Image Retrieval from Contextual Descriptions

Anonymous ACL submission

Abstract

The ability to integrate context, including perceptual and temporal cues, plays a pivotal role in grounding the meaning of a linguistic utterance. In order to measure to what extent current vision-and-language models master this ability, we devise a new multimodal challenge, Image Retrieval from Contextual Descriptions (IMAGeCoDE). In particular, models are tasked with retrieving the correct image from a set of 10 minimally contrastive candidates based on a contextual description. As such, each description contains only the details that help distinguish between images. Because of this, descriptions tend to be complex in terms of syntax and discourse and require drawing pragmatic inferences. Images are sourced from both static pictures and video frames. We benchmark several state-of-the-art models, including both cross-encoders such as ViLBERT and bi-encoders such as CLIP, on IMAGeCoDE. Our results reveal that these models dramatically lag behind human performance: the best variant achieves an accuracy of 20.9 on video frames and 59.4 on static pictures, compared with 90.8 in humans. Furthermore, we experiment with new model variants that are better equipped to incorporate visual and temporal context into their representations, which achieve modest gains. Our hope is that IMAGeCoDE will foster progress in grounded language understanding by encouraging models to focus on fine-grained visual differences.

1 Introduction

Natural languages are highly contextual (Fodor, 2001): for a listener, recovering the speaker’s intended meaning requires integrating information from different streams, such as grounding in perception (Pecher and Zwaan, 2005), shared world knowledge, and temporal reasoning (Wilson and Sperber, 1998). These processes, more generally, fall under the umbrella term of pragmatics (Grice, 1957). Despite recent progress in multimodal systems, it remains unclear to which extent they can handle settings where context plays a major role, such as in real-world communication.

To this end, we present a new challenge that requires multimodal models to leverage context to retrieve images from text. In particular, given a contextual description and a set of minimally contrastive candidate images, i.e. differing only in some details, the model has to retrieve the target image. In order to discriminate between similar images, human annotators naturally produce highly nuanced and grammatically complex descriptions. An example of our new challenging dataset, Image Retrieval from Contextual Descriptions (IMAGeCoDE), is shown in Figure 1.

During the data collection process, sets of similar images are selected among static pictures from Open Images (Kuznetsova et al., 2020) and (a larger portion) among video frames from diverse domains. Including both types of images allows for diversifying the dataset while representing different degrees of visual similarity within each set. Next, we
crowdsourced a contextual description of a target image (presented together with the rest of the set) that contains only differences relevant for retrieval. After a filtering phase involving human retrievers, we obtain a large-scale dataset with 94,020 images and 21,202 descriptions associated with image sets of size 10.

As a result of this annotation protocol, successfully completing the task requires models to integrate several kinds of context: i) the image set, as the descriptions only make sense in the context of several other images and are not suitable as stand-alone captions. In fact, aspects of the image that are very salient and that therefore would normally be emphasized are not useful in our proposed task. Instead, the focus of our descriptions are fine-grained details that help discriminate between images (see Figure 1); ii) the speaker’s intention. Due to their high degree of image similarity, contextual descriptions may be literally true for multiple images; however, once the speaker’s intention is taken into account, the correct image can be determined by virtue of pragmatics (see Figure 2); iii) temporal sequences: for video frames temporal reasoning is also required to compare different moments of an unfolding event.

On our new dataset IMAGECoDE, we benchmark a series of vision-and-language models that achieve state-of-the-art performance on other multimodal tasks, including both cross-encoders such as ViLBERT (Lu et al., 2019) and bi-encoders such as CLIP (Radford et al., 2021). We report several findings. First, accuracy on static images is vastly superior than on video frames. Therefore, the degree of similarity among the candidate images has an overwhelming impact on retrieval performance. Second, all state-of-the-art models generally struggle with image retrieval from contextual descriptions, whereas humans consistently achieve high accuracy.

Hence, we propose model variants capable of better taking context into account: i) once an image-description pair is encoded, we refine this representation by attending to the other images in the set; ii) we augment image encodings with special temporal embeddings. Based on our results, models take advantage of this additional information fruitfully but only to a limited degree.

Because of its challenging nature, due to the minimally contrastive images and complex descriptions, we believe that IMAGECoDE will help make visio-linguistic models more context-aware and sensitive to fine-grained details. The dataset and models would be publicly released with the camera-ready version.

2 Related Work

There is a long tradition of grounding language understanding on single images, in the form of visual question answering (Goyal et al., 2017; Hudson and Manning, 2019), visual dialogue (de Vries et al., 2017; Das et al., 2017), or visual entailment (Xie et al., 2019). Recently, more and more focus has been directed to settings where the visual context consists of multiple images, either conventional static pictures (Vedantam et al., 2017; Hu et al., 2019; Suhr et al., 2019; Forbes et al., 2019; Hendricks and Nematazadeh, 2021; Yan et al., 2021; Hosseinazadeh and Wang, 2021; Bogin et al., 2021; Liu et al., 2021), or video frames (Jhamtani and Berg-Kirkpatrick, 2018a; Bansal et al., 2020). While many of these benchmarks involve just two images, COVR (Bogin et al., 2021) and ISVQA (Bansal et al., 2020) provide more images, similar to our sets of 10 images.

ISVQA and Spot-the-diff (Jhamtani and Berg-Kirkpatrick, 2018a) are most similar to our dataset, IMAGECoDE. ISVQA is based on several video frames that are synthetic and cover a restricted domain, with short questions for Visual Question Answering. Spot-the-diff provides two frames from surveillance video cameras and descriptions of all their differences. IMAGECoDE is unique as a) we cover a wider range of domains; b) we construct image sets that are maximally similar while being distinguishable through natural language (Section 3) and c) we limit descriptions to relevant differences. This results in (a) diverse, (b) complex and (c) pragmatically informative descriptions.

IMAGECoDE elicits pragmatic reasoning (Andreas and Klein, 2016; Cohn-Gordon et al., 2018) as a listener has to consider the context and resolve ambiguities resulting from nuanced differences to solve the task.

3 Data Collection

Our data collection involves two steps with a human describer and retriever. The describer is given a set of 10 highly similar images $S = \{I_1, I_2, ..., I_{10}\}$, one of them marked as the target image $I_t$, and has to write a description $D$ that clearly distinguishes $I_t$ from the other distractor
images. In the second step, the retriever is given
the same 10 images and the description from the
first step and has to identify the target image based
on the description. S and D are only added to our
dataset if the retrieval is successful.

Below, we outline the main stages of data col-
collection: first, the collection of similar, contrastive
images in Section 3.1. Then, the crowdsourcing
of contextual descriptions in Section 3.2 and val-
ification of the examples via image retrieval (Sec-
tion 3.3). The final ImageCode dataset consists
of 94,020 images (partitioned into 9,402 sets) and
21,202 contextual descriptions (16,594 in the train
split, 2,302 and 2,306 in the validation and test split
respectively).

3.1 Collecting Similar Images

In the first stage, we collect sets of images that
are highly similar but still distinguishable from
each other by a human. To quantitatively measure
the pairwise similarity of two images, we compute
the Euclidean distance between their encodings ex-
tracted from a pre-trained CLIP model (Radford
et al., 2021).\(^1\) To study the effect of different de-
grees of similarity, further varigate our dataset,
and enable temporal reasoning, we source our can-
didate images from collections of static pictures as
well as videos, as detailed below.

Static Pictures. We obtain image sets from one
of the largest repositories of static pictures, the
Open Images Dataset V6 (Kuznetsova et al., 2020),
containing 1.74M images. For each image, we
retrieve the 9 closest images from the training set
based on their CLIP encodings. We then randomly
sample 4,845 of these image sets.

Video Frames. As sources for our video frames,
we use i) Video-Storytelling (Li et al., 2019), cov-
ering social events (wedding, birthday, Christmas,
camping); ii) general-domain MSR-VTT (Xu et al.,
2016); and iii) YouCook (Das et al., 2013), cover-
ing cooking events. We choose these datasets as
they contain publicly available and general-purpose
videos (not specific to downstream tasks). We re-
tain the original splits for train, validation, and test.

To obtain disjoint sets of 10 similar frames, we
first segment the videos into smaller scenes (also
known as shots) via the scene detection function-
ality of ffmpeg (Tomar, 2006). Then, for each
scene, we add its first frame to the set of selected

\(\)\(^1\)We also experimented with ResNet-50 features, but we
found CLIP results to be more similar to that of humans in
preliminary experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>After §3.1</th>
<th>After §3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR-VTT</td>
<td>11,643</td>
<td>8,045</td>
</tr>
<tr>
<td>Video-Storytelling</td>
<td>11,459</td>
<td>8,153</td>
</tr>
<tr>
<td>YouCook</td>
<td>894</td>
<td>588</td>
</tr>
<tr>
<td>Open Images</td>
<td>4,845</td>
<td>4,416</td>
</tr>
</tbody>
</table>

Table 1: Number of descriptions from each source of
images at different stages of the annotation process.

images. We then iterate over every following frame
and add it to the set if its pairwise Euclidean dis-
tance with each of the previously selected frames
is larger than a threshold.\(^2\) Once the set contains
10 images, we reiterate the procedure for a new set.
If the scene ends and the current set contains less
than 10 images, the set is discarded.

During this process, we additionally remove
frames that i) are too blurry, i.e. their BRISQUE
score (Mittal et al., 2012) is larger than 0.65; or
ii) contain too much text, which is detected with
the OCR tool Tesseract (Smith, 2007).\(^3\) We use
all of YouCook’s image sets and (due to cost con-
straints) randomly sample image sets from Video-
Storytelling and MSR-VTT for crowdsourcing (cf.
Table 1). We remark that image sets are further fil-
tered at the final stage of annotation (Section 3.3).

3.2 Crowdsourcing Contextual Descriptions

After creating sets of highly-similar images in Sec-
tion 3.1, we request annotators from Amazon Me-
chanical Turk (AMT) to write contextual descrip-
tions for each target image in a set. Each round,
a set of images is presented in random order for
static pictures and respecting temporal order for
video frames. This encourages annotators to take
the dynamics of the event into account. We then
(randomly) select 3 target images per set, and ask
annotators to produce a description that discrimi-
nates them from the other images in the set. To
eourage pragmatic reasoning, we do not ask for
all the differences (just those sufficient for retrieval)
and do not allow explicit mentions of other images
(see Figure 2). We select high-quality annotators
according to criteria in Appendix B and assign
partly disjoint sets of annotators to train and test in
order to avoid annotator bias (Geva et al., 2019).\(^4\)

\(\)\(^2\)The distance threshold was manually chosen as 0.35
based on qualitative results.

\(\)\(^3\)The rationale of the second criterion is to prevent workers
from focusing on the overlaid text rather than image content.

\(\)\(^4\)For further details on crowdsourcing instructions, analy-
sis of annotator bias and the AMT interface, please refer to

3
### Table 2: Distribution of challenging phenomena in IMAGECoDE based on 200 (or 1000 if underlined) manually annotated examples.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>all %</th>
<th>videos %</th>
<th>static %</th>
<th>Example from IMAGECoDE</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>47.3</td>
<td>57.3</td>
<td>6.6</td>
<td>Visual context or pragmatic inference required.</td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>15.0</td>
<td>18.5</td>
<td>4.1</td>
<td>A smiling boy just begins to look towards the dog.</td>
<td></td>
</tr>
<tr>
<td>Quantities</td>
<td>48.5</td>
<td>47.7</td>
<td>51.0</td>
<td>There is an equal amount of yellow and white between both hands.</td>
<td></td>
</tr>
<tr>
<td>Spatial Relations</td>
<td>70.5</td>
<td>72.2</td>
<td>65.3</td>
<td>The cloud on top left side of box only has half of it showing.</td>
<td></td>
</tr>
<tr>
<td>Negation</td>
<td>17.9</td>
<td>20.7</td>
<td>6.1</td>
<td>The spoon is at the top right corner, it is not moving any of the food.</td>
<td></td>
</tr>
<tr>
<td>Visibility/Oclusion</td>
<td>45.5</td>
<td>54.5</td>
<td>8.6</td>
<td>The flowers the woman in the teal strapless dress is carrying are completely obscured by the man in the black shirt’s head.</td>
<td></td>
</tr>
<tr>
<td>Nuances</td>
<td>26.3</td>
<td>31.6</td>
<td>5.1</td>
<td>There is the slightest of openings to see the end of the bridge through the obstruction.</td>
<td></td>
</tr>
<tr>
<td>Co-reference</td>
<td>41.5</td>
<td>42.4</td>
<td>38.8</td>
<td>The cloud on top left side of box only has half of it showing.</td>
<td></td>
</tr>
<tr>
<td>Meta Properties</td>
<td>12.0</td>
<td>13.9</td>
<td>6.1</td>
<td>Bright shot of a girl and boy standing up straight. Her eyes are closed.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3: Human performance (accuracy) and inter-annotator agreement (Krippendorff’s $\alpha$) on the validation and test splits of IMAGECoDE.

<table>
<thead>
<tr>
<th>Metric</th>
<th>val</th>
<th>test</th>
<th>ours</th>
<th>NLVR2</th>
<th>Spot-the-diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Accuracy</td>
<td></td>
<td></td>
<td>90.9</td>
<td>90.8</td>
<td></td>
</tr>
<tr>
<td>Krippendorff’s $\alpha$ (nominal)</td>
<td>.797</td>
<td>.795</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Krippendorff’s $\alpha$ (interval)</td>
<td>.872</td>
<td>.869</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4: Comparison of the text statistics of IMAGECoDE with other vision-and-language datasets.

<table>
<thead>
<tr>
<th></th>
<th>ours</th>
<th>NLVR2</th>
<th>Spot-the-diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average length</td>
<td>23.3</td>
<td>15.3</td>
<td>10.6</td>
</tr>
<tr>
<td>Word types</td>
<td>6,916</td>
<td>6,602</td>
<td>2,282</td>
</tr>
<tr>
<td>Average tree depth</td>
<td>5.1</td>
<td>4.8</td>
<td>4.3</td>
</tr>
<tr>
<td>Average sentences</td>
<td>1.6</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### 3.3 Human Validation via Image Retrieval

Finally, we validate the annotation crowdsourced in Section 3.2 by asking AMT workers to retrieve the correct target image from a set given its contextual description. For the final dataset, we retained only the examples that i) were retrieved successfully in the training set by a single worker or ii) were retrieved successfully by at least 2 out of 3 workers in the validation and test sets. As a consequence, we filtered out 26.5% of the contextual descriptions generated in Section 3.2. Table 1 compares the number of examples retained at each stage throughout the dataset creation.\(^5\)

### 4 Data Analysis

#### 4.1 Human Accuracy and Agreement

To quantify the reliability of the process outlined in Section 3, we report the inter-annotator agreement on our final dataset in Table 3. We use Krippendorff’s $\alpha$ as a metric (the higher the better), which accounts for incomplete data, since the number of annotators per example is not fixed. We treat the index of the target image either as a nominal variable for static images or as an ordinal variable for video frames. In both cases, we find a high degree of agreement. Moreover, in Table 3, we also report human accuracy—the percentage of times an annotator retrieved the correct target image from a contextual description (as described in Section 3.3). This provides an upper ceiling for the model performances (see Section 6).

#### 4.2 Language Statistics

In Table 4, we measure a series of statistics of the descriptions collected for IMAGECoDE and compare them with other vision-and-language datasets with multiple naturalistic images (cf. Section 2), such as NLVR2 (Suhr et al., 2019) and Spot-the-diff (Jhamtani and Berg-Kirkpatrick, 2018b).\(^6\) In particular, we count the average description length.

\(^5\)Again, the set of workers validating train and test sets were partly disjoint to avoid annotator bias.

\(^6\)For comparability, we measured the statistics for all the datasets with the same tools.
the number of distinct word types, the average dependency tree depth of each sentence, and the average number of sentences per description. Based on these metrics, we find evidence that IMAGE-CoDE’s descriptions are longer and more syntactically complex than in the other datasets. Moreover, they include multiple sentences (11.8% of examples have 3 or more).

4.3 Vision Statistics

By calculating the average pairwise Euclidean distance between CLIP-based encodings of images in the same set, we find that video frames are more similar than static pictures – as expected – by a factor of 1.13. Moreover, we find that descriptions of video frames mention human body parts (72.1%) more often than static pictures (30.2%). On the other hand, names of colors appear in descriptions of static pictures (61.4%) more frequently than video frames (33.6%). Thus, annotators resort to different strategies to discriminate between different types of image sets, focusing on the aspects that vary the most.

4.4 Challenging Phenomena

Finally, we identify 9 interesting and challenging phenomena in IMAGE-CoDE and annotate whether they are present in 200 examples from the validation set. We provide the definition of each phenomenon, its frequency, and an illustrative example in Table 2. More information is given in Appendix F. For 4 of these phenomena unique to IMAGE-CoDE, we further annotated 800 examples for the purpose of error analysis in Section 6. Inspecting these examples, we find a high number of cases where the visual context (47.0%) is required to complete the task. For instance, consider Figure 2: the description “No bridesmaid visible at all.” requires a retriever to resolve the co-references of the entities in 5 frames. In particular, the body parts of the bridesmaids (red boxes) visible in frames 2 and 4 would not be identifiable as such without frame 1 and 5, respectively (where they appear with matching dresses and flowers in their hands).

Another group of phenomena characteristic for IMAGE-CoDE originates from its minimally contrastive setup: annotators might focus on how an event unfolds over time (temporal context), on what is missing in a specific frame but visible in the others (negation), on what moved out of frame (visibility/occlusion), or on small regions and patches of pixels (nuances). Importantly, these phenomena are less prominent in static pictures than in video frames (cf. Table 2).

5 Methods

5.1 Baselines

In order to assess whether vision-and-language models can retrieve the correct image from a contextual description on a par with humans, we benchmark two state-of-the-art models that represent two main families of multimodal architectures (Bugliarello et al., 2021; Miech et al., 2021): i) ViLBERT, a cross-encoder where language and vision streams can interact via cross-attention at intermediate layers (Lu et al., 2019); ii) CLIP, a bi-encoder where language and vision streams are independent (Radford et al., 2021). It is worth noting that ViL-
CLIP / ViLBERT

Figure 3: Models with increasing levels of context integration: see Section 5 for more details. In the figure, we colour visual embeddings in red, text embeddings in blue, and positional embeddings in grey. POS is the score for the target image and NEG for the other candidates. @ represents dot product for CLIP and element-wise multiplication followed by a linear layer for ViLBERT. ⊙ represents element-wise multiplication. For ease of exposition, we show 3 images instead of 10.

5.2 Integrating Context into Vision-and-Language Models

For the fine-tuning regime, we further investigate some modifications in the training setup and model architecture that facilitate the integration of visual and temporal context into the model. First, we use an alternative objective where both CLIP and ViLBERT are trained on 10-class classification, but the 1 positive and 9 negatives are sourced from the same image set. The consequence of including positive and negative examples from the same image set in the same mini-batch is providing a wider visual context. We refer to this variant as +CONTEXTBATCH (second column of Figure 3).

This setup only conveys the visual context as a weak signal, since the model has no chance to directly compare the images in the same set. Hence, we experiment with enhancing the architecture of vision-and-language models with a mechanism inspired by Bogin et al. (2021). In particular, given an encoder (CLIP or ViLBERT), we obtain the representations of a contextual description $x_{V} \in \mathbb{R}^e$ (where $e$ is the model hidden size) and of the images in a set $(x_{V}^{(1)}, \ldots, x_{V}^{(10)})$, $x_{V}^{(i)} \in \mathbb{R}^e$ from their final layer. Then, we create a series of multimodal embeddings via element-wise multiplication: $m = (x_{L} \odot x_{V}^{(1)}, \ldots, x_{L} \odot x_{V}^{(10)})$. Finally, we feed these to a $l$-layer Transformer $T_{f}: \mathbb{R}^{10x e} \rightarrow \mathbb{R}^{10x e}$ to obtain context-aware multimodal embeddings $(T_{f}(m)_{1}, \ldots, T_{f}(m)_{10})$. Since each description–image pair can now attend on the others in a set, the model can fully exploit the visual context. We obtain the score for the $i$-th pair through a linear classifier head $W \in \mathbb{R}^{1x e}$. The target image is predicted as

$$\arg \max_i \text{softmax}[W(T_{f}(m)_i + m^{(i)})] \quad (1)$$

---

5.2 Integrating Context into Vision-and-Language Models

For the fine-tuning regime, we further investigate some modifications in the training setup and model architecture that facilitate the integration of visual and temporal context into the model. First, we use an alternative objective where both CLIP and ViLBERT are trained on 10-class classification, but the 1 positive and 9 negatives are sourced from the same image set. The consequence of including positive and negative examples from the same image set in the same mini-batch is providing a wider visual context. We refer to this variant as +CONTEXTBATCH (second column of Figure 3).

This setup only conveys the visual context as a weak signal, since the model has no chance to directly compare the images in the same set. Hence, we experiment with enhancing the architecture of vision-and-language models with a mechanism inspired by Bogin et al. (2021). In particular, given an encoder (CLIP or ViLBERT), we obtain the representations of a contextual description $x_{V} \in \mathbb{R}^e$ (where $e$ is the model hidden size) and of the images in a set $(x_{V}^{(1)}, \ldots, x_{V}^{(10)})$, $x_{V}^{(i)} \in \mathbb{R}^e$ from their final layer. Then, we create a series of multimodal embeddings via element-wise multiplication: $m = (x_{L} \odot x_{V}^{(1)}, \ldots, x_{L} \odot x_{V}^{(10)})$. Finally, we feed these to a $l$-layer Transformer $T_{f}: \mathbb{R}^{10x e} \rightarrow \mathbb{R}^{10x e}$ to obtain context-aware multimodal embeddings $(T_{f}(m)_{1}, \ldots, T_{f}(m)_{10})$. Since each description–image pair can now attend on the others in a set, the model can fully exploit the visual context. We obtain the score for the $i$-th pair through a linear classifier head $W \in \mathbb{R}^{1x e}$. The target image is predicted as

$$\arg \max_i \text{softmax}[W(T_{f}(m)_i + m^{(i)})] \quad (1)$$

---

5.2 Integrating Context into Vision-and-Language Models

For the fine-tuning regime, we further investigate some modifications in the training setup and model architecture that facilitate the integration of visual and temporal context into the model. First, we use an alternative objective where both CLIP and ViLBERT are trained on 10-class classification, but the 1 positive and 9 negatives are sourced from the same image set. The consequence of including positive and negative examples from the same image set in the same mini-batch is providing a wider visual context. We refer to this variant as +CONTEXTBATCH (second column of Figure 3).

This setup only conveys the visual context as a weak signal, since the model has no chance to directly compare the images in the same set. Hence, we experiment with enhancing the architecture of vision-and-language models with a mechanism inspired by Bogin et al. (2021). In particular, given an encoder (CLIP or ViLBERT), we obtain the representations of a contextual description $x_{V} \in \mathbb{R}^e$ (where $e$ is the model hidden size) and of the images in a set $(x_{V}^{(1)}, \ldots, x_{V}^{(10)})$, $x_{V}^{(i)} \in \mathbb{R}^e$ from their final layer. Then, we create a series of multimodal embeddings via element-wise multiplication: $m = (x_{L} \odot x_{V}^{(1)}, \ldots, x_{L} \odot x_{V}^{(10)})$. Finally, we feed these to a $l$-layer Transformer $T_{f}: \mathbb{R}^{10x e} \rightarrow \mathbb{R}^{10x e}$ to obtain context-aware multimodal embeddings $(T_{f}(m)_{1}, \ldots, T_{f}(m)_{10})$. Since each description–image pair can now attend on the others in a set, the model can fully exploit the visual context. We obtain the score for the $i$-th pair through a linear classifier head $W \in \mathbb{R}^{1x e}$. The target image is predicted as

$$\arg \max_i \text{softmax}[W(T_{f}(m)_i + m^{(i)})] \quad (1)$$

---

5.2 Integrating Context into Vision-and-Language Models

For the fine-tuning regime, we further investigate some modifications in the training setup and model architecture that facilitate the integration of visual and temporal context into the model. First, we use an alternative objective where both CLIP and ViLBERT are trained on 10-class classification, but the 1 positive and 9 negatives are sourced from the same image set. The consequence of including positive and negative examples from the same image set in the same mini-batch is providing a wider visual context. We refer to this variant as +CONTEXTBATCH (second column of Figure 3).

This setup only conveys the visual context as a weak signal, since the model has no chance to directly compare the images in the same set. Hence, we experiment with enhancing the architecture of vision-and-language models with a mechanism inspired by Bogin et al. (2021). In particular, given an encoder (CLIP or ViLBERT), we obtain the representations of a contextual description $x_{V} \in \mathbb{R}^e$ (where $e$ is the model hidden size) and of the images in a set $(x_{V}^{(1)}, \ldots, x_{V}^{(10)})$, $x_{V}^{(i)} \in \mathbb{R}^e$ from their final layer. Then, we create a series of multimodal embeddings via element-wise multiplication: $m = (x_{L} \odot x_{V}^{(1)}, \ldots, x_{L} \odot x_{V}^{(10)})$. Finally, we feed these to a $l$-layer Transformer $T_{f}: \mathbb{R}^{10x e} \rightarrow \mathbb{R}^{10x e}$ to obtain context-aware multimodal embeddings $(T_{f}(m)_{1}, \ldots, T_{f}(m)_{10})$. Since each description–image pair can now attend on the others in a set, the model can fully exploit the visual context. We obtain the score for the $i$-th pair through a linear classifier head $W \in \mathbb{R}^{1x e}$. The target image is predicted as

$$\arg \max_i \text{softmax}[W(T_{f}(m)_i + m^{(i)})] \quad (1)$$
Note that we add a highway layer from the input to the output of the Transformer. We label this model variant \texttt{+CONTEXTMODULE}.

Finally, in addition to visual context, we make models aware of the temporal context too, as shown in the fourth column of Figure 3. For video-based examples only, the multimodal embeddings of each description-image pair are summed with a learnable positional embedding $t \in \mathbb{R}^n$ that reflects the temporal order of the frames.\textsuperscript{11} Thus, $m = (x_L \odot x_V^{(1)} \oplus t^{(1)}, \ldots, x_L \odot x_V^{(10)} \oplus t^{(10)})$. Multimodal embeddings are then fed to a Transformer as above. We label this variant encapsulating both visual and temporal context \texttt{+TEMPORALEMBEDDINGS}.

### 5.3 Experimental Setup

For all CLIP experiments, we use a pre-trained model with the vision backbone ViT-B/16.\textsuperscript{12} We train the full models with a batch size of 360 examples (i.e., 36 image sets) for CLIP and 150 examples for ViLBERT. We perform early stopping based on the validation accuracy with a maximum of 30 epochs. In the variants that adopt the base version of a model, we select a learning rate of $4 \times 10^{-6}$ for CLIP, $5 \times 10^{-6}$ for ViLBERT, and $4 \times 10^{-5}$ for ViLBERT +\texttt{CONTEXT BATCH}. We find these values via hyper-parameter search on the range $[10^{-2}, 10^{-7}]$.

For all the model variants that modify the model architecture, we adopt the following setup: first, we fine-tune the full model in the +\texttt{CONTEXT BATCH} regime as detailed above. Afterwards, we freeze the encoder parameters and train the components responsible for processing the multimodal embeddings, described in Equation (1). More details are provided in Appendix E.

All descriptions in IMAGECODE exceeding the maximum length of CLIP and ViLBERT are truncated. Due to their negligible amount, this does not affect performance significantly.

### 6 Results

In Table 5, we report the performance of the models from Section 5 for all the examples in IMAGECODE as well as for the subsets containing only video frames or static pictures. Note that the random chance baseline has an accuracy of 10%. In what follows, we compare the results across several dimensions.

**Zero-shot vs. fine-tuning.** In the zero-shot setting, we observe that CLIP representations are surprisingly superior to ViLBERT even though CLIP has separate streams to encode an image and its description. In the simplest fine-tuning setting (i.e., if negatives are randomly sampled independent of the image set), we find that there is only a small increase in performance for both CLIP and ViLBERT (+5.4% and +8.3%, respectively) compared to zero-shot inference. This demonstrates that in the regime where images in the same set do not appear in the same batch during training, models cannot extrapolate how to leverage the visual context at inference time.

**Adding context.** For the fine-tuning regime, we observe instead a different trend once the visual context of the other images in a set is provided during training (+\texttt{CONTEXT MODULE}): CLIP receives a significant boost in performance (+18.7%), which is particularly accentuated for static pictures. On the other hand, ViLBERT’s performance remains the same, as this variant is beneficial for video frames but detrimental for static pictures. Stacking a special module for contextualizing multimodal representations on top of the encoders (+\texttt{CONTEXTMODULE}), instead, yields gains for ViLBERT compared to +CON-

---

Table 5: Performance (accuracy) on IMAGECODE across two training regimes (zero-shot and fine-tuning), two models (CLIP and ViLBERT) and 4 model variants. We report separate figures for all the examples and two disjoint subsets: video frames and static pictures.

<table>
<thead>
<tr>
<th></th>
<th>all</th>
<th>video</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ZERO-SHOT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIP</td>
<td>22.4</td>
<td>15.6</td>
<td>47.8</td>
</tr>
<tr>
<td><strong>FINE-TUNING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIP</td>
<td>23.6</td>
<td>17.1</td>
<td>48.2</td>
</tr>
<tr>
<td>+\texttt{CONTEXT BATCH}</td>
<td>28.0</td>
<td>19.7</td>
<td>59.2</td>
</tr>
<tr>
<td>+\texttt{CONTEXT MODULE}</td>
<td>28.2</td>
<td>19.9</td>
<td>59.4</td>
</tr>
<tr>
<td>+\texttt{TEMPORAL EMBEDDINGS}</td>
<td><strong>28.9</strong></td>
<td><strong>20.9</strong></td>
<td>58.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>all</th>
<th>video</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ZERO-SHOT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViLBERT</td>
<td>19.3</td>
<td>13.5</td>
<td>40.8</td>
</tr>
<tr>
<td><strong>FINE-TUNING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViLBERT</td>
<td>20.9</td>
<td>13.1</td>
<td><strong>49.9</strong></td>
</tr>
<tr>
<td>+\texttt{CONTEXT BATCH}</td>
<td>20.9</td>
<td>15.0</td>
<td>42.7</td>
</tr>
<tr>
<td>+\texttt{CONTEXT MODULE}</td>
<td>22.3</td>
<td>16.1</td>
<td>45.6</td>
</tr>
<tr>
<td>+\texttt{TEMPORAL EMBEDDINGS}</td>
<td><strong>24.5</strong></td>
<td><strong>18.0</strong></td>
<td>49.3</td>
</tr>
</tbody>
</table>

\textsuperscript{11}In the examples with static pictures, no temporal embedding is added.

\textsuperscript{12}https://github.com/openai/CLIP
TEXT BATCH, whereas CLIP is almost unaffected. This shows that all models can exploit visual context, but different strategies (contrastive training or dedicated modules) may be necessary.

Finally, both CLIP and ViLBERT achieve the highest performance when fine-tuned with both visual and temporal context. Adding temporal positional embeddings on top of the contextual module (+TEMPORAL EMBEDDINGS) yields an accuracy of 28.9 for CLIP and 24.5 for ViLBERT. Crucially, even the best-performing models lag significantly behind the (micro-averaged) human accuracy of 90.8 (cf. Table 3). Hence, despite some limited ability to integrate context, models are currently incapable of the fine-grained reasoning and pragmatic inferences needed to solve IMAGE CO DE.

Pre-trained model. Across all model variants and training regimes, CLIP consistently achieves higher accuracy than ViLBERT. This implies that a larger amount of parameters and pre-training examples are more beneficial than ViLBERT’s more expressive model architecture. Thus, these results violate the expectations that ViLBERT’s cross-attention would be more suitable to jointly encode highly nuanced visual details and descriptions (Miech et al., 2021).

Video frames vs. static pictures. The highest accuracy on the subset of the data with video frames (20.9) is far lower than that for static pictures (59.4). This confirms that videos represent the main challenge in IMAGE CO DE, both because of the higher similarity of images in a set and of the particular factors of variation that help differentiate among them (cf. Section 4.3 and examples in Appendix F). Additionally, model performance on video frames seems to increase more consistently as more context (both visual and temporal) is provided, whereas there is no clear trend in the case of static pictures.

Error Analysis. On a broad level, we have seen that video frames are much more challenging for models. Next, to identify more fine-grained causes for the overall low performance of the vision-and-language models on IMAGE CO DE, we compute the Pearson’s correlation between accuracy and a series of possible explanatory variables. In particular, we find a weak negative correlation with the number of tokens in the description ($\rho = -0.11$) and a weak positive correlation with the average pairwise Euclidean distance between CLIP encodings of the images in a set ($\rho = 0.22$), which represents visual similarity.

By focusing on the 1000 annotated examples in Table 2 we observe a stark drop from overall performance on the subset of examples containing nuances, visibility/occlusion, and negation (Figure 4). This confirms insights from Kassner and Schütze (2020) and Hosseini et al. (2021) on the difficulty of modeling negation in text-only models.

7 Conclusions and Future Work

We created a new challenge, Image Retrieval from Contextual Descriptions (IMAGE CO DE), which is designed to evaluate the ability of vision-and-language models to integrate visual, pragmatic, and temporal context into their predictions. In particular, given a complex and nuanced contextual description, a model is required to retrieve the corresponding image from a set of highly similar candidates. We benchmarked state-of-the-art bi-encoder and cross-encoder models, such as CLIP and ViLBERT. Moreover, we proposed new variants of these models that are more suitable to solve this task, by augmenting them with a module to attend on the other images in a set and temporal embeddings. We found that IMAGE CO DE is highly challenging for all variants: even the best model (28.9) lags behind human performance (90.8) dramatically. Images sourced from video frames display the largest gap in performance. The most challenging phenomena in IMAGE CO DE include pragmatics, negation, fine-grained distinctions between images, and occlusion among others.
References


A Length Distribution of the Image Descriptions

Figure 5: Distribution of the number of tokens across contextual descriptions in IMAGECODE.

B Criteria for Selecting Annotators

We keep data quality high through entry requirements (English speaking country, over 98% approval rate, etc.), qualification test, whitelisting workers and manually inspecting data. Most importantly our two-stage setup also allowed us to automate monitoring data quality as we could measure the description and retrieval accuracy of workers and only whitelisted those with high accuracy. We paid 0.25$ per description and 0.1$ per retrieval.

C Annotator Bias

The majority of descriptions in our test and validation split come from workers who did not work on the training set in order to avoid annotation bias. Our validation set contains 502 descriptions from workers “seen” from the training set and 1,800 description from “unseen” workers. In Table 6 we can see that models perform slightly better on seen workers across our CLIP model variants.

D Crowdsourcing Interface

Our AMT interface for the description task can be seen in Figure 6. The retriever interface looks conceptually similar, with a select-button for each image. Note that workers see images almost in almost half of full-screen (opposed to the shown examples in this PDF) and can quickly go back and forth between consecutive frames with arrow-keys, making it significantly easier to spot and compare nuanced changes.

<table>
<thead>
<tr>
<th></th>
<th>seen workers</th>
<th>unseen workers</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINE-TUNING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIP</td>
<td>23.9</td>
<td>23.8</td>
</tr>
<tr>
<td>+CONTEXT_BATCH</td>
<td>34.5</td>
<td>29.0</td>
</tr>
<tr>
<td>+CONTEXT_MODULE</td>
<td>33.3</td>
<td>29.2</td>
</tr>
<tr>
<td>+TEMPORAL EMBEDDINGS</td>
<td>32.1</td>
<td>30.8</td>
</tr>
</tbody>
</table>

Table 6: Performance (accuracy) on two subsets of the distinct validation split: seen workers (workers who also produced description on the train split) and unseen workers (who only worked on the test and validation data).

E Additional Hyper-parameters

The Transformer consists of 2 layers in CLIP variants and 4 layers in the ViLBERT variants, both employing gelu activation. The learning rate for the fine-tuning of the Transformer and linear heads is $2 \cdot 10^{-6}$ for the CLIP +CONTEXT_MODULE, $10^{-4}$ for CLIP +TEMPORAL EMBEDDINGS, and $2 \cdot 10^{-5}$ for both ViLBERT variants. We use the Volta-framework (Bugliarello et al., 2021) for the standardized ViLBERT model.

F Examples from IMAGECODE for all phenomena

For each phenomenon we provide 1 example and a definition we used for annotation purposes. Since most examples contain more than one phenomenon, some phenomena will be effectively showcased several times. Note that we picked examples that are relatively easy to understand and spot differences in.
**Figure 6:** AMT interface for the describer task.

**Figure 7:** Example of **Context:** “Both hands are on the piece of bread closest to the person.” Note: This is contextual since since without any context of other images, the description is also literally true for Frame 9. A model might even score it higher since the direct visual appearance is closer to typical bread. Definition: To understand the description, a listener has to consider other images and/or the speakers intention of describing only one of the images. In line with Grice’s maxim of quality, a description is contextual if it is literally true for several images but we know it was intended for only one image. A description is also contextual if an objects cannot clearly be identified in the target image directly but only through cross-referencing other images.

**Figure 8:** Example of **negation:** “The knife is most centrally placed to insert into the onion without having fully cut deeply into it yet.” Definition: Explicit linguistic negation (“not”, ”unseen”, ”non-“) or negation quantifiers (“no person”).
Figure 9: Example of quantifiers/quantities: “A yellow 3-way traffic light with a green arrow on the side facing closest to the camera.” Definition: We annotate for quantifiers (most, every, no, several,...) and absolute quantities (“five”) as well as relative quantities (ratios like “a third of his hand”).

Figure 10: Example of spatial relations/reasoning: “The small girl in front is looking directly to the right with her right hand on the side of her face.” Definition: Any relations or adjectives regarding space. Examples: “in the top left corner”, “left to the chair”, but also camera perspective, or body orientation (“turned towards...”)

Figure 11: Example of temporality: “A smiling boy just begins to look towards the dog.” Definition: While most examples based on video frames implicitly require some temporal knowledge, we focus on explicit textual mentions of 1) temporal markers (“after”, “during”, “about to”, etc) and 2) temporal verbs (“beginning to”, “end to”).

Figure 12: Example of visibility/occlusion: “The tire is directly on top of the person’s right shoe and you can just barely see fingers at the top.” Definition: A description that mentions objects/people being occluded, (partially) out of frame, or in the process of leaving the frame.
Figure 13: Example of **nuances** (we marked small details with red/green rectangles): “The person’s palm is towards us and touching the left bottom corner of the cake. There is a small amount of dark space between the right bottom corner of the photo and the edge of the cake.” Definition: Minor details, that are either a) not salient at all and would usually be left unmentioned and/or b) language reference is grounded on a small patch of pixels. Note that this phenomena is often linked with very minimally contrastive images.

Figure 14: Example of **coreference**: “A woman with a white background smiles at the camera. Most of her body is visible. She is wearing a black outfit.” Definition: Linguistic coreference.

Figure 15: Example of **meta properties**: “The cucumber is just to be cut into, you can see a transparent image covering the image.” Definition: Descriptions that mention aspects that stem from the way the photo/video was taken: two overlayed images (when a video transitions), black-and-white, blurriness, brightness.