Supplementary Material

Telling the What while Pointing to the Where: Multimodal Queries for Image Retrieval

Soravit Changpinyo Jordi Pont-Tuset Vittorio Ferrari Radu Soricut
Google Research
{schangpi,jponttuset,vittoferrari,rsoricut}@google.com

A. Additional Implementation Details

A.1. Model

We expand the details of our "embedder," "contextualizer," and "pooler," described in the main text. Please refer to Figure 4 of the main paper for a high-level overview.

We use the same hyperparameter across Image-Region Embedder (IRE), Text-Token Embedder (TTE), and Trace-Box Embedder (TBE), described in the main text. Our feed-forward network (FFN) is a 2-layer MLP with the ReLU activation function, hidden size of 512, and a dropout rate of 0.3 (applied during training only). The dimension of position embedding (TTE, TBE) is set to 512. To obtain a vector of size 512 from a visual box or a trace box of $xmin, xmax, ymin, ymax, area$ (IRE, TBE). We perform a linear projection of this 5D vector into 512D and concatenate the result before feeding this 2560D vector into FFN.

Our image (or text) transformer encoder has 6 layers, a vocab embedding of size 512, a hidden embedding of size 1024, a filter size of 4096. The number of attention heads is set to 8.

Our image (or text) pooler is a mean pooling layer, followed by a 2-layer MLP with the ReLU activation function, hidden size of 2048, and a dropout rate of 0.3 (applied during training only).

A.2. Learning

We expand the description of our learning procedure in the main text. For all experiments, we use Adam [1] with default hyperparameters. We use Google Cloud 32-core TPUs, with a batch size per core of 128. The FRCNN regional features are permuted during training. We use a linear warmup of 20 epochs, and a decay rate of 0.95 every 25 epochs for all experiments.

For all pre-training experiments, we use an initial learning rate of 0.000032 The maximum number of training steps is set to 1M for pre-training on Conceptual Captions and to 250K for pre-training on Open Images Localized Narratives.

We tune an initial learning rate when training or fine-tuning on Flickr30k Localized Narratives. For from-scratch experiments — with or without the mouse trace — we find that a slightly higher learning rate of 0.000096 works best. The maximum number of training steps is set to 70K.

For fine-tuning experiments, there are two cases. When both (latest) pre-training and fine-tuning stages use the same inputs, i.e., with or without the mouse trace in both stages, we use an initial learning rate of 0.0000032, with one exception — we find that CC–pre-trained model requires a higher learning rate of 0.000032 when fine-tuned on much longer captions from Flickr30k LocNar. We set the number of training steps to 10K (without the mouse trace) or 25K (with the mouse trace). When the mouse trace is only incorporated during the fine-tuning stage, we observe better performance with slightly higher initial learning rates of 0.000096 for OID LocNar- and CC→OID LocNar–pre-trained models, and an even higher 0.00032 for CC–pre-trained model. We set the number of training steps to 70K, as in the from-scratch experiments.

B. Additional Experiments

Architecture. In Table 1, we experiment with the number of layers of text (M) and image (L) transformer encoders of our model (Fig. 4 in the paper). We find that the benefit of the text+trace query modality over the text-only one generalizes to all our ablation studies. Further, depth is more important in the text(+trace) branch than in the image one.

C. Additional Qualitative Results

Figure 1 and Figure 2 show additional qualitative examples, comparing the best models from each modality.

As in the main text, the general trend is that using both the caption text and mouse trace (text+trace) is generally superior to the caption alone (text). For instance, text+trace correctly returns the target image with "water on the sand" in Figure 1 (top). We also find that the retrieval model

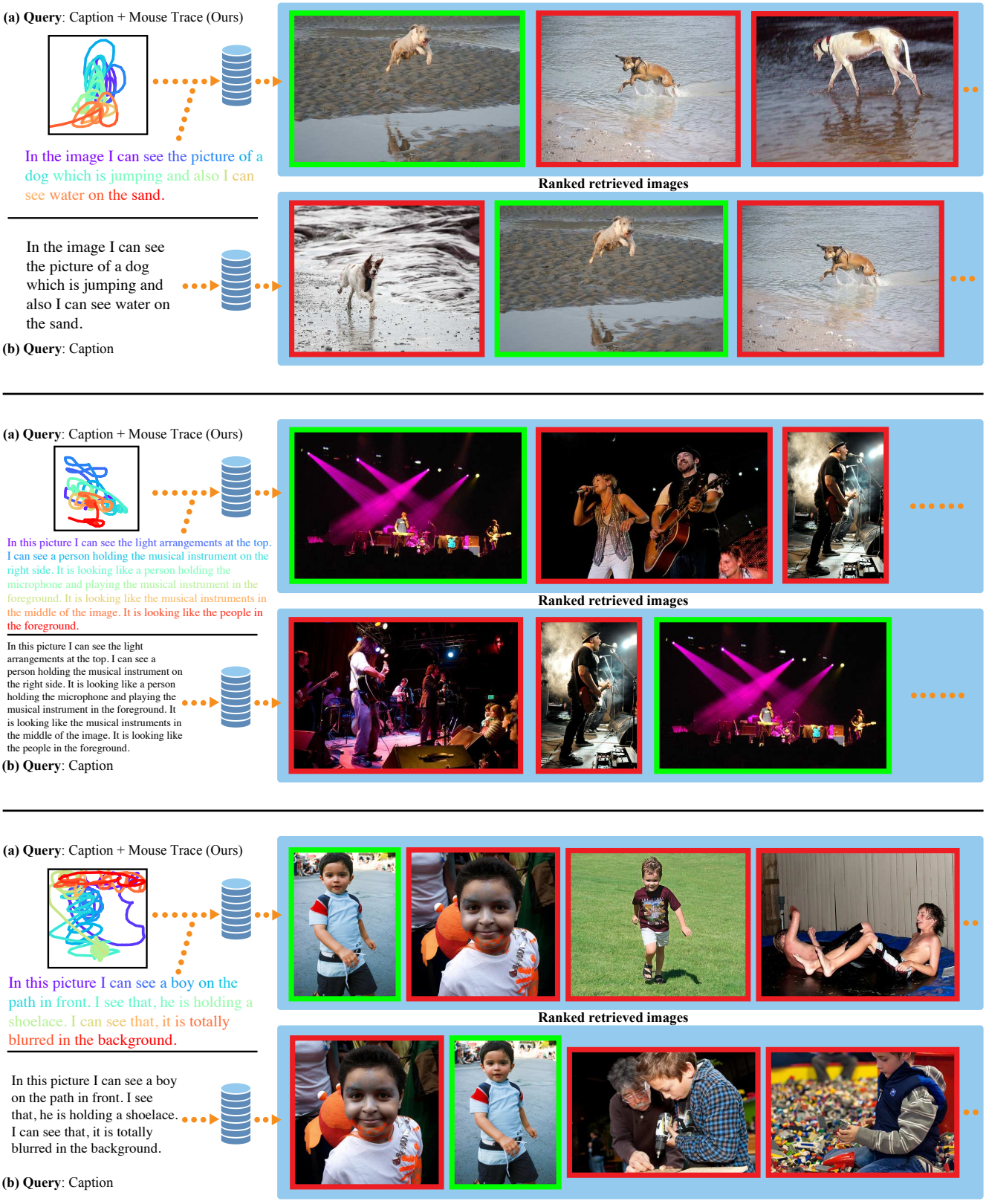

Figure 1: **Additional qualitative results (1 of 2)**: Comparison between the best model using our proposed text+trace query modality (a) to the best model using the text query modality (b). In green, the target image that corresponds to the query on the left.

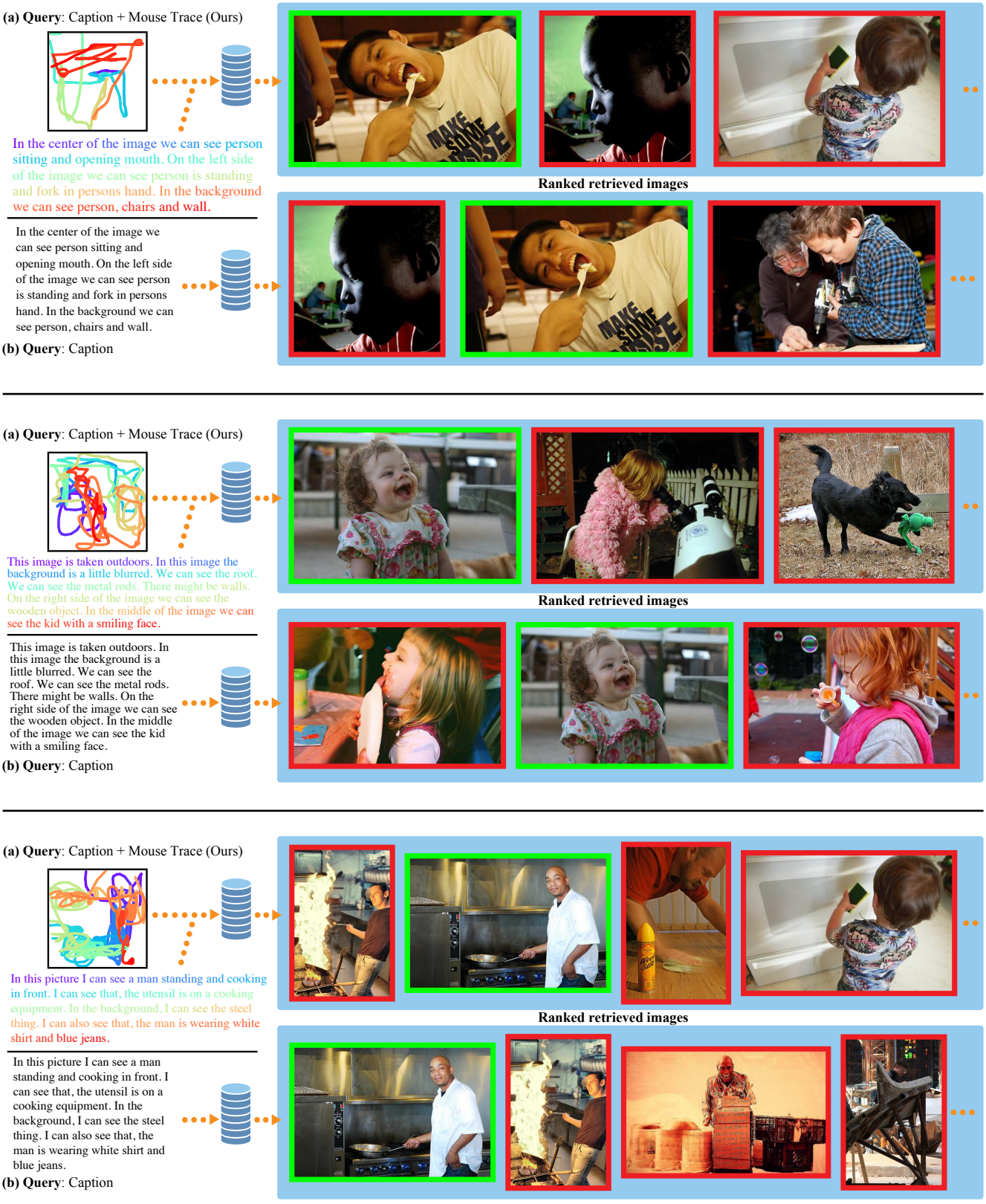

Figure 2: **Additional qualitative results (2 of 2)**: Comparison between the best model using our proposed text+trace query modality (a) to the best model using the text query modality (b). In green, the target image that corresponds to the query on the left.

Query	Model	Recall@K=		Model	Recall@K=	
	L=6	1	5	M=6	1	5
text	M=6	63.5	87.4	L=6	63.5	87.4
text+trace	M=6	68.2	88.8	L=6	68.2	88.8
text	M=4	59.5	86.5	L=4	61.7	87.2
text+trace	M=4	65.3	88.6	L=4	64.8	87.9
text	M=2	58.3	84.0	L=2	61.2	86.7
text+trace	M=2	60.5	85.7	L=2	64.6	88.6

Table 1: **Benefits of depth:** the number of layers of text (M) and image (L) transformer encoders on the image retrieval performance on the Flickr30k LocNar 1K test set.

struggles to use localization cues that are presented in the form of text such as "at the top," "on the right side," "in the foreground," and "in the middle of the image" in Figure 1 (middle), representing the setting in Figure 1(b) of the main text. In many cases, the retrieval model fails to retrieve relevant images, for instance, ignoring "shoelace" in Figure 1 (bottom), "person standing" in Figure 2 (top), "roof" in Figure 2 (middle). However, incorporating the trace helps correct this. We hypothesize that this is because the text+trace model is guided to focus on the right region, even in the presence of unfamiliar concepts and imperfect visual signals due to conditions such as poor lighting and extreme occlusion.

Finally, Figure 2 (bottom) shows our failure case. We observe that one main mode of failure is that the text+trace model can be over-reliant to the trace and ignore part of the input text ("white shirt"); in this example, the trace of "blue jeans," which covers a large space of the canvas, possibly misleads the model.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1