Cream 🍦: Visually-Situated Natural Language Understanding with Contrastive Reading Model and Frozen Large Language Models

Geewook Kim[1*] **Hodong Lee**[1*] **Daehee Kim**[1*] **Haeji Jung**[3*†] **Sanghee Park**[1*]
Yoonsik Kim[1*] **Sangdoo Yun**[2‡] **Taeho Kil**[1‡] **Bado Lee**[1‡] **Seunghyun Park**[1‡]
[1]NAVER Cloud Hyperscale AI [2]NAVER AI Lab [3]Korea University

Abstract

Advances in Large Language Models (LLMs) have inspired a surge of research exploring their expansion into the visual domain. While recent models exhibit promise in generating abstract captions for images and conducting natural conversations, their performance on text-rich images leaves room for improvement. In this paper, we propose the Contrastive Reading Model (Cream), a novel neural architecture designed to enhance the language-image understanding capability of LLMs by capturing intricate details typically overlooked by existing methods. Cream integrates vision and auxiliary encoders, complemented by a contrastive feature alignment technique, resulting in a more effective understanding of textual information within document images. Our approach, thus, seeks to bridge the gap between vision and language understanding, paving the way for more sophisticated Document Intelligence Assistants. Rigorous evaluations across diverse tasks, such as visual question answering on document images, demonstrate the efficacy of Cream as a state-of-the-art model in the field of visual document understanding. We provide our codebase and newly-generated datasets at https://github.com/naver-ai/cream.

Figure 1: **Comparsion results on a text-rich image.** While prior methods, such as BLIP-2 (Li et al., 2023) and LLAVA (Liu et al., 2023a), miss image details, our proposed Cream effectively captures these features for accurate LLM responses. On the other hand, simply incorporating OCR input (e.g., OCR + ChatGPT) has limitations due to its inability to fully comprehend visual context.

1 Introduction

Recent advances in large language models (LLMs) (Brown et al., 2020; OpenAI, 2023; Zhang et al., 2022; Touvron et al., 2023) have facilitated the development of numerous real-world applications, providing users with valuable and meaningful services. Researchers are increasingly focusing on extending these unimodal LLMs to multimodal LLMs, particularly large visual language models (LVLMs), leveraging vision encoders designed to tackle information-rich visual tasks (Radford et al., 2021a; Tsimpoukelli et al., 2021; Wang et al., 2022a; Alayrac et al., 2022; Wang et al., 2022b; Driess et al., 2023; Zhu et al., 2023).

To evaluate LVLMs, various downstream tasks, such as image captioning, visual dialogue, grounding, reasoning, and question generation, have been employed. Despite demonstrating impressive results on these tasks, a recent study (Liu et al.,

* Core contributions. Correspondence to Geewook Kim: gwkim.rsrch@gmail.com.
† Work done during internship at NAVER Cloud AI.
‡ Advisory contributions. A description of each author's contribution is available at the end of this paper.

Preprint. Work in progress.

2023b) indicated that LVLMs exhibit limitations when dealing with text-rich visual tasks, leading to reduced applicability in real-world applications such as document visual question answering (DocVQA). Visual document understanding (VDU) tasks require the comprehensive analysis of multiple information types, including text, objects (graphs and charts), and layout. However, existing LVLMs struggle to deliver satisfactory solutions in these specific contexts due to their ability to extract limited fine-grained features from images, as shown in Figure 1.

In this paper, we introduce **Cream**, ***C**ontrastive **rea**ding **m**odel*, specifically designed to effectively overcome these limitations. Cream features a streamlined and practical architecture, seamlessly integrating a general vision encoder with auxiliary encoders and innovative training techniques. In addition to a primary vision encoder for overall visual feature extraction from document images, Cream employs auxiliary encoders—such as OCR and object detectors—for text and object-specific feature extraction. Cream utilizes auxiliary encoders as well as a vision encoder to extract fine-grained features without missing image details while understanding the visual context. When combined with LLMs, Cream overcomes the limitations of LVLMs and achieves robust performance in text-rich visual tasks. To further enhance the model, we propose a contrastive feature alignment method to mitigate biases between the vision and auxiliary features extracted from each encoder during training.

We conduct extensive experiments across various VQA tasks. We perform experiments on two models: our standalone Cream model and a model that combines our Cream model with frozen LLMs. The experimental results demonstrate that standalone Cream achieves results comparable to the state-of-the-art in tasks that necessitate the extraction of specific text information from document images. Furthermore, we observe that when combined with LLMs, Cream demonstrates robust performance in VDU tasks, which are challenging for existing LVLMs. Lastly, we will open-source Cream's codebase and the newly built VQA datasets, TydiVQA and Wikipedia Key-Value VQA (WKVVQA), to foster further research and innovation in the field of visual document understanding.

Our contributions are summarized as follows:

- We present a novel model architecture and associated training techniques tailored for visual document understanding tasks which serves as the eye of LLMs for performing text-rich tasks, as it can provide both visual context and image details to LLMs.
- Through rigorous experimentation, we demonstrate Cream's superior performance on several downstream tasks requiring the extraction of text information from document images.
- We provide an accessible approach for integrating the proposed model with LLMs, highlighting improved performance across specific downstream tasks.
- By sharing the codebase and several newly-generated datasets, TydiVQA and WKVVQA, we contribute valuable resources to facilitate ongoing research and development in visual document understanding tasks.

2 Related Work

2.1 Visually-Situated Natual Language Understanding

Visually-situated Natural Language Understanding (NLU) combines computer vision and natural language processing to enable a more precise analysis of visual data through language. Early researches (Xu et al., 2020; Hong et al., 2022) mainly focused on performing OCR on visual document data and utilizing the extracted text for analysis. Subsequent studies (Kim et al., 2022; Davis et al., 2023; Lee et al., 2022; Liu et al., 2022) explored methods that directly process document images without relying on external OCR models. Donut (Kim et al., 2022) proposed a model that performs text reading directly from document images as a pre-training task, enabling document understanding without the need for an external OCR model. Pix2Struct (Lee et al., 2022) introduced the concept of screenshot parsing objectives, while MATCHA (Liu et al., 2022) incorporated chart derendering and math reasoning into the model training. For improved performance, approaches that leverage both image and text extracted by OCR have also been explored (Kil et al., 2022; Tang et al., 2022; Appalaraju et al., 2021; Xu et al., 2022; Huang et al., 2022). LayoutLMv3 (Huang et al., 2022) introduced Word-Patch Alignment technique to classify the alignment between texts and their corresponding image patches. UDOP (Tang et al., 2022) employed a unified encoder to represent features from both the image and texts, transforming

information from both modalities into vision-text embeddings by summing its image patch and the text features.

Cream, like other methods, performs document understanding from various modalities. However, Cream can extract aligned multi-modal features at a fine-grained level from each modality encoder enabled by contrastive learning (CL) without the necessity for an additional fusion encoder. Particularly in the field of VDU, where images are rich with various texts, it is important to extract fine-grained text and visual information as well as semantic information.

2.2 Applying LLMs to Visually-Situated NLU

By scaling up the training data and model parameters, LLMs have achieved significant success (Rae et al., 2021; Brown et al., 2020; Chowdhery et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023). Furthermore, InstructGPT (Ouyang et al., 2022) and ChatGPT (Shahriar and Hayawi, 2023) demonstrated that aligning large-scale pretrained language models with human intent allows them to provide contextually relevant answers to user queries across various tasks. Similarly, Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023), as open-source models, have shown similar performance through instruct tuning from LLaMA (Touvron et al., 2023).

Building upon this success, there have been various attempts to incorporate visual information into LLMs to address vision-language tasks (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Li et al., 2023; Zhu et al., 2023; Dai et al., 2023; Liu et al., 2023a; Ye et al., 2023). They involved extracting features from visual information through a vision encoder and utilizing LLMs to tackle various tasks as a reasoning module. BLIP-2 (Li et al., 2023) advanced this approach by training only a Q-former module that bridges the gap between the vision encoder and LLMs while keeping the vision encoder and LLMs frozen. MiniGPT-4 (Zhu et al., 2023) and InstructBLIP (Dai et al., 2023) froze the parameters of the Instruct-tuned LLMs when training vision-language models. LLAVA (Liu et al., 2023a) and mPLUG-Owl (Ye et al., 2023) also trained the parameters of LLMs along with vision-language instruction following data.

While these methods have demonstrated successful vision-language learning with LLMs, they have not yet shown strong performance in visually-situated NLU tasks, including question answering on visual documents. These methods have limited ability to extract fine-grained features in images, which has hindered their ability to match domain-specific methods in traditional text tasks (Liu et al., 2023b). Cream exhibits strong performance in analyzing text-rich images, where previous methods have encountered limitations, by leveraging its powerful visual understanding capability when combined with the LLMs.

3 Method

Our primary interest lies in accurately answering natural language questions given an image, based on specific evidence within the image. For instance, when answering a question that involves extracting specific information from a document image, the output can be meaningless unless the text in the image is correctly recognized and addressed even if the answer is linguistically plausible (see Figure 1). In this work, we investigate the system that effectively utilizes information embedded in images to accurately respond to given natural language queries. A crucial aspect of this process involves the model's ability to identify specific *feature evidences* in the image, such as texts, objects, and other relevant features. In this section, we describe our model (§ 3.1), explain its integration with frozen LLMs (§ 3.2), and provide details on model training and data preparation (§ 3.3).

3.1 Contrastive Reading Model

We propose **Cream** (**C**ontrastive **rea**ding **m**odel), a robust image understanding module designed for visually-situated language understanding applications. We consider two application scenarios. In the first scenario, Cream is deployed as a standalone model, where its decoder directly generates the required information in text form. In the second scenario, Cream is combined with an LLM, with its decoder serving as a soft visual prompter. Here, the decoder's output hidden state is utilized as a visual prompt for the LLM. Figure 2 depicts the overall pipeline, with the upper part illustrating the entire pipeline architecture and the bottom part representing Cream's feature alignment scheme. The pros, cons, and differences between these scenarios are examined in our experiments and analyses (§ 4).

3.1.1 Architecture

The Cream architecture comprises two encoders and one decoder. Given an input image, the vi-

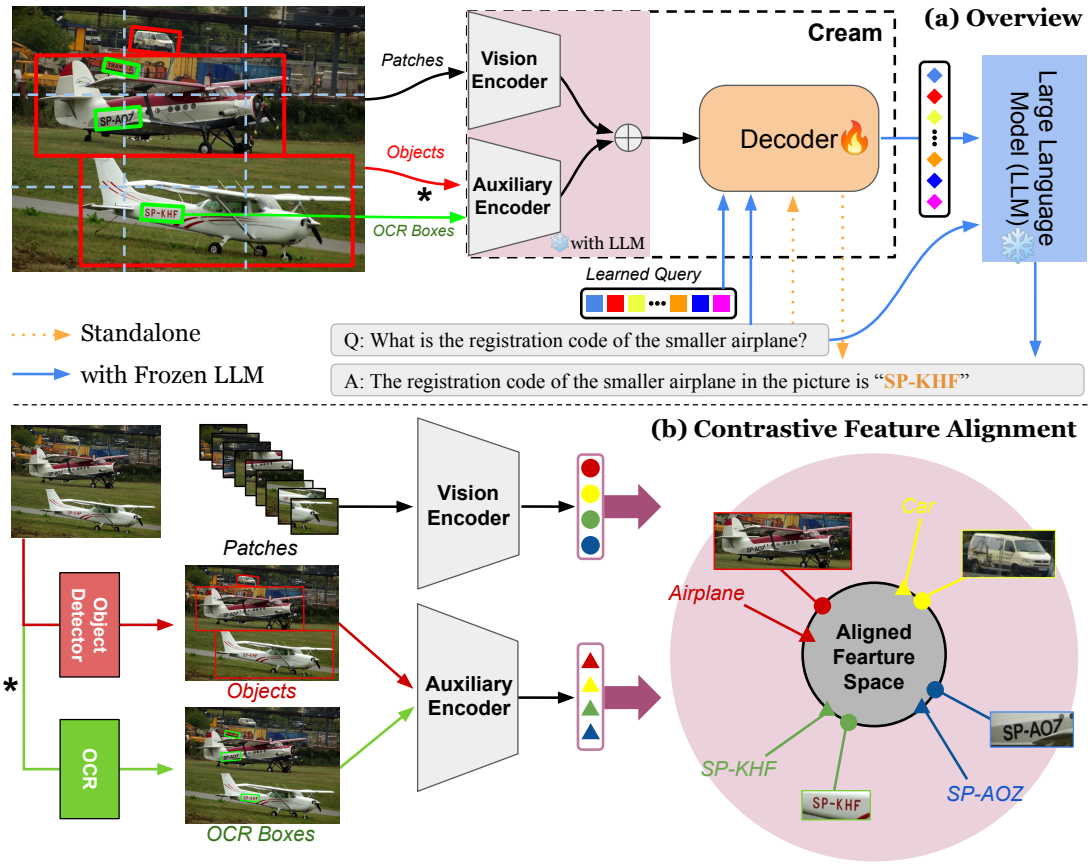


Figure 2: **Overview of Cream's framework.** (a) Image patches initially feed into the vision encoder. *The information extracted from off-the-shelf OCR and object detectors is delivered to auxiliary encoders if available. (b) The encoded vector representations are well-aligned using a contrastive learning scheme. For a given natural language query, the decoder either generates the answer by referring to the encoded vectors (yellow dotted line), or serves as a soft visual prompter for an LLM (blue solid line). Note that encoders are frozen when training with the LLM.

sion encoder processes the image into a set of embeddings. Additionally, if available, specific feature evidences (e.g., texts or objects within the image) are extracted by corresponding detectors (e.g., OCR or object detector) and the extracted items are passed to the auxiliary text encoder, where they are embedded into a common feature space. The output representations from the two encoders are trained to be aligned in the feature space through a proposed CL scheme (See the bottom part of Figure 2). The features are concatenated and fed to the cross-attention layers in the decoder. Given the embeddings alongside a natural language query, the decoder generates the desired information through attention mechanisms. The following sections provide details on each module.

Vision Encoder The vision converts the input image $\mathbf{x}\in\mathbb{R}^{H\times W\times C}$ into a set of embeddings $\{\mathbf{z}_i|\mathbf{z}_i\in\mathbb{R}^d, 1\leq i\leq n\}$, where n is the feature map size or the number of image patches, and d is the dimension of the resulting output vectors. CNN-based models (He et al., 2016) or Transformer-based models (Dosovitskiy et al., 2021; Liu et al., 2021) can be used as the encoder network. In this study, for simplicity, we employ the Vision Transformer (Dosovitskiy et al., 2021) with a 2D absolute position encoding (Xu et al., 2020) and a variable-resolution mechanism (Lee et al., 2022). The variable-resolution mechanism is an input image pre-processing strategy that converts the image into a constant number of patches without distorting the original image aspect ratio.

Auxiliary Encoder The auxiliary encoder encodes the information of extracted feature evidence, such as OCR boxes and general object boxes, into a set of embeddings $\{\hat{\mathbf{z}}_i|\hat{\mathbf{z}}_i\in\mathbb{R}^d, 1\leq i\leq\hat{n}\}$. As shown in Figure 3, the extracted feature evidence is converted into a sequence of token embeddings. For this conversion, the recognized text is used for OCR boxes, and the recognized semantic object label is

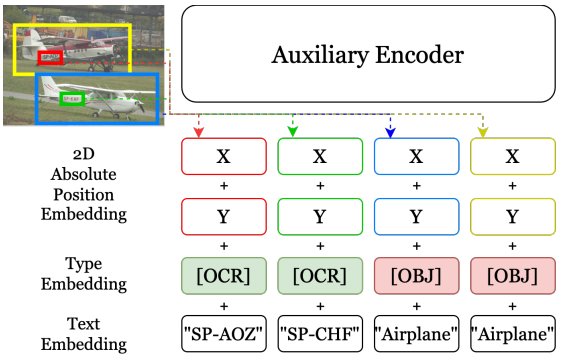


Figure 3: **Token embeddings of the auxiliary encoder.** The center point of each bounding box is utilized for positional embedding. The text labels, such as "SP-AOZ", "Airplane", are tokenized into subwords. Detailed process is omitted in the figure for simplicity.

used for general object boxes. Subsequently, a type embedding is added to distinguish OCR and general object boxes. In addition, a 2D absolute position encoding is applied to encode the location information. For the backbone, we adopt the BART (Lewis et al., 2020) encoder architecture in this work.

Decoder We employ BART (Lewis et al., 2020) as the decoder architecture, which processes the set of embeddings $\{\mathbf{z}_1, ..., \mathbf{z}_n, \hat{\mathbf{z}}_1, ..., \hat{\mathbf{z}}_{\hat{n}}\}$ and generates a sequence of vectors $\mathbf{h} \in \mathbb{R}^{m \times d}$, where m is the sequence length of the generated vectors. Referred to as the last hidden states, these vectors can be utilized in two different scenarios. (i) In the standalone scenario, a linear language modeling head, represented by a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times v}$, is applied to the hidden states, yielding the token sequence as follows: $\hat{\mathbf{Y}} = \mathbf{hW}$, where $\hat{\mathbf{Y}} \in \mathbb{R}^{m \times v}$ is the predicted sequence of tokens and v is the size of the token vocabulary. (ii) When Cream is integrated with an LLM, the last hidden states of the decoder are first linearly transformed using a weight matrix $\mathbf{U} \in \mathbb{R}^{d \times d'}$, where d' denotes the dimension of the LLM's input embeddings: $\mathbf{h}' = \mathbf{hU}$. The transformed hidden states $\mathbf{h}' \in \mathbb{R}^{m \times d'}$ are then used as input to the LLM, serving as a soft visual prompt that combines Cream's visual understanding capabilities with the LLM's language processing abilities. For both scenarios, Cream adopts the language modeling loss, where the decoder generates a sequence of token embeddings conditioned on the image. The details of Cream's training objective will be explained in Section 3.3.3 and 3.3.4.

3.1.2 Contrastive Feature Alignment

To assimilate information such as OCR and object data alongside image information within the decoder, we encode these inputs using an auxiliary encoder. However, in practice, it is uncertain whether features originating from different encoders will be well-aligned in common space. Our investigation exposes misalignment in the information encoded by the two encoders (will be presented and analyzed in Section 4), and to rectify this, we incorporate a simplistic form of CL objective into the model training.

Given a set of OCR or general object boxes obtained from an OCR or object detector, the embedding of the feature evidence and the embedding of the corresponding patch where the evidence is physically located in the image can be interpreted as containing semantically similar information. Based on this assumption, we apply CL, defining a positive pair as relationship between the embedding of the feature evidence and the corresponding image patch, while all other relationships are considered negative pairs. Specifically, consider a scenario where an image contains a 'book' with the title 'Apple'. In this case, the patch in the image where the book is located forms a positive pair with the bounding box information labeled 'book'. Additionally, the image containing the word 'Apple' forms a positive pair with the 'Apple' text label and its bounding box information. Any other relationships are considered as negative pairs. This approach allows us to obtain more pairwise relationships in one sample compared to image-level CL approach, e.g., CLIP (Radford et al., 2021b).

For CL, we use a 2-layer Multi-Layer Perceptron (MLP) $\mathbf{f}_\theta : \mathbb{R}^d \mapsto \mathbb{R}^{d^*}$, where d^* is a hyperparameter for a dimension of a common space. Most settings are similar to those of Khosla et al. (2020). The CL objective can be expressed as follows:

$$-\sum_{i=1}^{l} \log \frac{\exp(s(\mathbf{v}_i, \hat{\mathbf{v}}_i)/\tau)}{\sum_{j=1\ j\neq i}^{l} \exp(s(\mathbf{v}_i, \hat{\mathbf{v}}_j)/\tau)}, \quad (1)$$

where the sets $\{\mathbf{v}_i | 1 \leq i \leq l\}$ and $\{\hat{\mathbf{v}}_i | 1 \leq i \leq l\}$ represent features stacked in the order they are sampled as positive pairs from $\mathbf{z}$ and $\hat{\mathbf{z}}$, respectively. Here, l denotes the number of feature evidences and τ is the temperature parameter modulating the softmax sharpness. The function $s(\mathbf{x}, \mathbf{y}) = \cos(\mathbf{f}_\theta(\mathbf{x}), \mathbf{f}_\theta(\mathbf{y}))$ computes the cosine similarity between its vector inputs, using an MLP parameterized by θ. This CL objective encourages the

alignment of the embeddings from both encoders in the feature space. We validate the effectiveness of this objective in our experiments and analyses (§ 4).

3.2 Integration of Cream and LLMs

Large language models (LLMs) have demonstrated state-of-the-art performance on a wide range of natural language processing tasks, such as text classification, question answering, and machine translation. However, LLMs often face limitations in understanding and responding to context-specific language. To address this issue, we integrate Cream with LLMs following a similar approach proposed in the previous work, BLIP-2 (Li et al., 2023). The features extracted by the Cream decoder serve as a visual input prompt for the LLM, which subsequently generates a text response for a given input image and question.

To enhance this integration, we adopt the learned query mechanism introduced in BLIP-2 (Li et al., 2023). This mechanism utilizes a set of trainable embeddings as input for the Cream decoder to extract fixed size hidden states that are subsequently fed to the LLM. If available and appropriate, the natural language query is simultaneously input to the Cream decoder, enabling the decoder's last hidden states for the learned queries to encode more valuable information for answering the question. In multi-turn QA scenarios, we refrain from inputting the question to the Cream decoder, as its hidden states are not solely used for answering a single question. This approach allows the Cream and LLM combination to play more diverse roles in real-world applications.

Alternative methods have explored inputting visual embeddings and OCR tokens directly to the LLM (Li et al., 2023; Dai et al., 2023). However, these approaches are unsuitable for real-world applications due to their high computational cost. For example, answering questions in DocVQA (Tito et al., 2021) requires an average of nearly 400 and a maximum of 4,000 OCR tokens (See Section 4.3 and Figure 7). By incorporating the learned queries mechanism and considering the application context, the integration of Cream and LLMs becomes more flexible, enabling the LLM to focus on specific aspects of visual input while generating accurate and contextually appropriate responses. This approach not only enhances the LLM's understanding of visual context but also reduces the computational cost, as the learned queries can efficiently extract relevant information from input images.

3.3 Model Training

3.3.1 Training Tasks

Text Read Building upon the text reading pretraining task for visual document understanding models proposed by Kim et al. (2022), we utilize a large number of visual corpora (image and text information pairs) to train Cream for modeling texts within images. Due to the introduction of an auxiliary encoder, we adapt the task by replacing some OCR tokens with mask tokens when inputting OCR results to the auxiliary. This task shares similarities with the masked language modeling in UL2 (Tay et al., 2022). However, our task simultaneously employs the image modality and aims to read the entire text in the image, rather than just the masked tokens.

Masked Text Prediction To enhance Cream's comprehension of the overall context within the image, we introduce a masked text prediction (MTP) task for predicting hidden texts in the image. We randomly mask some OCR boxes in the image, and the objective is to predict the letters in the completely obscured areas of the image. This task can be interpreted as extending the masked language modeling task (Tay et al., 2022) to the visual domain.

Captioning The image captioning task requires the model to generate a natural language description of the image that captures the overall scene and object details. This task enhances Cream's ability to understand the overall situation in the image and recognize objects. The generated captions provide a comprehensive representation of the image content, which is crucial for visually-situated language understanding tasks.

Question Answering Question answering (QA) involves training Cream to process an image and a natural language question, then generate an appropriate answer. By incorporating this task, the model learns to focus on specific image regions and text information to provide accurate answers. This capability further improves Cream's understanding of the relationships between visual and textual information in the image.

Question Generation Question generation (QG) requires the model to generate a question sentence

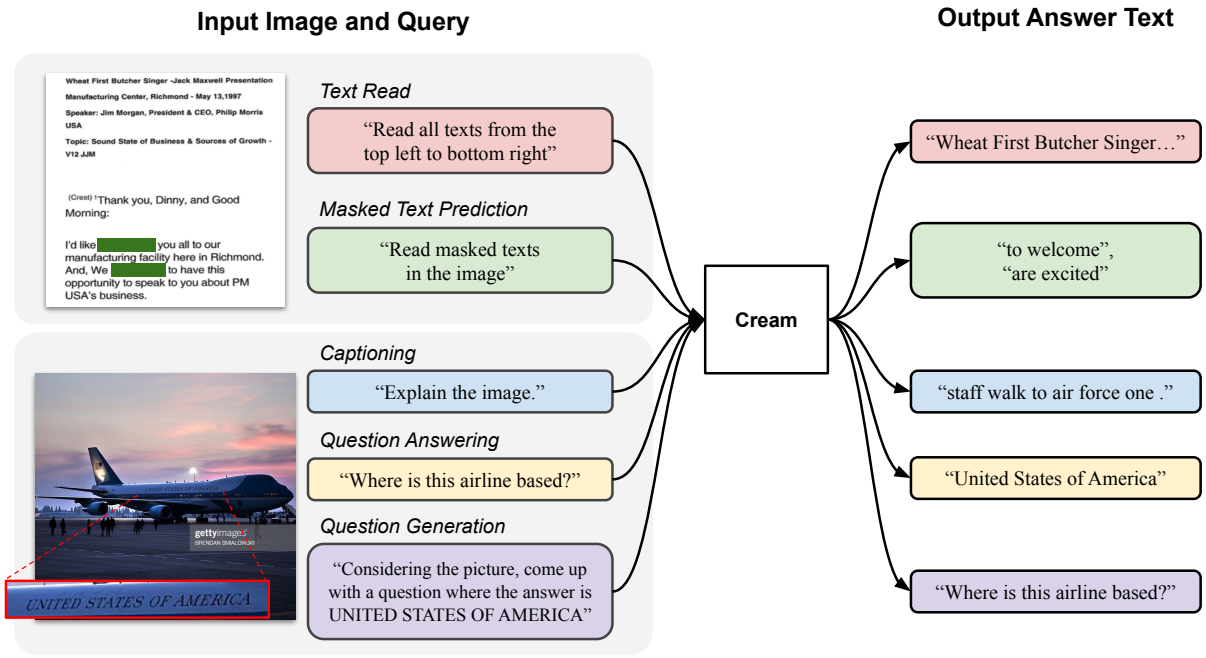


Figure 4: **Unified Multitask Model Training Framework.**

corresponding to a given answer text in the context of an image. This task promotes the model's ability to reason about the image content and answer text, facilitating the model's QA ability. QG training can be done by simply swapping the question and the answer.

3.3.2 Unified Multitask Training

The tasks discussed so far—text reading, MTP, image captioning, question answering, and question generation—are interrelated and can be addressed using similar approaches. Ultimately, they involve extracting a text sequence based on a given task command query when provided with an input image and feature evidence within the image. For instance, the query for text reading could be "please read all texts in the image from the top left to bottom right", while for masked LM, it could be "guess all hidden texts in the masked area". For image captioning, queries like "please describe the image" or "explain the image" can be used. Figure 4 illustrates our unified Cream training framework, where natural language prompt (query) and image are input and desired answer texts are generated for all tasks. Full prompt examples are available in Appendix A. Our prompt differs from other document understanding methods such as Donut (Kim et al., 2022) and UDOP (Tang et al., 2022), which employ single task-specified prompt. We believe that our Cream trained with natural language-based prompts can be more seamlessly integrated into LLMs.

We train the model using a combination of supervised fine-tuning VQA datasets and pre-training datasets for text reading, MTP, and image captioning. Some QA data demand more reasoning than merely reading the text in the image or describing the situation. Consequently, we restrict the use of such fine-tuning QA benchmarks in the early stage of training and increase the proportion of the QA data after the middle stages.

3.3.3 Training Objective

The training of the Cream consists of two main objectives: language modeling and contrastive loss. These objectives aim to align the embeddings generated by the vision and auxiliary encoders and improve the model's overall performance in visually-situated language understanding tasks.

The language modeling objective focuses on generating a sequence of token embeddings conditioned on the image. Cream employs a simple cross-entropy loss to measure the difference between the predicted token sequence and the ground truth. In line with the original Transformer (Vaswani et al., 2017), we utilize a teacher-forcing scheme (Williams and Zipser, 1989) during the training process. This strategy involves using the ground truth as input instead of the model's output from a previous time step, ensuring that the

model learns from accurate contextual information.

The CL objective encourages the alignment of the embeddings produced by the vision and auxiliary encoders in the feature space. This alignment is crucial for effectively assimilating OCR and object information alongside image information within the decoder. To achieve this, we define positive and negative pairs based on the location information of feature evidence in the image. Positive pairs consist of relationships where the feature evidence and the corresponding image patch share semantically similar information, while negative pairs comprise all other relationships. The CL objective was previously defined in Equation 1.

To combine both objectives during Cream's training, we use a weighted sum of the language modeling loss ($\mathcal{L}_{\text{LM}}$) and the CL loss ($\mathcal{L}_{\text{CL}}$), as follows:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{CL}}, \tag{2}$$

where λ is a hyperparameter that controls the relative importance of the two objectives. By incorporating this combined training objective, we ensure that the model effectively aligns the information encoded by both encoders and achieves high performance in visually-situated language understanding tasks, as validated in our experiments and analyses (§ 4).

3.3.4 Further Learning to Prompt LLMs

We adapt the integration method from BLIP-2 (Li et al., 2023) in order to align it with the objectives and architecture of Cream. The Q-former architecture suggested in BLIP-2 is not implemented in this work. Instead, our Cream decoder functions as a soft visual prompter. The transformed hidden states serve as a soft visual prompt, conditioning the LLM on the visual representation extracted by Cream's decoder. Throughout the integration process, we freeze both the LLM and Cream's encoders, ensuring that only the Cream decoder is updated via gradient descent-based training. In accordance with the BLIP-2 methodology, a new learnable parameter, referred to as the *vision query*, is introduced to extract a fixed number of vectors that act as a soft visual prompt. Given a desired number of vectors k, we create k new token embeddings, which serve as the vision queries. These queries are input into Cream's decoder, and the resulting output vectors are used as the visual prompt.

Upon integration with the LLM, Cream's decoder no longer functions as an autoregressive decoder. To address this, we modify its attention mechanism to allow bi-directional attention flow. This adjustment enhances the decoder's capability to effectively combine Cream's understanding of visual information with the LLM's language processing abilities, resulting in a more efficient model for visually-situated language understanding tasks.

4 Experiments and Analyses

In this section, we provide in-depth analyses of Cream. We first explain the details of model training (§ 4.1), present the benchmark results including ablations (§ 4.2), and visualization (§ 4.3).

4.1 Experimental Setup

4.1.1 Training Datasets

Our model is trained on a diverse range of datasets, which target text reading, MTP, image captioning, question answering (QA), and question generation (QG) tasks. We provide an overview of the relevant datasets and their statistics in Table 1, and present representative examples from each of these datasets in Figure 5.

Text Read and MTP In order to address the text read and MTP tasks, we utilize the IIT-CDIP (Lewis et al., 2006) (11M) and WEBVICOB (Kim et al., 2023) (30M) datasets. The IIT-CDIP dataset is a publicly available resource consisting of scanned document images and has been widely employed in various visual document understanding studies. WEBVICOB, on the other hand, is a recent dataset generator that constructs visual corpora from Wikipedia dumps. We produced a 30M-element visual corpus[1] using WEBVICOB. Both datasets are well-suited for training text read and MTP tasks, as they primarily consist of well-structured text in contrast to other data sources such as photographs or uniquely structured layouts.

Captioning We use CC3M (Sharma et al., 2018) (3M) dataset for the image captioning task, which constitutes a vast collection of web images paired with textual descriptions. CC3M has been employed extensively in the pre-training of numerous existing visual language models.

Question Answering and Question Generation In order to enhance Cream's visual document understanding capabilities, we utilize several widely

[1] https://github.com/clovaai/webvicob

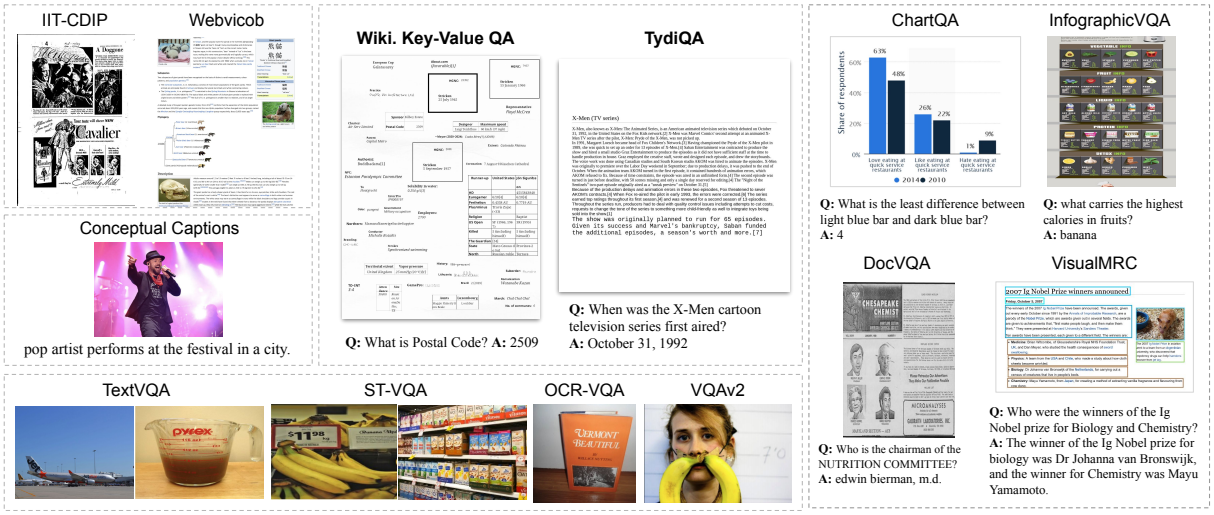


Figure 5: **Training Datasets.** This figure illustrates samples from the IIT-CDIP, WEBVICOB, Conceptual Captions (CC3M), document visual QA benchmarks, as well as general VQA and scene text VQA datasets. Furthermore, it displays samples from our open-source datasets, Wikipedia Key-Value VQA (WKVVQA) and TydiVQA.

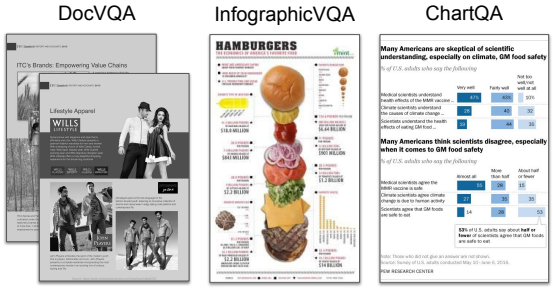


Figure 6: **Evaluation Benchmarks.** We evaluate models on ChartQA, InfographicVQA, and DocVQA to gauge their proficiency in answering queries using image details.

recognized public QA benchmark datasets, including ChartQA (Masry et al., 2022), InfographicVQA (Mathew et al., 2022), DocVQA (Tito et al., 2021), VisualMRC (Tanaka et al., 2021), DVQA (Kafle et al., 2018), and OCRVQA (Mishra et al., 2019). Additionally, we take advantage of STVQA (Biten et al., 2019), TextVQA (Singh et al., 2019), VizWizVQA (Gurari et al., 2018), and VQAv2 (Goyal et al., 2017) datasets in order to further improve Cream's scene text comprehension and general image understanding abilities.

We also introduce two synthetically generated VQA datasets, TydiVQA and Wikipedia Key-Value VQA (WKVVQA), depicted in Figure 5. TydiVQA is developed by extending TydiQA (Clark et al., 2020) into a multimodal context through associating each QA sample with its corresponding WEBVICOB resource (Kim et al., 2023). WKVVQA consists of synthetic document images containing various key-value pairs extracted from the Wikipedia dump. Both TydiVQA and WKVVQA will be publicly accessible via our GitHub repository.[2]

4.1.2 Test Datasets

Our models are evaluated using the test sets of ChartQA (Masry et al., 2022), InfographicVQA (Mathew et al., 2022), and DocVQA (Tito et al., 2021), in order to gauge their effectiveness in accurately answering natural language queries reliant on a profound understanding and recognition of various image elements, such as text, objects, and relationships. A sample of the test datasets used is depicted in Figure 6.

It is important to note that both InfographicVQA and ChartQA present significant challenges. ChartQA demands a degree of reasoning ability, while InfographicVQA is characterized by vast image sizes, necessitating a thorough comprehension of the images' content. It has been reported that GPT-4 (OpenAI, 2023) employed a chain-of-thought approach specifically designed for the ChartQA benchmark.

4.1.3 Configurations

Model Details Our primary vision encoder is initialized with the weights of OpenCLIP (Radford et al., 2021b; Ilharco et al., 2021) trained on LAION (Schuhmann et al., 2022) 2B data. In this paper, we create and utilize two sizes of Cream

[2] https://github.com/naver-ai/cream

Dataset	Task	Size (Images)
IIT-CDIP	Text Read / MTP	11M
WEBVICOB	Text Read / MTP	30M
CC3M	Image Captioning	3M
ChartQA	QA / QG	18K (train)
InfographicVQA	QA / QG	4K (train)
DocVQA	QA / QG	11K (train+val)
VisualMRC	QA / QG	9K (train+val)
DVQA	QA / QG	200K (train)
OCRVQA	QA / QG	146K (train)
STVQA	QA / QG	17K (train)
TextVQA	QA / QG	25K (train+val)
VizWizVQA	QA / QG	15K (train)
VQAv2	QA / QG	83K (train)
TydiQA	QA / QG	4K
Wiki. key-value QA	QA / QG	800K

Table 1: **Statistics of the training datasets.**

models. The main model has 18 layers for the primary vision encoder, 12 layers for the auxiliary encoder, and 12 layers for the decoder, with the patch size of 14×14. In contrast, the smaller model for the ablation study has 9 layers for the vision encoder, 6 layers for the auxiliary encoder, and 6 layers for the decoder, with the patch size of 32×32. For the main model, the primary vision encoder is initialized with ViT-L, while for the small model, it is initialized with ViT-B (Dosovitskiy et al., 2021). The auxiliary encoder and the decoder are initialized with the weights of MBart (Liu et al., 2020). Excluding token embeddings, the total size of the standalone Cream is 0.6B. For the LLM integration experiments, we use the 7B size model from Vicuna (Chiang et al., 2023).

Environment and Training Hyperparameters Table 2 provides the proportion of multiple tasks for each training phase. In the first phase, our main model is trained on 128 A100 GPUs with a batch size of 384, a fixed learning rate of 1e-4, and 220K steps. Once the loss has converged to a sufficient extent, we seamlessly transition to the next phase by changing the batch proportions and training hyperparameters. In the second phase, we use 32 A100 GPUs with a batch size of 96, a cosine scheduled initial learning rate of 5e-5, and 220K steps and take a higher proportion for QA. After sufficient convergence in this phase, we observe no significant change in QA performance when training with only QA data, so we conclude the Cream training with this phase. For integrating Cream with the LLM, we use the Cream model that has already completed the training process. As

Phase	Task Proportion
Cream-phase1	Text Read (22%), MTP (46%), Captioning (22%), QA (5%), QG (5%)
Cream-phase2	Text Read (7%), MTP (14%), Captioning (26%), QA (48%), QG (5%)
LLM Integration	QA (100%)

Table 2: **Task proportions according to training phases.**

described in Section 3.3.4, the integration process involves making only the Cream decoder learnable. During this process, we train the LLM integration using only QA datasets, as the QA performance quickly converges. For the integration, our model with LLM is trained on 128 A100 GPUs with a batch size of 1024 and 46K steps, using a cosine scheduled initial learning rate of 1e-4. For the rest training hyperparameters, we use $\lambda = 0.5$, $\tau = 0.07$, $d = 1024$, $d^* = 128$, $k = 192$, $d' = 4096$, and $l = 90$ in our experiments.

Off-the-Shelf Detectors During the experiments, in both training and test phases, we adopt the CLOVA OCR API[3] as a commercial OCR solution for the OCR module, and we utilize the OWL-ViT[4] model from Minderer et al. (2022) as the general object detector. For the semantic class label texts, we use the 80 class labels provided by the MS-COCO dataset (Lin et al., 2014).

4.2 Results

Table 3 displays the performance of diverse Frozen LLM integration models on the DocVQA, ChartQA, and InfoVQA benchmarks. Our proposed Cream integration exhibits significant improvement over other Frozen LLM integrations in benchmarks demanding advanced visual understanding capabilities.

A key feature of the Cream integration is the utilization of a fixed-size soft visual prompt, regardless of the number of texts within the image, which is set to 192 in our main experiments. Unlike approaches in which all OCR tokens are fed into the LLM, our method does not depend on excessively large token lengths (denoted as |OCR|) for processing document information, thereby enhancing efficiency. On average, the DocVQA (Tito et al., 2021) dataset requires 432 OCR tokens per

[3] https://clova.ai/ocr/en

[4] https://huggingface.co/google/owlvit-large-patch14

Model	Prompt Length	Use Auxiliary	DocVQA	ChartQA	InfoVQA
OCR-Vicuna7B (Chiang et al., 2023)	|OCR|	✓	29.2	6.2	13.6
OCR-Vicuna13B (Chiang et al., 2023)	|OCR|	✓	31.4	3.7	23.7
OCR-GPT3.5	|OCR|	✓	62.4	15.9	26.6
OCR-GPT4 (OpenAI, 2023)	|OCR|	✓	75.9	34.3	25.0
BLIP2-OPT-6.7B (Li et al., 2023)	32		3.7	4.6	11.0
BLIP2xOCR-OPT-6.7B (Li et al., 2023)	32+|OCR|	✓	6.2	17.5	30.4
BLIP2-FlanT5xxL-11B (Li et al., 2023)	32		8.6	4.4	11.4
BLIP2xOCR-FlanT5xxL-11B (Li et al., 2023)	32+|OCR|	✓	63.8	18.3	36.6
LLaVA-Vicuna7B (Liu et al., 2023a)	256		5.5	0.5	2.4
LLaVA-Vicuna13B (Liu et al., 2023a)	256		5.9	1.4	3.1
Cream-Vicuna7B (**Proposed**)	192	✓	**80.0**	**61.6**	**42.4**

Table 3: **Experimental results** for various models on visually-situated language understanding tasks. Cream, when integrated with the frozen Vicuna, significantly outperforms other LLM integrations with an efficient prompt length.

Model	DocVQA	ChartQA	InfoVQA
BLIP2-OPT-6.7B	3.7	4.6	11.0
BLIP2-FlanT5xxL-11B	8.6	4.4	11.4
LLaVA-Vicuna7B	5.5	0.5	2.4
LLaVA-Vicuna13B	5.9	1.4	3.1
Cream-Vicuna7B *w/o Aux*	**45.8**	**50.0**	**22.8**

Table 4: **Experimental results** show that Cream integrated with the frozen LLM notably outperforms other models when the vision information is only employed as an input.

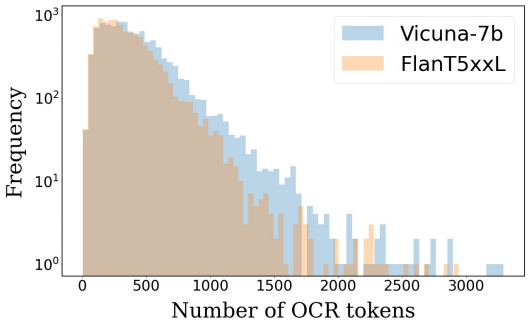


Figure 7: **Visualization of LLM Token Consumption induced by OCR.** The number of required OCR tokens (|OCR|) are shown. We use the DocVQA dataset (Tito et al., 2021) and tokenizers of Vicuna (Chiang et al., 2023) and FlanT5 (Chung et al., 2022).

sample, with a maximum of 3292 tokens in Vicuna model (Chiang et al., 2023). Further details can be found in Section 4.3.2.

Considering real-world applications, we also evaluate our approach in scenarios where off-the-shelf OCR and object detectors are not available. As illustrated in Table 4, our method yields outstanding performance improvements compared to preceding strategies. We confirm that Cream effectively extracts task-relevant information in text-rich tasks within the vision encoder, surpassing the existing methods.

Moreover, we assess the standalone performance of Cream, as depicted in Table 5. We include a selection of recent state-of-the-art models as points of comparison. Although Cream's standalone performance is marginally inferior to specialized standalone models, it demonstrates comparable results to contemporary state-of-the-art Visual Document Understanding models. Interestingly, we notice enhanced performance when combining the LLM with the standalone Cream model. Tables 3 and 5 verify that Frozen LLMs can achieve performance levels akin to state-of-the-art visual document understanding models, despite not directly observing the images or OCR results and relying on only fixed-size visual prompt.

Overall, the experimental results underscore the effectiveness of the Cream model and its integration with Vicuna7B for visually-situated text-rich image understanding tasks. Our proposed method successfully merges the strengths of visual understanding and language processing, culminating in a potent model that outperforms existing LLM integrations and exhibits competitive performance against cutting-edge standalone models across the evaluated benchmarks.

4.3 Analyses

4.3.1 Impact of Contrastive Learning and Auxiliary Encoding Scheme

We examine the influence of the auxiliary encoder and CL on the Cream model's performance. The results outlined in Table 6 lead to several observations:

Model	Aux	DocVQA	ChartQA	InfoVQA
BROS	✓	68.1	-	24.8
Donut		67.5	41.8	21.7
$\text{Pix2Struct}_{\text{Base}}$		72.1	56.0	38.2
$\text{Pix2Struct}_{\text{Large}}$		76.6	58.6	40.0
$\text{LayoutLMv3}_{\text{Base}}$	✓	78.8	-	-
$\text{LayoutLMv3}_{\text{Large}}$	✓	83.4	-	45.1
UDOP	✓	**84.7**	-	**47.4**
Cream	✓	81.3	**61.2**	39.8

Table 5: **Experimental results** for standalone models. The standalone Cream shows comparable results to the recent VDU models, including BROS (Hong et al., 2022), Donut (Kim et al., 2022), Pix2Struct (Lee et al., 2022), and UDOP (Tang et al., 2022).

Model	DocVQA	ChartQA	InfoVQA
$\text{Cream}_{\text{Small}}$	67.8	54.8	29.9
- *disable aux. at test*	38.9	41.4	13.1
Diff.	28.9	13.4	16.8
$\text{Cream}_{\text{Small}}$ w/o CL	65.3	52.7	31.5
- *disable aux. at test*	7.9	9.8	12.1
Diff.	57.4	42.9	19.4
Donut-like	49.8	47.6	16.8
Donut-like-Patch1700	60.5	52.2	20.9

Table 6: **Ablation study results** for $\text{Cream}_{\text{Small}}$. The table highlights the effect of CL that alleviates the performance gap when the auxiliary information is not employed.

- The performance difference between $\text{Cream}_{\text{Small}}$ and its vision-only counterpart (i.e., donut-like) underlines the significance of employing the auxiliary encoder, which integrates OCR and object detection results. This result supports the proposed approach's effectiveness for incorporating supplementary information to bolster visual understanding.

- Comparing models with and without CL highlights the effectiveness of the proposed CL method. In the absence of CL, the primary vision encoder is inadequately trained and utilized during the inference stage, causing reduced performance. The relatively smaller performance discrepancy in the CL setting exemplifies CL's capacity to thwart feature collapse or bias, establishing its importance in Cream's overall design.

- The gap between ($\text{Cream}_{\text{Small}}$)-(*disable aux. at test*) and ($\text{Cream}_{\text{Small}}$ w/o CL)-(*disable aux. at test*) indicates that the proposed CL mechanism contributes to balancing the two encoders, leading to more robust performance, even when auxiliary information is absent. This feature is crucial for real-world environment applications.

- The donut-like-patch1700 model, which adjusts the donut-like model to accommodate larger images, increases the patch number from 1024 to 1700 during the training phase. Although this modification necessitates substantial computational resources, it still results in lower scores than the base $\text{Cream}_{\text{Small}}$ model. This observation implies that bridging the performance gap between models with and without auxiliary encoding by merely scaling the vision encoder is challenging.

These analyses establish the effectiveness of leveraging the auxiliary encoder and CL within the Cream model. These components play a pivotal role in enhancing the model's performance on visually-situated language understanding tasks and contribute to its success in comparison to other state-of-the-art models.

Besides the quantitative analysis, we conduct a visualization analysis to qualitatively comprehend the importance of CL in achieving Cream's high performance. Figure 8 (a) presents the results of Principal Component Analysis (PCA) on the common space of the embeddings generated by the two encoders. The PCA results for the space with CL display improved alignment, particularly for the first principal component, which seems to represent the difference between the encoders. By excluding the first principal component and visualizing with the second and third components, we observe that each embedding exhibits enhanced alignment in the common space of Cream with CL, and semantically similar embeddings cluster more effectively.

Figure 8(b) presents the outcomes obtained from randomly selecting two embeddings in the shared embedding space, computing their cosine similarity, and constructing histograms. In accordance with the literature, a Gaussian distribution is expected for the histogram when dealing with random (unit) vectors in the embedding space (Spruill, 2007). The observed results imply that the embedding space, when subjected to contrastive learning, demonstrates a wider distribution of embeddings, suggesting a potential enhancement in the quality of the embedding space. The red line in the figures represents the Gaussian distribution with the

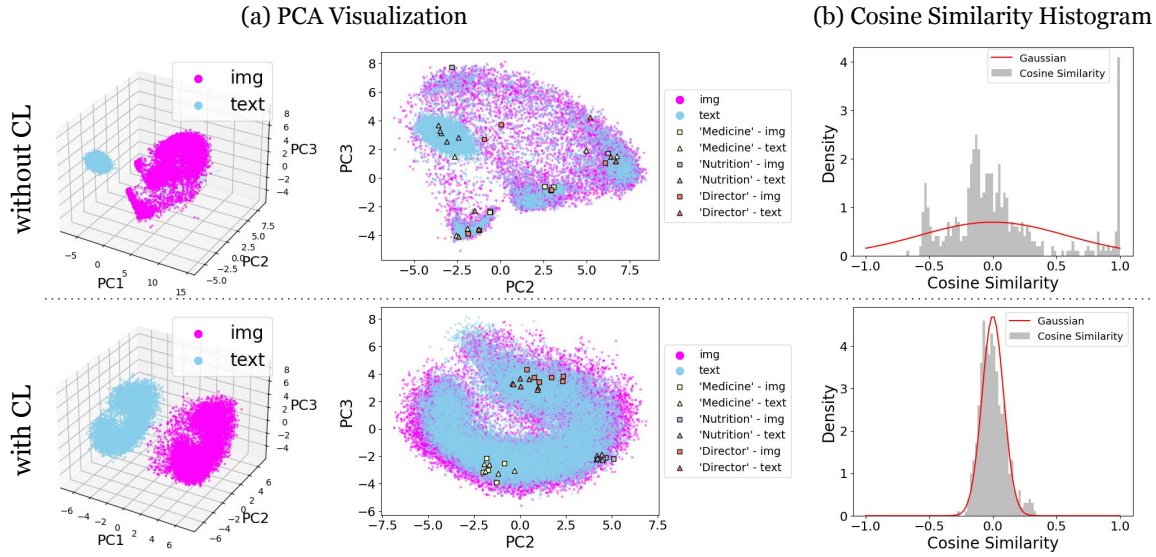


Figure 8: **Visualization of Contrastive Feature Alignment Effects in the Common Feature Space.**

minimal KL divergence for each histogram distribution. Specifically, the respective distributions are $\mathcal{N}(0, 1/140)$ for the contrastive learning case and $\mathcal{N}(0, 1/3)$ for the non-contrastive learning-applied case.

Through the aforementioned analysis, we infer that when CL is implemented, (i) the embeddings from the two encoders demonstrate better alignment, and (ii) they are more randomly (broadly) distributed within the embedding space. These characteristics appear to contribute significantly to the Cream model's performance improvement.

4.3.2 On Efficient LLM Integration

In addition to the previously discussed analyses, we would like to emphasize the efficiency of the Cream integration with LLM compared to other approaches. A significant advantage of the Cream model is its reduced computational cost due to the smaller visual prompt length than feeding all OCR tokens. To elaborate, the complexity per layer of attention can be mathematically expressed as $\mathcal{O}(\bar{n}^2\bar{d})$, where $\bar{n}$ represents the sequence length of tokens, and $\bar{d}$ denotes the hidden dimension of model. Particularly for LLMs (e.g., 175B) with substantial $\bar{d}$ values and large number of attentions, a decrease in the input token length yields a substantial reduction in complexity. This efficiency allows the model to achieve superior performance while consuming fewer resources.

Figure 7 illustrates the token consumption of the LLM when directly inputting OCR to the model. The figure shows the number of OCR boxes and the resulting OCR tokens on the DocVQA dataset (Tito et al., 2021). We visualized tokenizers of Vicuna (Chiang et al., 2023) and FlanT5 (Chung et al., 2022).

As seen in the Figure 7, directly inputting OCR to the LLM would induce high computational cost. In the analyses with DocVQA (Tito et al., 2021) dataset, the mean number of required tokens to input the OCR output is 432 and 361 for Vicuna and FlanT5, respectively, which are considerably large. In contrast, the Cream integration with LLM offers a more efficient solution by reducing the visual prompt length, allowing the model to achieve better performance while consuming fewer tokens and computational resources. This efficiency highlights the potential of the Cream model in visually-situated language understanding tasks, especially when compared to other LLM integration approaches.

4.3.3 On Qualitative Assessment

In this study, we aim to investigate the benefits of incorporating an LLM with a comprehensive understanding of illustrations and reasoning capabilities for samples requiring such an approach. To this end, we examine various samples and conducted a qualitative assessment. Our results indicate that the integrated LLM significantly improved comprehension, which we believe contributed to the improved performance observed in the quantitative evaluation. Detailed working examples from the qualitative evaluation can be found in Appendix B.

5 Conclusion

In this work, we introduce Cream, ***C**ontrastive **rea**ding **m**odel* designed to address the limitations of existing LVLMs in text-rich visual tasks. The model features a streamlined and practical architecture that seamlessly integrates a general vision encoder with auxiliary encoders and innovative training techniques, including a contrastive feature alignment method. Through extensive experiments across various visually-situated language understanding tasks, we demonstrated Cream's state-of-the-art performance in tasks requiring text information extraction from document images. Furthermore, we showcased the seamless integration of Cream with LLMs and provided valuable resources to the research community by open-sourcing Cream's codebase and the newly-built VQA datasets, TydiVQA and WKVVQA. Our work paves the way for new developments and breakthroughs in the visually-situated language understanding domain.

Acknowledgements

The authors specially thank Seung Ho Choi and members of NAVER Cloud Hyperscale AI Vision Understanding Team for helpful discussions and encouragements.

Ethics Consideration

In this study, we present Cream, a novel approach that integrates LLMs within the well-established paradigm of large-scale pre-training followed by fine-tuning. Consequently, our method inherits the ethical concerns commonly associated with existing LLMs, such as biases present in pre-training data and privacy considerations. To mitigate these issues, we recommend employing a stringent and thorough protocol during the curation of pre-training data, especially for applications designed for public utilization. Our model's pre-training is conducted using controlled public data sources.

A critical aspect of document processing is the management of privacy-sensitive documents, such as identification cards. It is imperative to exercise caution in ensuring that such samples are excluded from training datasets to prevent potential privacy breaches and unintended consequences. This highlights the necessity for responsible data handling practices in the development of LLMs for document processing.

Moreover, our current approach relies on the direct output of the autoregressive decoder as the final output, which offers the advantage of eliminating the need for complex post-processing. However, in the context of ethical considerations, it may be prudent to explore the incorporation of post-processing techniques designed to address potential biases and privacy issues. Such methods could provide an additional layer of protection, ensuring that the outputs generated by the model align with ethical guidelines and best practices in the field of Natural Language Processing.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1003.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion

Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2023. End-to-end document recognition and understanding with dessurt. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 280–296. Springer.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, and Radu Soricut. 2022. Prestu: Pre-training for scene-text understanding.

Donghyun Kim, Teakgyu Hong, Moonbin Yim, Yoonsik Kim, and Geewook Kim. 2023. On web-based visual corpus construction for visual document understanding. In *International Conference on Document Analysis and Recognition (ICDAR)*. Accepted, to appear.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *Computer Vision – ECCV 2022*, pages 498–517, Cham. Springer Nature Switzerland.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2022. Pix2struct: Screenshot parsing as pretraining for visual language understanding.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

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 *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2022. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, et al. 2023b. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1697–1706.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. 2022. Simple open-vocabulary object detection. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, page 728–755, Berlin, Heidelberg. Springer-Verlag.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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. 2021a. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021b. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sakib Shahriar and Kadhim Hayawi. 2023. Let's have a chat! a conversation with chatgpt: Technology, applications, and limitations.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marcus Spruill. 2007. Asymptotic Distribution of Coordinates on High Dimensional Spheres. *Electronic Communications in Probability*, 12(none):234 – 247.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *AAAI*.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2022. Unifying vision, text, and layout for universal document processing. *arXiv preprint arXiv:2212.02623*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier García, Dara Bahri, Tal Schuster, Huaixiu Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *ArXiv*, abs/2205.05131.

Rubèn Tito, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2021. Icdar 2021 competition on document visual question answering. In *Document Analysis and Recognition – ICDAR 2021*, pages 635–649, Cham. Springer International Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2022b. Image as a foreign

language: Beit pretraining for all vision and vision-language tasks.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2022. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models.

A Insights on Prompt Engineering

A.1 OCR-LLM Prompt Variations

We examined the impact of various prompts on model performance in the context of the DocVQA dataset (Tito et al., 2021), using OCR tokens combined with LLMs. Our investigation revealed that integrating more specific conditions into a prompt generally led to better performance. For extraction tasks in which the correct answer is present within the OCR tokens, specifying a condition that the response should be sourced from the OCR tokens resulted in substantial performance improvements. Interestingly, the performance differed across LLMs, even when using the same prompt.

Using the Vicuna model (Chiang et al., 2023), we could eliminate unnecessary sentences and words from generated results by implementing conditions such as "(please output answer text only)." In this case, the term "OCR tokens" proved more advantageous than alternatives like "Images," "Image Results," or "OCR Outputs." For certain benchmarks similar to chartQA, constraining the length and number of words in the correct answer proved effective. Adding a phrase such as "Answer:" at the end of the prompt facilitated addressing the QA task. As LLMs frequently generate questions as part of their responses, offering a condition that omits any question-related text is advantageous (see Table 8).

Prompt5 exhibited the best performance across all models consistently, as highlighted in Table 7.

Model	P1	P2	P3	P4	P5
OCR-Vicuna7B	25.6	19.2	20.1	6.4	**28.3**
OCR-Vicuna13B	28.2	24.8	**29.5**	7.5	29.0
OCR-GPT3.5	50.1	60.4	47.9	17.9	**60.5**
OCR-GPT4	52.1	63.8	60.2	30.9	**70.4**

Table 7: **Results by prompt** The evaluation metric is ANLS, measured after 500 samples on the validation set.

A.2 Image-OCR-LLM Prompts

Table 9 displays a prompt that addresses a QA task given an image tensor, OCR tokens, and a question. The LLaVA model (Liu et al., 2023a) initially processes a system message and progresses through two conversation turns. However, the actual number of turns is contingent on the data used for inference. As this model was trained to generate detailed and descriptive output, it is helpful to present an example of a concise answer in the first turn.

In the case of the BLIP-2 model (Li et al., 2023), we refrained from imposing any special conditions, as it was designed to generate succinct answers. We first provided the image tensor, followed by the question, and then the answer. Moreover, we observed no significant performance disparities based on the image tensor.

It is important to note that the prompts used in our experiments may not be optimal for our model and data, and that alternative prompts could result in varying performance levels.

A.3 Cream Query Variations

Table 10 lists the queries employed for addressing individual tasks during Cream model training. Instead of using a single query for all tasks, we randomly sampled an array of query types to enhance

No	Prompt
1	Image OCR Result: {ocr tokens} / Question: {question} / + (please output answer text only) + (with no more than five words) + Answer:
2	Image OCR Result: {ocr tokens} / Question: {question} / + (please output answer text only) + (Limit your answer to 50 characters or less) + (Answers should not include question text) + Extact Answer text in OCR Result:
3	Image OCR Result: {ocr tokens} / Question: {question} / + (please output answer text only) + (with no more than ten words) + (Answer should not include question text) + (The answer text must be included in the OCR text) + Short Answer:
4	OCR tokens: {ocr tokens} {question} OCR tokens: {ocr tokens} Question: {question} OCR tokens: {ocr tokens} {question} A short answer to the question is OCR tokens: {ocr tokens} Q: {question} A: OCR tokens: {ocr tokens} Question: {question} Short answer: OCR tokens: {ocr tokens} Given the image, answer the following question with no more than three words. {question} OCR tokens: {ocr tokens} Based on the image, respond to this question with a short answer: {question}. Answer: OCR tokens: {ocr tokens} Use the provided image to answer the question: {question} Provide your answer as short as possible: OCR tokens: {ocr tokens} What is the answer to the following question? "{question}" OCR tokens: {ocr tokens} The question "{question}" can be answered using the image. A short answer is
5	OCR tokens: {ocr tokens} / Question: {question} / + (Please output answer text only) + (With no more than 10 words) + (The answer must be a word that exists within the OCR tokens.) + Answer:

Table 8: **Inference prompt for OCR-LLM** The prompts for DocVQA (Tito et al., 2021). prompt4 are adapted from InstructBLIP (Dai et al., 2023).

the model's generalization capacity.

While the prompts for Task Reading, Masked LM, and Captioning tasks are relatively simple sentences, the placement of questions and answers in the prompts for the QA and QG tasks proved to be crucial. In our study, we mostly positioned them at the end of the prompt.

All prompts consist of a singular sentence structure and are no more than 100 characters long, excluding the question and answer in QA and QG tasks, respectively. The prompts used in Cream Training emphasize employing concise and straightforward sentences as a basic principle.

B On Working Examples

Figure 9 shows the working examples for ChartQA benchmarks. In Infographic VQA, we can also see that cream synthesize information from infographics and utilize knowledge LLM learnt and make inferences, but also utilize information previously learned. Figure 10 shows the working examples for ChartQA benchmarks. In Chart QA, it shows an improvement in the ability to identify elements within a graph and perform numerical operations between the information on each element by integrating an LLM.

C Contribution of Authors

Geewook Kim led the project as a task force manager, initiated the project, and made decisions on overall progress while organizing the research paper. **Hodong Lee** managed the overall dataset construction, co-initiated the project, organized model evaluations, and significantly contributed to code development. **Daehee Kim** contributed to the model architecture with a focus on contrastive feature alignment and played a key role in crafting the manuscript. **Haeji Jung** managed dataset construction at the project's beginning, co-initiated the project, and contributed to its proof of concept. **Sanghee Park** handled data processing, evaluated off-the-shelf LLMs, and made substantial contributions to prompt engineering. **Yoonsik Kim** provided critical advice on research direction and de-

Model	Prompt
LLaVA (Liu et al., 2023a)	You are LLaVA, a large language model trained by UW Madison WAIV Lab. + You are able to understand the visual content that the user provides, + and answer userś question using image and natural language. + Follow the instrutions carefully and provide answer + text only without question included, less than five words ###Human: What is the type of image? + (please output answer text only without question and explanation) + (with no more than five words) ###Assistant: The answer is a document image. ###Human: {question} {image tensor} ###Assistant:
BLIP-2 (Li et al., 2023)	{image tensor} Question: {question} Answer: {image tensor} OCR tokens: {ocr tokens} Question: {question} Answer:

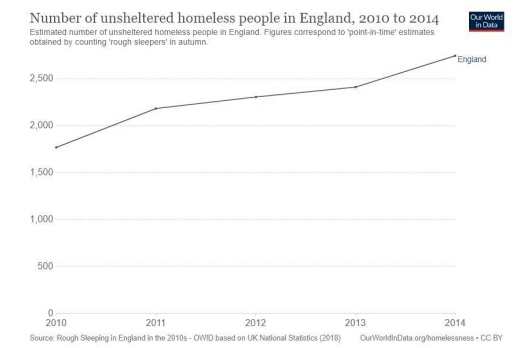
Table 9: **Inference prompt for Image-OCR-LLM** The prompts for DocVQA (Tito et al., 2021). We used a single prompt for each model.

velopment, heavily contributed to the manuscript, and participated in model architecture development. **Sangdoo Yun** shaped the overall research direction and contributed conceptualization of Cream as a senior researcher. **Taeho Kil** gave advice on overall research direction and development, and contributed to the manuscript as a senior researcher. **Bado Lee** advised the project from its beginning, co-initiated the project, and made significant contributions to creating the necessary environment and resources. **Seunghyun Park** advised the project from its inception, co-initiated the project, and significantly contributed to shaping the project's direction as a senior researcher.

All participants contributed to this manuscript.

Task	Queries
Text Reading	Read all texts. Read all texts in the image. Read all characters in the image. Given the image, read all texts. Given the image, read all characters.
Masked LM	Read masked texts. Read masked texts in the image. Given the image, read masked texts. Read all hidden texts that are covered by the mask area.
Captioning	Explain the image. Use a few words to illustrate what is happening in the picture. Using language, provide a short account of the image. Please provide a short depiction of the picture. Could you use a few words to describe what you perceive in the photo? Can you briefly explain what you see in the image? Briefly describe the content of the image. Provide a description of what is presented in the photo. Write a description for the photo. Write a short description for the image.
QA	{query} Q: {query} Question: {query} Given the image, answer the following question. {query} Based on the image, respond to this question with a short answer: {query}. Use the provided image to answer the question: {query}. Provide your answer as short as possible. What is the answer to the following question? "{query}" The question "{query}" can be answered using the image.
QG	Given the image, generate a question whose answer is: {answer}. Based on the image, provide a question with the answer: {answer}. Given the visual representation, create a question for which the answer is "{answer}". From the image provided, craft a question that leads to the reply: {answer}. Considering the picture, come up with a question where the answer is: {answer}. Taking the image into account, generate a question that has the answer: {answer}.

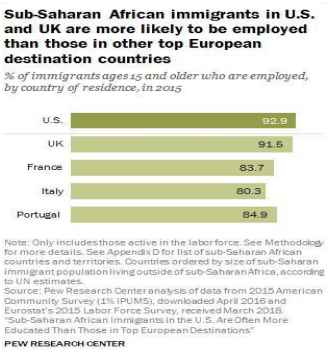
Table 10: **Task-specific queries (prompts) used in Cream training.** The prompts for Captioning, QA, and QG tasks are adapted from BLIP-2 (Li et al., 2023).



Q: When does the line have the sharpest increase?

✗ 2013 ✗ 2014 ✓ 2011

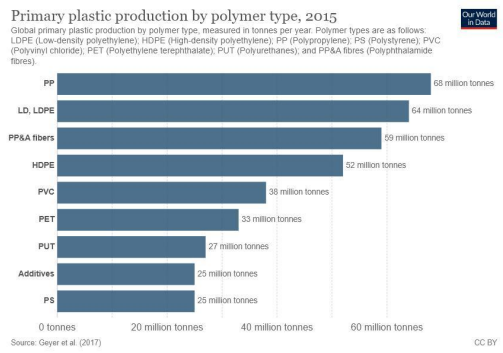
BLIP-2 Ours (Cream) **Ours** (Cream + LLM)



Q: Which country does the Dark green represent?

✗ France ✗ UK ✓ US

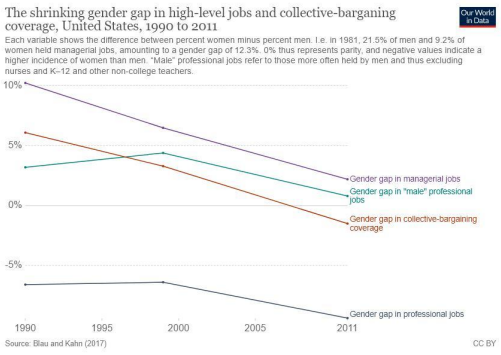
BLIP-2 Ours (Cream) **Ours** (Cream + LLM)



Q: What type of plastic was most produced?

✗ LDPE ✓ PP ✓ PP (Polypropylene)

BLIP-2 Ours (Cream) **Ours** (Cream + LLM)

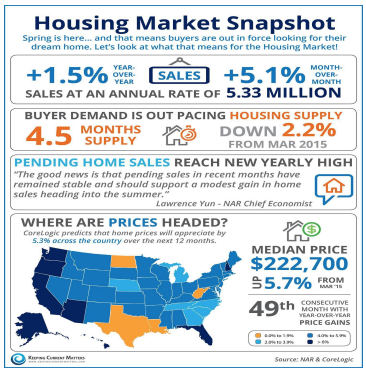


Q: When is the average value of all four gaps highest?

✗ 1981 ✗ [1990, 2011, 5, 10] ✓ 1990

BLIP-2 Ours (Cream) **Ours** (Cream + LLM)

Figure 9: **Working Examples for ChartQA Benchmarks.**



Q: Which state in the north-eastern region of the U.S. has greater than 6% of year-over-year home price gain - New York, Vermont, New Hampshire, Maine?

✗ Maine — BLIP-2

✗ New York — Ours (Cream)

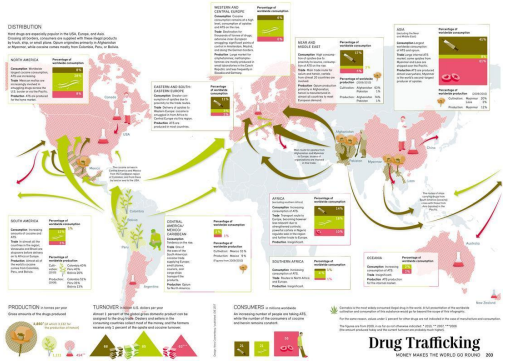
✓ New Hampshire — **Ours** (Cream + LLM)

Q: Who won the 2011 ICC Cricket World Cup?

✗ england — BLIP-2

✗ Australia — Ours (Cream)

✓ India — **Ours** (Cream + LLM)

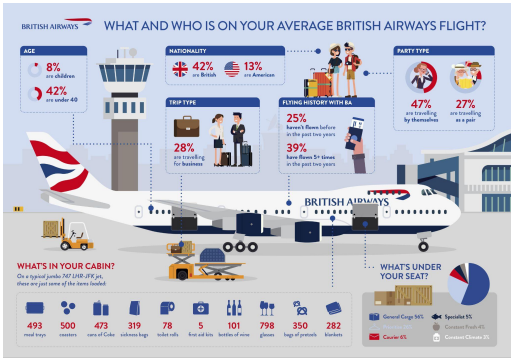


Q: Which continent has the largest consumption of opium?

✗ WESTERN AND of CENTRAL EUROPE — BLIP-2

✗ Eastern and South-East Europe — Ours (Cream)

✓ Asia — **Ours** (Cream + LLM)



Q: What percentage are not traveling by themselves?

✗ 47% are travelling — BLIP-2

✗ 73% — Ours (Cream)

✓ 53% — **Ours** (Cream + LLM)

Figure 10: **Working Examples for InfographicVQA Benchmarks.**