Composing Object Relations and Attributes for Image-Text Matching

Supplementary Material

6. Implementation Details

6.1. Visual features

We use the pre-extracted Faster R-CNN 2048-dimensional region features from BUTD [1, 23], which is the standard convention from prior work in the image-text matching literature [4, 12]. To transform them to have the same dimensions D as the joint embedding space, we implement a 2-layer MLP with residual connection. The region features are then pooled using the GPO [4] pooling operator into $\mathbb{R}^D$.

6.2. Textual features

Bi-GRU. The dimension of the word embedding is set to 300 for both experiments where we initialize the word embedding from GloVe or from scratch (refer to Sec. 8 for experiment of CORA without using GloVe). The GRU has 1 layer and its hidden dimension is also 300.

BERT. Similar to prior work, we use the `bert-base-uncased` architecture and pre-trained weights for the BERT semantic concept encoder. As mentioned in the main paper, using BERT to encode short phrases (*e.g.*, *construction worker, sitting*) does not take advantage of the full capability of BERT. BERT has never seen short text during its pre-training stage [8], and with its ability to capture long-range dependencies, BERT is more suitable for encoding long sentences. As a result, direct fine-tuning BERT for CORA leads to slightly lower results.

In our work, instead of fine-tuning the whole BERT model (with 110M params), we employ the prefix tuning technique P-Tuning v2 [29] in order to repurpose the pre-trained BERT model into encoding short phrases. With this technique, at every BERT encoding layer, a sequence of learnable N token embeddings $\mathbb{R}^{N\times 768}$ is added as prefix into the textual prompt. Intuitively, these tokens provide learnable context that assist BERT into learning the task at hand, which is encoding short phrases. The number of trainable params with P-Tuning v2 is only $2N \times L \times 768$ (where $L = 12$ is the number of BERT encoding layers, and $N = 24$ is the number of prefix tokens). In our experiment, we find fine-tuning the last BERT layer along with P-Tuning gives slightly better results. In overall, the number of trainable params of our BERT component is only 7M, which is much smaller than 110M params of the whole BERT model.

For both types of features (Bi-GRU and BERT), we implement an FC layer to transform the semantic encoded output into $\mathbb{R}^D$ before using them to initialize the node and edge features of the GATs.

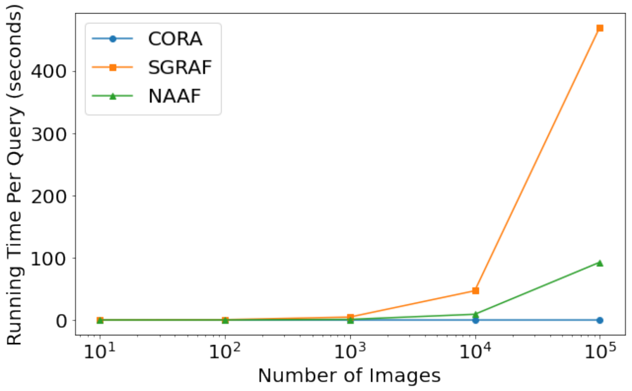


Figure 4. **Inference time comparison.** We compare the text-to-image retrieval inference time between our method CORA against two SOTA cross-attention methods SGRAF [9] and NAAF [57] (lower is better). The inference time is calculated with different number of images in the database. CORA with its dual-encoder architecture is much faster and scalable than cross-attention approaches.

6.3. Training and hyperparameter details

We use the AdamW optimizer [31] to train our model for 50 epochs. The learning rate is initialized at 5e-4, then decayed to 5e-5 after 15 epochs. The learning rate for the pre-trained components (*i.e.*, GloVe and BERT) is scaled by 0.1 w.r.t. the base learning rate. We set the batch size to 128 when training on Flickr30K, and 256 when training on MS-COCO. The margin α in the triplet loss is set to 0.4, while the cosine similarity in the contrastive loss is scaled by a temperature of 0.01 similar to CLIP [42]. Following [4], we perform size augmentation to randomly drop 35% region features. For data augmentation on the text, we perform subsampling on the scene graph by randomly dropping 10% of the nodes and edges and randomly masking 10% of the word tokens. We set $\lambda_{\text{CON}} = 0.25$ and $\lambda_{\text{SPEC}} = 3.0$.

7. Inference Time

We illustrate in Fig. 4 the inference time comparison between our method CORA against SOTA cross-attention methods SGRAF [9] and NAAF [57] with different number of images in the database (ranging from a very small to a very large number of images).

To conduct this experiment, for all methods, we first forward all images through the image encoder of each respective method in order to cache all image embeddings. Then, for CORA, when a text query arrives, it takes 0.04s to parse it into a scene graph, 0.014s to compute its scene graph em-

Table 5. **Our framework achieves the best results on the Flickr30K dataset when initializing the word embeddings fom scratch for the Bi-GRU semantic encoder.** Without the CA - “cross-attention”, our method still has competitive results to other baselines. † denotes methods that use ensembling of multiple models, and we highlight **the highest** and second-highest RSUM.

Method	Venue	CA	Image → Text			Text → Image			RSUM
			R@1	R@5	R@10	R@1	R@5	R@10	
Faster R-CNN + Bi-GRU									
SCAN† [23]	ECCV’18	✓	67.4	90.3	95.8	48.6	77.7	85.2	465.0
VSRN [24]	ICCV’19		71.3	90.6	96.0	54.7	81.8	88.2	482.6
SGM [51]	WACV’20	✓	71.8	91.7	95.5	53.5	79.6	86.5	478.6
GCN+DIST [25]	CVPR’20	✓	70.8	92.7	96.0	60.9	86.1	91.0	497.5
GSMN† [28]	CVPR’20	✓	76.4	94.3	97.3	57.4	82.3	89.0	496.8
CAAN [58]	CVPR’20	✓	70.1	91.6	97.2	52.8	79.0	87.9	478.6
VSE$_\infty$ [4]	CVPR’21		76.5	94.2	97.7	56.4	83.4	89.9	498.1
SGARF† [9]	AAAI’21	✓	77.8	94.1	97.4	58.5	83.0	88.8	499.6
MV-VSE† [26]	IJCAI’22		79.0	94.9	97.7	59.1	84.6	90.6	505.8
Ours			80.1	95.5	97.7	60.6	85.6	91.1	510.5
Ours†			81.7	95.5	98.1	62.0	86.6	91.8	**515.7**

bedding, then 0.01s to perform the vector-matrix multiplication with the image embeddings to find nearest neighbor results, which in total accounts to around 0.06s per query for all number of images from 10 to 10^5. On the other hand, for cross-attention approaches SGRAF and NAAF, when a text query arrives, these methods have to pair the text query with every image embedding in the database, then forward each pair through the cross-attention module in order to calculate their similarity. Fig. 4 shows that the inference time for SGRAF and NAAF scale up linearly w.r.t. the number of images in the database (*e.g.*, SGRAF takes 46s with 10^4 images, and 470s with 10^5 images), which is due to the iterative pairing of the input text with each image. Our model CORA enjoys the benefit of being fast and scalable of the dual-encoder architecture, while still achieving better retrieval results than SOTA cross-attention approaches (*e.g.*, SGRAF and NAAF).

8. More Ablation Studies

Initialize from GloVe vs. from scratch. When using Bi-GRU, we follow all recent studies [12, 20, 30, 35, 49, 57] to initialize the word embeddings using GloVe [37]. To fairly compare against other methods prior to these work, we also report our results when using Bi-GRU with word embeddings initialized from scratch in Tabs. 5 and 6 for the Flickr30K and MS-COCO dataset respectively. The results show that even when initializing the word embeddings from scratch, our method CORA still outperforms all previous work with and without cross-attention.

BERT P-Tuning v2. We compare between direct fine-tuning the whole BERT model against using P-Tuning v2 [29] to encode short phrases of semantic concepts. The results are displayed in Tab. 7. Note that this model is ablated without having multi-head self-attention in the visual encoder.

9. More Analysis

With larger visual backbone. We select ResNeXT-101 [55] pretrained on the Instagram dataset [33] as the larger visual extractor than the Faster R-CNN model used in our main experiments. This visual backbone is also reported in VSE$_\infty$ [4] and SDE [20]. The results of this experiment on the Flickr30K test set are displayed in Tab. 8, where it shows we obtain a large increase over region features and others.

Simulate parsing errors. As discussed in the conclusions, our CORA model is strongly dependent on the scene graph quality from the parser. To study this dependence, we simulate errors by performing the followings onto the parsed graphs: drop word tokens from nodes and edges, move attribute node to wrong object node, and move edge to wrong object pair. We randomly perform these onto 10%, 20%, 30% F30K captions and achieve 513.2, 512.2, 509.8 RSUM (original performance is 515.8). We observe that moving the edge affects performance more than moving the attribute.

Why consider CORA. CORA is a promising graph method that can supplement what CLIP (& other text encoders) may struggle against, *i.e.*, sentences with many semantics that are mixed among objects (discussed in Sec. 1). To show example, we select 100 sentences in Flickr30K with the highest number of attributes and relations, then evaluate image retrieval on them. We obtain results respectively for CLIP, HREM, CORA as 239.3, 240.0, 241.5 RSUM. This shows the large model CLIP is even slightly inferior to CORA on sentences that are rich in semantics.

Compare with pretrained image-text models. We scale up CORA on larger data and compare with prior SOTA image-text models in Tab. 9. We pretrain CORA on 1M image-text pairs in Conceptual Captions. All of the models in the table are finally fine-tuned on Flickr30K. Compared with CORA-BERT (refer to Tab. 1), CORA pretrained gets a +6.8 score. Despite smaller data and not using cross-attention, CORA is better than ViLBERT [32], UNITER [5], and can potentially reach Unicoder [17] with more data. However, it is inferior to CLIP zero-shot [42]. This shows the promising ability to scale up CORA further, *e.g.* by using ViT instead of regions, CLIP-text instead of BERT, and more data.

10. Qualitative Results

Image-to-text and image-to-entity retrieval. We illustrate some examples of successful and failed results when performing image-to-text retrieval using our CORA model with Faster R-CNN + BERT trained on the MS-COCO dataset in Fig. 5 and Fig. 6. Because CORA also has the ability to retrieve object entities, we also include image-to-

Table 6. **Our framework achieves the best results on the MS-COCO dataset when initializing the word embeddings fom scratch for the Bi-GRU semantic encoder.** Without the CA - "cross-attention", our method still has competitive results to other baselines. † denotes methods that use ensembling of multiple models, and we highlight **the highest** and second-highest RSUM.

			MS-COCO 5-fold 1K Test							MS-COCO 5K Test						
Method	Venue	Cross-	Image → Text			Text → Image			RSUM	Image → Text			Text → Image			RSUM
		Attention	R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10	
Faster R-CNN + Bi-GRU																
SCAN† [23]	ECCV'18	✓	72.7	94.8	98.4	58.8	88.4	94.8	507.9	50.4	82.2	90.0	38.6	69.3	80.4	410.9
VSRN [24]	ICCV'19		76.2	94.8	98.2	62.8	89.7	95.1	516.8	53.0	81.1	89.4	40.5	70.6	81.1	415.7
SGM [51]	WACV'20	✓	73.4	93.8	97.8	57.5	87.3	94.3	504.1	50.0	79.3	87.9	35.3	64.9	76.5	393.9
CAAN [58]	CVPR'20	✓	75.5	95.4	98.5	61.3	89.7	95.2	515.6	52.5	83.3	90.9	41.2	70.3	82.9	421.1
VSE$_\infty$ [4]	CVPR'21		78.5	96.0	98.7	61.7	90.3	95.6	520.8	56.6	83.6	91.4	39.3	69.9	81.1	421.9
SGARF† [9]	AAAI'21	✓	79.6	96.2	98.5	63.2	90.7	96.1	524.3	57.8	-	91.6	41.9	-	81.3	-
MV-VSE† [26]	IJCAI'22		78.7	95.7	98.7	62.7	90.4	95.7	521.9	56.7	84.1	91.4	40.3	70.6	81.6	424.6
Ours			80.5	96.0	98.6	62.9	90.6	96.0	524.6	60.4	85.1	91.7	40.5	70.6	81.2	429.5
Ours†			81.2	96.2	98.7	63.4	90.9	96.2	**526.6**	60.9	85.6	92.0	40.8	71.0	81.8	**432.1**

Table 7. **Ablation studies** to compare between fine-tuning the whole BERT model versus using P-Tuning v2 [29] to encode the short phrases of semantic concepts. The models are evaluated on the MS-COCO 1K Test set. Gray denotes our best model.

Method	Image-to-text			Text-to-image			RSUM
	R@1	R@5	R@10	R@1	R@5	R@10	
BERT	82.0	96.5	98.8	64.5	91.1	96.2	529.1
BERT P-Tuning, N = 8	81.7	96.5	99.0	64.5	91.1	96.1	528.9
BERT P-Tuning, N = 16	81.4	96.9	98.8	65.0	91.2	96.3	529.6
BERT P-Tuning, N = 24	81.9	96.6	98.9	65.0	91.2	96.4	530.0
BERT P-Tuning, N = 32	82.2	96.7	98.7	64.8	91.3	96.2	529.9

Table 8. Analysis on using the larger visual backbone ResNeXT-101 [55]. We plug ResNeXT-101 into our CORA model with BERT as the semantic concept encoder.

Method	Image-to-text			Text-to-image			RSUM
	R@1	R@5	R@10	R@1	R@5	R@10	
CORA-BERT	84.4	96.7	98.7	62.3	87.5	92.5	522.1
CORA-BERT + ResNeXT-101	90.9	99.1	99.8	76.8	95.1	97.7	559.4
VSE$_\infty$ + ResNeXT-101 [4]	88.7	98.9	99.8	76.1	94.5	97.1	555.1
SDE + ResNeXT-101 [16]	90.6	99.0	99.6	75.9	94.7	97.3	557.1

Table 9. Compare with pre-trained image-text models.

Method	RSUM
ViLBERT - NeurIPS'19 - Data: CC3M	502.7
UNITER - ECCV'20 - Data: CC3M, SBU	510.9
UNITER - ECCV'20 - Data: CC3M, SBU, VG, COCO	542.8
Unicoder-VL - AAAI'20 - Data: CC3M, SBU	538.8
CORA-BERT - Data: CC1M	530.1
CLIP zero-shot - ICML'21 - Data: CLIP 400M	540.6

entity retrieval results in the figures. The image-to-entity retrieval results also help display some of the biases of the model. One interesting application of image-to-entiy retrieval is for auto image tagging.

Among the examples in Fig. 5, the wrong matching texts and entities are understandable because they are still very semantically aligned with the input image. We explain each case below:

1. In the top image, all retrieved captions are correct. Among the retrieved entities, there are a few incorrect results which show that the model has not learned very accurately the visual appearance of *receipt, hairbrush, calendar*. Images of *toddler holding a hairbrush* is common in the training set, which must have made the model steered towards aligning *hairbrush* with something that *a toddler is holding*.
2. In the middle image, most matching captions are correctly retrieved except one that is incorrect due to object counting. Counting the correct number of objects is indeed a challenge for image-text matching model. The model also mistakenly recognizes the kite as a plane.
3. In the bottom image, the 1st caption is incorrect, but the model still ranks it at the top due to multiple semantic information in the text are still correct w.r.t. the image (*e.g*., *young boy, living room, cat*). All other captions are correctly retrieved. The image-to-entity retrievals show the concepts that the model does not grasp well.

We continue to explain the failure cases in Fig. 6 as following:

1. In the top image, all of the retrieved captions are incorrect matchings as determined by the ground truth data. However, we notice that the 2nd, 3rd captions still correctly describe the image to a certain extent. This is a weakness of the benchmark.
2. In the middle image, the model must have wrongly associated *skin* with *bikini*, hence why it retrieves captions with *bikini* at rank 4 and 5. In the entity retrieval results, interestingly, we notice the model returns *dental procedure* and *dental work*. We figure that the model must have aligned the action of *mouth opening* with *dental*, hence why these two entities are retrieved in this case.
3. In the bottom image, this is again an example of where the retrieved captions correctly describe the image, but because the ground truth data specify otherwise, they are

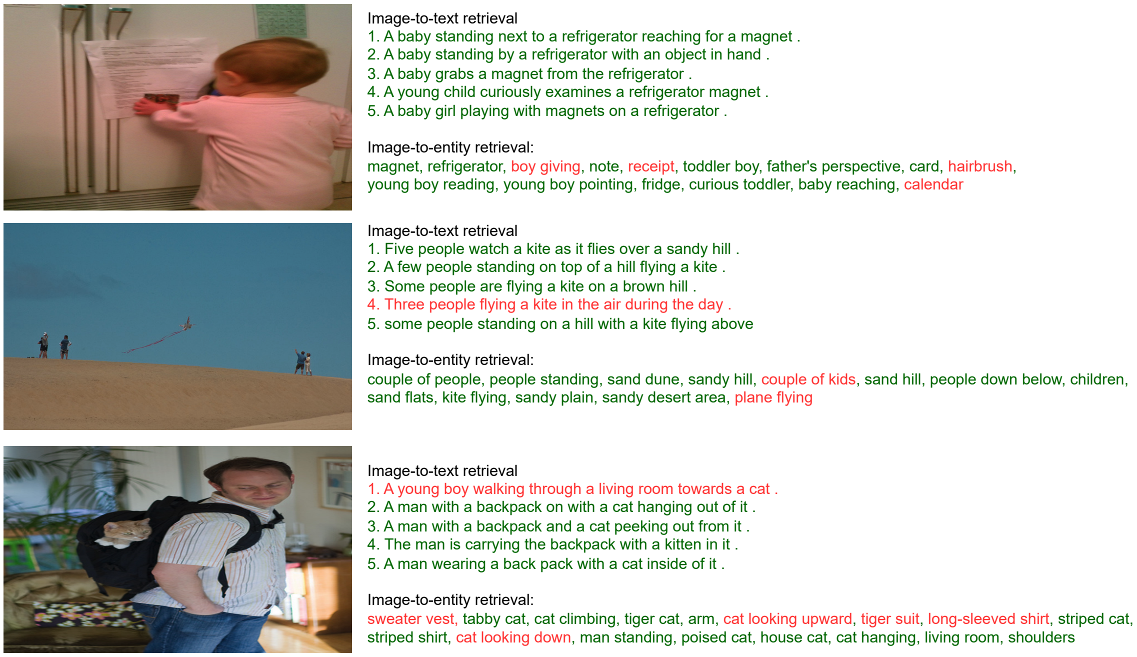


Image-to-text retrieval
1. A baby standing next to a refrigerator reaching for a magnet .
2. A baby standing by a refrigerator with an object in hand .
3. A baby grabs a magnet from the refrigerator .
4. A young child curiously examines a refrigerator magnet .
5. A baby girl playing with magnets on a refrigerator .

Image-to-entity retrieval:
magnet, refrigerator, boy giving, note, receipt, toddler boy, father's perspective, card, hairbrush, young boy reading, young boy pointing, fridge, curious toddler, baby reaching, calendar

Image-to-text retrieval
1. Five people watch a kite as it flies over a sandy hill .
2. A few people standing on top of a hill flying a kite .
3. Some people are flying a kite on a brown hill .
4. Three people flying a kite in the air during the day .
5. some people standing on a hill with a kite flying above

Image-to-entity retrieval:
couple of people, people standing, sand dune, sandy hill, couple of kids, sand hill, people down below, children, sand flats, kite flying, sandy plain, sandy desert area, plane flying

Image-to-text retrieval
1. A young boy walking through a living room towards a cat .
2. A man with a backpack on with a cat hanging out of it .
3. A man with a backpack and a cat peeking out from it .
4. The man is carrying the backpack with a kitten in it .
5. A man wearing a back pack with a cat inside of it .

Image-to-entity retrieval:
sweater vest, tabby cat, cat climbing, tiger cat, arm, cat looking upward, tiger suit, long-sleeved shirt, striped cat, striped shirt, cat looking down, man standing, poised cat, house cat, cat hanging, living room, shoulders

Figure 5. **Successful image-to-text and image-to-entity retrieval on MS-COCO.** In image-to-text retrieval, green denotes matching text according to the ground truth of MS-COCO, while red denotes incorrect matching. In image-to-entity retrieval, green and red denote correct and incorrect matching, respectively, as judged subjectively by us.

considered incorrect by the benchmark.

Text-to-image retrieval results. We illustrate some text-to-image retrieval results in Fig. 7. In both examples, our model is able to retrieve the correct image at rank 1. The images from rank 2 to rank 5 all exhibit visual traits that match partially with the input text.

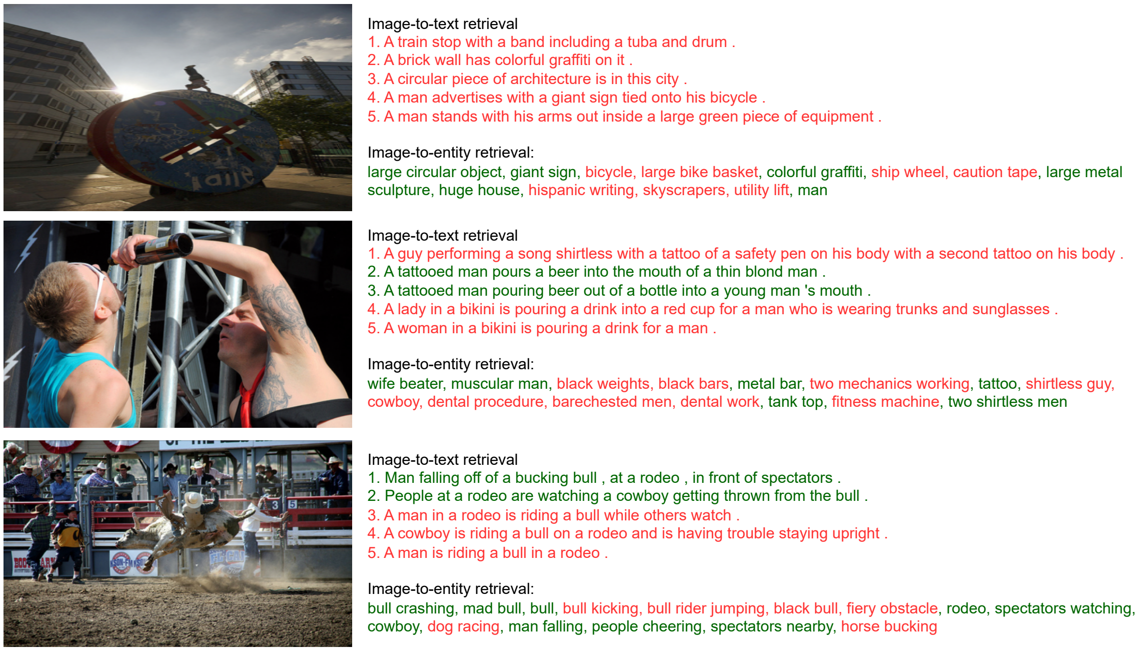


Image-to-text retrieval
1. A train stop with a band including a tuba and drum .
2. A brick wall has colorful graffiti on it .
3. A circular piece of architecture is in this city .
4. A man advertises with a giant sign tied onto his bicycle .
5. A man stands with his arms out inside a large green piece of equipment .

Image-to-entity retrieval:
large circular object, giant sign, bicycle, large bike basket, colorful graffiti, ship wheel, caution tape, large metal sculpture, huge house, hispanic writing, skyscrapers, utility lift, man

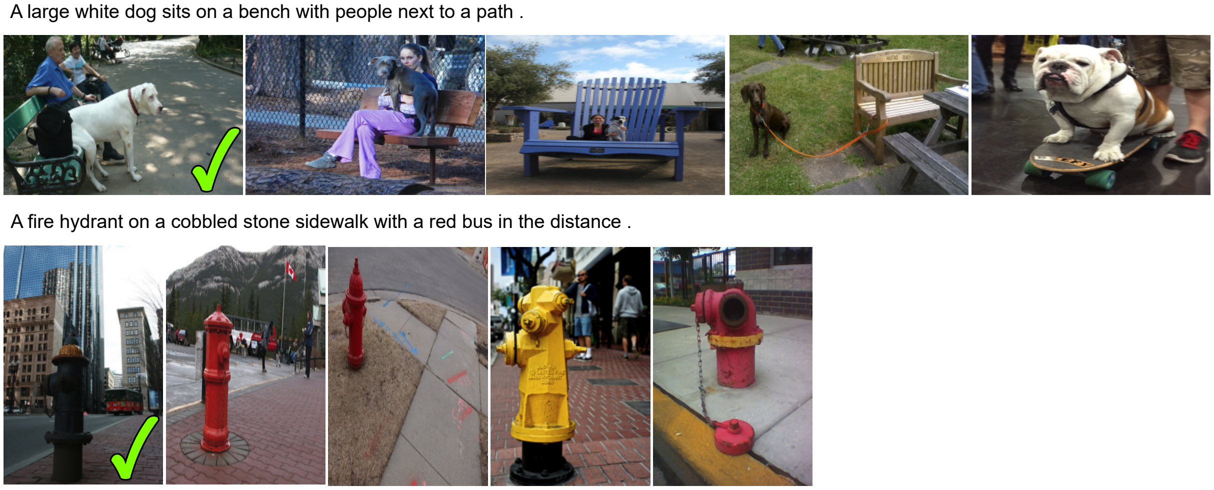
Image-to-text retrieval
1. A guy performing a song shirtless with a tattoo of a safety pen on his body with a second tattoo on his body .
2. A tattooed man pours a beer into the mouth of a thin blond man .
3. A tattooed man pouring beer out of a bottle into a young man 's mouth .
4. A lady in a bikini is pouring a drink into a red cup for a man who is wearing trunks and sunglasses .
5. A woman in a bikini is pouring a drink for a man .

Image-to-entity retrieval:
wife beater, muscular man, black weights, black bars, metal bar, two mechanics working, tattoo, shirtless guy, cowboy, dental procedure, barechested men, dental work, tank top, fitness machine, two shirtless men

Image-to-text retrieval
1. Man falling off of a bucking bull , at a rodeo , in front of spectators .
2. People at a rodeo are watching a cowboy getting thrown from the bull .
3. A man in a rodeo is riding a bull while others watch .
4. A cowboy is riding a bull on a rodeo and is having trouble staying upright .
5. A man is riding a bull in a rodeo .

Image-to-entity retrieval:
bull crashing, mad bull, bull, bull kicking, bull rider jumping, black bull, fiery obstacle, rodeo, spectators watching, cowboy, dog racing, man falling, people cheering, spectators nearby, horse bucking

Figure 6. **Failure cases of image-to-text and image-to-entity retrieval on Flickr30K.** In image-to-text retrieval, green denotes matching text according to the ground truth of Flickr30K, while red denotes incorrect matching. In image-to-entity retrieval, green and red denote correct and incorrect matching, respectively, as judged subjectively by us.



A large white dog sits on a bench with people next to a path .

A fire hydrant on a cobbled stone sidewalk with a red bus in the distance .

Figure 7. **Text-to-image retrieval on MS-COCO.** For every text, we show the top-5 retrieved images on MS-COCO. The image with the green tick mark is the correct matching according to ground truth in the dataset.