

ContextBLIP: Doubly Contextual Alignment for Contrastive Image Retrieval from Linguistically Complex Descriptions

Honglin Lin^{1*}, Siyu Li^{1*}, Guoshun Nan^{1†},
 Chaoyue Tang¹, Xuetong Wang¹, Jingxin Xu¹, Yankai Rong¹,
 Zhili Zhou², Yutong Gao³, Qimei Cui¹, Xiaofeng Tao¹

¹Beijing University of Posts and Telecommunications

²Guangzhou University ³Minzu University of China

Abstract

Image retrieval from contextual descriptions (IRCD) aims to identify an image within a set of minimally contrastive candidates based on linguistically complex text. Despite the success of VLMs, they still significantly lag behind human performance in IRCD. The main challenges lie in aligning key contextual cues in two modalities, where these subtle cues are concealed in tiny areas of multiple contrastive images and within the complex linguistics of textual descriptions. This motivates us to propose ContextBLIP, a simple yet effective method that relies on a doubly contextual alignment scheme for challenging IRCD. Specifically, 1) our model comprises a multi-scale adapter, a matching loss, and a text-guided masking loss. The adapter learns to capture fine-grained visual cues. The two losses enable iterative supervision for the adapter, gradually highlighting the focal patches of a single image to the key textual cues. We term such a way as *intra-contextual alignment*. 2) Then, ContextBLIP further employs an inter-context encoder to learn dependencies among candidates, facilitating alignment between the text to multiple images. We term this step as *inter-contextual alignment*. Consequently, the nuanced cues concealed in each modality can be effectively aligned. Experiments on two benchmarks show the superiority of our method. We observe that ContextBLIP can yield comparable results with GPT-4V, despite involving about 7,500 times fewer parameters. Our code is available at <https://github.com/LHL3341/ContextBLIP>.

1 Introduction

Text-to-image retrieval is a fundamental cross-modal task that aims to search images for textual queries. Early studies relied on convolutional



Figure 1: An instance selected from a public benchmark of IRCD, which involves six very similar contrastive image candidates, and the query “*Middle girl’s hand is blurry and shoulder level, her eyes are almost shut, the girl on the right is looking at the middle girl’s hand*”. The target image is the 4-th one in red rectangular box.

neural networks (CNN) (Lecun et al., 1998) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) to learn to extract image and text features and then proceeded to align the representations of two modalities for retrieval. Recent proliferated large vision-language models (VLMs), such as vision Transformers (ViTs) (Dosovitskiy et al., 2020), UNITER (Chen et al., 2020) and CLIP (Radford et al., 2021), which are trained on large-scale short text-image corpus, have made remarkable progress for retrieving images from sentences with few objects and simple linguistic.

However, in the real world, natural languages are highly contextual (Fodor, 2001; Krojer et al., 2022) with long utterances. Context, including perceptual and temporal cues, plays a pivotal role in grounding the implication of a linguistically complex text (Li et al., 2023b). Figure 1 demonstrates such a case that identifies an image from six very similar candidates with a grammatically complicated description. Two major challenges of the retrieval are: 1) A model needs to understand nuanced textual cues, such as “*hand is blurry*”, “*eyes are almost shut*”, and “*looking at...*” across three grammatically complex sentences and align them with various context

*Equal Contribution.

†Corresponding author.

cues in each image. 2) Long-range dependencies among candidate images need to be captured to perform cross-modal reasoning for further alignment. The above interesting and challenging task is known as image retrieval from contextual descriptions (IRCD) (Krojer et al., 2022).

Despite the great success of VLMs, they can hardly tackle the above two challenges, and significantly lag behind human performance. Existing VLMs applied to text-image retrieval mainly include contrastive-based ones such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), randomly mask-based ones such as M3AE (Geng et al., 2022) and MaskVLM (Kwon et al., 2022), and attention-based ones such as ALBEF (Li et al., 2021) and BLIP (Li et al., 2022). These models have some limitations. 1) The former two focus more on high-level semantic alignment, while the fine-grained contextual cues may be largely ignored. 2) Existing mask-based ones randomly remove image patches, without specifically learning to concentrate on the key objects associated with text tokens, e.g., “*hand*” and “*eyes*” in Figure 1. 3) Dependencies among images are not specifically considered.

Previous state-of-the-art NDCR (Li et al., 2023b) proposed for IRCD divided the complex alignment into multiple simple ones and then combined them for final retrieval. However, the performance is highly dependent on the candidate’s distributions and is poor for fine-grained alignment on static images with a large variance. We observe that NDCR can hardly capture the key contextual cues in grammatically complex long sentences. Further, it also lacks zero-shot capability. Consequently, NDCR still suffers from the two challenges of IRCD. Details are discussed in Table 1 of Experiments.

To this end, we introduce ContextBLIP, a novel doubly contextual alignment scheme for the challenging IRCD task based on BLIP (Li et al., 2022). 1) Specifically, our ContextBLIP comprises a multi-scale adapter, a matching loss, and a text-guided masking loss. The learnable adapter, which is inserted into frozen BLIP, aims to capture higher-level and lower-level visual features of the candidate images. The two losses enable iterative supervision during the training stage, gradually allowing the adapter to highlight the focal patches of a single image to the linguistically complex textual cues. Such a way is termed as *intra-contextual alignment* that aims to tackle the first challenge issue of the IRCD task. 2) Then, we further fine-tune

ContextBLIP with a temporal Transformer to learn dependencies among candidate images, facilitating alignment between text to multiple images. This step is termed as *inter-contextual alignment* that aims to address the second challenge issue of IRCD. Experiments on a public benchmark show the effectiveness of the proposed ContextBLIP. The main contributions of this paper are listed as follows.

- We propose ContextBLIP, a simple yet effective method that relies on a doubly contextual alignment scheme for IRCD. It consists of a multi-scale adapter, a matching loss, and a text-guided masking loss, to learn to align the nuanced visual and textual cues, thus effectively tackling the first challenge of IRCD.
- We apply ContextBLIP for the zero-shot IRCD, and further fine-tune ContextBLIP with a temporal Transformer to learn the dependencies among different candidates, thus properly addressing the second challenge of IRCD.
- We conduct extensive experiments under various settings to show the superiority of our method. Our ContextBLIP can achieve comparable performance with proliferated GPT-4V (Yang et al., 2023) under various prompts. We also evaluate our ContextBLIP on a very recent benchmark MMVP-VLM and the results further confirm the effectiveness of the proposed method.

2 Related Work

Vision-language models (VLMs): VLMs (Radford et al., 2021; Li et al., 2021; Wang et al., 2022; Li et al., 2022, 2023a; Zhai et al., 2023; Fang et al., 2023; Xu et al., 2023; Sun et al., 2023) have shown great potential on image-text retrieval (Lin et al., 2014; Krishna et al., 2017; Plummer et al., 2015). However, tuning these models for new tasks is expensive. Adapters (Houlsby et al., 2019; Gao et al., 2021; Zhang et al., 2021; Chen et al., 2022; Lu et al., 2023), which can be inserted into a pre-trained VLM model, facilitate efficient adaptation to new tasks. The key difference between our adapter and previous ones: ours comprises multiple down-projection adapter layer (DPAL) that connect to the same up-projection adapter layer (UPAL). Each DPAL is inserted into distinct VLM layers, while the UPAL resides outside of the VLM. Such a design can effectively capture multi-level

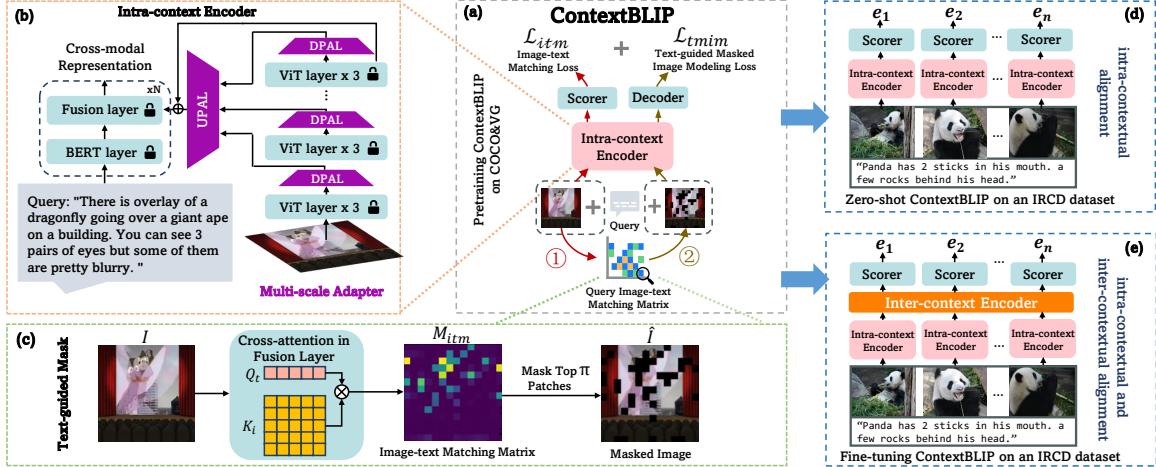


Figure 2: (a) Architecture of our ContextBLIP, including a BLIP-based **intra-context encoder**, a scorer for image-text matching (ITM, \mathcal{L}_{itm}), and a Transformer-based decoder for text-guided masked image modeling (TMIM, \mathcal{L}_{tmim}). (b) The **multi-scale adapter** in the encoder is co-supervised by \mathcal{L}_{itm} and \mathcal{L}_{tmim} on COCO&VG datasets, while BLIP is frozen. (c) The learnable **text-guided mask** is iteratively updated under the co-supervision. (d) Zero-shot ContextBLIP on the IRCD task. (e) Fine-tuning ContextBLIP for IRCD with the **inter-context encoder**.

subtle cues. While previous ones include multiple pairs of DPAL and UPAL inserted in VLM.

Masked Image Modeling (MIM): MIM, such as ViBERT (Lu et al., 2019) and MAE (He et al., 2022), refers to predicting the missing pixels in an image by using the surrounding pixels as context. MIM has been applied for various tasks, such as robust learning (Wang et al., 2023) and generation target (Bao et al., 2021, 2022). Recent studies, such as MaskVLM (Kwon et al., 2022), M3AE (Geng et al., 2022), VL-BEiT (Bao et al., 2022) extend MIM for both image and language masking. The key difference between ours and existing ones is: we employ a text-guided masking scheme that learns to generate masks under iterative supervision with two losses, thus gradually highlighting the focal patches of a single image to the key textual cues. While the previous ones randomly remove image patches, without specifically learning to focus the key visual objects associated with text tokens.

3 Our ContextBLIP

3.1 Overall Architecture

Figure 2(a) presents the overall architecture of the proposed ContextBLIP for the challenging IRCD task, which consists of an intra-context encoder based on frozen BLIP, a multilayer perceptron (MLP) module, and an image decoder. The overall procedure can be described in three steps. **(1) Pretraining:** We first pre-train the proposed ContextBLIP on the large-scale COCO (Lin et al.,

2014) and VG (Krishna et al., 2017) datasets.

Aiming at tackling the first challenge of IRCD, ContextBLIP includes three key ingredients, i.e., a multi-scale adapter (Figure 2(b)), and two co-supervision losses including image-text matching (ITM, \mathcal{L}_{itm}) loss and text-guided masked image modeling (TMIM, \mathcal{L}_{tmim}) loss. **(2) Intra-context Alignment:**

We directly apply pre-trained ContextBLIP for zero-shot IRCD (Figure 2(d)) on the public benchmark (e.g., IMAGECODE), effectively aligning the nuanced visual and textual cues. **(3) Intra- and Inter-context Alignment:** Then, we further fine-tune ContextBLIP (Figure 2(e)) on an IRCD benchmark with an inter-context encoder to learn long-range dependencies among image candidates, thus effectively addressing the second challenge of IRCD. Next, we detail four key components of our method, including the multi-scale adapter, \mathcal{L}_{itm} , \mathcal{L}_{tmim} and the inter-context encoder.

3.2 Multi-Scale Adapter

Our multi-scale adapter, which resides in the intra-context encoder of ContextBLIP, aims to learn to align nuanced visual and textual cues from tiny areas of an image and within linguistically complex descriptions, respectively. Figure 2 (b) illustrates the architecture of the proposed adapter. It comprises multiple down-projection adapter layer (DPAL) and a up-projection adapter layer (UPAL). DPALs are inserted into different ViT layers of frozen vanilla BLIP, while the UPAL resides out-

side of ViT layers. All DPALs connect to the same UPAL, such that both higher-level and lower-level features of candidate images can be effectively captured. We give the detailed formulation as follows.

For a pair of image-text (I, T) , $I \in \mathbb{R}^{h \times w \times c}$, $T \in \mathbb{R}^{1 \times t}$, where h, w, c are height, width and the number of channels of the image, t is the number of tokens in sentence T . We split an image into $p \times p$ patches and then augment them with positional encoding. We feed these image patches into the m -layer ViT. The intermediate visual representations generated at the l -th layer of the ViT can be denoted as $X^l = [x_1^l, \dots, x_i^l, \dots, x_{p^2}^l]$, where $l \in [1, m]$, $i \in [1, p^2]$ and $x_i^l \in \mathbb{R}^d$. Here d indicates the representation dimension of the ViT at the l -th layer. We feed X^l into a down-project adapter layer (DPAL) for mapping to lower-dimensional space. The output of DPAL can be expressed as follows,

$$\tilde{X}^l = \text{DPAL}(X^l), \quad (1)$$

where DPAL is an MLP network and $\tilde{X}^l \in \mathbb{R}^{p^2 \times \tilde{d}}$. Here $\tilde{d} = d/\delta$, where δ is the downsampling rate of DPAL in our adapter. We use the same rate for all DPALs and obtain the output representations from other DPALs. We aggregate these representations by simply concatenating or adding. Then, the aggregated representations \tilde{X} will be fed into the proposed up-projection adapter layer (UPAL) for up-projection mapping. The output of the UPAL can be expressed as follows,

$$Y = \text{UPAL}(\tilde{X}), \quad (2)$$

where $Y \in \mathbb{R}^{p^2 \times d}$ and UPAL is an MLP network.

Finally, we add the output of ViT to Y to obtain the final representations of the image for cross-modal matching. The textual query is encoded by frozen BERT (Devlin et al., 2019) of vanilla BLIP and then is fed into the fusion layer. We feed the representations of the fusion layer’s output to the scorer of the intra-context encoder to get a matching score e . By doing so, our multi-scale adapter can facilitate fine-grained interactions between subtle visual regions and linguistic concepts.

3.3 Co-supervision under \mathcal{L}_{itm} and \mathcal{L}_{tmim}

We train the proposed adapter with two losses, i.e., \mathcal{L}_{itm} and \mathcal{L}_{tmim} , that offer collaborative supervision to highlight key contextual cues in two modalities. Our ContextBLIP performs two separate forward computations to calculate \mathcal{L}_{itm} and \mathcal{L}_{tmim} . We detail them as follows.

Step 1: Computing \mathcal{L}_{itm} : We sequentially feed a pair of image-text into the intra-context encoder and the MLP-based scorer, and obtain the matching score for the pair. The matching loss \mathcal{L}_{itm} can be expressed as follows.

$$\mathcal{L}_{itm} = \frac{1}{3N} \sum_{i=1}^{3N} \text{CrossEntropy}(e_i, q_i), \quad (3)$$

where $e_i \in \mathbb{R}^2$, $i \in [1, 3N]$, indicates the matching score of the i -th image-text pair, and $q_i \in \mathbb{R}^2$, $i \in [1, 3N]$, refers to the groundtruth label that consists of 0 and 1. Here N is the training batch size. Inspired by vanilla BLIP, we additionally generate $2N$ pairs of negative samples based on cosine distances to N pairs. By distinguishing much more similar image candidates, our ContextBLIP can learn to align the nuanced textual and visual context concealed in tiny areas and within complex descriptions.

Step 2: Generating masks and computing

\mathcal{L}_{tmim} : We rely on cross-attentions outputted by the fusion layer to generate the text-guided image masking matrix. Figure 2 (c) demonstrates the generation procedure. Specifically, we manually define a mask ratio π , $\pi \in [0, 1]$, to determine the number of patches to be masked. Top π patches with the highest attention scores will be masked and then we can get the masking matrix. We remove the patches according to the mask matrix, feed the masked image to the intra-context encoder, and then use a Transformer-based decoder to reconstruct the image. The pixel-level reconstruction loss \mathcal{L}_{tmim} based on mean squared error (MSE) for a N -size training batch can be expressed as follows.

$$\mathcal{L}_{tmim} = \frac{1}{N} \frac{1}{\mu} \sum_{i=1}^N \sum_{j=1}^{\mu} \sum_{s=1}^S \text{MSE}(y_{ijs}, \hat{y}_{ijs}), \quad (4)$$

where μ is the number of masked patches and S is the number of pixels in each patch. Here \hat{y}_{ijs} and y_{ijs} refer to the original and corresponding reconstructed pixel respectively for the s -th pixel of the j -th masked patch in i -th instance of a training batch.

Step 3: Iterative refinement: The total loss of our ContextBLIP can be formulated as:

$$\mathcal{L} = \mathcal{L}_{itm} + \mathcal{L}_{tmim}. \quad (5)$$

Details are available in Appendix A.3.

We iteratively perform the above steps supervised by the two losses to learn to update the parameters of the multi-scale adapter, and refine the text-guided matrix, allowing the proposed ContextBLIP to gradually concentrate on the focal visual contextual keys associated with the textual cues.

3.4 Inter-context Encoder

The task of image retrieval from contextual descriptions (IRCD) requires understanding long-range contextual dependencies among candidate images. Keeping this in mind, we introduce a simple yet effective inter-context encoder, which aims to capture rich interactions between candidate images, as well as contextual alignment between a textual query to multiple images. We employ a two-layer Transformer that stacks on top of the intra-context encoder. The underlying design principle is general, and more advanced encoders can be used here for inter-context alignment. Thus, we can effectively tackle the second challenging issue of IRCD.

4 Experiments

4.1 Experimental Settings

We conduct experiments on four datasets, including large-scale COCO (Lin et al., 2014) and Visual Genome (VG) (Krishna et al., 2017) for pre-training, IMAGECODE (Krojer et al., 2022) for zero-shot and fine-tuning, and MMVP-VLM (Tong et al., 2024) for evaluating our fine-tuned ContextBLIP. We pre-train ContextBLIP on $4 \times$ A100 GPU cards, and other experiments on a RTX3090 GPU card. During the pre-training stage, we configure adapter downsampling rate δ as 2, the mask ratio π as 0.25. We use vanilla BLIP-129M checkpoint as our backbone, which involves 223M parameters. We implement our model on the PyTorch platform.

We select eight strong baselines including CLIP (Radford et al., 2021), UNITER (Chen et al., 2020), ViBERT (Lu et al., 2019), OFA (Wang et al., 2022), ALBEF (Li et al., 2021), BLIP (Li et al., 2022), BLIP-2 (Li et al., 2023a), and NDCR (Li et al., 2023b). Note that NDCR is the previous state-of-the-art method on IMAGECODE. We follow the previous work (Krojer et al., 2022) to use accuracy as the evaluation metric. The IMAGECODE dataset involves three categorizations in the test set, including “Video” which indicates the candidate images are collected from video frames, “Im-

Method	Params	Zero-shot			Fine-tuned		
		All	Video	Image	All	Video	Image
CLIP	473M	22.4	15.6	47.8	29.9	22.0	59.8
UNITER	-	19.8	13.6	42.9	25.7	19.1	50.5
ViLBERT	-	19.3	13.5	40.8	24.5	18.0	49.3
OFA [†]	-	-	-	-	27.2	21.0	52.1
ALBEF [†]	-	27.7	15.7	73.3	-	-	-
BLIP [†]	223M	28.1	15.9	74.4	<u>34.1</u>	22.7	77.4
BLIP-2 [†]	1.2B	<u>29.4</u>	<u>16.3</u>	79.2	-	-	-
NDCR	440M	-	-	-	<u>34.1</u>	26.1	64.3
Ours	240M	31.0	18.8	<u>77.1</u>	35.7	<u>24.4</u>	78.5
Human						90.8	

Table 1: Comparisons on IMAGECODE. Our model achieves state-of-the-art accuracy, with only 2.4M more parameters on vanilla BLIP. Baselines marked with [†] indicate that we reproduced the scores as no results are publicly available. The best and second-best results are highlighted with **bold** and underline, respectively.

age” which represents the ones that are constructed based on static images, and “All” is the hybrid of the above two datasets.

4.2 Main Results

Zero-shot on IMAGECODE: We per-train the proposed ContextBLIP on COCO and VG. Table 1 reports the comparisons between ours and the baselines for zero-shot ContextBLIP on the IMAGECODE dataset. Equipped with our multi-scale adapter that only involves 2.4M parameters, our ContextBLIP achieves state-of-the-art performance on all test instances and video frames. Compared to the existing CLIP that uses global information in text and images for alignment, our ContextBLIP obtains 8.6% higher accuracy. Our method also outperforms existing UNITER and ViLBERT by 10.2% and 10.7%, respectively. The two methods employ random masks for cross-modal alignment. We attribute the improvement to *intra-context alignment* based on multi-scale adapter and text-guided masking. The former learns both higher- and lower-level visual features, and their rich interactions at each level. The latter enables our ContextBLIP to concentrate on focal contextual cues. We also observe that existing BLIP-2 performs better than ours by 2.1 points on static images, and this is not a surprise as BLIP-2 is 50 larger than ContextBLIP.

Fine-tuned on IMAGECODE: The right side of Table 1 reports the comparisons of our fine-tuned ContextBLIP to baselines. Our ContextBLIP can achieve state-of-the-art accuracy on test sets of “All” and “Image” and this further confirms the

[†]More hyperparameters are available in Appendix A.1

	Context	Quantities	Spatial	Negation	Occlusion	Nuances	Co-reference	Meta Properties
CLIP (Zero-shot)	13.3	23.7	20.6	11.7	14.1	9.9	25.3	29.2
BLIP (Zero-shot)	15.4	<u>30.9</u>	27.7	11.4	16.7	11.4	28.9	12.5
Ours (Zero-shot)	17.3	39.2	36.9	<u>19.0</u>	<u>19.1</u>	11.0	37.4	25.0
CLIP (Fine-tuned)	<u>19.2</u>	<u>30.9</u>	30.5	17.3	18.6	<u>14.8</u>	32.5	<u>33.3</u>
Ours (Fine-tuned)	25.1	39.2	<u>31.9</u>	25.2	23.7	19.7	<u>36.1</u>	37.5

Table 2: Comparison of challenging samples in the IMAGECODE benchmark under zero-shot and fine-tuned settings. The samples involve challenging contextual alignment such as “Context”, “Nuances” and “Co-reference”.

	Image Size	Params	⌚	🔍	⚡	⬆️	💡	🎥	⚙️	🅰️	📷	MMVP Average
DFN ViT-H-14 (Fang et al., 2023)	224 ²	986.1M	20.0	26.7	73.3	26.7	26.7	66.7	46.7	<u>13.3</u>	53.3	<u>39.3</u>
MetaCLIP ViT-H-14 (Xu et al., 2023)	224 ²	986.1M	6.7	13.3	60.0	13.3	6.7	53.3	26.7	<u>13.3</u>	33.3	25.2
EVA02 ViT-bigE-14+ (Sun et al., 2023)	224 ²	5044.9M	13.3	20.0	<u>66.7</u>	13.3	<u>26.7</u>	<u>66.7</u>	26.7	20.0	33.3	33.3
BLIP (Li et al., 2022)	224 ²	223M	13.3	6.7	40.0	20.0	26.7	66.7	46.7	20.0	46.7	31.9
ContextBLIP (Ours)	224 ²	240M	26.7	46.7	60.0	40.0	46.7	93.3	<u>40.0</u>	20.0	53.3	47.4

Table 3: Comparisons on challenging visual patterns on the MMVP-VLM benchmark. We follow the previous work (Tong et al., 2024) to use the symbols, such as ⌚ and 🔎, to indicate 9 challenging visual patterns. More details can be found in Appendix B. The best and second-best results are marked in **bold** and underline.

superiority of our method. ContextBLIP outperforms exiting CLIP and OFA by 18.7 and 26.4 points on static images, showing the effectiveness of both intra-context and inter-context alignment. We also observe that the proposed ContextBLIP performs slightly worse than the previous state-of-the-art NDCR by 1.7 points on video frames. However, NDCR pays more attention to the cross-modal alignment of video frames and complex text, while the ability of static images is largely underexplored. We observe that ours is better than NDCR by 14.2 points on “Image”, with nearly half fewer parameters. We also attribute this to the model’s capability to handle distributions of the candidate image, where static ones present a large variance.

Comaprison of challenging samples of IMAGECODE: We compare our method with the existing CLIP and BLIP on the 200 challenging samples highlighted in IMAGECODE. These samples are manually labeled with high-quality annotations. Table 2 shows that our ContextBLIP consistently performs best under the fine-tuned setting on various scenarios, and performs best on most scenarios under the zero-shot setting. These results further confirm the superiority of our ContextBLIP in tackling the two challenges of the IRCD task.

Comparisons on MMVP-VLM Benchmark: We compare our ContextBLIP with a very recent benchmark MMVP-VLM (Tong et al., 2024),

which aims to evaluate how well a VLM model handles various challenging visual patterns. These manually defined patterns such as specific features, and positional and relational context, require a model to capture contextual details of visual cues or perform cross-modal reasoning. Table 3 shows that our ContextBLIP, which is fine-tuned on the IMAGECODE dataset, achieves the best accuracy on most of the patterns. Further, the proposed ContextBLIP involves much fewer parameters, e.g, the number of parameters in MetaCLIP ViT-H-14 (Xu et al., 2023) is nearly 25 times more than ContextBLIP, while average accuracy is lower than ours by 22.2 points.

	Zero-shot	Fine-tuned
w/o Multi-scale Adapter	15.9	22.7
w/o \mathcal{L}_{tmim}	<u>18.2</u>	23.0
w/o Inter-context Encoder	-	<u>23.4</u>
ContextBLIP (Ours)	18.8	24.4

Table 4: Ablation study on the IMAGECODE dataset.

4.3 Ablation Study

We conduct an ablation study on IMAGECODE to measure the contribution of each component to IRCD. Table 4 reports the results. Under both zero-shot and fine-tuned settings, we observe that the removal of the multi-scale adapter leads to a significant performance decrease, i.e., 2.9 points and 1.7

points, indicating the effectiveness of the adapter. Under the fine-tuned setting, it shows that the removal of the inter-context encoder leads to a 1 point performance drop, suggesting the effectiveness of long-range dependencies for the retrieval.

4.4 Sensitivity Analysis

We conduct experiments for sensitivity analysis on IMAGECODE under the zero-shot setting.

Masking ratio: To evaluate how mask ratio π affect the retrieval, we configure π as 0.25, 0.50 and 0.75, respectively. Table 5 shows lower masking can increase the accuracy, e.g., from 75.4% to 79.5%. This interesting finding on the IRCD task does not align with previous studies that advocated for higher masking ratios (He et al., 2022; Geng et al., 2022; Bao et al., 2022; Kwon et al., 2022). One possible underlying reason is the challenging alignment in IC RD require more dense visual cues among similar candidates.

Mask Ratio	All	Video	Image
0.25	31.1	19.9	79.5
0.50	31.0	20.6	76.3
0.75	29.8	19.4	75.4

Table 5: Sensitivity analysis of the mask ratio π .

Position of adapter inserted: We analyze the impact of adapters inserted into different layers on the retrieval. We include three cases, i.e., inserting both down- and up-projection layers in the top layer of BLIP, inserting down-projection layers in 3-th, 6-th, 9-th, 12-th layers, and inserting down-projection layers in each layer. Table 6 shows inserting adapters at multiple layers of BCLIP achieved the highest overall accuracy. This suggests there exists a tradeoff for the number of layers to be inserted.

Downsampling rate δ : We evaluate how δ affects the retrieval. We configure δ as 1, 2, 4, and 8, respectively. Table 7 shows that our ContextBLIP achieves the best performance when δ is set as 2. The results present a large variance, e.g., 77.0 for $\delta = 8$ and 79.5 for $\delta = 2$. This suggests that a small rate δ leads to better performance.

Layer	All	Video	Image	Params
[12]	30.4	19.4	78.1	223.6M
[3,6,9,12]	31.1	19.9	79.5	225.4M
[1-12]	30.5	19.8	77.0	230.1M

Table 6: Sensitivity analysis of the position of adapter.

Type	1	2	4	8
All	30.2	31.1	30.8	30.1
Video	19.6	19.9	20.1	19.3
Static	76.5	79.5	77.2	77.0

Table 7: Sensitivity analysis of downsampling rate δ .

4.5 Case Study



Query : Two girls in the frame. The one with black hair has her hand covering her face.

P_{BLIP} [0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100] (\times)

P_{Ours} [0.090, 0.072, 0.131, 0.070, 0.074, 0.073, 0.103, 0.108, 0.106, **0.172**] (\checkmark)

Key contextual cue : Two girls in the frame.

P_{BLIP} [0.100, 0.099, 0.100, 0.099, 0.099, 0.099, 0.101, 0.101, 0.101, 0.100] (\times)

$P_{Ours'}$ [0.092, 0.066, 0.102, 0.058, 0.063, 0.063, **0.151**, **0.152**, 0.106, **0.148**] (\checkmark)



Query : There is a person in white on the left side of the image who is just about to serve the ball with one hand.

P_{BLIP} [0.122, 0.120, 0.044, **0.161**, 0.089, 0.133, 0.059, 0.115, 0.083, 0.075] (\times)

P_{Ours} [**0.206**, 0.106, 0.026, 0.090, 0.079, 0.104, 0.036, 0.106, 0.127, 0.121] (\checkmark)

Key contextual cue : There is a person who is just about to serve the ball with one hand.

P_{BLIP} [0.171, 0.120, 0.033, **0.182**, 0.062, 0.106, 0.036, 0.150, 0.086, 0.054] (\times)

$P_{Ours'}$ [**0.304**, 0.085, 0.012, 0.069, 0.048, 0.088, 0.020, 0.136, 0.145, 0.092] (\checkmark)

Figure 3: (a) Zero-shot: P_{BLIP} and P_{Ours} are two matching scores of BLIP and ours, and $P_{BLIP'}$, $P_{Ours'}$ are scores for the key contextual cue. (b) Fine-tuned: P_{BLIP} and P_{Ours} are two matching scores of BLIP and ours, and $P_{BLIP'}$, $P_{Ours'}$ are scores for the key contextual cue.

Figure 3 two cases to visually show why our ContextBLIP performs better for the challenging IRCD task. Both the zero-shot case and fine-tuned case demonstrate that our ContextBLIP not only yields the highest matching score for the golden candidate image. More importantly, it is also capable of aligning the key context cues in two modalities. For example, our method can understand textual cues “two girls in the frame” in the long query, and yield more accurate alignment to the 7-th, 8-th, and 10-th candidate images with higher matching scores. Equipped with the proposed inter-context encoder, our ContextBLIP can accurately identify the 1-st candidate with the highest score.

4.6 Comparisons with GPT-4V

	Video			Static		
	1	2	3	1	2	3
GPT-4V	22%	20%	22%	46%	54%	44%
ContextBLIP (Ours)	20%	24%	26%	82%	80%	78%

Table 8: Comparison with GPT-4V on random samples.

We compare our ContextBLIP with the OpenAI multi-modal large language model (MLLM) GPT-4V (Yang et al., 2023). We randomly sample 50 instances three times from the test set of IMAGECODE, and use a prompt in the form of “Which image fits the text description best, please output the serial number of the image:< *text_query* >” to identify the image. Table 8 shows that our model can achieve comparable results to GPT-4V, despite involving about 7,500 times fewer parameters. This also aligns with some existing studies (Tong et al., 2024) that GPT-4V may fail to understand subtle cues. Figure 4 demonstrates how we prompt GPT-4V for the challenging IRCD task. Table 9 reports the matching scores of three instances selected from the test set of IMAGECODE. For the challenging “ID 8129”, we observe that GTP-4V performs poorly.



Figure 4: Case illustration of how we prompt GPT-4V for IRCD. The red boxes represent the GPT-4V’s response and the yellow one indicates our prediction.

Instance ID	Wrong Matching	Right Matching	All Results
7407	0	20	20
7836	1	19	20
8129	11	9	20

Table 9: Statistics of retrieval performance of GPT-4V under 20 different prompts. “Instance ID” indicates the instance number in the IMAGECODE test set.

More cases are provided in Appendix D.2

4.7 More discussion

We also compare our ContextBLIP with the previous state-of-the-art NDCR (Li et al., 2023b) model on the IRCD task. We follow the existing NDCR to divide the linguistic complex descriptions into multiple segments with different lengths. We are interested in such a setting and evaluate how well the proposed ContextBLIP performs over sentences that are split from the same long description. Figure 10 illustrates that the proposed ContextBLIP consistently outperforms the existing NDCR under various sentence lengths.

Nums of props	1	2	3	4	5	6
All	72	899	1215	99	14	3
NDCR	29	327	384	28	2	0
ContextBLIP	36	391	416	37	3	1
Improvement (%)	9.7	7.1	2.6	9.1	7.1	33.3

Table 10: Comparisons of ContextBLIP with NDCR over various lengths of textual propositions, where 2 indicates the number of segments split from a long query.

5 Conclusion

This paper presents ContextBLIP, a simple yet effective doubly contextual alignment scheme for the challenging IRCD. Our model comprises a multi-scale adapter, a matching loss, and a text-guided masking loss. The adapter learns to capture fine-grained visual cues. The two losses enable iterative supervision for the adapter, gradually highlighting the focal patches of a single image to the key textual cues. Then, ContextBLIP further employs an inter-context encoder to learn dependencies among candidates, facilitating accurate alignment between text to multiple images. Consequently, the nuanced cues concealed in textual and visual modalities can be effectively aligned. Experiments on two benchmarks show the effectiveness of our method. We observe that our ContextBLIP can yield comparable results with GPT-4V, despite involving about 7,500 times fewer parameters. In the future, we plan to extend our method to text-to-video retrieval.

6 Limitation

The adaptive mask ratio is worth considering in the future, as a fixed masking ratio in our paper may not dynamically adapt to different cross-modal interactions. The proposed method may also have limitations for fine-grained retrieval for long videos, as pre-training on long videos is time-expensive and requires very large GPUs.

References

- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.
- Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. 2022. Vi-beit: Generative vision-language pretraining. *arXiv preprint arXiv:2206.01127*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120.
- Zhe Chen, Yuchen Duan, Wenhui Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **Bert**: Pre-training of deep bidirectional transformers for language understanding.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. 2023. **Data filtering networks**.
- Jerry Fodor. 2001. **Language, thought and compositionality**. *Royal Institute of Philosophy Supplement*, 48:227–242.
- Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Jiao Qiao. 2021. Clip-adapter: Better vision-language models with feature adapters. *ArXiv*, abs/2110.04544.
- Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurmans, Sergey Levine, and Pieter Abbeel. 2022. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. **Parameter-efficient transfer learning for NLP**. In *ICML*, pages 2790–2799.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. **Scaling up visual and vision-language representation learning with noisy text supervision**. In *ICML*, pages 4904–4916.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73.
- Benno Krojer, Vaibhav Adlakha, Vibhav Vineet, Yash Goyal, Edoardo Ponti, and Siva Reddy. 2022. Image retrieval from contextual descriptions. *arXiv preprint arXiv:2203.15867*.
- Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. 2022. Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*.
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. **Gradient-based learning applied to document recognition**. *IEEE*, 86(11):2278–2324.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. **BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation**. In *ICML*, pages 12888–12900.
- Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. **Align before fuse: Vision and language representation learning with momentum distillation**. In *NeurIPS*, volume 34, pages 9694–9705.
- Yunxin Li, Baotian Hu, Yunxin Ding, Lin Ma, and Min Zhang. 2023b. A neural divide-and-conquer reasoning framework for image retrieval from linguistically complex text. *arXiv preprint arXiv:2305.02265*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755.
- Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. 2023. **Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling**. *arXiv preprint arXiv:2302.06605*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. *Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*. In *NeurIPS*, volume 32.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. *Learning transferable visual models from natural language supervision*. In *ICML*, volume 139, pages 8748–8763.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*.

Haoqing Wang, Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhi-Hong Deng, and Kai Han. 2023. Masked image modeling with local multi-scale reconstruction. In *CVPR*, pages 2122–2131.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. *OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework*. In *ICML*, volume 162, pages 23318–23340.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2023. Demystifying clip data. *arXiv preprint arXiv:2309.16671*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of Imms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*.

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-Adapter: Training-free CLIP-Adapter for Better Vision-Language Modeling.

A Implementation details

A.1 Training Hyper-parameters

Pre-training. We conducted pre-training on 4 Nvidia A100 GPUs for about 10 hours, randomly sampling 224^2 patches from images and using RandAug for data augmentation.

Hyperparameters	
max epoch	20
batch size	256
vit	Vit-B/16
input resolution	224^2
augmentation	RandAug
optimizer	AdamW
base learning rate	3e-4
warmup learning rate	1e-6
minimize learning rate	1e-6
momentum	(0.9, 0.999)
warmup steps	3000
weight decay	1e-4
mask ratio	0.25
reduction	2
random seed	42

Table 11: Pretraining setting on IMAGECODE

Fine-tuning. Compared to Krojer *et al.*’s multi-step approach involving +Context Batch, +Context Module, and +Temporal Embeddings, we streamline Krojer *et al.*’s procedure by directly fine-tuning our model on IMAGECODE and then separately training a contextual modeling module between the backbone and the prediction head to enable contextual comparison and reasoning.

During the first stage, we conduct full fine-tuning for 20 epochs. After that, we add a two-layer transformer as a context/temporal module, which is identical to Krojer *et al.*. To optimize the module while keeping other parameters fixed, we freeze all parts of the model except the inserted module and train it alone for 5 epochs.

We set the batch size to 360 images (36 batches of 10-image-set) and use the Adam (Kingma and Ba, 2014) optimizer with learning rate of 2e-6 and weight decay of 0.01. We conduct the experiment on one NVIDIA Geforce 3090 for about one day. All other baselines also adopted the same settings.

A.2 Dataset Details

Table 13 provides statistics for the datasets used in pre-training and fine-tuning. It includes three datasets: COCO, VG (Visual Genome), and IMAGECODE. The number of images (#image) and texts (#text) for each dataset are listed.

Hyperparameters	
max epoch	25
batch size	36
input resolution	224 ²
augmentation	RandAug
optimizer	Adam
backbone learning rate	2e-6
head learning rate	1e-4
momentum	(0.9, 0.999)
learning rate scheduler	ExponentialLR(0.95)
weight decay	1e-2
random seed	10

Table 12: Fine-tuning setting on IMAGECODE

	COCO	VG	IMAGECODE
#image	113K	100K	94020
#text	567K	769K	21202

Table 13: Details of datasets.

A.3 Model Details

In section 3.2, the scorer is a 1x768 linear layer. In section 3.3 (Step 2), the decoder is a four-layer transformer, with each layer having twelve attention heads and a feature dimension of 768.

In section 4.2, for CLIP, ViLBERT, and UNITER, we utilized the models with added context modules and temporal embeddings as proposed by Krojer et al. Our-Context model incorporated a context module in the same manner, while other models did not have such a module added. For ALBEF, due to its pretraining on images with a resolution of 256x256, we conducted experiments using a slightly higher resolution (224x224) compared to other models.

B Validation Performance

Comparison with state-of-the-art methods on the valid set of IMAGECODE is shown on Table 15.

C Additional Case Studies

For tasks involving high image similarity and detailed textual descriptions, such as contrastive image retrieval, the challenge is considerable. We will demonstrate the superior performance of our MIM adaption method in more scenarios and provide the following examples to further illustrate the doubly contextual alignment capabilities of our model.

Figure 5 presents examples of two zero-shot experiments. In the left image, when two men are facing each other, our model assigns significantly higher matching scores compared to other images.

In the right image, despite the difficulty in estimating the proportion of the man’s right eye visible, our model assigns obviously higher scores to the first two images showing the right eye. In contrast, BLIP assigns similar scores to all ten images in both samples, indicating its difficulty in attending to the textual cues used to align the intra-contextual information.

Figure 6 showcases examples of fine-tuning experiments. The left image in Figure 6 illustrates an example of the Quantity phenomenon, with the subheading "Two thumbs on the egg." We observe that our model infers significantly higher confidence scores for images 2 and 3 (matching the description of thumb quantity) and infers higher BLIP confidence scores for images 0, 2, and 3 (where the description of thumb quantity does not match). Thus, we demonstrate that our model indeed performs better in dual-contextual alignment.

The right image in Figure 6 demonstrates an example of the Meta-property phenomenon, with the subheading "The man’s face is blurry." We find that our model infers higher confidence scores for images 2, 3, 7, and 8 (all relatively matching the description) while BLIP infers higher confidence scores for images 1, 2, 3, and 8 (where image 1 is clear but does not match the description). Hence, it indicates that our model indeed performs better in understanding intra-contextual alignment.

In conjunction with the main text, we demonstrate the advantages of our model in two challenges: better intra-contextual alignment, better inter-contextual alignment. It can be seen that in many scenarios, our model exhibits better dual-contextual alignment capabilities compared to BLIP.

D More about GPT-4V Experiment

D.1 Sampling Details

Due to GPT-4V’s high computational demands, we created distinct test datasets from IMAGECODE’s static and video-shot sections. We randomly chose 50 samples from each, repeating thrice to avoid bias. Each sample included a ground-truth image plus nine adjacent ones, since GPT-4V processes up to ten images simultaneously. We prompted GPT-4V in form of "Which image fits the text description best? Please output the serial number of the image:< *text_query* >" to select the matching image. We used three random seeds: 1, 10, 100, to explore the datasets with the text length

Method	Params	Zero-shot			Fine-tune		
		All	Video	Image	All	Video	Image
CLIP (Radford et al., 2021)	473M	22.4	15.6	47.8	29.9	22.0	59.8
UNITER (Chen et al., 2020)	-	19.8	13.6	42.9	25.7	19.1	50.5
ViLBERT (Lu et al., 2019)	-	19.3	13.5	40.8	24.5	18.0	49.3
OFA (Wang et al., 2022)	-	-	-	-	27.2	21.0	52.1
ALBEF (Li et al., 2021)	-	27.7	15.7	73.3	-	-	-
M3AE (Geng et al., 2022)	-	-	-	-	-	-	-
BLIP (Li et al., 2022)	223M	28.1	15.9	74.4	<u>34.1</u>	22.7	<u>77.4</u>
BLIP-2 (Li et al., 2023a)	1.2B	<u>29.4</u>	<u>16.3</u>	79.2	-	-	-
NDCR (Li et al., 2023b)	440M	-	-	-	<u>34.1</u>	26.1	64.3
ContextBLIP (Ours)	240M	31.0	18.8	<u>77.1</u>	35.7	24.4	78.5
Human Performance					90.8		

Table 14: Comparison with state-of-the-art methods on IMAGECODE task. Our model achieve highest zero-shot and fine-tune performance while requiring a relatively fewer number of parameters. The best and second-best results are marked in **bold** and underline.

Method	Params	Zero-shot			Fine-tune		
		All	Video	Image	All	Video	Image
CLIP (Radford et al., 2021)	473M	21.8	14.9	51.6	30.6	22.3	67.0
UNITER (Chen et al., 2020)	-	19.8	13.6	42.9	26.0	19.9	52.8
ViLBERT (Lu et al., 2019)	-	18.5	14.0	37.9	25.1	19.4	49.5
ALBEF (Li et al., 2021)	-	28.2	<u>17.0</u>	<u>77.0</u>	-	-	-
OFA (Wang et al., 2022)	-	-	-	-	27.2	21.0	52.1
BLIP (Li et al., 2022)	223M	<u>28.4</u>	<u>17.0</u>	77.9	<u>36.2</u>	<u>26.3</u>	<u>79.3</u>
ContextBLIP (Ours)	240M	30.6	19.9	<u>77.0</u>	38.5	28.7	81.2

Table 15: Validation Performance on IMAGECODE

distribution (in tokens) illustrated in the Figure 7.

In this experiment, using Python scripts with the API interface of the gpt-4-vision-preview model, we make a request to GPT-4V. If encounter a refusal to answer, we use the browser version to ask again until a result is obtained.

D.2 More Comparison with GPT-4V

This part provides some more examples of GPT4V prediction errors, where our model predicts correctly. The first two samples are from video shot set, and the last two are from static pictures.

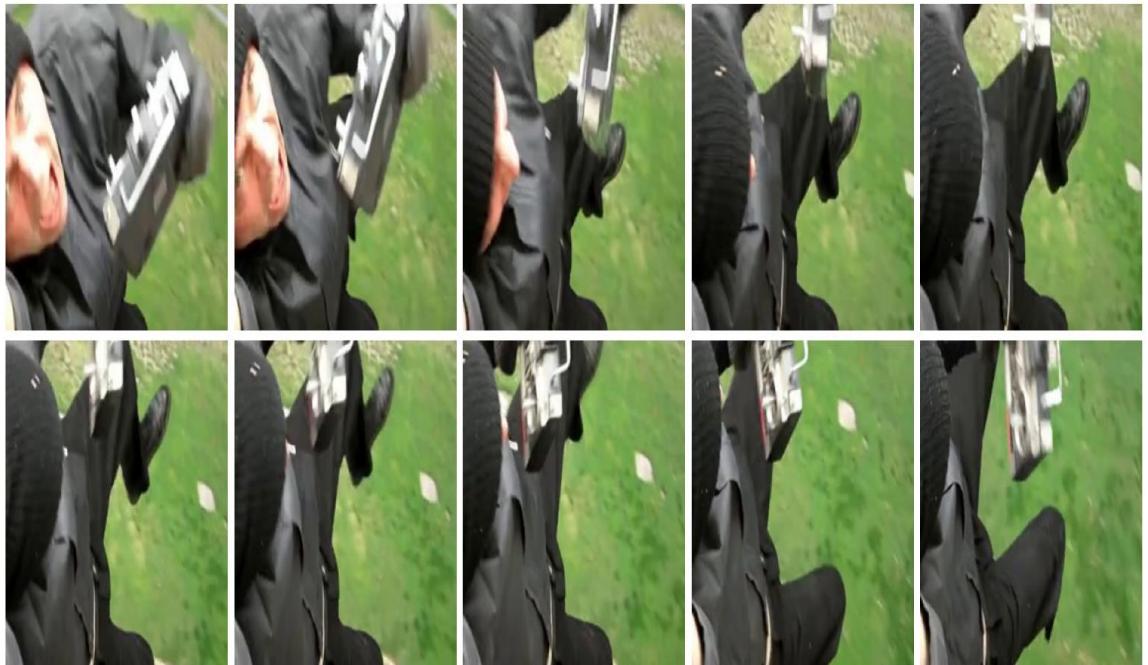
	Image Size	Params	Orientation and Direction	Presence of Specific Features	State and Condition	Quantity and Count	Positional and Relational Context	Color and Appearance	Structural and Physical Characteristics	Text	Viewpoint and Perspective	MMVP Average
OpenAI ViT-L-14 (Radford et al., 2021)	224 ²	427.6M	13.3 13.3 20.0 20.0 13.3 53.3 20.0 6.7 13.3									19.3
OpenAI ViT-L-14 (Radford et al., 2021)	336 ²	427.9M	0.0 20.0 40.0 20.0 6.7 20.0 33.3 6.7 33.3									20.0
SigLIP ViT-SO-14 (Zhai et al., 2023)	224 ²	877.4M	26.7 20.0 53.3 40.0 20.0 <u>66.7</u> <u>40.0</u> <u>20.0</u> 53.3									37.8
SigLIP ViT-SO-14 (Zhai et al., 2023)	384 ²	878.0M	<u>20.0</u> <u>26.7</u> <u>60.0</u> <u>33.3</u> 13.3 <u>66.7</u> 33.3 26.7 53.3									37.0
DFN ViT-H-14 (Fang et al., 2023)	224 ²	986.1M	<u>20.0</u> <u>26.7</u> 73.3 26.7 26.7 <u>66.7</u> 46.7 13.3 53.3									<u>39.3</u>
DFN ViT-H-14	378 ²	986.7M	13.3 20.0 53.3 <u>33.3</u> 26.7 <u>66.7</u> <u>40.0</u> <u>20.0</u> 40.0									34.8
MetaCLIP ViT-L-14 (Xu et al., 2023)	224 ²	427.6M	13.3 6.7 66.7 6.7 <u>33.3</u> 46.7 20.0 6.7 13.3									23.7
MetaCLIP ViT-H-14 (Xu et al., 2023)	224 ²	986.1M	6.7 13.3 <u>60.0</u> 13.3 6.7 53.3 26.7 13.3 33.3									25.2
EVA01 ViT-g-14 (Sun et al., 2023)	224 ²	1136.4M	6.7 <u>26.7</u> 40.0 6.7 13.3 <u>66.7</u> 13.3 13.3 20.0									23.0
EVA02 ViT-bigE-14+ (Sun et al., 2023)	224 ²	5044.9M	13.3 20.0 66.7 13.3 26.7 <u>66.7</u> 26.7 <u>20.0</u> 33.3									33.3
BLIP (Li et al., 2022)	224 ²	223M	13.3 6.7 40.0 20.0 26.7 <u>66.7</u> 46.7 <u>20.0</u> <u>46.7</u>									31.9
ContextBLIP (Ours)	224 ²	240M	26.7 46.7 <u>60.0</u> 40.0 46.7 93.3 40.0 <u>20.0</u> 53.3									47.4

Table 16: Comparison with various VLMs on different visual patterns in MMVP-VLM benchmark. The best and second-best results are marked in **bold** and underline. We identify 9 visual patterns: : Orientation and Direction, : Presence of Specific Features, : State and Condition, : Quantity and Count, : Positional and Relational Context, : Color and Appearance, : Structural and Physical Characteristics, : Text and : Viewpoint and Perspective.



Query : *Two men facing each other.*

P_{BLIP} [0.100, 0.101, 0.101, 0.100, 0.100, 0.100, 0.099, 0.100, 0.100, 0.100] (✗)
 P_{Ours} [0.076, 0.089, 0.094, 0.090, 0.084, 0.099, **0.142**, 0.121, 0.093, 0.113] (✓)



Query : *The man's right eye is 40% visible.*

P_{BLIP} [0.100, 0.099, 0.103, 0.099, 0.099, 0.099, 0.099, **0.105**, 0.099, 0.099] (✗)
 P_{Ours} [**0.145**, 0.129, 0.105, 0.086, 0.090, 0.083, 0.084, 0.085, 0.105, 0.089] (✓)

Figure 5: Zero-shot cases from the test set. Our model has advantages over BLIP in both confidence scores.



Query : *Two thumbs on egg, no blur, bottom right blue dial has just started to be covered.*

P_{BLIP} [0.118, 0.047, 0.104, **0.134**, 0.079, **0.131**, 0.095, 0.105, 0.075, 0.111] (✗)

P_{Ours} [0.097, 0.063, **0.137**, **0.131**, 0.068, 0.107, 0.106, 0.111, 0.087, 0.092] (✓)

Key contextual cue : *Two thumbs on egg.*

$P_{\text{BLIP}'}$ [**0.187**, 0.034, **0.196**, **0.242**, 0.025, 0.045, 0.065, 0.048, 0.025, 0.132] (✗)

$P_{\text{Ours}'}$ [0.086, 0.036, **0.412**, **0.22**, 0.016, 0.031, 0.045, 0.034, 0.025, 0.095] (✓)



Query : *The man's face is visible but very blurry. He is punching downward but you can't see any of his elbow because it is blocked by his fist. His fist is visible but blurry.*

P_{BLIP} [0.063, 0.076, 0.193, **0.262**, 0.016, 0.017, 0.063, 0.063, 0.229, 0.019] (✗)

P_{Ours} [0.087, 0.067, **0.24**, 0.207, 0.014, 0.008, 0.069, 0.126, 0.168, 0.014] (✓)

Key contextual cue : *The man's face is very blurry.*

$P_{\text{BLIP}'}$ [0.011, **0.117**, **0.141**, **0.39**, 0.006, 0.002, 0.046, 0.084, **0.199**, 0.003] (✗)

$P_{\text{Ours}'}$ [0.021, 0.113, **0.194**, **0.215**, 0.008, 0.001, 0.051, **0.2**, **0.194**, 0.003] (✓)

Figure 6: Two cases from the test set after fine-tuning. Our model outperforms BLIP in confidence scores for both compound and simple proposition texts.

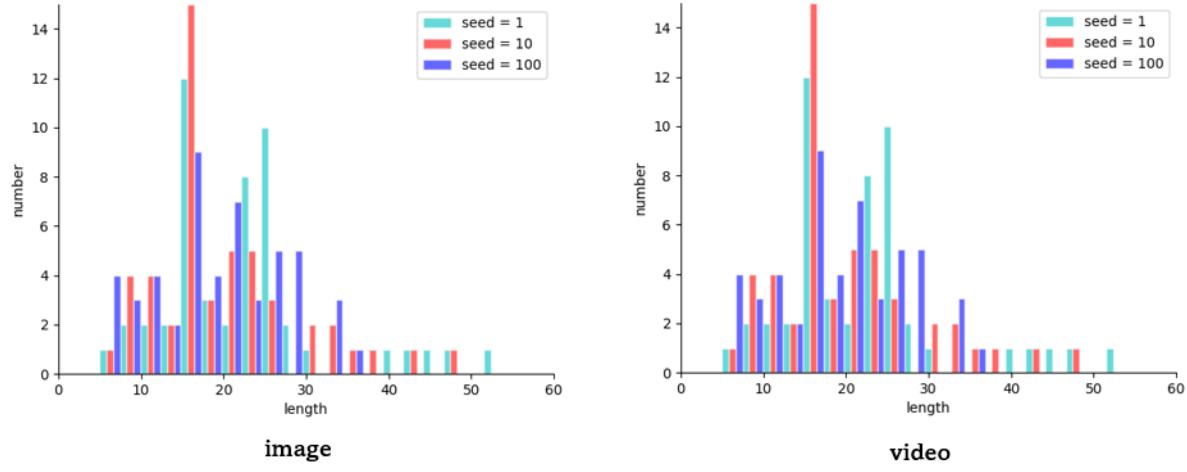


Figure 7: Distribution of the number of tokens across contextual descriptions in the subset of IMAGECODE. The distributions of the three sampling results are similar to the overall distribution described in the IMAGECODE text, and the average text length is about 20.



Q: Which image fits the text description best? Please output the serial number of the image : The SUV is right along side of the yellow and red truck. You can still barely see the cab of the truck.

 ChatGPT

Image 4 matches the description best. The SUV is right alongside the yellow and red truck, and you can still barely see the cab of the truck.

Figure 8: A case from the video-shot set. This is an enlarged version of the example in the main text. The yellow boxes indicate the correct image and our model's result, while the red boxes represent the model's output. We will see our model's result is correct but the GPT-4V's output is wrong.



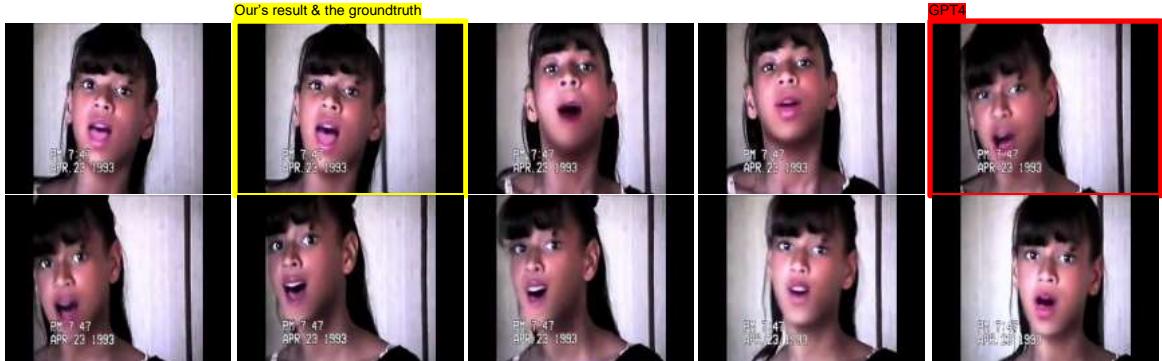
Q: Which image fits the text description best? Please output the serial number of the image : No hands are visible and the hanger on the silver ball on the train is pointing almost straight down.



ChatGPT

After reviewing the provided images, the one that best fits the description "No hands are visible and the hanger on the silver ball on the train is pointing almost straight down" is : **Image 3** . This image does not show any hands and the hanger on the silver ball is oriented almost straight down.

Figure 9: A case from the video-shot set. The yellow boxes indicate the correct image and our model's result, while the red boxes represent the model's output. We will see our model's result is correct but the GPT-4V's output is wrong.



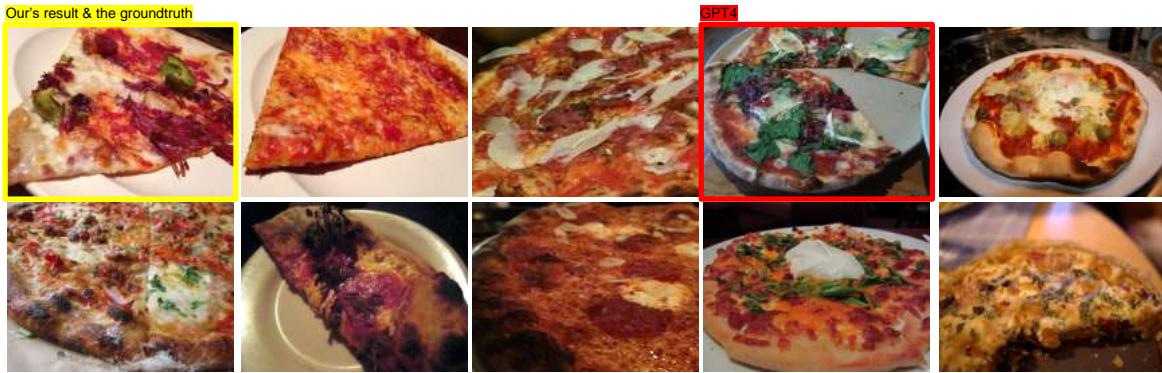
Q: Which image fits the text description best? Please output the serial number of the image : The girl has her mouth wide open but you can't see any teeth at all.



ChatGPT

The image that fits the text description best, where the girl has her mouth wide open but no teeth are visible at all, is image **number 6**.

Figure 10: A case from the video-shot set. The yellow boxes indicate the correct image and our model's result, while the red boxes represent the model's output. We will see our model's result is correct but the GPT-4V's output is wrong.



Q: Which image fits the text description best? Please output the serial number of the image : There is a single slice of pizza on a white plate with green things on it.



ChatGPT

The image that fits the text description best is **Serial Number 4**.

Figure 11: A case from the static pictures set. The yellow boxes indicate the correct image and our model's result, while the red boxes represent the model's output. We will see our model's result is correct but the GPT-4V's output is wrong.



Q: Which image fits the text description best? Please output the serial number of the image : Three sheep are in the foreground of an image with many more sheep visible in the distance .



ChatGPT

The image that best fits the text description is **image 1**. In this image, **three sheep are prominent in the foreground, and many more sheep can be seen in the distance**. This matches the description provided.

Figure 12: A case from the static pictures set. The yellow boxes indicate the correct image and our model's result, while the red boxes represent the model's output. We will see our model's result is correct but the GPT-4V's output is wrong.