Flamingo: 🦩
a Visual Language Model for Few-Shot Learning

Andrea Wynn and Xindi Wu
11/21/2022
Overview

Motivation

Flamingo Model Architecture

Training Data & Objective

In-Context Learning & Fine Tuning

Evaluation & Ablation Results

Limitations

Related Work: CM3 & Frozen

Discussion
Motivation

GPT-3  VIT  VisualBERT  CLIP  ?

Motivation

GPT-3   VIT   VisualBERT   CLIP   ?

Motivation

Motivation

GPT-3  VIT  VisualBERT  CLIP  ?

Motivation

GPT-3  VIT  VisualBERT  CLIP  ?

Motivation

GPT-3       VIT       VisualBERT       CLIP

Flamingo 🦩

The first vision-language model that has in-context learning ability
Motivation | Challenges

Challenges of multimodal generative modelling

- Unifying strong single-modal models
  - Interleave cross-attention layers with language only self-attention layers
Motivation | Challenges

<table>
<thead>
<tr>
<th>GPT-3</th>
<th>VIT</th>
<th>VisualBERT</th>
<th>CLIP</th>
<th>Flamingo</th>
</tr>
</thead>
</table>

Challenges of multimodal generative modelling

- Unifying strong single-modal models
  - Interleave **cross-attention** layers with language only self-attention layers
- Supporting images and videos
  - **Perceiver-based** architecture with a fixed number of visual tokens
Motivation | Challenges

- Unifying strong single-modal models
  - Interleave cross-attention layers with language only self-attention layers
- Supporting images and videos
  - Perceiver-based architecture with a fixed number of visual tokens
- Heterogeneous training data
  - Combine web scraping with existing image-text or video-text datasets.

Challenges of multimodal generative modelling

<table>
<thead>
<tr>
<th>GPT-3</th>
<th>VIT</th>
<th>VisualBERT</th>
<th>CLIP</th>
<th>Flamingo</th>
</tr>
</thead>
</table>

Inference (667x356 to 711x397)
Flamingo Overview

Separately trained image + language models, with novel layers in between
Input/Output

**Interleaved inputs:** text/images/video

**Outputs:** free-form text

Selected single image samples

**Input Prompt**

Question: What is the title of the book? Answer:

**Completion**

The House Book.

Selected dialogue samples

**Input Prompt**

Question: What is in this picture?

- It’s a bowl of soup with a monster face on it.

**Completion**

What is the monster made out of?

- It’s made out of vegetables.

**Completion**

No, it’s made out of a kind of fabric. Can you see what kind?

**Completion**

It’s made out of a woolen fabric.

Selected video samples.

**Input Prompt**

Question: What is happening here? Answer:

**Completion**

The dachshund puppy is being weighed on a scale.
Flamingo Overview

Separately trained image + language models, with novel layers in between
Flamingo Overview

Separately trained image + language models, with novel layers in between
Flamingo Overview

$$p(y|x) = \prod_{\ell=1}^{L} p(y_{\ell}|y<\ell, x\leq \ell),$$
Vision Encoder

Pretrained and frozen Normalizer Free ResNet (NFNet)

Perceiver Resampler
def perceiver_resampler:
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
    ):  

    """The Perceiver Resampler model."
    
    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f) # [T, S, d] -> [T*S, d]

    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)

    return x
Perceiver Resampler

```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
    """The Perceiver Resampler model."""
    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)
    return x
```
Perceiver Resampler

```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
    """The Perceiver Resampler model."""

    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]

    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)

    return x
```
Conditioning the Language Model
Gated XATTN-Dense layers
Gated XATTN-Dense layers

LM layer

GATED XATTN-DENSE

Vision input

Language input

self attention

FFW

tanh gating

cross attention

K=V=[X]  Q=[Y]

K=V=[Y]  Q=[Y]

Vision input X

Language input Y
Gated XATTN-Dense layers

```python
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn,  # xattn gating parameter - init at 0.
    alpha_dense,  # ffw gating parameter - init at 0.
):
    """Applies a GATED XATTN-DENSE layer."""

    # 1. Gated Cross Attention
    y = y + tanh(alpha_xattn) * attention(q=y, kv=x)

    # 2. Gated Feed Forward (dense) Layer
    y = y + tanh(alpha_dense) * ffw(y)

    # Regular self-attention + FFW on language
    y = y + frozen_attention(q=y, kv=y)
    y = y + frozen_ffw(y)

    return y  # output visually informed language features
```
Gated XATTN-Dense layers

```python
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter - init at 0.
    alpha_dense, # ffw gating parameter - init at 0.
):
    
    """Applies a GATED XATTN-DENSE layer."""

    # 1. Gated Cross Attention
    y = y + tanh(alpha_xattn) * attention(q=y, kv=x)

    # 2. Gated Feed Forward (dense) Layer
    y = y + tanh(alpha_dense) * ffw(y)

    # Regular self-attention + FFW on language
    y = y + frozen_attention(q=y, kv=y)
    y = y + frozen_ffw(y)

    return y  # output visually informed language features
```

Vision input X

Language input Y
Gated XATTN-Dense layers

```python
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn,  # xattn gating parameter - init at 0.
    alpha_dense,  # ffw gating parameter - init at 0.
)

    """Applies a GATED XATTN-DENSE layer."""

    # 1. Gated Cross Attention
    y = y + tanh(alpha_xattn) * attention(q=y, kv=x)

    # 2. Gated Feed Forward (dense) Layer
    y = y + tanh(alpha_dense) * ffw(y)

    # Regular self-attention + FFW on language
    y = y + frozen_attention(q=y, kv=y)
    y = y + frozen_ffw(y)

    return y  # output visually informed language features
```

Multi-Visual Input Support
Multi-Visual Input Support

Processed text: <image> tags are inserted and special tokens are added
Multi-Visual Input Support

Input webpage

Processed text: <image> tags are inserted and special tokens are added

My puppy sitting in the grass.

My cat looking very dignified.
Multi-Visual Input Support
Pre-Lecture Question

Describe how Flamingo handles input sequences of arbitrarily interleaved textual and visual data, and combines pre-trained text-only and vision-only models.

Answer:

For example, the input contains an image of a dog together with a text description and an image of a cat with an incomplete text description. The text is parsed from the input with images replaced with placeholders the images are also extracted from the input passed through a frozen vision encoder and then mapped through the perceiver resampler to produce a fixed number of visual tokens per input.
Training Data
Mixture of Datasets

- N: Number of visual inputs for a single example
- T: Number of video frames
- H, W, C: height, width, color channels
Interleaved Image/Text: MultiModal MassiveWeb (M3W)

- Interleaved text and image training data
- Compiled from webpage HTML
- Randomly sample 256 token subsequence and extract first 5 images

Example:

Multi-Modal Massive Web (M3W) dataset

\[N>1, \ T=1, \ H, \ W, \ C\]
Image-Text Pairs: ALIGN

“motorcycle front wheel”  “thumbnail for version as of 21 57 29 june 2010”  “file frankfurt airport skyline 2017 05 jpg”

“file london barge race 2 jpg”  “moustache seamless wallpaper design”  “st oswalds way and shops”

Image-Text Pairs: Long Text & Image Pairs (LTIP)

This is an image of a flamingo.
Video & Text Pairs (VTP)

Input Prompt

Question: What is happening here? Answer:

Completion

The dachshund puppy is being weighed on a scale.

A sword.
Data Augmentation & Preprocessing

- Visual inputs resized to 320x320
- M3W Data Augmentation: Randomizing image placement

(a) This is my dog! <dog image>       This is my cat! <cat image>
(b) <dog image> That was my dog!      <cat image> That was my cat!
Training Objective

\[
\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y) \sim D_m} \left[ - \sum_{\ell=1}^{L} \log p(y_{\ell} | y_{< \ell}, x_{\leq \ell}) \right]
\]

- Weighted sum of dataset specific expected negative log likelihood of text, given some visual inputs
- AdamW optimizer
- No weight decay for Perceiver Resampler
- Weight decay of 0.1 for other parameters
Pre-Lecture Question

Describe what datasets are used for mixed training. How important is each type of dataset empirically?

Answer:

Datasets - M3W (interleaved images and text), ALIGN (large, lower quality image + text pairs), LTIP (image + text pairs), VTP (video + text pairs)

Importance (lambda weights) - 1.0 (M3W), 0.2 (ALIGN), 0.2 (LTIP), 0.03 (VTP)

Number of datasets (M) - 4
Flamingo Evaluation
## Benchmark Tasks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DEV</th>
<th>Gen.</th>
<th>Custom prompt</th>
<th>Task description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet-1k [94]</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>Object classification</td>
</tr>
<tr>
<td>MS-COCO [15]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Scene description</td>
</tr>
<tr>
<td>VQAv2 [3]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Scene understanding QA</td>
</tr>
<tr>
<td>OKVQA [69]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>External knowledge QA</td>
</tr>
<tr>
<td>Flickr30k [139]</td>
<td></td>
<td></td>
<td>✓</td>
<td>Scene description</td>
</tr>
<tr>
<td>VizWiz [35]</td>
<td>✓</td>
<td></td>
<td></td>
<td>Scene understanding QA</td>
</tr>
<tr>
<td>TextVQA [100]</td>
<td></td>
<td></td>
<td>✓</td>
<td>Text reading QA</td>
</tr>
<tr>
<td>HatefulMemes [54]</td>
<td></td>
<td>✓</td>
<td></td>
<td>Meme classification</td>
</tr>
<tr>
<td>Video</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kinetics700 2020 [102]</td>
<td>✓</td>
<td></td>
<td></td>
<td>Action classification</td>
</tr>
<tr>
<td>VATEX [122]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Event description</td>
</tr>
<tr>
<td>MSVDQA [130]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Event understanding QA</td>
</tr>
<tr>
<td>YouCook2 [149]</td>
<td></td>
<td></td>
<td>✓</td>
<td>Event description</td>
</tr>
<tr>
<td>MSRVTITQA [130]</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Event understanding QA</td>
</tr>
<tr>
<td>iVQA [135]</td>
<td>✓</td>
<td></td>
<td></td>
<td>Event understanding QA</td>
</tr>
<tr>
<td>RareAct [73]</td>
<td></td>
<td></td>
<td>✓</td>
<td>Composite action retrieval</td>
</tr>
<tr>
<td>NextQA [129]</td>
<td>✓</td>
<td></td>
<td></td>
<td>Temporal/Causal QA</td>
</tr>
<tr>
<td>STAR [128]</td>
<td></td>
<td></td>
<td>✓</td>
<td>Multiple-choice QA</td>
</tr>
</tbody>
</table>
Benchmark Tasks: ImageNet-1k

Benchmark Tasks: Visual Question Answering (VQA)

Benchmark Tasks: Kinetics700 2020

- Taken from YouTube videos
- Format: label, youtube_id, start time, end time

<table>
<thead>
<tr>
<th>label</th>
<th>youtube_id</th>
<th>time_start</th>
<th>time_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>clay pottery making</td>
<td>--0dWlqvvl</td>
<td>19</td>
<td>29</td>
</tr>
<tr>
<td>javelin throw</td>
<td>--07WQ2iBlw</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>climbing a rope</td>
<td>--0NTAs-fA0</td>
<td>29</td>
<td>39</td>
</tr>
<tr>
<td>sipping cup</td>
<td>--0l35AkU34</td>
<td>68</td>
<td>78</td>
</tr>
<tr>
<td>flipping pancake</td>
<td>--33Lscn6sk</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>tickling</td>
<td>--3OAsUWtU</td>
<td>45</td>
<td>55</td>
</tr>
</tbody>
</table>

Benchmark Tasks: MSVDQA

Q: what is a man with long hair and a beard is playing?  
A: guitar

Q: what are two people doing?  
A: dance

Q: what are some guys playing in a ground?  
A: football

Q: who talks to judges?  
A: girl

Q: what is a kid doing stunts on?  
A: motorcycle

Q: what is a dog doing?  
A: swim

Q: what is a man using to slice up small pieces of meat for cooking?  
A: knife

Q: what is a batter doing?  
A: hit

## Classification Task Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>Prompt size</th>
<th>shots/class</th>
<th>ImageNet top 1</th>
<th>Kinetics700 avg top1/5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SotA</td>
<td>Fine-tuned</td>
<td>-</td>
<td>full</td>
<td>90.9 [127]</td>
<td>89.0 [134]</td>
</tr>
<tr>
<td>SotA</td>
<td>Contrastive</td>
<td>-</td>
<td>0</td>
<td>85.7 [82]</td>
<td>69.6 [85]</td>
</tr>
<tr>
<td>NFNetF6</td>
<td>Our contrastive</td>
<td>-</td>
<td>0</td>
<td>77.9</td>
<td>62.9</td>
</tr>
<tr>
<td>Flamingo-3B</td>
<td>RICES</td>
<td>8</td>
<td>1</td>
<td>70.9</td>
<td>55.9</td>
</tr>
<tr>
<td>Flamingo-3B</td>
<td>RICES</td>
<td>16</td>
<td>1</td>
<td>71.0</td>
<td>56.9</td>
</tr>
<tr>
<td>Flamingo-3B</td>
<td>RICES</td>
<td>16</td>
<td>5</td>
<td>72.7</td>
<td>58.3</td>
</tr>
<tr>
<td>Flamingo-9B</td>
<td>RICES</td>
<td>8</td>
<td>1</td>
<td>71.2</td>
<td>58.0</td>
</tr>
<tr>
<td>Flamingo-9B</td>
<td>RICES</td>
<td>16</td>
<td>1</td>
<td>71.7</td>
<td>59.4</td>
</tr>
<tr>
<td>Flamingo-9B</td>
<td>RICES</td>
<td>16</td>
<td>5</td>
<td>75.2</td>
<td>60.9</td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td>16</td>
<td>≤ 0.02</td>
<td>66.4</td>
<td>51.2</td>
</tr>
<tr>
<td>Flamingo-80B</td>
<td>RICES</td>
<td>8</td>
<td>1</td>
<td>71.9</td>
<td>60.4</td>
</tr>
<tr>
<td>Flamingo-80B</td>
<td>RICES</td>
<td>16</td>
<td>1</td>
<td>71.7</td>
<td>62.7</td>
</tr>
<tr>
<td>Flamingo-80B</td>
<td>RICES</td>
<td>16</td>
<td>5</td>
<td>76.0</td>
<td>63.5</td>
</tr>
<tr>
<td>RICES+ensembling</td>
<td></td>
<td>16</td>
<td>5</td>
<td>77.3</td>
<td>64.2</td>
</tr>
</tbody>
</table>
## Fine Tuning Results

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA2</th>
<th>COCO</th>
<th>VATEX</th>
<th>VizWiz</th>
<th>MSRVTQA</th>
<th>VisDialog</th>
<th>YouCook2</th>
<th>TextVQA</th>
<th>HatefulMemes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test-dev</td>
<td>test</td>
<td>test</td>
<td>test-dev</td>
<td>test</td>
<td>valid</td>
<td>test</td>
<td>valid</td>
<td>test</td>
</tr>
<tr>
<td>Flamingo - 32 shots</td>
<td>67.6</td>
<td>-</td>
<td>113.8</td>
<td>65.1</td>
<td>49.8</td>
<td>-</td>
<td>31.0</td>
<td>56.8</td>
<td>-</td>
</tr>
<tr>
<td>SimVLM [124]</td>
<td>80.0</td>
<td>80.3</td>
<td>143.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OFA [119]</td>
<td>79.9</td>
<td>80.0</td>
<td>149.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Florence [140]</td>
<td>80.2</td>
<td>80.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Flamingo Fine-tuned</strong></td>
<td><strong>82.0</strong></td>
<td><strong>82.1</strong></td>
<td><strong>138.1</strong></td>
<td><strong>84.2</strong></td>
<td><strong>65.7</strong></td>
<td><strong>65.4</strong></td>
<td><strong>47.4</strong></td>
<td><strong>61.8</strong></td>
<td><strong>59.7</strong></td>
</tr>
<tr>
<td>Restricted SotA†</td>
<td>80.2</td>
<td>80.4</td>
<td>143.3</td>
<td>76.3</td>
<td>-</td>
<td>-</td>
<td>46.8</td>
<td><strong>75.2</strong></td>
<td><strong>74.5</strong></td>
</tr>
<tr>
<td>[140]</td>
<td>[140]</td>
<td>[124]</td>
<td>[153]</td>
<td>-</td>
<td>[51]</td>
<td>[79]</td>
<td>[79]</td>
<td>[132]</td>
<td>[137]</td>
</tr>
<tr>
<td>Unrestricted SotA</td>
<td>81.3</td>
<td>81.3</td>
<td>149.6</td>
<td>81.4</td>
<td>57.2</td>
<td>60.6</td>
<td>-</td>
<td>75.4</td>
<td>-</td>
</tr>
<tr>
<td>[133]</td>
<td>[133]</td>
<td>[119]</td>
<td>[153]</td>
<td>[65]</td>
<td>[65]</td>
<td>[123]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
## Model Scaling

<table>
<thead>
<tr>
<th>Model</th>
<th>Requires model sharding</th>
<th>Frozen Language</th>
<th>Frozen Vision</th>
<th>Trainable Gated</th>
<th>Trainable XATTN-DENSE</th>
<th>Trainable Resampler</th>
<th>Total count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flamingo-3B</td>
<td>✗</td>
<td>1.4B</td>
<td>435M</td>
<td>1.2B (every)</td>
<td>194M</td>
<td></td>
<td><strong>3.2B</strong></td>
</tr>
<tr>
<td>Flamingo-9B</td>
<td>✗</td>
<td>7.1B</td>
<td>435M</td>
<td>1.6B (every 4th)</td>
<td>194M</td>
<td></td>
<td><strong>9.3B</strong></td>
</tr>
<tr>
<td>Flamingo</td>
<td>✓</td>
<td>70B</td>
<td>435M</td>
<td>10B (every 7th)</td>
<td>194M</td>
<td></td>
<td><strong>80B</strong></td>
</tr>
</tbody>
</table>
Number of Shots

![Graph showing aggregated performance across different numbers of shots for Flamingo-80B, Flamingo-9B, and Flamingo 3B.](image)
Ablation Studies
Ablation Studies

<table>
<thead>
<tr>
<th>Ablated setting</th>
<th>Flamingo-3B original value</th>
<th>Changed value</th>
<th>Overall score↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) Training data</td>
<td>All data</td>
<td>w/o Video-Text pairs</td>
<td>67.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w/o Image-Text pairs</td>
<td>60.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image-Text pairs → LAION</td>
<td>66.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w/o M3W</td>
<td>53.4</td>
</tr>
<tr>
<td>(ii) Optimisation</td>
<td>Accumulation</td>
<td>Round Robin</td>
<td>62.9</td>
</tr>
<tr>
<td>(iii) Tanh gating</td>
<td>✓</td>
<td>✗</td>
<td>66.5</td>
</tr>
<tr>
<td>(iv) Cross-attention architecture</td>
<td>GATED XATTN-DENSE</td>
<td>VANILLA XATTN GRAFTING</td>
<td>66.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>63.1</td>
</tr>
<tr>
<td>(v) Cross-attention frequency</td>
<td>Every</td>
<td>Single in middle</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Every 4th</td>
<td>68.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Every 2nd</td>
<td>68.2</td>
</tr>
<tr>
<td>(vi) Resampler</td>
<td>Perceiver</td>
<td>MLP Transformer</td>
<td>66.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>66.7</td>
</tr>
<tr>
<td>(vii) Vision encoder</td>
<td>NFNet-F6</td>
<td>CLIP ViT-L/14 NFNet-F0</td>
<td>64.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.7</td>
</tr>
<tr>
<td>(viii) Freezing LM</td>
<td>✓</td>
<td>✗ (random init)</td>
<td>57.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✗ (pretrained)</td>
<td>62.7</td>
</tr>
</tbody>
</table>
# Pre-Training Dataset Ablation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Combination strategy</th>
<th>ImageNet accuracy top-1</th>
<th>COCO image-to-text</th>
<th>COCO text-to-image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>LTIP</td>
<td>None</td>
<td>40.8</td>
<td>38.6</td>
<td>66.4</td>
</tr>
<tr>
<td>ALIGN</td>
<td>None</td>
<td>35.2</td>
<td>32.2</td>
<td>58.9</td>
</tr>
<tr>
<td>LTIP + ALIGN</td>
<td>Accumulation</td>
<td>45.6</td>
<td>42.3</td>
<td>68.3</td>
</tr>
<tr>
<td>LTIP + ALIGN</td>
<td>Data merged</td>
<td>38.6</td>
<td>36.9</td>
<td>65.8</td>
</tr>
<tr>
<td>LTIP + ALIGN</td>
<td>Round-robin</td>
<td>41.2</td>
<td>40.1</td>
<td>66.7</td>
</tr>
</tbody>
</table>
# Frozen Language Model

<table>
<thead>
<tr>
<th>Ablated setting</th>
<th>Flamingo 3B value</th>
<th>Changed value</th>
<th>Overall score↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flamingo 3B model (short training)</strong></td>
<td></td>
<td></td>
<td><strong>70.7</strong></td>
</tr>
<tr>
<td>(i) Resampler size</td>
<td>Medium</td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>(ii) Multi-Img att.</td>
<td>Only last</td>
<td>All previous</td>
<td>63.5</td>
</tr>
<tr>
<td>(iii) $p_{next}$</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>(iv) LM pretraining</td>
<td>MassiveText</td>
<td>C4</td>
<td>62.8</td>
</tr>
<tr>
<td>(v) Freezing Vision</td>
<td>✓</td>
<td>✓ (random init)</td>
<td>✓ (pretrained)</td>
</tr>
<tr>
<td>(vi) Co-train LM on MassiveText</td>
<td>✗</td>
<td>✓ (random init)</td>
<td>✓ (pretrained)</td>
</tr>
<tr>
<td>(vii) Dataset and Vision encoder</td>
<td>M3W+1TF+VTP and NFNetF6</td>
<td>LAION400M and CLIP</td>
<td>M3W+LAION400M+VTP and CLIP</td>
</tr>
</tbody>
</table>
0-initialized tanh gating

<table>
<thead>
<tr>
<th>Ablated setting</th>
<th>Flamingo-3B original value</th>
<th>Changed value</th>
<th>Overall score↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training data</td>
<td>All data</td>
<td>w/o Video-Text pairs</td>
<td>67.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w/o Image-Text pairs</td>
<td>60.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image-Text pairs → LAIO</td>
<td>66.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w/o M3W</td>
<td>53.4</td>
</tr>
<tr>
<td>Optimisation</td>
<td>Accumulation</td>
<td>Round Robin</td>
<td>62.9</td>
</tr>
<tr>
<td>(iii) Tanh gating</td>
<td>√</td>
<td>×</td>
<td>66.5</td>
</tr>
<tr>
<td>Cross-attention architecture</td>
<td>GATED XATTN-DENSE</td>
<td>VANILLA XATTN GRAFTING</td>
<td>66.9</td>
</tr>
<tr>
<td>Cross-attention frequency</td>
<td>Every</td>
<td>Single in middle</td>
<td>59.8</td>
</tr>
<tr>
<td>Resampler</td>
<td>Perceiver</td>
<td>MLP Transformer</td>
<td>66.6</td>
</tr>
</tbody>
</table>
Failures: Hallucinations

Input Prompt

Question: What is on the phone screen? Answer:

Question: What can you see out the window? Answer:

Question: Whom is the person texting? Answer:

Output

A text message from a friend.

A parking lot.

The driver.
Survey of Visual LMs
CM3

- Causally Masked Multimodal Modeling

Learning Image Embeddings on Frozen LM Prefix

- Multimodal few shot learning for interleaved vision and text

Discussion

If you are going to build a visual LM for few-shot learning, what are the other ways of fusing visual and textual data? What pre-training data would you consider?