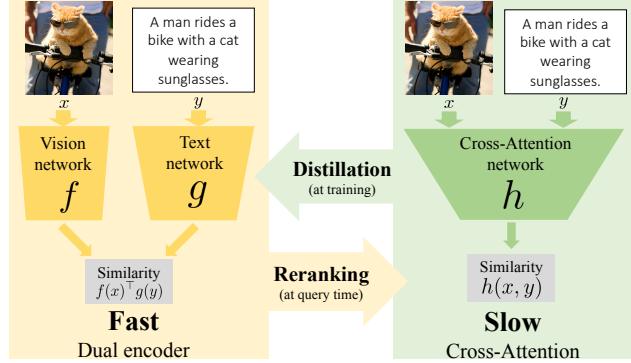


Figure 1: On the left, the *Fast* models, a.k.a dual encoders, independently process the input image and text to compute a similarity score via a single dot product, which can be efficiently indexed and is thus amenable to large-scale search. On the right, the *Slow* models, a.k.a cross-attention models, jointly process the input image and text with cross-modal attention to compute a similarity score. *Fast* and indexable models are improved by *Slow* models via distillation at training time (offline). *Slow* models are accelerated and improved with the distilled *Fast* approaches using a re-ranking strategy at query time.

have clearly nothing in common with the description. In the second phase, you may start paying more attention to image details with a *slow* process, *e.g.* by grounding individual words of a query sentence to make sure the scrutinized image is the best match.

Analogous to the *fast* process above, fast retrieval systems can be implemented by separately encoding visual and textual inputs into a joint embedding vector space where similarities can be computed by dot product. Such methods are regarded as *indexable*, *i.e.* they allow application of fast approximate nearest neighbour search [11, 32, 52, 64] and enable efficient billion-scale image retrieval. However, the accuracy of such methods is limited due to the simplicity of vision-text interaction model defined by the dot product in the joint embedding space. We refer to these techniques as *Dual Encoders (DE)* or *Fast approaches*.

Vision-text transformers compare each word to all loca-

tions in the image using cross-attention [12, 29, 46], allowing for grounding, and can be related to the *slow* process mentioned earlier. Such methods, referred to here as *Cross-attention (CA) or Slow approaches*, significantly boost retrieval performance. Modeling text-vision interactions with attention, however, makes these models slow and impractical for large-scale image retrieval given the cost of the cross-attention mechanisms required for each sample at test time. Hence, the challenge we consider is the following: How to benefit from accurate cross-attention mechanisms while preserving the fast and scalable visual search?

Our short answer is: By *thinking Fast and Slow* [10]. As illustrated in Figure 1, we propose to combine dual encoder approaches with cross-attention via two complementary mechanisms. First, we improve *Fast* DE models with a novel *distillation* objective that transfers knowledge from accurate but *Slow* CA models to the *Fast* and indexable dual encoders. Second, we propose to combine DE and CA models with re-ranking where a few most promising candidates obtained with the *Fast* model are re-ranked using the *Slow* model. Our resulting approach is both fast and accurate. Since the speed of CA is not a bottleneck anymore, we further improve performance by enriching the vision-text cross-attention model with a novel feature map upsampling mechanism enabling fine-grained attention. Note that our work can also be applied to vision-to-text retrieval. However, we focus on text-to-vision retrieval due to its wider practical application.

**Contributions.** (i) We first propose a gradual feature upsampling architecture for improved and fine-grained vision and text cross-attention. Our model is trained with a bi-directional captioning loss which is remarkably competitive for retrieval compared to standard cross-modal matching objectives. (ii) We introduce a generic approach for scaling-up transformer-based vision-text retrieval using two core ideas: a method to distill the knowledge of *Slow* cross-attention models into *Fast* dual-encoders, and re-ranking top results of the *Fast* models with the *Slow* ones. (iii) Finally, we validate our approach on image retrieval with the COCO [43] and Flickr30K [59] datasets and show we can reduce the inference time of powerful transformer-based models by 100× whilst also getting competitive results to the state of the art. We also successfully extend our approach to text-to-video retrieval and improve state of the art on the challenging VATEX [72] dataset.

## 2. Related work

**Vision and Language models.** Driven by the significant advances in language understanding lead by Transformers [13, 69], recent works have explored the use of these architectures for vision and language tasks. Many of them in image [8, 37, 39, 40, 46, 66, 67, 77] or video [78] rely on pretrained object detectors used for extracting ROIs

that are viewed as individual visual words. A few other works, such as PixelBERT [29] and VirTex [12] for images or HERO [38] for video, operate directly over dense feature maps instead of relying on object detectors. In these approaches, both vision and text inputs are fed into a Transformer-based model usually pretrained with multiple losses such as a cross-modal matching loss, a masked language modelling or a masked region modelling loss. Other non-Transformer based vision and text approaches used recurrent neural networks [14, 15, 36], MLP [70, 71], or bag-of-words [19, 50] text models. These models are then usually optimized with objectives such as CCA [19], max margin triplet loss [15, 70, 71, 73, 74], contrastive loss [23] and, more related to our work, by maximizing text log-likelihoods conditioned on the image [14]. In our work, we focus on the powerful vision-text Transformer models for retrieval and particularly address their scalability, which was frequently neglected by prior work.

**Language-based visual search.** A large number of vision and language retrieval models [15, 19, 20, 36, 49, 50, 54, 58, 70, 71, 73, 74, 76] use a dual encoder architecture where the text and vision inputs are separately embedded into a joint space. These approaches can efficiently benefit from numerous approximate nearest neighbour search methods such as: product quantization [32], inverted indexes [64], hierarchical clustering [52] or locality sensitive hashing [11], for fast and scalable visual search. In contrast, state-of-the-art retrieval models rely on large vision-text multimodal transformers [8, 29, 37, 39, 46, 47, 66, 67, 77]. In these approaches, both vision and text inputs are fed into a cross-modal attention branch to compute the similarity between the two inputs. This scoring mechanism based on cross-modal attention makes it particularly inadequate for indexing and thus challenging to deploy at a large scale. Our work aims at addressing this issue by connecting scalable visual search techniques with these powerful yet non-indexable vision-text cross-attention based models.

**Re-ranking.** Re-ranking retrieval results is standard in retrieval systems. In computer vision, the idea of geometric verification [31, 56] is used in object retrieval to re-rank objects that better match the query given spatial consistency criteria. Query expansion [9] is another re-ranking technique where the query is reformulated given top retrieved candidates, and recent work has brought attention mechanisms into deep learning methods for query expansion [21]. Related to language-based visual search, re-ranking by a video-language temporal alignment model has been used to improve efficient moment retrieval in video [16]. In contrast, we focus on transformer-based cross-attention models and develop a distillation objective for efficient retrieval.

**Distillation.** Knowledge distillation [3, 28] has proven to be effective for improving performance in various computer vision domains such as weakly-supervised learn-

ing [41, 60], depth estimation [22], action recognition [65], semantic segmentation [44], self-supervised learning [57] or self-training [75]. One major application of distillation is in compressing large and computationally expensive models in language analysis [62], object detection [5], image classification or speech recognition [28] into smaller and computationally less demanding models. In this work, we describe a distillation mechanism for the compression of powerful but non-indexable vision-text models into indexable models suitable for efficient retrieval.

### 3. Thinking Fast and Slow for Retrieval

This section describes our proposed approach to learn both fast and accurate model for language-based image retrieval.

Our goal is to train the model to output a similarity score between an input image  $x$  and a textual description  $y$ . In this work, we focus on two families of models: the *Fast* and the *Slow* models, as illustrated in Figure 1.

The *Fast* model, referred to as the *dual encoder approach*, consists of extracting modality-specific embeddings:  $f(x) \in \mathbb{R}^d$  for the image and  $g(y) \in \mathbb{R}^d$  for the text. The core property of this approach is that the similarity between an image  $x$  and a text  $y$  can be computed via a single dot product  $f(x)^\top g(y)$ . Hence, these methods can benefit from approximate nearest neighbour search for efficient large-scale retrieval [32, 52, 64].

The *Slow* model, referred to as the *cross-attention* approach differs by a more complex modality merging strategy based on cross-modal attention. We assume the given similarity score  $h(x, y)$  cannot be decomposed as a dot product and as such is not indexable. These models allow for richer interactions between the visual and textual representations, which leads to better scoring mechanisms, though at a higher computational cost.

Section 3.1 introduces the (*Slow*) cross-attention model considered in this work and details our contribution on the model architecture that leads to a more accurate text-to-image retrieval system. Section 3.2 describes how we obtain both a *fast* and *accurate* retrieval method by combining the advantages of the two families of models.

#### 3.1. Thinking Slow with cross-attention

Given an image  $x$  and a text description  $y$ , a *Slow* cross-attention retrieval model  $h$  computes a similarity score between the image and text as:

$$h(x, y) = A(\phi(x), y), \quad (1)$$

where  $\phi$  is a visual encoder (e.g. a CNN).  $A$  is a network that computes a similarity score between  $\phi(x)$  and  $y$  using cross-attention [46, 69] mechanisms, *i.e.* the text attends to the image or vice versa via multiple non-linear functions involving both the visual and language representations. Such

models emulate a *slow* process of attention which results in better text-to-image retrieval.

We propose two important innovations to improve such models. First, we introduce a novel architecture that enables fine-grained visual-text cross-attention by efficiently increasing the resolution of the attended high-level image features. Second, we propose to revisit the use of a captioning loss [14] to train retrieval models and discuss the benefits over standard alternatives that use classification or ranking loss [8, 37, 39, 46, 66, 67, 77].

#### A novel architecture for fine-grained vision-text cross-attention.

A typical approach to attend to visual features produced by a CNN is to consider the last convolutional layer [12, 29]. The feature map is flattened into a set of feature vectors that are used as input to vision-language cross-attention modules. For example, a  $224 \times 224$  input image passed through a ResNet-50 [26] outputs a  $7 \times 7$  feature map that is flattened into 49 vectors. While the last feature map produces high-level semantic information crucial for grounding text description into images, this last feature map is also severely downsampled. As a result, useful fine-grained visual information for grounding text descriptions might be lost in this downsampling process.

One solution to the problem is to increase the input image resolution. However, this significantly raises the cost of running the visual backbone. Inspired by previous work in segmentation [2, 25, 61] and human pose estimation [53], we instead propose to gradually upsample the last convolutional feature map conditioned on earlier higher resolution feature maps, as illustrated in Figure 2. We choose a lightweight architecture for this upsampling process inspired by recent advances in efficient object detection [68]. In Section 4, we show large improvements of this approach over several baselines and also show its complementarity to having higher resolution input images, clearly demonstrating the benefits of the proposed fine-grained vision-language cross-attention.

**Bi-directional captioning objective for retrieval.** A majority of text-vision transformer-based retrieval models [8, 37, 39, 46, 66, 67, 77] rely on a cross-modal image-text matching loss to discriminate positive image-text pairs  $(x, y)$  from negative ones. In this work, we instead explore the use of a captioning model for retrieval. Given an input text query  $y$ , retrieval can be done by searching the image collection for the image  $x$  that leads to the highest likelihood of  $y$  given  $x$  according to the model. In detail, we take inspiration from VirTex [12] and design the cross-attention module  $A$  as a stack of Transformer decoders [69] taking the visual feature map  $\phi(x)$  as an encoding state. Each layer of the decoder is composed of a masked text self-attention layer, followed by a cross-attention layer that enables the text to attend to the visual features and finally a feed forward

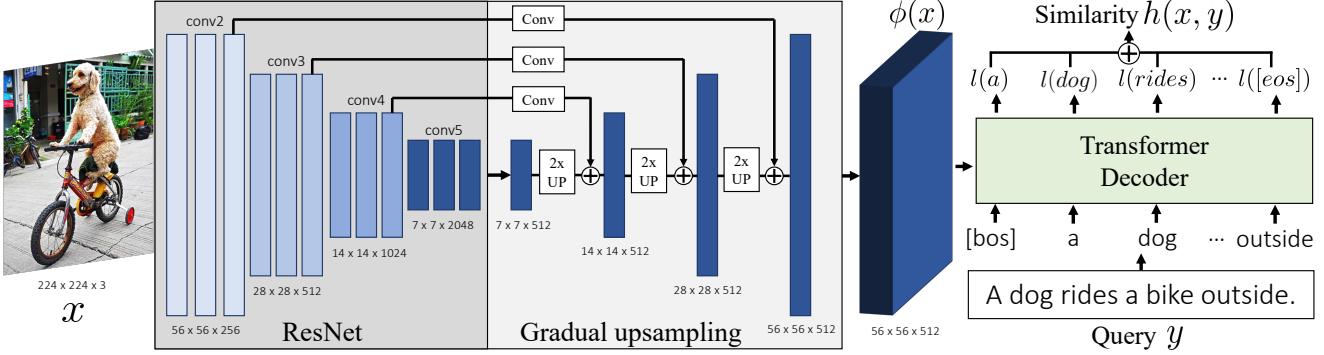


Figure 2: **Our Slow retrieval model** computes a similarity score  $h(x, y)$  between image  $x$  and query text description  $y$  by estimating the log-likelihood of  $y$  conditioned on  $x$ . In other words, given an input text query  $y$ , we perform retrieval by searching for an image  $x$  that is the most likely to decode caption  $y$ .  $l(\cdot)$  denotes the log probability of a word given preceding words and the image. The decoder is a Transformer that takes as the conditioning signal a high-resolution (here  $56 \times 56$ ) feature map  $\phi(x)$ . In this example,  $\phi(x)$  is obtained by gradually upsampling the last convolutional layer of ResNet ( $7 \times 7$ ) while incorporating features from earlier high-resolution feature maps. The decoder performs bidirectional captioning but, for the sake of simplicity, we only illustrate here the forward decoding transformer.

layer. One advantage of this architecture compared to standard multimodal transformers [8, 37, 39, 46, 66, 67, 77] is the absence of self-attention layers on visual features, which allows the resolution of the visual feature map  $\phi(x)$  to be scaled to thousands of vectors. We write the input text as  $y = [y^1, \dots, y^L]$  where  $L$  is the number of words. Formally, the model  $h$  scores a pair of image and text  $(x, y)$  as:

$$h(x, y) = h_{fwd}(x, y) + h_{bwd}(x, y), \quad (2)$$

where  $h_{fwd}(x, y)$  (resp.  $h_{bwd}(x, y)$ ) is the forward (resp. backward) log-likelihood of the caption  $y$  given the image  $x$  according to the model:

$$h_{fwd}(x, y) = \sum_{l=1}^L \log(p(y^l | y^{l-1}, \dots, y^1, \phi(x); \theta_{fwd})), \quad (3)$$

where  $p(y^l | y^{l-1}, \dots, y^1, \phi(x); \theta)$  corresponds to the output probability of a decoder model parametrized by  $\theta$  for the token  $y^l$  at position  $l$  given the previously fed tokens  $y^{l-1}, \dots, y^1$  and the encoded image  $\phi(x)$ .  $\theta_{fwd}$  is the parameters of the forward transformer models.  $h_{bwd}(x, y)$  is the same but with the sequence  $y^1, \dots, y^L$  in reverse order.

The parameters of the visual backbone, the forward and backward transformer models are obtained by minimizing  $\mathcal{L}_{CA} = -\sum_{i=1}^n h(x_i, y_i)$  where  $n$  is the number of annotated pairs of images and text descriptions  $\{(x_i, y_i)\}_{i \in [1, n]}$ .

We show in Section 4 that models trained for captioning can perform on-par with models trained with the usual contrastive image-text matching loss. At first sight this may appear surprising as the image-text matching loss seems more suited for retrieval, notably because it explicitly integrates negative examples. However, when looked at more closely, the captioning loss actually shares similarities with a contrastive loss: for each ground truth token of the sequence a

cross entropy loss is taken (see Eq. (3)) which effectively means that all other tokens in the vocabulary are considered as negatives.

In this section, we have described the architecture and the chosen loss for training our accurate *Slow* cross-attention model for retrieval. One key remaining challenge is in the scaling of  $h(x, y)$  using Eq. (1) to large image datasets as: (i) the network  $A$  is expensive to run and (ii) the resulting intermediate encoded image,  $\phi(x)$ , is too large to fit the entire encoded dataset in memory. Next, we introduce a generic method, effective beyond the scope of our proposed *Slow* model, for efficiently running such cross-modal attention-based models at a large scale.

### 3.2. Thinking Faster and better for retrieval

In this section, we introduce an approach to scale-up the *Slow* transformer-based cross-attention model, described in the previous section, using two complementary ideas. First, we distill the knowledge of the *Slow* cross-attention model into a *Fast* dual-encoder model that can be efficiently indexed. Second, we combine the *Fast* dual-encoder model with the *Slow* cross-attention model via a re-ranking mechanism. The outcome is more than  $100 \times$  speed-up and, interestingly, an improved retrieval accuracy of the combined *Fast* and *Slow* model. Next, we give details of the *Fast* dual encoder model, then explain the distillation of the *Slow* model into the *Fast* model using a teacher-student approach, and finally describe the re-ranking mechanism to combine the outputs of the two models. Because our approach is model agnostic, the *Slow* model can refer to any vision-text transformer and the *Fast* to any dual-encoder model. An overview of the approach is illustrated in Figure 1.

**Fast indexable dual encoder models.** We consider *Fast* dual encoder models, that extract modality specific embeddings:  $f(x) \in \mathbb{R}^d$  from image  $x$ , and  $g(y) \in \mathbb{R}^d$  from text  $y$ . The core property of this approach is that the similarity between the embedded image  $x$  and text  $y$  is measured with a dot product,  $f(x)^\top g(y)$ . The objective is to learn embeddings  $f(x)$  and  $g(y)$  so that semantically related images and text have high similarity and the similarity of unrelated images and text is low. To achieve that we train these embeddings by minimizing the standard noise contrastive estimation (NCE) [24, 33] objective:

$$\mathcal{L}_{\text{DE}} = -\sum_{i=1}^n \log \left( \frac{e^{f(x_i)^\top g(y_i)}}{e^{f(x_i)^\top g(y_i)} + \sum_{(x',y') \in \mathcal{N}_i} e^{f(x')^\top g(y')}} \right), \quad (4)$$

which contrasts the score of the positive pair  $(x_i, y_i)$  to a set of negative pairs sampled from a negative set  $\mathcal{N}_i$ . In our case, the image encoder  $f$  is a globally pooled output of a CNN while the text encoder  $g$  is either a bag-of-words [50] representation or a more sophisticated BERT [13] encoder. Implementation details are provided in Section 4.1.

**Distilling the *Slow* model into the *Fast* model.** Given the superiority of cross-attention models over dual encoders for retrieval, we investigate how to distill [28] the knowledge of the cross-attention model to a dual encoder. To achieve that we introduce a novel loss.

In detail, the key challenge is that, as opposed to standard distillation used for classification models, here we do not have a small finite set of classes but potentially an infinite set of possible sequences of words describing an image. Therefore, we cannot directly apply the standard formulation of distillation proposed in [28].

To address this issue, we introduce the following extension of distillation for our image-text setup. Given an image-text pair  $(x_i, y_i)$ , we sample a finite subset of image-text pairs  $\mathcal{B}_i = \{(x_i, y_i)\} \cup \{(x, y_i) | x \neq x_i\}$ , where we construct additional image-text pairs with the same text query  $y_i$  but different images  $x$ . Note that this is similar to the setup that would be used to perform retrieval of images  $x$  given a text query  $y_i$ . In practice, we sample different images  $x$  within the same training batch. We can write a probability distribution measuring the likelihood of the pair  $(x, y) \in \mathcal{B}_i$  according to the *Slow* teacher model  $h(x, y)$  (given by eq. (1)) over subset  $\mathcal{B}_i$  as:

$$p(\mathcal{B}_i)(x, y) = \frac{\exp(h(x, y)/\tau)}{\sum_{(x',y') \in \mathcal{B}_i} \exp(h(x', y')/\tau)}, \quad (5)$$

where  $\tau > 0$  is a temperature parameter controlling the smoothness of the distribution. We can obtain a similar distribution from the *Fast* student model, by replacing  $h(x, y)$

from Eq. (5) by  $f(x)^\top g(y)$ :

$$q(\mathcal{B}_i)(x, y) = \frac{\exp(f(x)^\top g(y)/\tau)}{\sum_{(x',y') \in \mathcal{B}_i} \exp(f(x')^\top g(y')/\tau)}. \quad (6)$$

Given the above definition of the sampled distributions, we use the following distillation loss that measures the similarity between the teacher distribution  $p(\mathcal{B}_i)$  and the student distribution  $q(\mathcal{B}_i)$  as :

$$\mathcal{L}_{\text{distill}} = \sum_{i=1}^n \mathcal{H}(p(\mathcal{B}_i), q(\mathcal{B}_i)), \quad (7)$$

where  $\mathcal{H}$  is the cross entropy between the two distributions. The intuition is that the teacher model provides soft targets over the sampled image-text pairs as opposed to binary targets in the case of a single positive pair and the rest of the pairs being negative. Similarly to the standard distillation [28], we combine the distillation loss (7) with the DE loss (4) weighted with  $\alpha > 0$  to get our final objective as:

$$\min_{f,g} \mathcal{L}_{\text{distill}} + \alpha \mathcal{L}_{\text{DE}}. \quad (8)$$

**Re-ranking the *Fast* results with the *Slow* model.** The distillation alone is usually not sufficient to recover the full accuracy of the *Slow* model using the *Fast* model. To address this issue, we use the *Slow* model at inference time to re-rank a few of the top retrieved candidates obtained using the *Fast* model. First, the entire dataset is ranked by the (Distilled) *Fast* model that can be done efficiently using approximate nearest neighbour search, which often has only sub-linear complexity in the dataset size. Then the top  $K$  (e.g. 10 or 50) results are re-ranked by the *Slow* model. As the *Slow* model is applied only to the top  $K$  results its application does not depend on the size of the database.

More precisely, given an input text query  $y$  and an image database  $\mathcal{X}$  containing a large number of  $m$  images, we first obtain a subset of  $K$  images  $\mathcal{X}_K$  (where  $K \ll m$ ) that have the highest score according to the *Fast* dual encoder model. We then retrieve the final top ranked image by re-ranking the candidates using the *Slow* model:

$$\arg \max_{x \in \mathcal{X}_K} h(x, y) + \beta f(x)^\top g(y), \quad (9)$$

where  $\beta$  is a positive hyper-parameter that weights the output scores of the two models. In the experimental Section 4, we show that *combined with distillation*, re-ranking less than ten examples out of thousands can be sufficient to recover the performance of the *Slow* model.

## 4. Experiments

In this section, we evaluate the benefits of our approach on the task of text-to-vision retrieval. We describe the

datasets and baselines used for evaluation in Section 4.1. In Section 4.2 we validate the advantages of cross-attention models with captioning objectives as well as our use of gradually upsampled features for retrieval. Section 4.3 evaluates the benefit of the distillation and re-ranking. In Section 4.4, we compare our approach to other published state-of-the-art retrieval methods in the image domain and show state of the art results in the video domain.

## 4.1. Datasets and models

**MS-COCO [43].** We use this image-caption dataset for training and validating our approach. We use the splits of [7] (118K/5K images for train/validation with 5 captions per image). We only use the first caption of each image to make validation faster for slow models. C-R@1 (resp. C-R@5) refers to recall at 1 (resp. 5) on the validation set.

**Conceptual Captions (CC) [63].** We use this dataset for training our models (2.7M training images (out of the 3.2M) at the time of submission). CC contains images and captions automatically scraped from the web which shows our method can work in a weakly-supervised training regime.

**Flickr30K [59].** We use this dataset for zero-shot evaluation (*i.e.* we train on COCO or CC and test on Flickr) in the ablation study, as well as fine-tuning when comparing to the state of the art. We use the splits of [34] (29K/1014/1K for train/validation/test with 5 captions per image). We report results on the validation set except in Section 4.4 where we report on the test split. We abbreviate F-R@1 (resp. F-R@5) as the R@1 (resp. R@5) scores on Flickr.

**VATEX [72].** VATEX contains around 40K short 10 seconds clip from the Kinetics-600 dataset [4] annotated with multiple descriptions. In this work, we only use the 10 English captions per video clip and ignore the additional Chinese captions. We use the retrieval setup and splits from [6].

**Models.** For each model, the visual backbone is a ResNet-50 v2 CNN [27] trained from scratch. Inputs are  $224 \times 224$  crops for most of the validation experiments unless specified otherwise. Models are optimized with ADAM [35], and a cosine learning rate decay [45] with linear warm-up is employed for the learning rate. The four main models used in this work are described next.

**NCE BoW** is a dual-encoder (DE) approach where the text encoder is a bag-of-words [50] on top of word2vec [51] pre-trained embeddings. The model is trained with the NCE loss given in Eq. (4) where the negative set  $\mathcal{N}_i$  is constructed as in [48]. We refer to **NCE BoW** as the *Fast* approach.

**NCE BERT** is a DE approach where the text encoder is a pretrained BERT base model [13]. We take the [CLS] output for aggregating the text representation. The model is also trained with the NCE loss given in Eq. (4).

**VirTex** [12] is a cross-attention (CA) based approach that originally aims at learning visual representations from text data using a captioning pretext task. We chose this visual

| Model                                  | Type | Train | F-R@1       | F-R@5       | C-R@1       | C-R@5       |
|--|------|-------|-------------|-------------|-------------|-------------|
| <i>Fast</i> NCE BoW<br>NCE BERT        | DE   | COCO  | 27.2        | 54.1        | 24.8        | 53.7        |
|  |      |       | 24.4        | 48.0        | 24.2        | 52.0        |
| PixelBERT<br>VirTex Fwd only<br>VirTex | CA   | COCO  | 30.0        | 55.1        | 25.1        | 52.5        |
|  |      |       | 33.4        | 58.1        | 31.8        | 61.2        |
|  |      |       | <b>38.1</b> | <b>62.8</b> | <b>35.1</b> | <b>64.6</b> |
| <i>Fast</i> NCE BoW<br>NCE BERT        | DE   | CC    | 32.4        | 59.6        | 14.9        | 35.0        |
|  |      |       | 25.8        | 50.7        | 12.2        | 29.8        |
|  |      |       | 30.4        | 57.7        | 14.1        | 33.6        |
| PixelBERT<br>VirTex Fwd only<br>VirTex | CA   | CC    | 32.2        | 58.4        | 14.7        | 32.9        |
|  |      |       | <b>35.0</b> | <b>60.7</b> | <b>16.1</b> | <b>36.4</b> |

Table 1: Dual encoder (DE) and Cross-attention (CA) comparison. F-R@K corresponds to the recall at K on Flickr while C-R@K is the recall at K on COCO.

| Feature map          | Size | F-R@1       | F-R@5       | C-R@1       | C-R@5       |
|----------------------|------|-------------|-------------|-------------|-------------|
| Slow 96x96           | 384  | <b>44.8</b> | <b>70.5</b> | <b>39.0</b> | <b>67.7</b> |
| Slow 56x56           |      | 42.2        | 66.8        | 38.5        | 65.2        |
| Slow 28x28           | 224  | 40.4        | 66.3        | 37.4        | 66.8        |
| Slow 14x14           |      | 39.2        | 63.8        | 36.8        | 64.9        |
| VirTex conv5 (7x7)   |      | 38.1        | 62.8        | 35.1        | 64.6        |
| VirTex conv4 (14x14) | 224  | 38.9        | 64.4        | 34.9        | 63.5        |
| VirTex conv3 (28x28) |      | 32.4        | 57.9        | 30.4        | 58.3        |
| VirTex conv2 (56x56) |      | 20.6        | 41.1        | 18.3        | 43.0        |

Table 2: Gradual upsampling with different feature map size. Size denotes the input image size. Models are trained on COCO.

captioning model as another point of comparison for the effectiveness of Transformer-based captioning models for text-to-vision retrieval.

**PixelBERT** [29] is a CA approach trained with the standard masked language modelling (MLM) and image-text matching (ITM) losses for retrieval. One difference between our implementation and the original PixelBERT is the use of  $224 \times 224$  images for a fair comparison with other models. Note that the main difference with VirTex is in the vision-text Transformer architecture: PixelBERT uses a deep 12-layer Transformer encoder while VirTex uses a shallow 3-layer Transformer decoder to merge vision and language.

We chose PixelBERT and VirTex for their complementarity and their simplicity since they do not rely on object detectors. We reimplemented both methods so that we could ensure that they were comparable. Next, we describe the details of our proposed CA approach.

**Slow model architecture.** For the upsampling, we follow a similar strategy as used in BiFPN [68]. For the decoder, we use a stack of 3 Transformer decoders with hidden dimension 512 and 8 attention heads. Full details about the architecture are provided in Appendix D.

## 4.2. Improving cross-attention for retrieval

In this section, we provide an experimental study on the use of cross-attention models for retrieval. All our results are validated on the COCO and the Flickr30K validation

| Student     | Teacher                 | Train | F-R@1       | F-R@5       | C-R@1       | C-R@5       |
|-------------|-------------------------|-------|-------------|-------------|-------------|-------------|
| <i>Fast</i> | None                    | COCO  | 27.2        | 54.1        | 24.8        | 53.7        |
|             | <i>Slow</i>             |       | <b>37.7</b> | <b>64.7</b> | <b>32.5</b> | <b>62.1</b> |
|             | <i>Slow</i> upper bound |       | 42.2        | 66.8        | 38.5        | 65.2        |
| <i>Fast</i> | None                    | CC    | 32.4        | 59.6        | 14.9        | 35.0        |
|             | <i>Slow</i>             |       | <b>33.4</b> | <b>60.1</b> | <b>17.2</b> | <b>38.1</b> |
|             | <i>Slow</i> upper bound |       | 41.7        | 67.5        | 19.8        | 40.9        |

Table 3: Distillation experiment with our proposed *Slow* approach as teacher and the *Fast* NCE BoW as student.

sets with models pretrained on COCO and CC training sets. Our main findings are summarized below.

**Cross-attention models are better than Dual Encoders.** Table 1 compares various approaches for retrieval. We observe that cross-attention models (PixelBERT and the VirTex variants), overall, outperform the dual encoders (NCE BoW and BERT). Interestingly, using a simple BoW text encoder performs better than using a BERT text encoder for the DE models. This suggests that the complexity of the language model is not the key factor for good performance but instead that complex merging strategy obtained from text-vision cross-attention may matter most for retrieval.

**Captioning models are surprisingly good for retrieval.** Comparing ‘PixelBERT’ against the ‘VirTex Fwd only’ in Table 1 with the exact same input dimensions and visual backbones, we see that using a captioning loss leads to better results than using an image-text matching loss coupled with a masked language modelling loss. Backward captioning further improves retrieval performance. This result demonstrates that captioning can be a strong alternative to the usual image-text matching losses for retrieval.

**Benefits of our gradual upsampling architecture design.** In Table 2, we provide the results using the proposed upsampling strategy for our *Slow* model presented in Section 3.1 and illustrated in Figure 2. We observe significant improvements over the VirTex baseline, denoted with conv5 (7x7), (more than 4% for R@1 on Flickr and more than 3% on COCO) for the largest upsampling  $56 \times 56$ . We also confirm that the performance gap does not just come from having a larger input feature map to attend to as the baseline with the output of ResNet conv2, which has a resolution of  $56 \times 56$ , performs poorly. We believe it is important to keep high-level abstraction in the feature maps while having high resolution which our proposed architecture allows. It is also important to highlight that the proposed architecture leads to our best performing model and can be combined with higher input resolution for further improvements. However, our proposed changes increase the inference time. Next, we explore how to recover the speed.

### 4.3. Thinking Fast and Slow

This section focuses on getting both a fast and accurate model for retrieval. First, we evaluate the benefit of the dis-

| Model                  | Top K | Dist. | Train | F-R@1       | F-R@5       | C-R@1       | C-R@5       |               | F-Qt          | C-Qt |
|------------------------|-------|-------|-------|-------------|-------------|-------------|-------------|---------------|---------------|------|
| <i>Slow</i>            | X     | X     | COCO  | 44.8        | 70.4        | 39.0        | 67.7        | 4 s           | 19 s          |      |
|                        | 10    | X     |       | 44.0        | 63.0        | 38.6        | 61.5        | <b>0.12 s</b> | <b>0.12 s</b> |      |
|                        | 10    | ✓     |       | 47.2        | 70.1        | 40.5        | 67.8        | <b>0.12 s</b> | <b>0.12 s</b> |      |
| <i>Fast &amp; Slow</i> | 50    | X     | CC    | 46.7        | 65.6        | 40.2        | 68.2        | 0.60 s        | 0.60 s        |      |
|                        | 50    | ✓     |       | <b>47.6</b> | <b>73.2</b> | <b>40.9</b> | <b>70.0</b> | 0.60 s        | 0.60 s        |      |
|                        | 50    | ✓     |       | 46.9        | 71.5        | 21.0        | 43.3        | 4 s           | 19 s          |      |
| <i>Slow</i>            | 10    | X     | CC    | 47.7        | 66.6        | 22.6        | 41.1        | <b>0.12 s</b> | <b>0.12 s</b> |      |
|                        | 10    | ✓     |       | 48.4        | 67.4        | 22.7        | 43.4        | <b>0.12 s</b> | <b>0.12 s</b> |      |
|                        | 50    | X     |       | 50.2        | 73.4        | <b>23.8</b> | <b>46.9</b> | 0.60 s        | 0.60 s        |      |
|                        | 50    | ✓     |       | <b>50.5</b> | <b>73.6</b> | <b>23.8</b> | <b>46.9</b> | 0.60 s        | 0.60 s        |      |

Table 4: Combination of re-ranking and distillation. Dist.: distillation. F-Qt (resp. C-Qt) is the query time in seconds on Flickr with 1k images (resp. COCO with 5k images) using 1x V100 GPU.

tillation from the *Slow* to the *Fast* model. Next, we evaluate the benefit of the re-ranking strategy and validate our combined approach on a large-scale retrieval experiment.

**Distillation improves dual encoder models.** In Table 3, we use our approach, denoted as *Slow*, to distill the knowledge to a *Fast* NCE BoW student dual encoder. The distillation improves the performance of the *Fast* model with improvements of over 10% on R@1 when training on COCO, significantly reducing the gap between the *Slow* and *Fast* models. On the other hand, the improvements when training on CC are moderate, but we believe the gap can be further reduced by training longer on CC as we found the distillation often takes significantly longer to converge.

**Benefits of re-ranking.** Table 4 provides the results from re-ranking. We see that with  $K$  as low as 10, we are able to recover or outperform the performance of the *Slow* model in terms of R@1 while significantly decreasing the query time. Combining re-ranking with distillation leads to further improvements: on COCO, we can significantly decrease from  $K = 50$  to  $K = 10$  the number of examples to re-rank to outperform the *Slow* model thanks to the distillation. In particular, we see a  $100\times$  reduction in retrieval time on COCO from our *Slow* to our *Fast & Slow* ( $K=10$ ) model. Note that for the rest of the experimental section, the *Slow* model runs with an increased image resolution of  $384 \times 384$  for better results, albeit with slower inference.

Figure 3 provides a more detailed visualization of the effect of re-ranking with respect to the number of top  $K$  examples returned from the *Fast* distilled model. Notably, we see on COCO that *re-ranking as few as five images out of five thousand* from the distilled *Fast* model is enough to reach the *Slow* model R@1 performance. More quantitative and qualitative results are given in Appendix B.

**Discussion of scalability.** We would like to emphasize that the combination of the distillation and re-ranking would be even more appealing in the large-scale retrieval regime as our method allows application of fast approximate nearest neighbour search [11, 32, 52, 64] and hence can potentially scale to billion-scale image retrieval. As a result, our method scales sub-linearly with the number of test images and the time complexity mostly depends on the top  $K$ ,

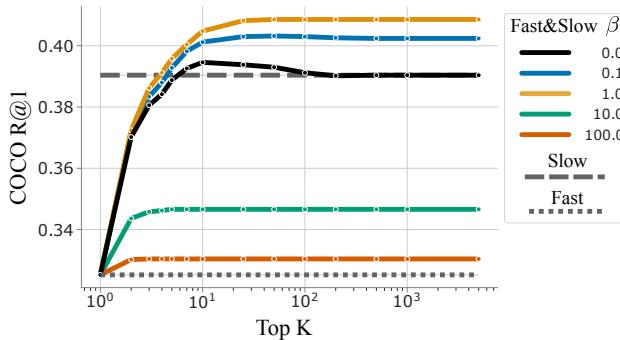


Figure 3: Retrieval result when varying the top-K retrieved examples from the distilled *Fast* model with varying  $\beta$  (See Eq. (9)).

which is the number of calls to the *Slow* model.

#### 4.4. Comparison to the state of the art

We compare to the state of the art on Flickr30K in Table 5 for the zero-shot and fine-tuning setting.

The *Fast* model is distilled from the *Slow* model on the pretraining dataset (COCO or CC). The *Fast* model and the *Slow*  $384 \times 384$  models are then fine tuned on the Flickr30K training set. When pretraining on CC, we significantly outperform the VilBERT [46] approach despite not using extra information contained in object detectors. On COCO, we outperform PixelBERT [29] with the same ResNet-50 backbone while neither training on Visual Genome (VG) annotations nor using high image resolution. Finally, we are still below the performance reported in UNITER [8] and OSCAR [40]. We believe this remaining gap can be attributed to (*i*) not using the same amount of pretraining data (UNITER was trained on the combination of four datasets: COCO, CC but also Visual Genome (VG) and SBU and OSCAR is trained on Flickr, CC, SBU and GQA [30]), (*ii*) not using the same high input image resolution, (*iii*) not relying on pre-trained object detectors, and (*iv*) having a smaller model (3 layers transformer with hidden dimension 512 vs. 24 layers with dimension 1024 for UNITER). However our proposed approach enables fast retrieval at scale which is not possible out of the box with any of the previously mentioned methods. More importantly, our scaling approach (distillation and re-ranking) can also be applied to other multimodal transformers including UNITER and OSCAR.

**Extension to video.** Our approach can also be applied to video. To do so, we extend the architecture introduced in Section 3.1 to a TSM ResNet50 model [42] with the following modifications. The input of the network is now a sequence of 32 frames at resolution  $224 \times 224$ . Due to memory constraints, we only upsample the last feature map to a  $14 \times 14$  grid and allow the decoder to attend to the resulting spatio-temporal volume representing the video of shape  $32 \times 14 \times 14$  (details in Appendix D.2). We use a pretrained

| Method                            | Object Det. | Size | Train                       | Zero-shot | F-R@1       | F-R@5       | F-R@10      |
|-----------------------------------|-------------|------|-----------------------------|-----------|-------------|-------------|-------------|
| VILBERT [46]                      | ✓           | Full |                             | ✓         | 31.9        | 61.1        | 72.8        |
| <i>Fast and Slow</i> (K=100)      | ✗           | 384  | CC                          |           | <b>48.7</b> | <b>74.2</b> | <b>82.4</b> |
| VILBERT [46]                      | ✓           | Full |                             | ✗         | 58.2        | 84.9        | 91.5        |
| <i>Fast and Slow</i> (K=100)      | ✗           | 384  |                             |           | <b>68.2</b> | <b>89.7</b> | <b>93.9</b> |
| PixelBERT (R50) [29]              | ✗           | 800  | COCO<br>+VG                 | ✗         | 59.8        | 85.5        | <b>91.6</b> |
| <i>Fast and Slow</i> (R50, K=100) |             | 384  | COCO                        | ✗         | <b>62.9</b> | <b>85.8</b> | 91.3        |
| Unicoder-VL [37]                  | ✓           | Full | CC + SBU                    | ✗         | 71.5        | 90.9        | 94.9        |
| UNITER [8]                        | ✓           | Full | COCO<br>+CC<br>+SBU<br>+VG  | ✗         | 75.6        | <b>94.1</b> | <b>96.8</b> |
| OSCAR [40]                        | ✓           | Full | COCO<br>+CC<br>+SBU<br>+GQA | ✗         | <b>75.9</b> | 93.3        | 96.6        |
| <i>Fast and Slow</i> (K=100)      | ✗           | 384  | COCO<br>+CC                 | ✗         | 72.1        | 91.5        | 95.2        |

Table 5: Comparison to state of the art for text-to-image retrieval. OSCAR results were reproduced from recent work [17].

| Method   | R@1         | R@5         | R@10        |
|--|-------------|-------------|-------------|
| Dual [15]                                      | 31.1        | 67.4        | 78.9        |
| HGR [6]  | 35.1        | 73.5        | 83.5        |
| Support-set [55]                               | 45.9        | 82.4        | 90.4        |
| <i>Fast NCE BoW</i>                            | 42.3        | 79.1        | 88.0        |
| <i>Fast and Slow</i> ( $7 \times 7$ ) (K=10)   | 47.5        | 81.4        | 88.0        |
| <i>Fast and Slow</i> ( $14 \times 14$ ) (K=10) | <b>50.5</b> | 83.4        | 88.0        |
| <i>Fast and Slow</i> ( $14 \times 14$ ) (K=50) | <b>50.5</b> | <b>84.6</b> | <b>91.7</b> |

Table 6: Comparison to state of the art retrieval on VATEX.

TSM ResNet-50 network [1] on HowTo100M [48] and AudioSet [18] datasets. Results are given in Table 6. We observe that: (i) the upsampling architecture is also beneficial for video, and (ii) our Fast and Slow model sets a new state of the art on this benchmark.

## 5. Conclusion

We have shown how to scale-up powerful vision-text transformer-based models for retrieval. In particular, we have introduced an accurate but *Slow* text-vision transformer-based architecture with fine-grained cross-attention for retrieval. To make it scalable for text-to-visual search, we have augmented this *Slow* model with a *Fast* dual encoder model through a combination of distillation and re-ranking. As a result, the combined *Fast & Slow* approach achieves better results than the *Slow* model while significantly reducing the inference time by several orders of magnitude on large datasets. We emphasize that our approach is model agnostic and can be applied to any vision-text Transformer *Slow* model and dual-encoder *Fast* retrieval model.

**Acknowledgements.** We would like to thank Lisa Anne Hendricks for feedback. The project was partially funded by the French ANR as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15\_003/0000468).

## References

- [1] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-Supervised MultiModal Versatile Networks. In *NeurIPS*, 2020. 8
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 3
- [3] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006. 2
- [4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 6
- [5] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, 2017. 3
- [6] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, 2020. 6, 8
- [7] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6
- [8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In *ECCV*, 2020. 2, 3, 4, 8
- [9] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007. 2
- [10] Kahneman Daniel. Thinking, fast and slow, 2017. 2
- [11] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, 2004. 1, 2, 7
- [12] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. *arXiv preprint arXiv:2006.06666*, 2020. 2, 3, 6
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 2, 5, 6
- [14] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2, 3
- [15] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 2, 8
- [16] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*, 2019. 2
- [17] Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulic, and Iryna Gurevych. Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval. *arXiv preprint arXiv:2103.11920*, 2021. 8
- [18] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 8
- [19] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 2
- [20] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 2
- [21] Albert Gordo, Filip Radenovic, and Tamara Berg. Attention-based query expansion learning. In *ECCV*, 2020. 2
- [22] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016. 3
- [23] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *ECCV*, 2020. 2
- [24] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 5
- [25] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 3
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 6
- [28] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3, 5, 11
- [29] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020. 2, 3, 6, 8, 12
- [30] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 8
- [31] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 2
- [32] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *PAMI*, 2010. 1, 2, 3, 7
- [33] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016. 5
- [34] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 6
- [35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6
- [36] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 2
- [37] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Dixin Jiang,

- and Ming Zhou. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training. In *AAAI*, 2020. 2, 3, 4, 8
- [38] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020. 2
- [39] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2, 3, 4
- [40] Xiuju Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 2, 8
- [41] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, 2017. 3
- [42] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 8, 13
- [43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 2, 6
- [44] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *CVPR*, 2019. 3
- [45] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6
- [46] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2, 3, 4, 8
- [47] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020. 2
- [48] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020. 6, 8
- [49] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv preprint arXiv:1804.02516*, 2018. 2
- [50] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 2, 5, 6
- [51] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 6
- [52] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP*, 2009. 1, 2, 3, 7
- [53] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3
- [54] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2
- [55] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020. 8
- [56] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2
- [57] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020. 3
- [58] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *CVPR*, 2017. 2
- [59] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2, 6
- [60] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, 2018. 3
- [61] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3
- [62] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 3
- [63] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 6
- [64] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1, 2, 3, 7
- [65] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *WACV*, 2020. 3
- [66] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2019. 2, 3, 4
- [67] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 2, 3, 4
- [68] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020. 3, 6, 12
- [69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3
- [70] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *PAMI*, 2018. 2
- [71] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2
- [72] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research.

- In ICCV, 2019. 2, 6
- [73] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 2
- [74] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *ICCV*, 2017. 2
- [75] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 3
- [76] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015. 2
- [77] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, 2020. 2, 3, 4
- [78] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020. 2

## A. Appendix

We provide here more details about the main paper. Section B gives additional ablation results for the distillation method and the Flickr re-ranking curve (similarly to the COCO re-ranking curve in the main paper). In Section C, we provide additional details about our training hyperparameters. Section D describes in more details the proposed architecture for upsampling as well as the video architecture extension. Finally, in Section E we provide qualitative results of our approach.

## B. Additional quantitative results

**What matters for good distillation.** In Table 7, we explore various text models for the *Fast* dual-encoder student when performing distillation. Interestingly, the BoW model still seems to be the best fit for distillation, hinting that complex language models are not necessarily the most important for the task we consider. This is in line with our findings in the main paper about the use of more complex language models for **Noise-Contrastive Estimation (NCE)** (Equation 4 from the main paper) that did not lead to improvements. Table 8 shows that care should be given to the choice of temperature used when performing the distillation and that combining the distillation loss with the original loss is crucial to ensuring improvements. Note that we follow [28] and adapt the combination factor  $\alpha$  with respect to the temperature parameter  $\tau$ . Based on that study, we use  $\tau = 10$  and  $\frac{\alpha}{\tau^2} = 0.001$  for all other distillation experiments of the paper. The models were trained on COCO for 50k steps.

**Flickr re-ranking results when varying K.** Figure 4 provides a more detailed visualization of the effect of re-ranking with respect to the number of top K examples returned from the *Fast* distilled model on the Flickr validation

| Text model              | Depth | F-R@1       | F-R@5       | C-R@1       | C-R@5       |
|-------------------------|-------|-------------|-------------|-------------|-------------|
| Bag-of-words            | 1     | <b>35.6</b> | <b>60.8</b> | <b>31.2</b> | <b>61.1</b> |
|                         | 1     | 27.4        | 51.7        | 20.1        | 45.0        |
| Transformer             | 3     | 26.3        | 49.7        | 19.4        | 43.6        |
|                         | 6     | 27.1        | 49.9        | 20.0        | 44.0        |
|                         | 12    | 28.9        | 50.0        | 19.6        | 43.5        |
| <i>Slow</i> upper bound |       | 42.2        | 66.8        | 38.5        | 65.2        |

Table 7: Distillation: Which text model to use for the dual encoder approach on COCO.

| $\tau$ | $\frac{\alpha}{\tau^2}$ | F-R@1 | F-R@5 | C-R@1 | C-R@5 |
|--------|-------------------------|-------|-------|-------|-------|
| 1.0    | 0.0                     | 26.8  | 54.0  | 24.1  | 52.7  |
|        | 0.1                     | 28.2  | 54.0  | 24.9  | 55.8  |
|        | 1.0                     | 27.2  | 51.1  | 25.2  | 52.8  |
|        | 10.0                    | 19.1  | 39.6  | 19.0  | 42.6  |
| 10.0   | 0.0                     | 34.0  | 61.6  | 31.1  | 61.0  |
|        | 0.1                     | 35.7  | 61.0  | 30.9  | 61.2  |
|        | 1.0                     | 34.5  | 61.0  | 31.7  | 61.0  |
|        | 10.0                    | 27.2  | 53.4  | 26.2  | 54.6  |

Table 8: Distillation temperature and loss weighting ablation study. The models were trained on COCO for 50k steps.

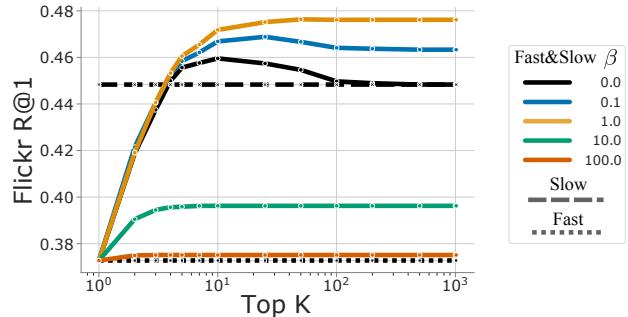


Figure 4: Flickr retrieval result when varying the top-K retrieved examples from the distilled *Fast* model with varying  $\beta$ .

set. The models were trained on COCO. Here again, we see that *re-ranking as few as four images out of thousand* from the distilled *Fast* model is enough to reach the *Slow* model R@1 performance.

## C. Experiment details

**Training time data augmentation.** We randomly crop images using the tensorflow function `tf.image.sample_distorted_bounding_box`<sup>1</sup> with the following parameters:

<sup>1</sup>[https://www.tensorflow.org/api\\_docs/python/tf/image/sample\\_distorted\\_bounding\\_box](https://www.tensorflow.org/api_docs/python/tf/image/sample_distorted_bounding_box)

- `min_object_covered=0.2`,
- `aspect_ratio_range=(3 / 4, 4 / 3)`,
- `area_range=(0.2, 1.0)`.

The crops are then resized to  $224 \times 224$  and randomly flipped from left to right with a probability of 0.5. Note that while flipping an image from left to right is a common data augmentation technique in classification, it can be problematic in captioning annotation that mention parts of images specifically to the left or right. However, we counted that on Conceptual Caption around 1.3% of the captions either mention the word `left` or `right` while this percentage is down on COCO to 0.6% which is sufficiently low to not cause a problem. For our video experiment we use the same spatial augmentation but additionally subsample temporal clips of 32 frames at 10 frame per second (3.2 seconds) from the original 10 seconds clips of VATEX.

**Test time data augmentation.** For all image experiments, we simply take the central crop of the image to perform the retrieval. For videos, we sample 4 temporally uniformly clips over the video. Each clip has 32 frames that are centrally cropped. We average the visual-text score  $h(x, y)$  over the 4 clips to obtain the final score.

**Training hyper-parameters.** In the following, we provide the optimization details for each of the trained models on COCO and CC. We recall that each model is trained using the ADAM optimizer with a cosine learning rate decay and 5k steps of warm up.

- **NCE BoW/BERT:** The models are trained both on COCO and CC using a total batch size of 1024 and a base learning rate of 0.001. On COCO, the model is trained for 20k steps while it is trained for 140k steps on CC. A gradient clip of 30.0 is applied. When performing distillation, we instead train longer for COCO until 100k steps. The weight decay is set to 0.0001.
- **VirTex and Slow:** The models are trained both on COCO and CC using a base learning rate of 0.0004. On COCO, the model is trained for 250k steps with a batch size of 512 while it is trained for 500k steps with a batch size of 1024 on CC. A gradient clip of 100.0 is applied. The weight decay is set to 0.0001.
- **PixelBERT:** The models are trained both on COCO and CC using a total batch size of 1024 and a base learning rate of 0.0001. On both CC and COCO, the model is trained for 800k steps. A gradient clip of 30.0 is applied. The weight decay is set to 0.0001. Note that because we worked with smaller resolution images compared to the original PixelBERT work, we

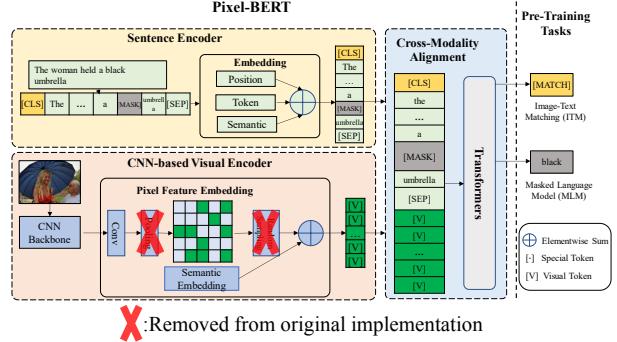


Figure 5: Illustration of the PixelBERT [29], architecture with the pooling and sampling modules we have removed from the original implementation.

removed the downsampling block (which follows the ResNet backbone and is composed of a max pooling and random sampling of pixels from the feature map). Instead we feed all the  $7 \times 7$  visual features to the transformer. Figure 5 provides an illustration of the architecture, taken from the original work [29], with the modules we have removed to deal with smaller image resolution.

The models are trained using the JAX deep learning framework.

## D. Architecture details

### D.1. Upsampling strategy

We follow the same upsampling mechanism used in the BiFPN architecture [68]. In particular we perform the Fast normalized fusion approach:

$$P^{out} = \text{SepConv} \left( \frac{w_1 \cdot P^{in} + w_2 \cdot \text{Resize}(P^{prev})}{w_1 + w_2 + \epsilon} \right)$$

where  $w_i \geq 0$  is ensured by applying a ReLU after each  $w_i$ ,  $\epsilon = 0.0001$  is a small value to avoid numerical instability,  $\text{Resize}$  is a  $2 \times$  upsampling using a nearest neighbour interpolation,  $P^{in}$  is the feature map input of the upsampling block and  $P^{prev}$  is the feature map from the previous ResNet feature maps with its dimensionality reduced to 512 through a  $1 \times 1$  convolution. SepConv is a separable depth wise convolution layer followed by a batch normalization and ReLU. A more detailed illustration of the upsampling architecture is illustrated in Figure 2.

## D.2. Video architecture with upsampling

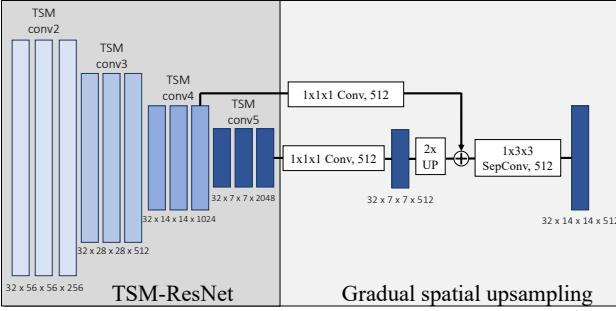


Figure 6: **Upsampling architecture for videos:** adaptation of the upsampling architecture for the TSM network.

In order to apply our architecture to video for our experiment on VATEX, we adapt the image only architecture to a video one that can handle spatio-temporal attention. For that, we start from the TSM ResNet-50 architecture [42]. This architecture consists of a standard ResNet-50 model where Temporal Shift Modules (TSM) are inserted within each residual block. The Temporal Shift Module enables temporal modeling by moving features along the time dimension (forward and backward in time). One particularity of the model is that there is no temporal pooling hence the temporal resolution stays the same everywhere in the network. For that reason, we use the same spatial upsampling strategy that we develop for the image network as illustrated in Figure 6 and simply adapts it to deal with the additional temporal dimension. Due to memory constraints, we only spatially upsample the feature map to a  $14 \times 14$ . We use clips of 32 frames sampled at 10 frame per second for our VATEX experiment (3.2 seconds of video). As a result, the Transformer can attend to a  $32 \times 14 \times 14$  spatio-temporal feature map.

## E. Qualitative results

### E.1. Retrieval results

We also provide retrieval results on Flickr using four approaches: *Fast* w/o distillation, *Fast*, *Slow* and *Fast and Slow* ( $K=50$ ). Figure 7 illustrates one retrieval example, where we show the top-5 retrieved examples for each model (first row: *Fast* w/o distillation, second row: *Fast*, third row: *Slow*). The last row shows the top  $K=50$  re-ranked examples from the *Fast* using the *Slow* model. The models are trained on COCO and evaluated on Flickr in a zero-shot manner. Note that we have biased the results towards examples that failed for the *Fast* w/o distillation model but are successfully retrieved with re-ranking.

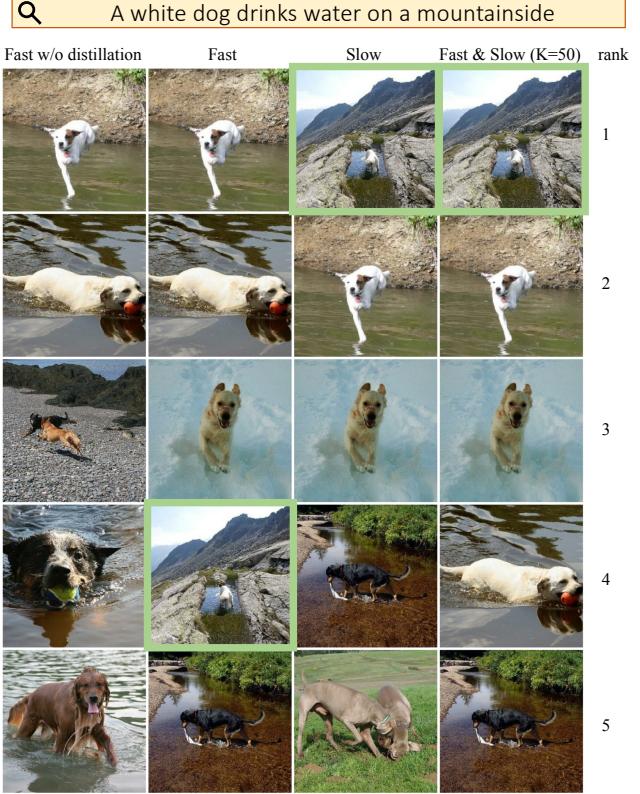


Figure 7: Retrieval qualitative examples on Flickr. First column: *Fast* w/o distillation, Second column: *Fast*, Third column: *Slow* and the last column shows top  $K=50$  re-ranked examples from the *Fast* using the *Slow* model. The image with the green bounding box is the annotated groundtruth for the query. Note that the groundtruth image is not retrieved in the top-5 with the *Fast* model without distillation and is retrieved at the 4th place with distillation. The *Slow* model correctly retrieves it at the first place similarly to the re-ranking model. Moreover, we can see that the *Slow & Fast* approach is the only one that either retrieves a white dog or a dog on water.

### E.2. Attention maps

In this section, we provide a qualitative analysis of the attention maps between the text and the input image in the Transformer model. We start by describing how we obtain these attention maps and how we display them. Next, we provide our main observations and findings from analyzing these feature maps. To conduct this qualitative study, we compare two models trained on COCO: (i) our *Slow* model (see Figure 8) which can attend to the  $56 \times 56$  upsampled feature map and (ii) our reimplementation of the VirTeX model (see Figure 9) with decoder heads attending to a  $7 \times 7$  feature map.

**Extracting and visualizing attention maps.** Recall that our textual decoder is a 3 layer Transformer. Each of these layers can perform cross-attention between an input word token and the whole image feature map of size  $H \times H$  in order to output prediction scores for the next word. In detail, at each layer and for each input text token, a *query* text vector is produced and compared to the precomputed  $H \times H$  visual *keys* via dot product. The resulting unnormalized  $H \times H$  scores are then normalized with a softmax. These normalized weights are then used to aggregate  $H \times H$  visual vectors, or *values*, that are used to update the current token representation in order to predict the scores of the next word. This process is done in parallel for 8 attention heads before concatenating the outputs of all heads in a single vector.

This gives an opportunity to visualize the attention maps to see what are the regions of the images that are attended to in order to output a given word. We provide such visualization in Figure 8 and Figure 9. Each figure has multiple examples shown in different rows. On the left, we show the input image in color. On the bottom right, we show the input tokens shifted backward by one word so that we directly visualize what the attention maps look like for the prediction of the current word. On the right, we show the attention maps for the different layers of the transformer in yellow overlaid over the gray image. The attention map rows are ordered so that the bottom one corresponds to the layer closest to the input text. The attention maps are resized to the original input image resolution ( $224 \times 224$ ) via bicubic sampling. We only show a single attention head per token and per layer by selecting the one that has the highest average score over the  $H \times H$  grid prior to applying the softmax. For that study, we use as inputs the ground truth captions and images from the Flickr validation set in order to emulate what happens when the Transformer model is used for retrieval.

**Analysis.** Looking at Figure 8 and Figure 9 we make the following observations.

First of all, in both cases we see that there is some level of coherency between the attended regions and the tokens being predicted. For example, in the second row of Figure 8 (the woman with the bicycle), we see that the last layer of the transformer attends to the mouth of the woman for the word “smiling”, at the shirt of the woman to predict the color “peach” of the top, and finally attends to the region containing the bicycle to predict the last word.

Second, we observe that, as expected, the attention maps obtained from our *Slow* model in Figure 8 is indeed fine grained when compared to the original VirTeX attention maps of Figure 9. This can notably be seen on the last row of the figures, where the *Slow* model can attend more precisely to the faces of the children thanks to the higher

$56 \times 56$  resolution feature map compared to the crude  $7 \times 7$  feature map of the original VirTeX.

Third, we note that for the *Slow* model, the attention becomes more focused for the higher layers that are closer to the output. This is notably true when being input the first word where the attention systematically covers the full image for the first layer before being refined on a specific region in the image.

Finally, while these visualizations are not always perfectly interpretable, we believe similar studies and inspections are valuable to better understand how these models relate text and vision.

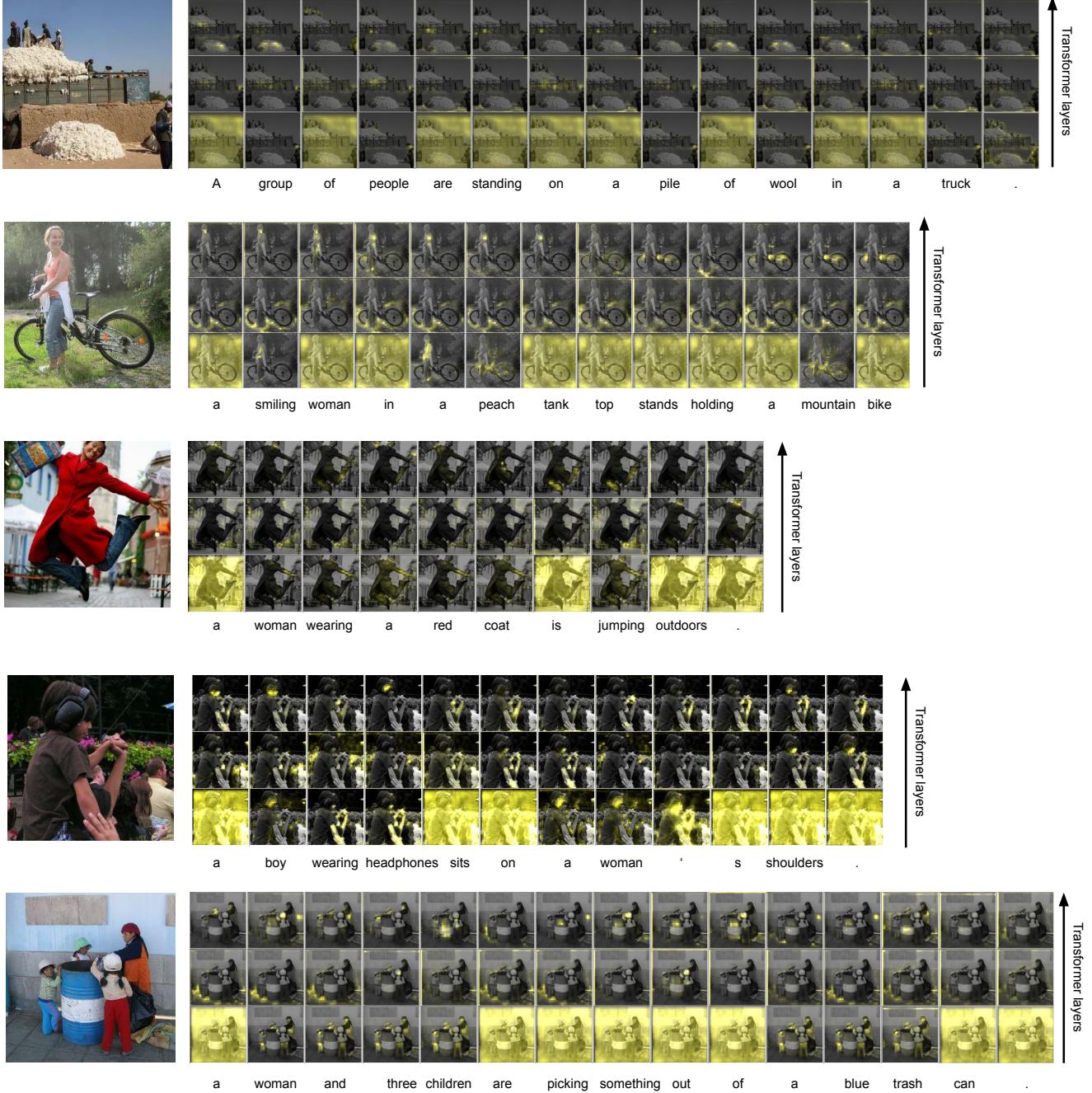


Figure 8: Attention maps visualization for the *Slow* model with the  $56 \times 56$  upsampled feature map. For each image, the attention maps are given so that the bottom row corresponds to the transformer layer closest to the input text. See main text in Appendix E.2 for details. Best seen in color on a screen.

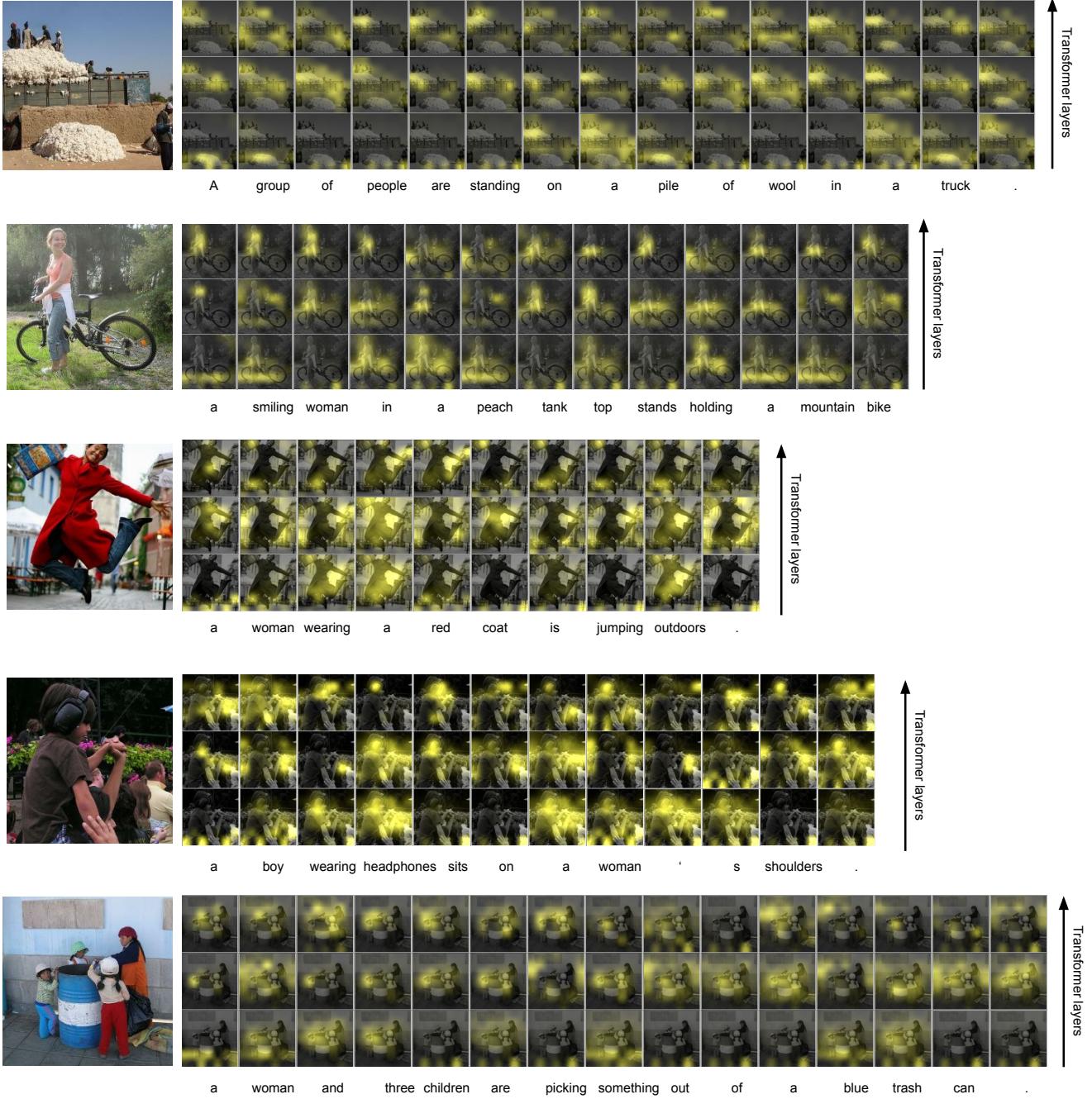


Figure 9: Attention maps visualization for VirTex (7 × 7 attention map). For each image, the attention maps are given so that the bottom row corresponds to the transformer layer closest to the input text. See main text in Appendix E.2 for details. Best seen in color on a screen.