MAGMA – Multimodal Augmentation of Generative Models through Adapter-based Finetuning

Constantin Eichenberg* and **Sidney Black*** and **Samuel Weinbach**
Aleph Alpha
{constantin.eichenberg, samuel.weinbach}@aleph-alpha.com, sdtblck@gmail.com

Letitia Parcalabescu and **Anette Frank**
Heidelberg University
{parcalabescu, frank}@cl.uni-heidelberg.de

Abstract

Large-scale pretraining is fast becoming the norm in Vision-Language (VL) modeling. However, prevailing VL approaches are limited by the requirement for labeled data and the use of complex multi-step pretraining objectives. We present MAGMA – a simple method for augmenting generative language models with additional modalities using adapter-based finetuning. Building on *Frozen* (Tsimpoukelli et al., 2021), we train a series of VL models that autoregressively generate text from arbitrary combinations of visual and textual input. The pretraining is entirely end-to-end using a single language modeling objective, simplifying optimization compared to previous approaches. Importantly, the language model weights remain unchanged during training, allowing for transfer of encyclopedic knowledge and in-context learning abilities from language pretraining. MAGMA outperforms Frozen on open-ended generative tasks, achieving state of the art results on the OKVQA benchmark and competitive results on a range of other popular VL benchmarks, while pretraining on $\sim 0.2\%$ of the number of samples used to train SimVLM (Wang et al., 2021).

1 Introduction

Self-supervised representation learning with transformer models (Vaswani et al., 2017) has become the dominant technique in Natural Language Processing in recent years, with encoder transformer models trained using a Masked Language Modeling (MLM) objective (Devlin et al., 2019) excelling at Natural Language Understanding tasks, and autoregressive decoder models (Radford et al., 2018, 2019; Brown et al., 2020) displaying impressive Natural Language Generation at increasingly large scales. Vision Language (VL) modeling – the modeling of joint image-text representations for tasks such as image captioning or visual question answering (VQA) – has followed suit, with the transformer encoder becoming the prevalent architecture in recent research. A popular approach among the latest state of the art VL models is to use a BERT-style encoder language model (LM) in combination with an object detection backbone such as Faster-RCNN (Ren et al., 2015). This approach, while displaying impressive performance on challenging benchmarks, has a number of drawbacks (see Section 2), in particular not being able solve VL tasks in an open-ended, generative fashion.

Figure 1: An example output produced by MAGMA. For this and all following examples the input text is displayed in black, and the model's response in green.

A recent line of work (Tsimpoukelli et al., 2021; Wang et al., 2021; Sollami and Jain, 2021) explores VL modeling using autoregressive decoder models trained with a language modeling objective. *SimVLM* (Wang et al., 2021) shows impressive performance, but requires prohibitively large amounts of pretraining data and the training of language and vision components in tandem. *Frozen* (Tsimpoukelli et al., 2021) shows that a pretrained autoregressive language model can, without any finetuning to the LM weights themselves, be harnessed to train a visual prefix which enables images to be used as its input. While the performance of *Frozen* on VL benchmarks falls short compared to the state of the art, we feel the approach is promis-

*Equal contribution

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2416–2428
December 7-11, 2022

ing due to its practicality, and the public availability of large, pretrained LMs such as GPT-J (Wang and Komatsuzaki, 2021), PanGu-α (Zeng et al., 2021), and GPT-Neo (Black et al., 2021).

Extending the *Frozen* approach, in this paper we introduce a framework to combine existing unimodal language and unimodal vision models pretrained on large web datasets into a powerful multimodal model. Specifically, our contributions are:

i) We introduce MAGMA: An autoregressive VL model that is able to generate text from an arbitrary combination of visual and textual input. Like *Frozen*, we start from a fixed large LM and a visual encoder-prefix stack. MAGMA differs from *Frozen* by additionally augmenting the LM with adapter layers, and using CLIP's (Radford et al., 2021) visual component as encoder. Only training the adapters and visual components, the method is parameter efficient and naturally retains the LM's encyclopedic knowledge and *in-context* learning abilities.

ii) Pretrained on a simple next token prediction objective, MAGMA is competitive in several VL downstream tasks, significantly outperforming its predecessor, *Frozen*, while pretraining on $\sim$0.2 % of the number of samples used for *SimVLM* (Wang et al., 2021). In particular, MAGMA achieves state of the art accuracy on the OKVQA benchmark, which we evaluate as a fully open-ended generative task.

iii) Our extensive ablations on the vision encoder and adapter components show i) that a pretrained CLIP ResNet encoder outperforms other visual backbones, ii) that an adapter-tuned model outperforms a visual prefix-only method, and iii) that different adapter configurations excel at different downstream tasks.

iv) We show that a carefully curated pretraining dataset – including around 25 million image-text pairs from a wide range of sources, including downstream task training data – can dramatically increase downstream performance when compared to a noisier, web-scraped dataset (CC12M Changpinyo et al. (2021a)).

We only explore the VL domain in this work, but we expect the general method of a modality-specific prefix in combination with adapter layers and a frozen LM to apply equally well to other combinations of modalities, such as audio-text pairs.

With this publication, we open-source our code and release a trained model checkpoint.[1]

2 Related Work

VL models of the past years (Zhang et al., 2021; Li et al., 2020; Chen et al., 2019; Li et al., 2019; Su et al., 2020; Tan and Bansal, 2019) harness a BERT-like encoder transformer as the language component, trained with a MLM objective – where random words in the input are masked out, and the model is tasked with predicting them. Encoder VL models are often also pretrained with auxiliary objectives or custom cross-modal losses, such as the Masked Region Modeling, Image-Text Matching and Word-Region Alignment of UNITER (Chen et al., 2019), or the contrastive loss of OSCAR (Li et al., 2020). Using auxiliary cross-modal loss functions and pretraining tasks complicates the pretraining procedure by requiring these losses to be properly balanced. Additionally, encoder models need extra task-specific finetuning for each task to perform effectively, limiting their accessibility. In comparison, autoregressive VL models like MAGMA are trained on a single, simple next token prediction objective, and can perform well on a wide range of tasks without further finetuning.

Two predecessors to our method are *Frozen* (Tsimpoukelli et al., 2021) and *SimVLM* (Wang et al., 2021), two autoregressive decoder models trained with a next token prediction language modeling objective. *Frozen* affixes an NFResnet (Brock et al., 2021) vision encoder to a pretrained autoregressive LM and, keeping the LM weights frozen, trains the vision encoder along with a *visual prefix* that linearly maps the output of the vision encoder to the dimensionality of the LM's token embeddings. *Frozen* shows that autoregressive VL models have the ability to adapt to examples *in-context*, like their language only counterparts (Brown et al., 2020), without performing any gradient updates. When shown multiple examples of a task in its context window in *Few-Shot learning*, its performance on that task improves, it appears to 'learn' from the presented examples without task-specific finetuning. Our model has similar *in-context* learning capabilities, but the addition of adapters and the different choice of visual backbone results in a model with improved performance when trained on a comparable dataset, see Section 3.3.

SimVLM is similar to *Frozen*, but pretrains the

[1] https://github.com/Aleph-Alpha/magma

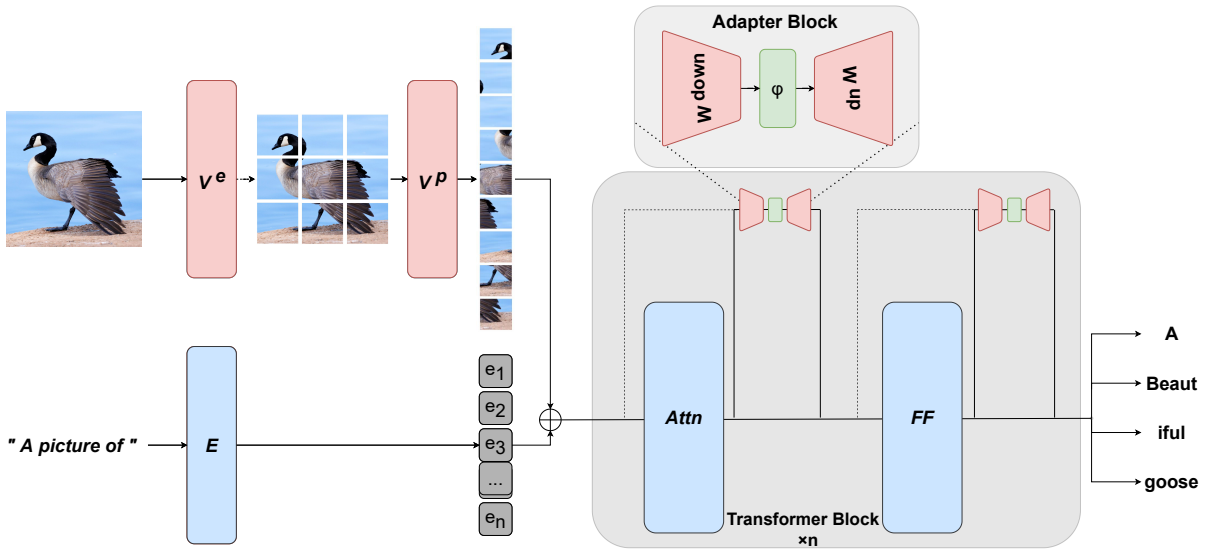

Figure 2: MAGMA's architecture. The layers in red are trained, and the layers in blue remain frozen.

vision and language components in tandem using a prefix LM objective. *SimVLM* consists of an encoder-decoder transformer with a combined ResNet (He et al., 2015) and ViT (Dosovitskiy et al., 2020) backbone as encoder, and a transformer decoder for language modeling. It extends the state of the art on a wide range of benchmarks, showing that a simple language modeling task can outperform MLM approaches. However, the joint pretraining requires prohibitively large uni- and multimodal datasets (1.8 Billion+ image-text pairs and ∼800GB of raw text), and long training times (∼4 Billion image-text pairs and ∼130 Billion text tokens). Aside from using orders of magnitude less data than *SimVLM*, MAGMA allows for the full recovery of the underlying LM's performance by simply removing the adapter layers.

Our work builds on recent advances in parameter efficient finetuning of LMs (Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021; He et al., 2021; Hu et al., 2021), specifically with adapter layers (Houlsby et al., 2019; He et al., 2021; Pfeiffer et al., 2020), which are small modules inserted in between the elements of a transformer layer which are finetuned instead of the model weights as a form of parameter efficient fine-tuning.

For a visual backbone, it is common to use region features from a pretrained object detection model such as Faster-RCNN (Ren et al., 2015). These are generally trained using expensive human labeled data on a bounded set of object classes, limiting the number of object types the resulting model can recognize. On the other hand, contrastive models such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) present a more robust approach to learning visual features by learning joint representations between image-text pairs. They show strong performance on a wide variety of vision tasks as well as impressive generalization abilities that can provide powerful semantic guidance to image generation (Esser et al., 2021). But since they were only trained to match image-text pairs, they cannot inherently be used for tasks that require text generation as output (Shen et al., 2021).

However, Shen et al. (2021) show that the weights of contrastive language-image models contain useful semantic information for VL tasks. By replacing the conventional region-based backbone with CLIP's visual encoder in popular VL architectures, the authors achieve SOTA results across a wide variety of VL tasks without needing region-based features, motivating us to use CLIP's visual component as a vision encoder for MAGMA. Notably, we confirm their finding that the ViT variant of CLIP underperforms on VL tasks when compared to the ResNet variant, particularly in tasks that require localization within an image.

3 Method

Our general approach is an image conditioned variant of soft-prompting or prefix tuning (Lester et al., 2021; Shin et al., 2020; Qin and Eisner, 2021) for language transformers and extends the *Frozen* method (Tsimpoukelli et al., 2021). The core idea is to translate image features into language embeddings carrying visual information which can therefore be interpreted by the language transformer without need to retrain the latter from scratch.

3.1 Architecture

The model can be broken down into four main components, see Figure 2. First, images are fed into a *Visual Encoder*, which processes the raw image input and outputs a sequence of feature vectors. Then an *Image Prefix* module maps image features into a sequence of embedding vectors that are input to the third model component, an auto-regressive *Language Model*. The fourth component is a series of *Adapter* layers which are inserted into the transformer LM, and tuned during training. We discuss the four components in more detail below.

Visual Encoder – V^e The visual encoder is a network used to extract condensed semantic information about an image. In principle, the visual encoder could take the form of any deep vision network whose output can be mapped to a sequence of embedding vectors. For our ablations, we use the visual backbone of several variants of CLIP. We also train a model with an NFResnet encoder trained from scratch, which is analogous to the model presented in *Frozen*, see §4.2. The visual encoder output is then passed into the *Image Prefix*.

Image Prefix – V^p Before the encoder output can be input to the LM, it needs to be translated into a sequence of n d_h-dimensional vectors, where d_h is the LM's hidden dimension. For the CLIP encoders, we extract the feature grid before the pooling layers, resulting in an $N \times N$ grid, where $N = 7, 7, 12$ for the ViT-B/32, RN50x4 and RN50x16 variants of CLIP respectively. We then flatten the feature grid into a sequence of N^2 vectors, and linearly transform the vectors' channel dimension to d_h. For the NFResnet variant, we follow the procedure described in *Frozen* by linearly transforming the output to $d_h \cdot n$, where n can be an arbitrary sequence length which we set to 2. Finally, we apply dropout regularization to the output of the image prefix, followed by Layer Normalization. We also explored non-linear variants of prefix mappings, replacing the linear transformation with an MLP and a transformer encoder, but found no improvements.

Language Model – E, T, H The language backbone of our architecture is initialized from a pre-trained auto-regressive transformer LM similar to GPT (Radford et al., 2018).

A text input y is converted into a sequence of tokens $t_1, ..., t_m$. Then a word embedding layer E maps each token t_k to a unique vector $e_k = E(t_k) \in \mathbb{R}^{d_h}$, obtaining a sequence of embeddings $e_1, ..., e_m$ which are input to a transformer-decoder module T with a causal attention mask. A language model head H maps the final output embeddings of the transformer to logits over the token space which can be used in a cross-entropy loss function for a next-token-prediction training objective and to auto-regressively generate text during inference. Because any sequence of vectors $v_1, ..., v_m \in \mathbb{R}^{d_h}$ can be used as input to the transformer, we can use images as input after mapping them through the encoder and the prefix as described above.

For the LM component, we use the open sourced weights of the 6 Billion parameter GPT-J (Wang and Komatsuzaki, 2021) LM. Since its architecture is largely similar to that described in Radford et al. (2018), we will not cover it in this paper, but do note two key differences of GPT-J compared to the original GPT architecture. Firstly, GPT-J replaces learned positional embeddings with rotary positional embeddings (Su et al., 2021), a form of relative positional embedding. As noted in (Tsimpoukelli et al., 2021), relative positional embeddings enable the transformer to generalize to inputs with more than one image, or a different image-text ordering compared to the training distribution, which is key to the VL model's ability to perform in-context learning with multiple image examples. Secondly, the attention layer and the feedforward layer are computed in parallel for decreased communication costs (Wang and Komatsuzaki, 2021).

Adapters – $\{A_i\}$ Adapters are a series of small modules placed in between elements of a transformer model (Houlsby et al., 2019), that can be finetuned instead of the model weights as a form of parameter efficient fine-tuning. We use the framework of He et al. (2021), where the adapter layers take the form of a scaled residual bottleneck MLP:

$$A_i(h) = h + \lambda_i W_i^{up} \varphi \left(W_i^{down} h \right). \quad (1)$$

The matrices $W^{down} \in \mathbb{R}^{d_b \times d_h}$ and $W^{up} \in \mathbb{R}^{d_h \times d_b}$ with $d_b < d_h$ constitute the bottleneck, φ is an activation function (in our case ReLU) and λ_i is a scaling parameter that is either trained or set equal to 1. We refer to the ratio d_h/d_b as the **downsample factor** of the adapter.

Given a set of adapters $\{A_i\}$ and a transformer module T, we denote the adapted version of T by $\tilde{T}$, which means replacing the attention and/or feed-forward blocks B_i of T by their adapted version $\tilde{B}_i$,

either obtained from adding the adapters in parallel or sequentially:

$$\tilde{B}_i \colon h \mapsto \begin{cases} B_i(h) + A_i(h) & \text{(parallel)} \\ B_i(h) + A_i(B_i(h)) & \text{(seq.)} \end{cases} \quad (2)$$

We experiment with both parallel and sequential adapter variants, see Section 4.2.1 for results.

3.2 Training

During training, the weights of the LM E, T, H remain unchanged, whereas the weights of the image encoder V^e, image prefix V^p and the adapters $\{A_i\}$ are optimized. The language model components are initialized with weights from the pretrained GPT-J model and the image encoder is initialized with pretrained CLIP weights except for the NFResnet ablation, where the image encoder is randomly initialized. The image prefix and adapters are always trained from scratch. In the following we denote the trainable parameters of a module by the subscript θ. As described in 3.1, a set of trainable adapters $\{A_{i,\theta}\}$ gives rise to the modified transformer module $\tilde{T}_\theta$.

The training objective is a captioning task: given an image-caption pair (x, y), we embed the image as $v_{1,\theta}, ..., v_{n,\theta} = V^p_\theta \circ V^e_\theta(x)$ and the text as $e_1, ..., e_m = E(t_1), ..., E(t_m)$, where $\{t_k\}$ is the tokenized caption y. Note that the image sequence length n is fixed while the length of the caption m is variable. The image embeddings are then prepended to the text embeddings and fed through the adapted transformer module. Denoting the embedding-to-logits function as $l_\theta = H \circ \tilde{T}_\theta$, we then compute the loss

$$L_\theta(x, y) = -\sum_{i=1}^{m} l_\theta(v_{1,\theta}, ..., v_{n,\theta}, e_1, ..., e_i), \quad (3)$$

where $l_\theta(v_{1,\theta}, ..., v_{n,\theta}, e_1, ..., e_i)$ is interpreted as next-token log-probability conditioned on the previous sequence elements

$$l_\theta(v_{1,\theta}, ..., v_{n,\theta}, e_1, ..., e_i) = \log p_\theta(t_i \mid x, t_1, ..., t_{i-1}). \quad (4)$$

For technical details regarding training, see A.

3.3 Dataset

For **pretraining** we use two different large scale datasets, one for the ablations and another one for our final model MAGMA_{base}, respectively MAGMA_{long}. For all **ablations** in 4.2 we train on CC12M (Changpinyo et al., 2021a) for a total of around 3M samples ensuring comparability with *Frozen*. Unfortunately, CC12M performs hypernyming, replacing people names with ⟨PERSON⟩. This causes downstream models to output ⟨PERSON⟩ overwhelmingly often, even when the inputs do not contain people or places.

This failure mode, as well as recent research suggesting that increased training dataset diversity improves downstream generalization capabilities (Zhang et al., 2021; Radford et al., 2021; Brown et al., 2020; Gao et al., 2021), prompted us to construct another large-scale pretraining dataset from various publicly available image-text datasets, including a heavily filtered subset of LAION (Schuhmann et al., 2021), Wikipedia Image-Text (Srinivasan et al., 2021), CC3M (Changpinyo et al., 2021b), Visual Genome (Krishna et al., 2016), Localized Narratives (Pont-Tuset et al., 2020).

Following research showing that LMs become strong zero-shot learners after being finetuned on collections of structured, task-based datasets (Wei et al., 2021; Sanh et al., 2021), we also include the training splits of the following downstream tasks: VQA (Antol et al., 2015), GQA (Hudson and Manning, 2019), OKVQA (Marino et al., 2019), VizWiz (Gurari et al., 2018), Hateful Memes (Kiela et al., 2020), CoCo Captions (Chen et al., 2015). This results in a dataset of around 25 million image-text pairs to train our final model, see §4.3.

4 Experiments and Analysis

To evaluate our methodology, we first train a series of ablations (cf. §4.2), to break down the effects of the vision encoder and adapter choice. We evaluate these ablations, and all subsequent models on a range of *visual question answering* and *image captioning* tasks designed to quantify the model's ability to adapt to new tasks using *in-context* learning, recognize a wide variety of objects, and reason in detail about an image – often involving complex spatial understanding, encyclopedic world knowledge, and optical character recognition (OCR).

4.1 Evaluations

4.1.1 Visual Question Answering (VQA)

VQA tasks require the model to answer a question about the input image. Breaking from previous works, which generally formulate VQA tasks as classification tasks over the most frequent re-

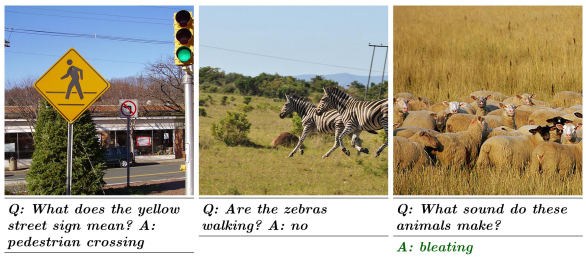

Figure 3: An example of a *2-shot* prompt for OKVQA.

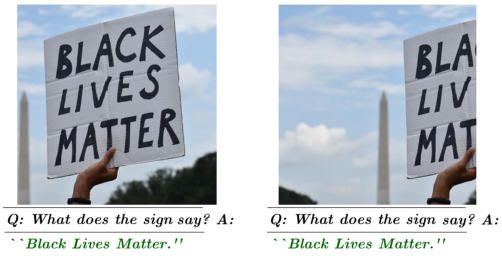

Figure 4: MAGMA's OCR capabilities. Even when text is obscured, MAGMA imputes the missing values.

sponses in the training set, we formulate all VQA tasks as open-ended generative tasks to enable few-shot prompting. We use the following datasets:
VQA 2.0 (Antol et al., 2015). A large and commonly used dataset for VQA where samples consist of an image, a question regarding the content of the image and 10 corresponding ground-truth answers.
OKVQA (Marino et al., 2019). A VQA dataset where correct answers require explicit outside world knowledge not contained in the picture.
GQA (Hudson and Manning, 2019). A large VQA dataset focusing on visual and spatial reasoning.
VizWiz (Gurari et al., 2018). A dataset in the same format as VQA with questions asked by visually impaired people. The ground-truth to a question about an image may be "unanswerable" or "unsuitable", which has to be recognized by the model.

To compare the generated model output with the provided ground-truths, we apply the normalization procedure of the official VQA 2.0 repo, [2] and truncate the model output to the length of the longest ground truth answer. For VQA, OKVQA, and VizWiz we calculate the accuracy metric from the official VQA paper (Antol et al., 2015), and for GQA we use the canonical accuracy score.

For few-shot settings, we use the procedure described in Tsimpoukelli et al. (2021), prepending n random examples of completed tasks before each question answer pair. We preprend "Q: " and "A: " to each question and answer respectively, improving performance (as exemplified in Figure 3).

4.1.2 Image Captioning

Image captioning tasks require the model to generate accurate descriptions of input images in natural language. We evaluate on two datasets – CoCo Captions (Chen et al., 2015) and NoCaps (Agrawal et al., 2019), measuring performance using the BLEU@4 and CIDEr metrics. NoCaps is designed to evaluate a model's ability to caption images containing uncommon or novel object classes that don't appear in CoCo.

Like with *SimVLM*, prompting with "A picture of" dramatically increases downstream scores, e.g. for MAGMA_{long}, increasing the CIDEr score on CoCo Captions from 7.5 to 57.1. All scores reported in Table 1 use this as a prefix. Other prefixes, such as "Caption:" have a similar effect.

4.1.3 Visual Entailment

We test Visual Entailment performance on SNLI-VE (Xie et al., 2018), a task built on top of SNLI (Bowman et al., 2015). SNLI-VE requires the model to reason about the relationship between an *image premise*, P_{image}, and a *text hypothesis*, H_{text}. Given P_{image} and H_{text} as input, the task is to label their relationship as either entailment, neutral or contradiction. We formulate SNLI-VE as a classification task by finetuning the model together with a linear classification head on the last-layer transformer embedding of the last text token.

4.2 Ablations

We run two series of ablations: i) One designed to test the impact of the adapter layers and their precise configuration, and ii) another designed to test the impact of the vision encoder choice. We also independently replicate the *Frozen* model (see Table 1), using the pretraining setup described in their paper (with the exception that we pretrain on CC12M) to use as a baseline. All ablations are trained for a total of 15k steps, or around 3.8 million image-text pairs.

4.2.1 Adapter Types

We run ablations with several different adapter configurations, motivated by He et al. (2021) showing that the precise formulation of the adapter layer can have a large impact on the performance of a model on downstream tasks. Also, different adapter layers can perform better than others depending on the task. Since an exhaustive sweep in the parameter space of adapters is very expensive, we decided

[2] https://github.com/GT-Vision-Lab/VQA

Adapter ablations					n-shot-VQA				n-shot-OKVQA				n-shot-GQA				n-shot-VizWiz				Avg.
Type	λ	Attn	FF	Params	0	1	2	4	0	1	2	4	0	1	2	4	0	1	2	4	
–	–	–	–	0.1	36.4	41.5	41.7	41.8	12.5	16.2	16.0	16.5	12.6	**20.8**	23.6	26.9	2.9	5.3	5.5	6.7	20.4
s	1	–	2	2	34.7	40.1	42.2	43.2	12.4	16.9	18.6	21.5	8.2	14.1	19.2	24.6	5.3	7.4	7.8	9.7	20.4
s	1	–	4	1	32.7	40.2	42.5	43.8	11.7	16.3	19.1	21.2	6.8	15.6	22.1	27.7	4.2	6.7	6.9	8.6	20.0
s	1	8	8	1	36.6	41.7	**43.8**	45.2	**13.9**	17.1	20.0	22.5	**14.3**	20.7	**24.9**	**28.4**	**5.6**	8.5	8.6	9.8	**22.6**
s	1	12	6	1	**36.9**	41.2	43.6	44.7	13.9	**19.4**	**21.6**	23.2	12.8	18.8	22.5	25.8	5.3	**9.6**	**9.8**	**10.6**	22.5
p	1	–	4	1	36.5	41.7	43.1	43.8	14.5	18.4	20.3	21.8	11.2	16.3	19.9	23.2	4.6	8.4	8.4	9.2	21.3
p	t	8	8	1	34.9	**42.2**	44.1	**45.4**	12.9	17.7	21.4	**23.4**	8.8	15.6	20.2	24.5	4.3	7.9	8.5	9.9	21.4
Encoder ablations																					
NFResnet					32.0	37.0	39.0	39.7	9.8	15.8	18.9	20.8	9.1	**20.2**	**27.1**	**28.7**	2.8	5.6	6.5	8.2	20.1
CLIP-ViT					32.8	33.9	36.7	37.7	10.5	9.2	12.4	14.2	8.4	14.9	22.2	25.7	2.7	5.1	5.2	7.7	17.5
CLIP-RN50x4					**35.2**	40.0	**42.6**	**44.2**	**12.6**	**17.7**	19.0	**21.8**	**10.5**	13.0	16.1	20.5	**5.0**	6.2	6.6	8.3	20.0
CLIP-RN50x16					32.7	**40.2**	42.5	43.8	11.7	16.3	**19.1**	21.2	6.8	15.6	22.1	27.7	4.2	**6.7**	**6.9**	**8.6**	**20.4**
Frozen (NFResnet + no adapters)					28.6	36.7	37.9	38.1	6.2	15.1	16.2	15.8	8.7	23.5	27.0	27.5	1.7	5.4	6.2	8.0	18.9
MAGMA pretrained																					
MAGMA$_{base}$					60.0	–	–	–	37.6	–	–	–	47.4	–	–	–	15.9	–	–	–	40.3
MAGMA$_{long}$					**61.5**	–	–	–	**40.3**	–	–	–	**49.6**	–	–	–	**16.7**	–	–	–	**42.0**

Adapter ablations					NoCaps - CIDEr				NoCaps - B@4				CoCo - CIDEr	CoCo - B@4	
Type	λ	Attn	FF	params	In	Out	Near	All	In	Out	Near	All			
–	–	–	–	0.1	45.1	53.7	43.3	45.7	9.9	5.8	7.9	7.8	36.7	10.3	
s	1	–	2	2	37.7	55.5	40.6	43.2	6.2	6.1	6.5	6.4	33.4	9.4	
s	1	–	4	1	39.3	56.2	44.0	45.8	6.3	6.7	7.7	7.3	39.6	11.2	
s	1	8	8	1	38.2	49.5	40.9	42.2	6.4	4.9	6.7	6.3	37.1	10.6	
s	1	12	6	1	**51.9**	**64.8**	**54.6**	**56.2**	**11.4**	**8.4**	**11.3**	**10.8**	**46.3**	**13.9**	
p	1	–	4	1	37.5	38.1	35.9	36.0	7.2	5.1	6.7	6.4	36.3	10.8	
p	t	8	8	1	40.6	58.3	45.0	47.1	8.0	6.6	7.9	7.7	39.5	11.2	
Encoder ablations															
NFResnet					22.5	16.2	22.0	20.9	5.0	1.6	5.3	4.5	22.4	8.2	
CLIP-ViT					33.2	44.2	35.3	36.8	5.9	5.2	5.8	5.7	27.2	7.7	
CLIP-RN50x4					**47.7**	43.6	**48.1**	**50.2**	**9.3**	6.7	**9.2**	**8.7**	**41.9**	**13.1**	
CLIP-RN50x16					39.3	**56.2**	44.0	45.8	6.3	**6.7**	7.7	7.3	39.6	11.2	
MAGMA pretrained															
MAGMA$_{base}$					55.8	56.5	49.9	52.1	11.1	6.1	10.3	9.5	51.1	15.8	
MAGMA$_{long}$					**58.1**	**62.0**	**56.9**	**58.1**	**13.3**	**8.5**	**13.2**	**12.3**	**57.0**	**17.6**	

Table 1: Performance evaluation on downstream tasks. Open-ended few-shot evaluation on VQA-val, OKVQA-val, GQA-testdev and VizWiz-val. Captioning evaluation on NoCaps-val and CoCo-val. Models under **MAGMA pretrained** are trained on the mixed dataset detailed in Section 3.3, all other models are trained on CC12M. Notation for adapter ablations. **Type**: (s)caled or (p)arallel. **λ**: 1 or (t)rained. **Attn, FF**: Downsample factor of the bottleneck in the resp. position. – means not applied. **Params**: Number of trainable parameters relative to the ablation with sequential FF adapters with downsample factor 4.

on seven configurations, including models with no adapters, to get a qualitative picture of the effect on downstream performance. We use the same visual encoder (CLIP 'RN50x16') for all adapter ablations and evaluate the open-ended few-shot scores on the VQA and Image Captioning tasks described in 4.1.1 and 4.1.2 respectively. The results are shown in Table 1. Although there is no adapter configuration which clearly outperforms the rest, we observe three key points:

Applying adapters to the attention layer is key. Adapter configurations with no adapters on the attention layer underperform, particularly at few shot prompting.

More adapter parameters to the feed forward layer increases performance on knowledge-based tasks. The adapter variant with more parameters allocated to the feed forward adapter outperforms other variants on OKVQA and NoCaps tasks requiring outside knowledge and uncommon object classes recognition respectively. This supports preliminary research indicating that the feed-forward blocks are important in storing implicit knowledge in pretrained transformers (Dai et al., 2021).

Balancing attention and feed-forward parameter allocation aids scene understanding. The adapter variant with equal number of parameters allocated to the attention and the feed forward adapters excels at the GQA benchmark, a QA benchmark built around scene graphs and designed to focus on skills such as spatial reasoning, comparisons, and object and attribute recognition.

4.2.2 Visual Encoders

We run ablations with four different image encoders: NFResnet, CLIP-ViT-B/32, CLIP-RN50x4 and CLIP-RN50x16. All visual encoder ablations are trained using the adapter configuration with sequential adapters on the feed-forward block and a downsample factor of 4. The results are shown in Table 1. Our findings are the following:

CLIP-RN50x16, on average, performs best

	VQA	OKVQA	GQA	VizWiz	SNLI-VE	NoCaps		Coco	
						CIDEr	B@4	CIDEr	B@4
MAGMA	68.0	**49.2**	54.5	35.4	79.0	93.6	27.8	91.2	31.4
SOTA	**75.5**	48.0	**72.1**	**54.7**	**86.3**	**112.2**	**33.1**	**143.3**	**41.7**
SOTA model	*SimVLM*	*PICa*	*CFR*	*Pythia*	*SimVLM*	*SimVLM*	*VIVO*	*SimVLM*	*OSCAR*

Table 2: MAGMA finetuned performance. **B@4**: NoCaps-all score. SOTA scores are to the best of our knowledge at the time of writing. If available/applicable, we compare to the SOTA score of models solving the task in an open-ended generative fashion like MAGMA (notably *SimVLM* on VQA), otherwise we compare to the general SOTA (classification setting). Models: *SimVLM* (Wang et al., 2021), *PICa* (Yang et al., 2021), *CFR* (Nguyen et al., 2021), *Pythia* (Singh et al., 2019), *VIVO* (Hu et al., 2020), *OSCAR* (Li et al., 2020).

Figure 5: An example of an adversarial *typographic attack* which MAGMA appears robust to, unlike CLIP.

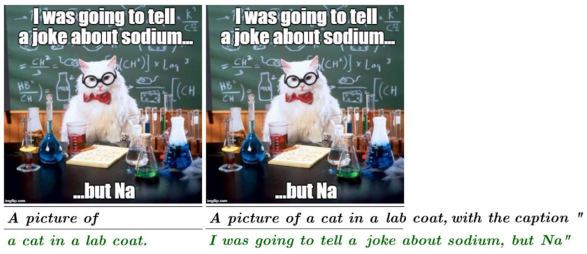

Figure 6: Example of *multi-step prompting*. Using the output of the model (left) again as the input (right), the generation procedure is broken down into atomic steps.

at VQA tasks. However, the difference between RN50x16 and RN50x4 is slight, with the smaller encoder performing better on VQA and OKVQA, while the larger encoder has a much higher GQA accuracy. We hypothesize that the increased resolution of the larger feature grid results in a more detailed scene understanding, while the smaller grid is better at condensing global visual information, which also shows in the Image Captioning scores, where CLIP-RN50x4 excels.

CLIP-ViT has the worst average score across question answering tasks. This reinforces the finding of Shen et al. (2021), who find that the CLIP-ViT model struggles at tasks which require localization within an image.

Recall that the image prefix length varies between image encoders which may have a confounding effect on the results – further study is needed to disentangle the effects of sequence length and the choice of the vision encoder.

4.3 Final Model

Based on our ablation studies, in particular the average VQA scores, we opt to train a final MAGMA model using the CLIP-RN50x16 encoder and sequential adapters with a downsample factor of 8 applied to the feed-forward and attention layers. We train on the dataset detailed in §3.3 and see that evaluation loss does not plateau after ∼3M samples as reported in *Frozen*, and so continue training, resulting in two model variants – MAGMA_{base} trained for 15k steps for comparability to *Frozen*, and MAGMA_{long} trained for 7.6M samples.

Due to the inclusion of training splits of tasks like VQA in the pretraining dataset, the performance of MAGMA_{base} significantly exceeds the downstream performance of previously trained ablations. The evaluation is conducted in the same way as the zero-shot procedure for the ablations and to avoid cluttered notation, we refer to it as such, although "zero-shot" usually refers to solving tasks unseen in pretraining. We stress that *the pretraining set and the eval sets are still disjoint*.

While the scores of MAGMA_{long} already surpass the VQA-finetuned variants reported in *Frozen*, we find that we can further increase the single-task performance on the training sets of each benchmark described in §4.1 by finetuning on them. After finetuning, MAGMA achieves competitive scores across all benchmarks, setting a new state of the art accuracy on OKVQA, as well as attaining strong scores on the NoCaps benchmark – to our knowledge, being surpassed only by *SimVLM* and *VinVL* (Zhang et al., 2021), see Table 2.

We include several **qualitative results**, which highlight strengths of the model we feel are not sufficiently reflected by the evaluations in Table 1. Notably, MAGMA appears to be less easily fooled by the adversarial *typographic attacks* to which CLIP is susceptible (Goh et al., 2021), see Figure 5. Additionally, MAGMA shows impressive OCR

capabilities even without supervised finetuning, see Figure 4, which warrants further quantitative evaluation. Interestingly, if a word or phrase is truncated, MAGMA can often impute the missing text. We also include an example of a multi-step *factored cognition* prompt (Mishra et al., 2021), see Figure 6, where a challenging task is broken down into atomic steps. We suspect that task decomposition may enable MAGMA to perform complex tasks that it would otherwise be unable to solve.

5 Conclusion

We propose a simple framework for Multimodal Augmentation of Generative Models through Adapter-based Finetuning – demonstrating that it is possible to transform multiple unimodal models into a powerful multimodal VL model while keeping the weights of the language component frozen. Our model, MAGMA, trained using adapter layers and a simple next token prediction objective, can perform competitively with state of the art VL models on a wide range of benchmarks, excelling at tasks requiring external knowledge and the recognition of uncommon object classes.

We hope our results will act as a starting point for further research into augmenting pretrained language models with additional modalities.

6 Limitations

Although the performance of MAGMA is impressive, we note some current limitations with the model and autoregressive VL models in general. Firstly, as we observed in the Image Captioning tasks, LMs can be sensitive to input – performance is heavily dependent on the prompt format.

Secondly, although the model can perform in-context learning with multiple examples in its context window, it struggles to reason over multiple images, as it was only pretrained on single image-caption pairs.

Finally, MAGMA shows similar capabilities to large LMs like GPT3, about which there are ongoing ethical concerns regarding their reproduction of biases from the training data, as well as concerns relating to how to effectively align their outputs to human goals. As such, further research into the reproduction of visual biases, and the guiding of model outputs is needed.

Acknowledgements

We would like to thank Mayukh Deb for his help with setting up and maintaining the public repository which makes MAGMA available to the research community. We would additionally like to thank the research team at Aleph Alpha for providing a stimulating and supportive environment.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. nocaps: novel object captioning at scale. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. 2021. High-performance large-scale image recognition without normalization. *CoRR*, abs/2102.06171.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021a. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *CoRR*, abs/2102.08981.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021b. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *CoRR*, abs/2102.08981.

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. *CoRR*, abs/1504.00325.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Universal image-text representation learning.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *CoRR*, abs/2104.08696.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. 2021. Multimodal neurons in artificial neural networks. *Distill*. Https://distill.pub/2021/multimodal-neurons.

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. *CoRR*, abs/1802.08218.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020. VIVO: surpassing human performance in novel object captioning with visual vocabulary pre-training. *CoRR*, abs/2009.13682.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *CoRR*, abs/2005.04790.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language.

Binh X. Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D. Tran, and Anh Nguyen. 2021. Coarse-to-fine reasoning for visual question answering.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. *CoRR*, abs/2007.07779.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *ECCV*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 5203–5212, Online. Best Short Paper Award.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2019. Zero: Memory optimization towards training A trillion parameter models. *CoRR*, abs/1910.02054.

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*, 28:91–99.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

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.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks?

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. *CoRR*, abs/1904.08920.

Michael Sollami and Aashish Jain. 2021. Multimodal conditionality for natural language generation.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. WIT: wikipedia-based image text dataset for multimodal multilingual machine learning. *CoRR*, abs/2103.01913.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *CoRR*, abs/2106.13884.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2018. Visual entailment task for visually-grounded language learning. *CoRR*, abs/1811.10582.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021. An empirical study of gpt-3 for few-shot knowledge-based vqa.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jianfeng Yu, Qi Guo, Yue Yu, Yan Zhang, Jin Wang, Hengtao Tao, Dasen Yan, Zexuan Yi, Fang Peng, Fangqing Jiang, Han Zhang, Lingfeng Deng, Yehong Zhang, Zhe Lin, Chao Zhang, Shaojie Zhang, Mingyue Guo, Shanzhi Gu, Gaojun Fan, Yaowei Wang, Xuefeng Jin, Qun Liu, and Yonghong Tian. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation.

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.

A Appendix: Training details

During pretraining for the ablations and all subsequent models, we update the parameters θ by minimizing the loss (3) per mini-batch using the Adam optimizer in combination with *ZeRO* (Rajbhandari et al., 2019) to parallelize gradients and optimizer states across devices. We train all models with a batch size of 256, a dropout probability of 0.1, a weight decay of 0, and use learning rates of $2 \cdot 10^{-6}$ for V_θ^e and $8 \cdot 10^{-4}$ for $(V_\theta^p, \{A_{i,\theta}\})$, annealing both to 10% of their original value using a cosine decay schedule throughout training. When finetuning on downstream tasks (see Section 4.3) we do early stopping based on validation loss, and use the same hyperparameters as above, aside from decreasing the learning rates for V_θ^e, $(V_\theta^p, \{A_{i,\theta}\})$ to $1.5 \cdot 10^{-6}$, $7 \cdot 10^{-4}$ for generative tasks and $1.5 \cdot 10^{-6}$, $3 \cdot 10^{-4}$ for SNLI-VE classification. We build our model using the PyTorch framework with Deepspeed for data-parallel training – training all ablations on 32 A100 GPUs for around 1.25 days each.