MorphVLM: Towards more Efficient and Robust Multimodal Vision Language Models

Chirag Khatri
ckhatri@usc.edu

Mihir Mangesh Pavuskar
pavuksar@usc.edu

Pothula Punith Krishna
pothulap@usc.edu

Prince Verma
princeve@usc.edu

Lavrenti Mikaelyan
mikaelya@usc.edu

1 Tasks Performed

1.1 Datasets being explored

We analyzed several datasets to finetune Flamingo(Alayrac et al., 2022) models and compare the score with the original model.

The Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) a benchmark dataset designed to measure knowledge acquired in zero-shot and few-shot settings. It covers 57 subjects across STEM, the humanities, the social sciences, history, law, and ethics. We analyzed the state-of-the-art models on MMLU subtasks, their compatibility with Flamingo model framework so that they could be used as substitute for the language component.

VLStereoSet (Zhou et al., 2022) is a vision-language probing dataset to measure stereotypical bias in vision-language models. We are using the dataset to detect and mitigate bias from the Flamingo model.

The SuperGLUE (Sarlin et al., 2020) dataset was discarded from the project due to lack of task relevance in favor of the **Multimodal C4 (mmC4)** (Zhu et al., 2023) dataset as it extends the C4 language dataset by interleaving images. This dataset is suitable for Flamingo training and testing as it focuses on and enhances the in-context few-shot learning capability of the Flamingo model.

We have also chosen to move away from the **MSCOCO** dataset (Lin et al., 2014) as we look to use **VQA (Visual Question Answering)**(Goyal et al., 2017) as our benchmark, a dataset comprising 265,016 images drawn from COCO and abstract scenes, each associated with a minimum of three open-ended questions.

1.2 Creating a baseline

The next engineering task involved setting up the datasets for evaluation. An evaluation script is being developed to receive predictions from Open Flamingo based on samples within the datasets, and output performance metrics. This script is a crucial component in assessing the model's capabilities and comparing our implementation's performance with the SOTA.

Next Steps: We are currently employing OpenAI CLIP ViT-L/14 (Ilharco et al., 2021) vision encoder and the MPT 1B RedPajama (Computer, 2023) language decoder to assess a benchmark performance on the above datasets. We will experiment performance by substituting with other small frozen models. The decision to avoid testing on larger frozen models was primarily due to the excessive computation times and GPU resource consumption associated with such experiments.

1.3 Efficient Fine-tuning

We will be utilizing **PEFT** Parameter-Efficient Fine-Tuning of Billion-Scale Models on Low-Resource Hardware (Mangrulkar et al., 2022) that has the following approaches:

LoRA: LoRA (Hu et al., 2021) approach involves reducing computation by approximating the updates to the LLM's weights with a low-rank decomposition matrix, significantly the number of trainable parameters and training times.

Adapter Layer: Introduced in CLIP Adapter (Gao et al., 2021), this method involves adding a few additional layers to the visual and language encoders and finetuning only these additional layers and freezing the rest of the models to achieve comparable performance to a model with all fine-tuned layers.

PEFT also introduces DeepSpeed (Li et al., 2022) for distributed training across various hardware.

2 Risks and Challenges

2.1 Infrastructure Limitations

Our experiments revealed the following challenges:

Model Details: We experimented with the small-

est flamingo model with 3B parameters - Open-Flamingo 3B Vision Model: ViT-L-14 (Ilharco et al., 2021), Language Model: MPT 1B Redpajama 200B (Computer, 2023)
Local Machines: Lacking GPU support, our local machines were inadequate for executing any model sizes, even the smallest Open-Flamingo 3B.
Colab Pro: This paid version successfully loaded the 3B Open-Flamingo model. Utilizing an Nvidia Tesla A100 GPU (as opposed to the T4 in Colab Free version). At a cost of $10 for 100 credits, this is projected to provide approximately 16 hours for experimentation.

2.2 Lack of compatible Datasets and Models

Our aim is to change the Flamingo architecture such that there is a performance boost in niche domains such as medical, law, etc. whilst minimizing bias.
Lack of Open-source Models: Unavailability of official open-source implementations
Large and Incompatible Models: Incompatibility due to difference in tokenization method and input formats as compared to the original Flamingo

2.3 Qualitative Performance

We compared the outputs of the pre-trained Open Flamingo (3B) model with the original Flamingo (80B) models. On comparing the outputs Fig. 1, we see that the Open Flamingo model generates decent output in the first example and an unrelated output in the third example. More extensive quantitative analyses will follow.

3 Plans to Mitigate

Each of the team members have some free allocated credits for GPU usage across different platforms and can utilize those for specific GPU-intensive tasks such as experimentation with larger frozen models and fine-tuning. One of the team members also has quite a capable GPU to run larger models or more complex computations. Thus, we plan to organize and coordinate the use of all of our available resources more strategically to minimize out-of-pocket expenses for **Colab Pro GPUs**. We are utilizing finetuning optimization methods introduced by **PEFT** such as **Adapter Layer** and **LoRA**. Upon preliminary testing, alternative versions of Flamingo such as **Mini-Flamingo** and **Tiny-Flamingo** are capable of running on most of our machines and Google Colab.

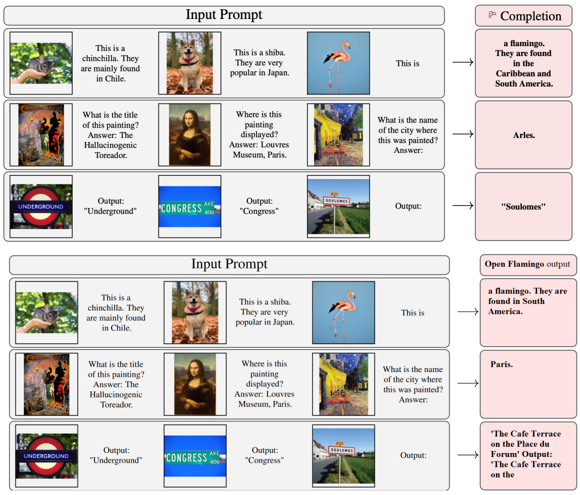

Figure 1: *Above*: The result of Flamingo 80B model from the original paper on 3 image-text examples. *Below*: The output from pre-trained Open Flamingo 3B model on the same inputs run on a A100 GPU machine on Colab Pro

A potential workaround to using alternative, currently-incompatible frozen models could be creating new or extending existing interfaces within the base **Open-Flamingo** (Awadalla et al., 2023) implementation such that the incompatibilities with alternative vision or language models.

4 Contributions

Mihir Mangesh Pavuskar:
Prepare MMLU dataset and analyze SOTA models
Experiment with CLIP Adapter methodology
Implement Mini-Flamingo and Tiny-Flamingo
Chirag Khatri:
Implement VLStereoset dataset for bias mitigation
Evaluation framework for benchmarking
Identify alternate VL components
Pothula Punith Krishna:
Explore and validate available implementations of LoRA
Identify and experiment with PEFT-LoRA
Acquire cheap GPU hardware - Azure ML
Lavrenti Mikaelyan:
VQAv2 Dataset Exploration and Preparation
Explore SuperGLUE, determined relevance, identified Multimodal C4 as better alternative
Implemente evaluation script on VQAv2 dataset
Prince Verma:
Experiment with Open Flamingo on local machine
Explore LoRA finetuning techniques
Identify alternate challenger models

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021. Clip-adapter: Better vision-language models with feature adapters.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

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

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

Conglong Li, Zhewei Yao, Xiaoxia Wu, Minjia Zhang, and Yuxiong He. 2022. Deepspeed data efficiency: Improving deep learning model quality and training efficiency via efficient data sampling and routing.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks.

Kankan Zhou, Eason Lai, and Jing Jiang. 2022. VL-StereoSet: A study of stereotypical bias in pre-trained vision-language models. Association for Computational Linguistics.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023. Multimodal c4: An open, billion-scale corpus of images interleaved with text.