MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, William W. Cohen
Google Research
{wenhuchen,hexiang,patverga,wcohen}@google.com

Abstract

While language Models store a massive amount of world knowledge implicitly in their parameters, even very large models often fail to encode information about rare entities and events, while incurring huge computational costs. Recently, retrieval-augmented models, such as REALM, RAG, and RETRO, have incorporated world knowledge into language generation by leveraging an external non-parametric index and have demonstrated impressive performance with constrained model sizes. However, these methods are restricted to retrieving only textual knowledge, neglecting the ubiquitous amount of knowledge in other modalities like images – much of which contains information not covered by any text. To address this limitation, we propose the first Multimodal Retrieval-Augmented Transformer (MuRAG), which accesses an external non-parametric multimodal memory to augment language generation. MuRAG is pre-trained with a mixture of large-scale image-text and text-only corpora using a joint contrastive and generative loss. We perform experiments on two different datasets that require retrieving and reasoning over both images and text to answer a given query: WebQA, and MultimodalQA. Our results show that MuRAG achieves state-of-the-art accuracy, outperforming existing models by 10-20% absolute on both datasets and under both distractor and full-wiki settings.

1 Introduction

Pre-trained language models like GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), etc have been shown to capture a massive amount of world knowledge implicitly in their parameters. However, using such large models incurs an extremely high computation cost. As an alternative to a singular monolithic transformer, retrieval-augmented architectures like KNN-LM (Khandelwal et al., 2019), REALM (Guu et al., 2020), RAG (Lewis et al., 2020), FiD (Izacard and Grave, 2021), and RETRO (Borgeaud et al., 2021) have been proposed to decouple world knowledge from the model's parameters. More specifically, these models are trained to access an external memory to enhance the model's predictions. Such retrieval-augmented architectures have multiple beneficial properties including: decreased model size (Borgeaud et al., 2021), better attribution/-explanation for model predictions (Lewis et al., 2020), and adaptability to new information without retraining (Verga et al., 2021). However, previous retrieval-augmented models are limited to memories that contain only text or structured data and hence cannot make use of the massive amount of multimodal knowledge available on the web—much of which contains information only available in non-text modalities.

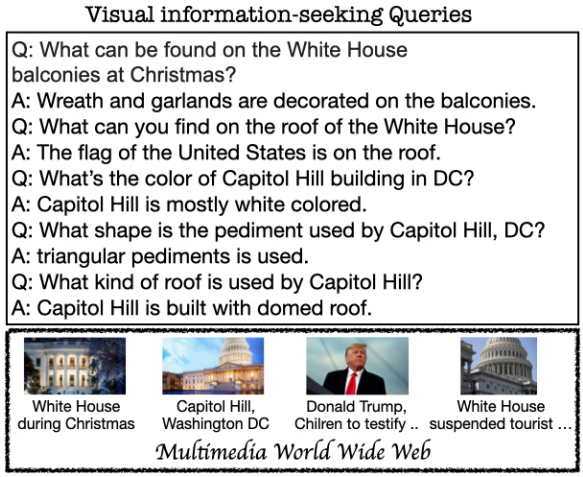

Figure 1: **Visual information-seeking queries**: These queries are unanswerable with text-only retrieval and require retrieving and reasoning over images.

Figure 1, shows several information-seeking queries that require retrieving and reasoning over visual knowledge. Here, a user first poses a question such as *"What can be found on the White House balconies at Christmas"*. The system then retrieves relevant items from its memory, for exam-

ple, the first image of Figure 1 with the caption *"White House during Christmas"*, which it uses to produce the answer *"wreaths and garlands"*. Existing text retrieval-augmented models would struggle with such queries because, in many cases, they would simply not have access to the answer as some knowledge does not exist in text form. That, coupled with the abundance of multimodal knowledge that exists, leads to the conclusion that retrieval-augmented models should ultimately be developed to retrieve and reason over multiple modalities.

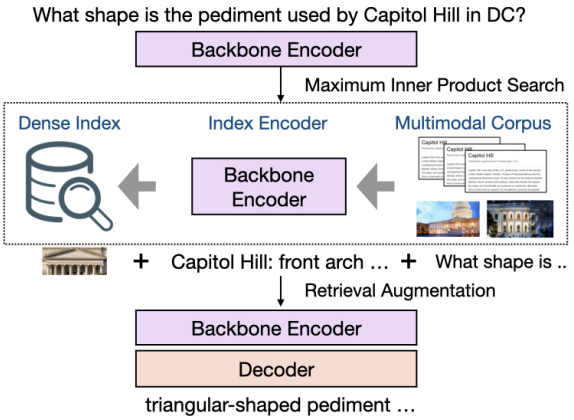

Figure 2: **Model Overview**: retrieval-and-predict process of MuRAG on downstream datasets.

In this paper, we are specifically interested in endowing pre-trained language models with a non-parametric multimodal memory containing images, text, or image-text pairs. To accomplish this, we first combine pre-trained T5 (Raffel et al., 2020) and ViT (Dosovitskiy et al., 2020) models to build a backbone encoder (Figure 3), which encodes image-text pairs, image-only, and text-only inputs into a multimodal representation. MuRAG uses the backbone encoder to embed items into an external memory as well as queries to retrieve multimodal knowledge from that memory. These retrievals then augment a language model to generate more visually-grounded outputs.

We pre-train MuRAG with a mixture of image-text and text-only datasets including LAION (Schuhmann et al., 2021), Conceptual-Caption (Sharma et al., 2018), VQA (Antol et al., 2015) and Probably-Asked-Questions (PAQ) (Lewis et al., 2021). More specifically, we reformulate these datasets in a retrieve-and-predict format. Here, the model's input is an image along with a text prompt. The model then retrieves from a memory containing captions and passages, which it uses to generate a target token sequence. The model is trained with both a contrastive and a generative loss; this teaches the model to discriminate relevant from irrelevant memory entries, and guides the model to leverage the multimodal knowledge into generation.

Unlike the pre-training stage, during fine-tuning Figure 2 the model's input is a question, and the memory contains a collection of captioned images and text snippets. We fine-tune MuRAG on the downstream datasets with a contrastive and generative loss similar to pre-training. To avoid excessive computation cost, we develop a two-stage training pipeline to first train with small in-batch memory, and then with a statically encoded and indexed large global memory.

Our experiments show that MuRAG achieves state-of-the-art performance on two different open-multimodal-QA datasets, both of which require retrieving images and text from a large corpus to answer factoid questions: WebQA (Chang et al., 2022) and MultimodalQA (Talmor et al., 2021). On both datasets, we outperform sophisticated baselines (Li et al., 2020; Radford et al., 2021; Zhang et al., 2021) by 10-20% accuracy under both distractor (from 40+ candidates) and full-wiki settings (from 1M candidates). We also perform a comprehensive study to ablate different components of the pre-training to see their contributions. These empirical results demonstrate the effectiveness of our proposed models to integrate multimodal knowledge into pre-trained generation models and pave the way to unified retrieval-augmented frameworks.

2 Related Work

Retrieval Augmented Models Retrieval augmented models are hybrid models containing both parameterized sequence models and a non-parametric memory, infusing world knowledge into existing language models. Among them, KNN-LM (Khandelwal et al., 2019) was first proposed to retrieve instances from a text training corpus to help language modeling. Later, RETRO (Borgeaud et al., 2021) was proposed to scale up the text corpus to trillions of tokens, enabling the model to achieve similar perplexity to GPT-3 (Brown et al., 2020) with 25x fewer model parameters. Another family of models, such as REALM (Guu et al., 2020), RAG (Lewis et al., 2020), and FiD (Izacard and Grave, 2021), integrate Wikipedia passages as a datastore to benefit downstream knowledge intensive tasks (*e.g.* Question Answering). REALM is an encoder-only model trained with masked lan-

guage modeling, while RAG and FiD adopt an encoder-decoder model with a generative language modeling objective. Compared to them, MuRAG is the first retrieval-augmented model that is capable of using knowledge presented in multiple modalities (*i.e.* visual and textual knowledge data), whereas all prior methods are restricted to using text-only knowledge.

Multimodal Transformers Multimodal transformers have demonstrated strong performances in learning cross-modal representation that are generally beneficial on downstream vision and language tasks, such as image-text retrieval (Karpathy and Fei-Fei, 2015), image captioning (Chen et al., 2015), and VQA (Antol et al., 2015). These methods typically learn a joint transformer model on top of unimodal visual and textual backbones, via fusing deep features from each modality. The early version of multimodal transformers (Lu et al., 2019; Chen et al., 2020; Li et al., 2020) usually learns a Transformer on pre-extracted unimodal features for contextualization, which makes it impossible to adjust those unimodal features to the target tasks. Recently, SimVLM (Wang et al., 2022) and COCA (Yu et al., 2022) proposed end-to-end training for both deep multimodal transformers and unimodal featurization networks and demonstrated strong performance in both multimodal and unimodal downstream tasks. The multimodal memory encoder of MuRAG is broadly similar to SimVLM and CoCa, but has a different focus to encode and retrieve multimodal knowledge (*i.e.* images and texts) to augment language generation models.

Multimodal Question Answering The problem of multimodal question answering has been extensively studied. VQA was the first proposed to answer questions from visual-only inputs. Later, OK-VQA (Marino et al., 2019) enlarged VQA's scope to annotate questions requiring both image and implicit textual/common-sense knowledge to answer. More recently, MuMuQA (Reddy et al., 2021), ManyModelQA (Hannan et al., 2020) and MIMOQA (Singh et al., 2021) provide questions which require reasoning over images and explicitly provided text snippets. However, these datasets are restricted to dealing with given text and images without requiring any retrieval from the web: they are analogous to machine-reading approaches to QA from text like SQuAD, rather than open-book QA. To study the more realistic open multimodal QA task, WebQA (Chang et al., 2022) and MultimodalQA (Talmor et al., 2021) have been proposed to evaluate answers to open queries which require retrieving and reasoning over a large-scale web multimodal corpus. Our model uses these datasets to study open-world multimodal question answering, obtaining state-of-the-art results.

3 Model

3.1 Backbone Encoder

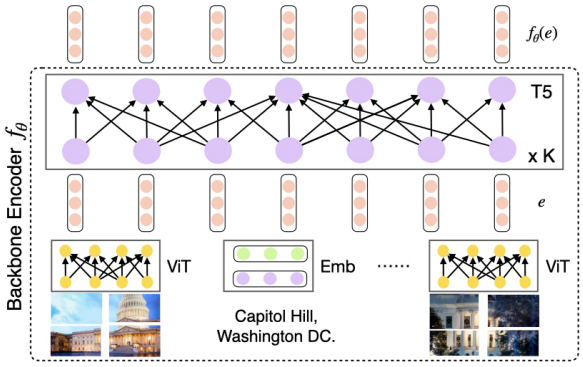

Figure 3: Backbone encoder: ViT encodes image patches into a sequence of vectors $\mathbf{e}_I$, while word embedding converts text tokens into another sequence of vectors $\mathbf{e}_T$. These vectors are concatenated to form $f_\theta(e)$ and fed to a decoder for text generation.

MuRAG is built on top of a simpler model we call a "backbone" model, which is pre-trained to encode image-text pairs such that they are suitable for both answer generation and retrieval. The backbone model's encoder is used as a component of the MuRAG model. The backbone model is built with a pre-trained visual Transformer (Dosovitskiy et al., 2020) and a T5 text Transformer (Raffel et al., 2020), and consists of a multimodal encoder f_θ and decoder g_θ. The encoder takes as input a sequence of image-text pairs, where either the image or the text component can be empty to accommodate text-only and image-only cases.

As depicted in Figure 3, the encoder can take a sequence of images and text. For image input, we first split each into 16x16 patches and feed them to a ViT (Dosovitskiy et al., 2020) transformer to generate a sequence of visual embedding denoted as $\mathbf{e}_I \in \mathbb{R}^{L_i \times D}$, where L_i is the length of the image tokens. For text input, we use word embedding to produce another sequence of textual embedding $\mathbf{e}_T \in \mathbb{R}^{L_t \times D}$. For k images and n text inputs, we concatenate all their embeddings in the input order as $\mathbf{e} = [\mathbf{e}_I^1; \mathbf{e}_T^1; \cdots; \mathbf{e}_I^k; \mathbf{e}_T^n] \in \mathbb{R}^{(kL_t + nL_i) \times D}$, which is fed to another bi-directional transformer f_θ initialized from T5. We enable cross-attention

between the two modalities to produce a fused representation, denoted as $f_\theta(\mathbf{e}) \in \mathbb{R}^{(kL_t+nL_i)\times D}$. We add a [CLS] token to obtain a pooled representation $f_\theta(\mathbf{e})_{[\text{CLS}]} \in \mathbb{R}^D$ for dense retrieval.

3.2 MuRAG

We build MuRAG (shown in Figure 4) on top of the backbone model. During the retriever stage, MuRAG takes a query q of any modality as input and retrieves from a memory $\mathcal{M}$ of image-text pairs. Specifically, we apply the backbone encoder f_θ to encode a query q, and use maximum inner product search (MIPS (Guo et al., 2020)) over all of the memory candidates $m \in \mathcal{M}$ to find the Top-K nearest neighbors $Top_K(\mathcal{M}|q) = [m_1, \cdots, m_k]$. Formally, we define $Top_K(\mathcal{M}|q)$ as follows:

$$Top_K(\mathcal{M}|q) = \underset{m\in\mathcal{M}}{TopK} \quad f_\theta(q)_{[\text{CLS}]} \cdot f_\theta(m)_{[\text{CLS}]}$$

During the reader stage, the retrievals (the raw image patches) are combined with the query q as an augmented input $[m_1, \cdots, m_k, q]$, which is fed to the backbone encoder f_θ to produce retrieval-augmented encoding. The decoder model g_θ uses attention over this representation to generate textual outputs $\mathbf{y} = y_1, \cdots, y_n$ token by token.

$$p(y_i|y_{i-1}) = g_\theta(y_i|f_\theta(Top_K(\mathcal{M}|q); q); y_{1:i-1})$$

where y is decoded from a given vocabulary $\mathcal{V}$.

3.3 Pre-training

The pre-training implementation is depicted in the upper portion of Figure 4, where the input query is an image x_I plus a text prompt x_p. The external memory $\mathcal{M}$ contains textual-only entries m^T. The Top-K retrievals $m_1^T, \cdots, m_k^T$ are leveraged to generate the textual output. To avoid the excessive computation cost of backpropagation over the massive external memory, we adopt an in-batch memory $\mathcal{M}_B$, dynamically constructed from the input examples in a batch. The small in-batch memory enables MuRAG to continuously update the memory encoder efficiently similar to TOME (de Jong et al., 2022) and QAMAT (Chen et al., 2022).

Dataset The pre-training corpus consists of LAION (Schuhmann et al., 2021), Conceptual-Caption-12M+3M (CC) (Sharma et al., 2018; Changpinyo et al., 2021), VQA (Antol et al., 2015) and PAQ (Lewis et al., 2021) Table 1. LAION is a publicly-released image-text dataset containing crawled image-text pairs filtered by CLIP (Radford et al., 2021). We apply rules to filter LAION from 400M to 200M by removing text with HTTP-URLs or image width/height beyond 1000 pixels. CC contains 15M (image, anonymized alt-text) pairs crawled from the web but filtered more extensively to maintain high alignment quality. VQA contains annotated QA pairs aligned to MSCOCO images. We further add captions to each image from MSCOCO-Captioning (Lin et al., 2014) to create (Image, Caption, QA) triples. PAQ is a text-only dataset containing 65M machine-generated QA pairs along with their source Wikipedia passage.

Dataset	#Size	Format	Source
CC	15M	(Image, Caption)	Crawled
LAION	200M	(Image, Alt-Text)	Crawled
PAQ	65M	(Passage, QA)	Generated
VQA	400K	(Image, Caption, QA)	Annotated

Table 1: Pre-training Dataset Statistics

For LAION and CC, we use the input image as x_I, and 'generate caption:' as the text prompt x_p. For VQA, we use the input image as x_I and the question as the prompt x_p. For PAQ, we use an empty array as the input image and the question as the prompt. The in-batch memory $\mathcal{M}_B$ is constructed by stacking the captions associated with the input images in LAION/CC/VQA and the passages associated with the questions in PAQ. Each textual memory entry is denoted as m^T. The decoder is optimized to generate either a caption or an answer, depending on the source dataset. Since the four dataset sizes are highly unbalanced, we use fixed mixture sampling ratios to balance their presence during pre-training.

We train the model with a joint loss $L = L_{gen} + L_{con}$ as follows:

$$L_{con} = -\log \frac{exp(f_\theta(x_I, x_p) \cdot f_\theta(m^T))}{\sum_{m\in\mathcal{M}_B} exp(f_\theta(x_I, x_p) \cdot f_\theta(m^T))}$$

$$L_{gen} = -\log g_\theta(\mathbf{y}|f_\theta(M_p; x_I; x_p))$$

$$M_p = \begin{cases} Top_K(\mathcal{M}_B|x_I, x_p) & \text{If } (x_I, x_p) \in \text{PAQ/VQA} \\ \emptyset & \text{If } (x_I, x_p) \in \text{LAION/CC} \end{cases}$$

where M_p is the retrieved augmentation: if the input query is from PAQ/VQA, we use the retrieved memory entries, otherwise, we use null. The reason for setting it to null for LAION/CC is to avoid a trivial solution when the generation target (caption) also exactly appears in the memory.

The contrastive loss L_{con} is minimized to discriminate between the positive query-memory pairs

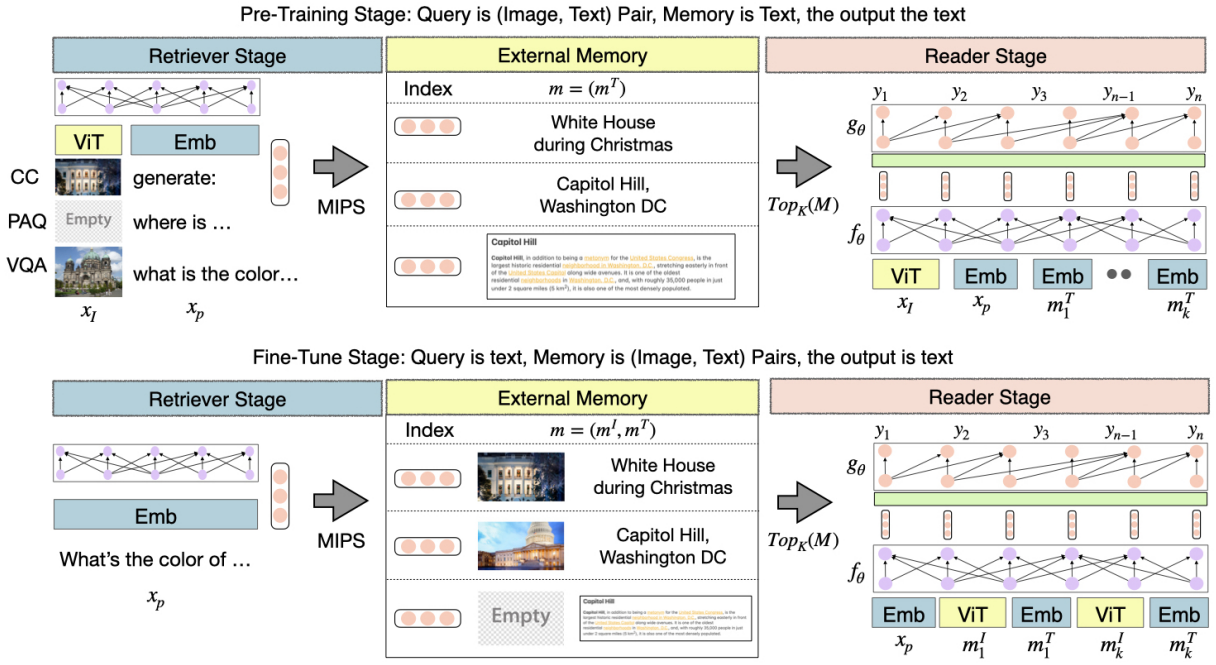

Figure 4: Model Architecture: the model accesses an external memory to obtain multimodal knowledge contained in images or text snippets, which is used to augment the generation. The upper part defines the pre-training implementation, while the lower part defines fine-tuning implementation.

and all other query-memory pairs from the memory. The pairwise matching score is computed as the dot product between query $f_\theta(x_I; x_p)_{\text{[CLS]}}$ and candidates $f_\theta(m^T)_{\text{[CLS]}}$. This objective enables the model to retrieve the most relevant knowledge from the memory. The generative loss L_{gen} is minimized to generate target tokens $\mathbf{y}$ conditioned on the retrieval-augmented representation. This objective enables the model to combine information across different modalities for text generation.

3.4 Fine-tuning

We finetune MuRAG to align with the expected inputs of the downstream datasets which require answering text questions by retrieving image-caption pairs or text snippets from the external knowledge datastore. As depicted in the lower part of Figure 4, the input query for the downstream task is a text question x_q, and the memory $\mathcal{M}$ containing (image, text) pairs (m^I, m^T).[1] The Top-K retrievals $\{(m_1^I, m_1^T), \cdots, (m_k^I, m_k^T)\}$ are leveraged to generate the answer a. To minimize the computation cost, we develop a two-stage pipeline to optimize with an in-batch memory and then resume with fixed retrieval from global memory.

In-Batch Training In this stage, we aim to minimize the joint loss function $L = L_{con} + L_{gen}$ based on the in-batch memory $\mathcal{M}_B$ as follows:

$$L_{con} = -\log \frac{exp(f_\theta(x_q) \cdot f_\theta(m^I; m^T))}{\sum_{m \in \mathcal{M}_B} exp(f_\theta(x_q) \cdot f_\theta(m^I; m^T))}$$

$$L_{gen} = -\log g_\theta(\mathbf{y} | f_\theta(Top_K(\mathcal{M}_B | x_q); x_q))$$

The in-batch memory $\mathcal{M}_B$ is constructed in the following way: the k-th example in the dataset is represented as $(x_{q,k}, y_k, \{m_i^I, m_i^I\}_k, \{\bar{m}_j^I, \bar{m}_j^T\}_k)$, where m represents the positive (image, text) source, and $\bar{m}$ represents the hard negative (image, text) source provided by the dataset[2]. For a batch with B examples, we assemble all the associated positive and negative knowledge source as our in-batch memory $\mathcal{M}_B = \{\{m_i^I, m_i^I\}_1, \{\bar{m}_j^I, \bar{m}_j^T\}_1, \cdots, \{\bar{m}_j^I, \bar{m}_j^T\}_B\}$.

Fixed-Retrieval Training After in-batch training, we encode all available cross-modal pairs, and index these encodings for fast MIPS retrieval. We then apply the trained retriever to search over the full multimodal corpus $\mathcal{M}$ to obtain the global top-K retrievals $Top_K(\mathcal{M}|x_q)$ and continue to optimize L_{gen}. During this training phase, the stored encodings are not updated. During inference time, we use fixed encodings to generate the answers.

[1] We set the image to a zero array if the memory entry is a text snippet.

[2] These hard negatives are mined through Bing Search API and Wikipedia page, refer to (Chang et al., 2022) for details.

4 Experiments

4.1 Implementation Details

The backbone model uses T5-base (Raffel et al., 2020) and a ViT-large model (Dosovitskiy et al., 2020) as described in Table 2. We adopt the sentence-piece model from T5 with a vocabulary size of 32128. The ViT model was pre-trained on the JFT dataset. We resize every image into 224x224 pixels and split them into a sequence of 16x16 patches. The output of ViT is a sequence of 1024-dimension vectors, which are projected to 768-dimension for consistency with T5 model. MuRAG reuses the model as retriever and reader, thus the full model size is 527M parameters.

Model	#Enc	#Dec	Hidden	Heads	Params
ViT-large	24	0	1024	16	307M
T5-base	12	12	768	12	220M

Table 2: The model size and configurations, with #Enc/#Dec denoting encoder/decoder layers.

Our model is implemented in JAX (Bradbury et al., 2018), based on the T5X codebase (Roberts et al., 2022). During pre-training, we first train the model on LAION for 1M steps, and then continue training on CC/PAQ/VQA with 1:1:1 sample ratio for another 200K steps. We optimize the model with Adafactor (Shazeer and Stern, 2018). For both stages, we adopt a constant learning rate of 5e-4 and a batch size of 4096. The models are trained on 64 Cloud v4 TPUs (Jouppi et al., 2020).

We then fine-tune MuRAG on WebQA and MultimodalQA with a constant learning rate of 3e-4 for 20K steps. The checkpoint with the highest validation score is run on the test set. We use a batch size of 64 and set TopK=4 for both in-batch training and fixed-retrieval training. We noticed that increasing Top-K further does not yield further improvement. We use a beam size of 2 to search for the best hypothesis for both datasets (increasing it further doesn't yield better performance).

4.2 Datasets

For evaluation, we choose two multimodal QA datasets: WebQA (Chang et al., 2022) and MultimodalQA (Talmor et al., 2021) and demonstrate their statistics in Table 3.

WebQA This dataset contains multi-hop, multimodal question-answer pairs where all questions are knowledge-seeking queries. The queries require 1-2 images or 1-2 text snippets to answer.

Dataset	Train Image/Text	Dev Image/Text	Test Image/Text
WebQA	18K/17K	2.5K/2.4K	3.4K/4K
MultimodalQA	2.1K/7.4K	230/721	-

Table 3: Overall Statistics of downstream dataset.

Each query in WebQA is associated with a set of visual/text distractors (hard negatives). The answers in WebQA are normally complete sentences to better assess the model's generation capability. Two evaluation setups are used, namely distractor and full-wiki. Under the distractor setup, the model needs to retrieve from these hard negatives + positives to answer the question. Under the full-wiki setup, the model needs to search over 1.1M text and visual sources from Wikipedia to answer the question. For evaluation, WebQA uses BARTScore (Yuan et al., 2021) to measure the fluency between the generation and the reference, and keyword accuracy score to measure the correctness/truthfulness of the generation. These two scores are multiplied to calculate the overall score.

MultimodalQA-Subset This dataset contains human-annotated multimodal questions over different modalities including tables, text, and images. Wikipedia tables are used as anchors to connect different modalities. The authors first use the template to generate questions and then ask crowd-workers to filter and paraphrase the generated questions. Since tables are outside the scope of our paper, we focus on the subset of queries requiring only text and image information. Specifically, we choose the questions with types of 'TextQ' and 'ImageQ' to construct the subset. The query requires 1 image or 1 text snippet to answer. Each query in MultimodalQA is also associated with visual and text distractors (hard negatives). Similarly, two evaluation setups are used as before. Under a full-wiki setup, MultimodalQA uses a database containing 500K text and visual sources. The evaluation scores are based on Exact Match and F1.

4.3 Baselines

For WebQA and MultimodalQA, we mainly compare different variants of pre-trained vision-language models.

VLP In WebQA, VLP-like models (Zhou et al., 2020) like Oscar (Li et al., 2020) and VinvL (Zhang et al., 2021) are used as the standard baselines. These models were pre-trained on Conceptual

3M (Sharma et al., 2018) with a masked language objective. During fine-tuning, the VLP model takes a set of token inputs <[CLS], s_i, [SEP], Q, [SEP]> first to select the most plausible source s_i, and then feed s_i in the form of <[CLS], S, Q, A, [SEP]> to autoregressively decode answer A with masked language model prediction.

AutoRouting In MultimodalQA, this method first applies a question type classifier to detect the modality of the question (either a passage or an image), and then routes the question to its sub-model. The method uses RoBERTa-large (Roberts et al., 2022) for text-questions and VilBERT (Lu et al., 2019) with features extracted from Faster-RCNN (Ren et al., 2015) for image questions.

CLIP (K) CLIP (Radford et al., 2021) is used for full-wiki retrieval. Specifically, the baselines systems adopt CLIP to encode queries and all the image/text candidates separately into vectors and then run approximated nearest neighbor searches to find a set of K potential candidates. After the coarse-level retrieval without cross-attention, it adopts a reranker to further narrow down to the 1-2 candidates to feed as input S to the QA model.

4.4 Experimental Results

We demonstrate WebQA's results in Table 4. All results reported are the medium score from three runs with different random seeds, and the variance of the Overall score is within 0.2%. We can observe that MuRAG can significantly outperform VLP with different backends including Oscar, ResNet, and VinVL. In retrieval performance, our model outperforms VLP by 15% in the full-wiki setting. For Fluency, our model outperforms VLP by 12% under the distractor setting and 14% under the full-wiki setting. For Accuracy, our model manages to achieve 16% under the distractor setting and even 20% the under the full-wiki setting. These improvements reflect the high fluency and accuracy of MuRAG's generation, and the improvement is more pronounced for full wiki.

We show the MultimodalQA results in Table 5. We can see that MuRAG is also able to vastly outperform the routing-based multimodality QA model. For text questions, our model improves over AutoRouting by 10+% EM under both settings. For image questions, the gap becomes more significant, with 20+% improvement under both settings. Similarly, we find that our model is more capable of handling full-wiki corpus.

Evaluation Metrics	Distractor Retr	FL	Accuracy	Overall
Question-Only	-	34.9	22.2	13.4
VLP (Oscar)	68.9	42.6	36.7	22.6
VLP + ResNeXt	69.0	43.0	37.0	23.0
VLP + VinVL	70.9	44.2	38.9	24.1
MuRAG	**74.6**	**55.7**	**54.6**	**36.1**
Evaluation	Full-Wiki			
CLIP (2) + VLP	11.9	34.2	24.1	14.6
CLIP (20) + VLP	24.0	36.1	27.2	16.1
MuRAG	**39.7**	**50.7**	**47.8**	**31.5**

Table 4: WebQA official test-set results indicated on leaderboard[3] as of May 2022. Retr denotes the retrieval-F1 score. FL refers to fluency metric BARTSCcore, and Accuracy refers to keyword matching F1 score, they are combined as Overall.

Evaluation	Distractor				
Metrics	Text		Image		All
	EM	F1	EM	F1	EM
Question-Only	15.4	18.4	11.0	15.6	13.8
AutoRouting	49.5	56.9	37.8	37.8	46.6
MuRAG	**60.8**	**67.5**	**58.2**	**58.2**	**60.2**
Evaluation	Full-Wiki				
Metrics	Text		Image		All
	EM	F1	EM	F1	EM
CLIP (10) + AutoRouting	35.6	40.2	32.5	32.5	34.7
MuRAG	**49.7**	**56.1**	**56.5**	**56.5**	**51.4**

Table 5: Multimodal dev-set results on the subset.

4.5 Ablation Study

Here we ablate the properties of MuRAG to better understand our experimental results.

Pre-training Corpus In order to study the contributions of different pre-training corpora, we investigated several pre-training corpus combinations. We report their fine-tuned results on WebQA test set in Table 6. As can be seen, without any pre-training, our model only achieves an overall score of 23.5, which lags behind the baseline models. After pre-training on different singular datasets, MuRAG is able to achieve better performance than the baselines. Among the individual datasets, LAION is shown to yield the highest score, and adding CC, PAQ, and VQA to the pre-training corpus set one by one produces steady improvements.

Two-Stage Fine-tuning In order to study the necessity of the two-stage fine-tuning, we perform an ablation study to see the impact of the two stages. We display our results in Table 7. (Only In-Batch)

Pre-train Dataset	FL	Accuracy	Overall
None	42.5	36.1	23.5
CC	46.4	41.3	25.6
LAION	47.8	44.8	28.3
VQA	47.0	44.4	27.4
PAQ	46.8	42.8	27.0
LAION+CC	49.5	47.4	30.7
LAION+CC+PAQ	53.7	51.8	34.4
LAION+CC+PAQ+VQA	55.7	54.6	36.1

Table 6: Ablation Study for different pre-training corpus, score under distractor setting.

Model	WebQA	Multimodal
MuRAG (Only In-Batch)	29.4	49.6
MuRAG (Only Fixed-Retrieval)	25.8	40.7
MuRAG (Two Stage)	31.5	51.4

Table 7: Ablation Study for different fine-tuning stages to see their contributions. WebQA uses the overall score, and MultimodalQA refers to EM-all score.

Evaluation	Model	Correct	Wrong
Distractor	MuRAG (Text)	80%	20%
	MuRAG (Image)	64%	36%
Full-Wiki	MuRAG (Text)	72%	28%
	MuRAG (Image)	54%	46%

Table 8: The human evaluation results on WebQA dataset separately for image/text queries.

refers to the model trained only with in-batch memory are directly used to generate outputs by accessing the global memory. Without further tuning, the performance will drop by roughly 2% on both datasets. (Only Fixed-Retrieval) refers to using the pre-trained retriever directly to obtain Top-K and then optimize the generative loss. As can be seen, the performance drop is more severe in this case for both datasets. This is understandable due the misalignment between pre-training retrieval is (image + text->text) while the fine-tuning retrieval is (text -> image+text). Thus, it is necessary to adapt the MuRAG's pre-trained retriever to different use cases depending on the downstream datasets.

4.6 Human Analysis

In order to better understand the model's performance, we manually study 200 model outputs and classify them into three categories and show our manual analysis results in Table 8. As can be seen, image queries are much harder than text queries. MuRAG only achieves 64% accuracy for the distractor setting and 54% accuracy for the full-wiki setting, falling significantly behind text accuracy.

We further categorize the image-query errors manually into the categories of Table 9. Counting is the most difficult question type, and constitutes 52% of the total errors, while object recognition errors rank second, constituting 29% of errors. In contrast, identifying color, shape, and gender is comparatively easier, with fairly low error rates. We demonstrate some correct and typical error cases in Figure 5 including miscounting and misrecognizing objects. We observe that these errors are mostly due to several reasons: 1) the question is related to infrequent objects, thus making recognition errors, 2) the image scene is highly complex with a large number of objects, thus grounding to a specific region is difficult, 3) the questions require optical character recognition ability from images. Hence, the bottleneck of MuRAG is still in the visual understanding module.

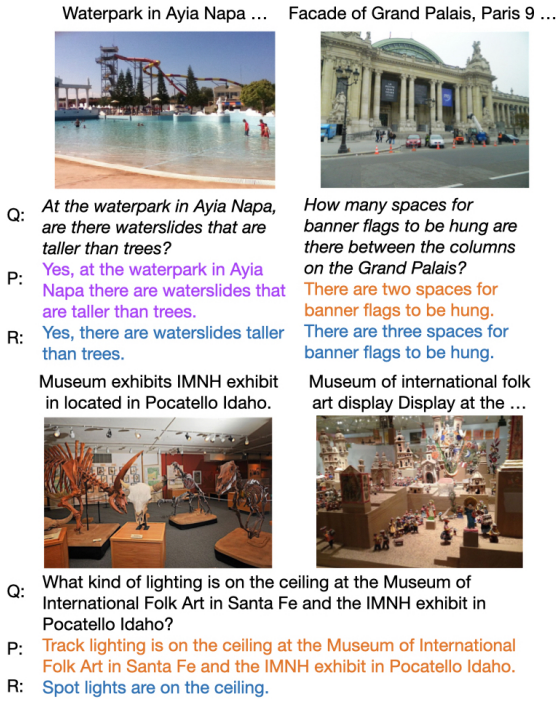

Figure 5: Upper left: correct prediction, Upper Right: error due to miscounting, Lower: error due to misrecognition (multiple image reasoning). Q refers to the question, P refers to prediction and R refers to the reference.

Category	Count	Object	Color	Shape	Gender
Ratio	52%	29.4%	5.8%	5.8%	5.8%

Table 9: Error categorization and their ratios on sampled WebQA-dev image queries.

5 Examples

We list more examples in Figure 6 and Figure 7. As can be seen, in the first example, the model is

grounded on the oracle image-text pair to make the correct prediction. However, in the second example, though the model retrieves the wrong image-text pair, it is able to make the correct prediction of 'the angel is holding a dead body'. We conjecture that the model utilizes textual clues to make the prediction rather than grounding on the image itself. Such shortcut learning is concerning and needs to be addressed through better learning algorithms.

Figure 6: Examples: we demonstrate model retrieval vs. groundtruth and model answer vs. reference.

6 Conclusion

In this paper, we build the first visually-grounded language generator capable of retrieving multimodal knowledge from a large-scale corpus. Our experiments show the promise of this approach, as it outperforms existing baselines by a large margin. At the same time, the performance on knowledge-seeking queries that require reasoning over images is still significantly lower than the performance on queries requiring only text. This indicates that there is still ample room for further improvements and we hope our study can motivate more research on better multimodal retrieval-augmented models.

Limitations

The current approach has several limitations: 1) since we do not mine hard negatives during pre-training, negatives come from other examples within the same batch. This requires that we set the batch size sufficiently large enough to collect hard-enough negatives. This results in the pre-training

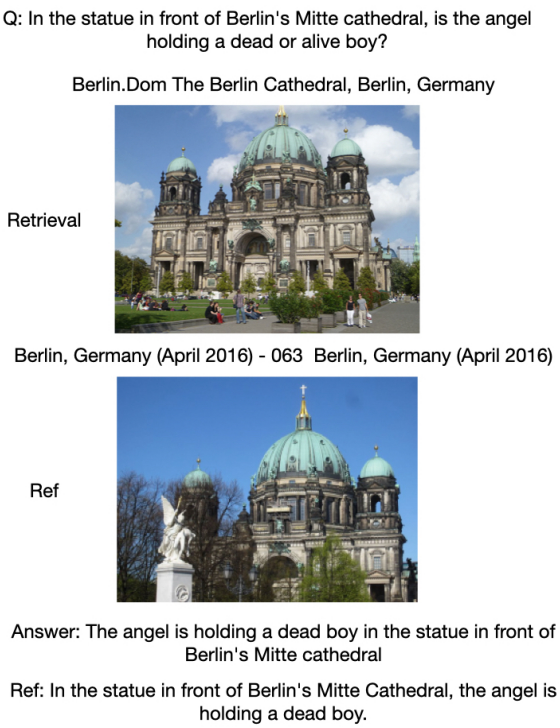

Figure 7: Examples: we demonstrate model retrieval vs. groundtruth, and model answer vs. reference.

requiring a large number of computation resources to reach competitive retrieval abilities. 2) our pre-training corpus's format (image -> text) is different from fine-tuning (text -> image+text). This misalignment limits the model's performance. Future work should consider how to design a better-aligned pre-training objective to achieve better performance. 3) Current visual representation in the reader stage is relatively expensive, i.e. 16x16=196 tokens per image, which poses great challenges for the transformer encoder to scale up to large Top-K values due to the quadratic attention complexity.

Ethical Statement

Our work uses the LAION dataset, a widely-used and publicly available large-scale visual-language corpus crawled from the web. The authors have conducted automatic filtering to greatly reduce harmful content. However, it is not possible to fully remove all of the potential risks from the data given its tremendous size. Being trained on this dataset, we anticipate our model to contain some biases (racial, gender, etc.). During our manual inspection, we saw some such biases, for example, 5% of errors are caused by misrecognition of gender. However, there are other many other forms of biases that we cannot fully enumerate or observe

explicitly.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. *The Conference on Computer Vision and Pattern Recognition*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568.

Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William Cohen. 2022. Augmenting pre-trained language models with qa-memory for open-domain question answering. *arXiv preprint arXiv:2204.04581*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

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 *European conference on computer vision*, pages 104–120. Springer.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William Cohen. 2022. Mention memory: incorporating textual knowledge into transformers through entity mention attention. *ICLR*.

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. In *International Conference on Learning Representations*.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896. PMLR.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Darryl Hannan, Akshay Jain, and Mohit Bansal. 2020. Manymodalqa: Modality disambiguation and qa over diverse inputs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7879–7886.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Norman P Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. 2020. A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, 63(7):67–78.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

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 *European conference on computer vision*, pages 740–755. Springer.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3195–3204.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Revanth Gangi Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, Shih-Fu Chang, et al. 2021. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. *arXiv preprint arXiv:2112.10728*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. 2022. Scaling up models and data with t5x and seqio. *arXiv preprint arXiv:2203.17189*.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

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.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Hrituraj Singh, Anshul Nasery, Denil Mehta, Aishwarya Agarwal, Jatin Lamba, and Balaji Vasan Srinivasan. 2021. Mimoqa: Multimodal input multimodal output question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5317–5332.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: complex question answering over text, tables and images. In *ICLR*.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Weston Cohen. 2021. Adaptable and interpretable neural memory over symbolic knowledge. In *Proceedings of NAACL-HLT*, pages 3678–3691.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022. Simvlm: Simple visual language model pretraining with weak supervision. *ICLR*.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

A Pre-training

During Pre-trainnig, we found that directly training with a mixture of all four datasets will lead to instability. We experimented with different variants and found that a scheduled pre-training can lead to a stable solution. We propose to first pre-train the model on the largest LAION dataset for 1M steps, and then continue training on the other three datasets with a fixed sample ratio. We plot the first stage of LAION training in Figure 8. We monitor the generation quality (LAION image -> text captioning), and the retrieval quality (image -> 4096 in-batch caption retrieval). As can be seen, the LAION pre-training converges after 1M steps, where we first warm up and then decrease the learning rate using a scheduler.

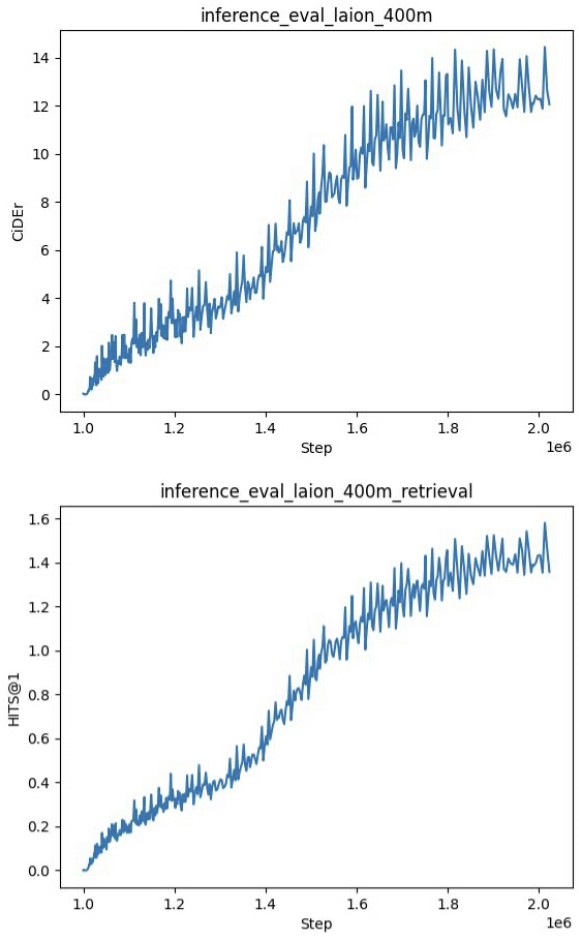

Figure 8: LAION Pre-training, validation accuracy, generation Cider score and retrieval recall score from the in-batch memory.

We further the pre-training on a mixture of the other three datasets. We plot their inference evaluation scores in Figure 9. We can see that the model is able to achieve very strong performance on these datasets, i.e. higher than 1.2 CiDEr on CC12M+3M validation set. The model also achieves strong performance on text-only reading comprehension on PAQ (similar to NQ), i.e. higher than 55% EM score. On the VQA dataset, the model is able to achieve higher than 72% VQA accuracy on the validation set. These results demonstrate the efficiency and multi-tasking capabilities of the pre-trained model. The overall retrieval accuracy from the multimodal memory consisting of captions, and passages are plotted in Figure 10, where the model is able to achieve 85% RECALL@1 from a 4K memory.

B Model Configuration

We demonstrate the ViT configuration as follows:

```
"vit_config": {
  "model": "ViT",
  "patches": {
    "size": [16, 16]
  },
  "hidden_size": 1024,
  "image_size": [224, 224],
  "num_heads": 16,
  "num_layers": 24,
  "mlp_dim": 4096,
  "return_pooled_output": false,
  "dropout_rate": 0.1
},
```

We demonstrate the T5-EncDec configuration as follows:

```
"model_config": {
  "vocab_size": 32128,
  "hidden_size": 768,
  "intermediate_dim": 2048,
  "num_attention_heads": 12,
  "memory_key_dim": 768,
  "encoder_layers": 12,
  "decoder_layers": 12,
  "dropout_rate": 0.1,
  "max_distance": 128,
  "num_buckets": 32,
  "scale": 1.0,
  "retrieval_weight": 0.5,
}
```

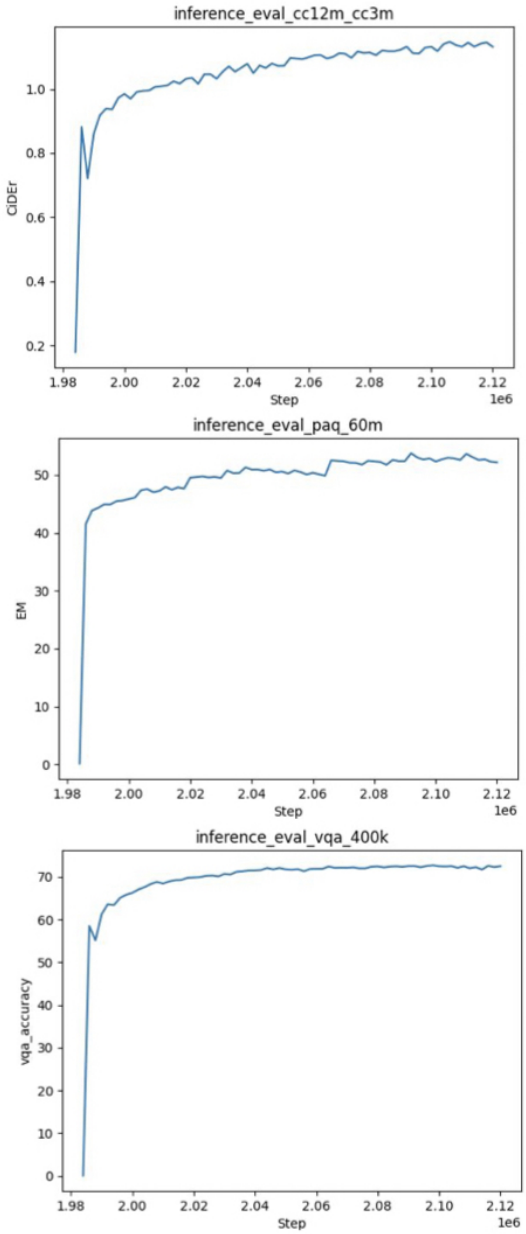

Figure 9: Mixture Pre-training, CiDEr, EM, and VQA accuracy for CC, PAQ, and VQA datasets.

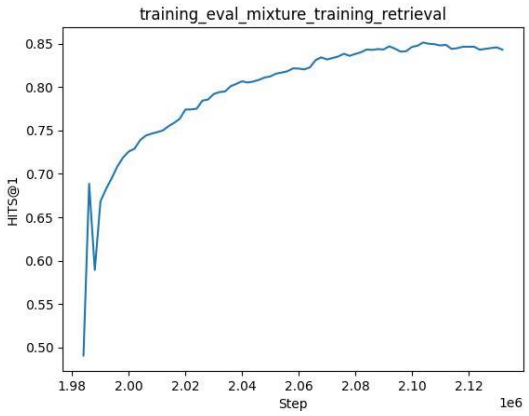

Figure 10: Mixture Pre-training retrieval accuracy over CC, PAQ, and VQA datasets.