**Flamingo: A Visual Language Model for Few-Shot Learning (NeurIPS 2022)**

**https://chatgpt.com/share/67e45b14-d214-8005-9357-f41b66d180a9**

**Method Summary:** Flamingo is a **large-scale multimodal model (80B parameters)** that combines a pretrained vision encoder and a pretrained language model to tackle image+text tasks in a few-shot manner​

[openreview.net](https://openreview.net/forum?id=EbMuimAbPbs#:~:text=%28VLM%29%20with%20this%20ability,answering)

. A key innovation is a **two-stage architecture**: a **Perceiver Resampler** module takes the raw visual features (e.g. from a ConvNet or ViT) and compresses them into a small set of latent visual tokens, which are then fed into a frozen language model through **gated cross-attention** layers​

[medium.com](https://medium.com/@paluchasz/understanding-flamingo-visual-language-models-bea5eeb05268#:~:text=representations%20produced%20by%20the%20Perceiver,architecture%20can%20be%20visualised%20below)

​

[medium.com](https://medium.com/@paluchasz/understanding-flamingo-visual-language-models-bea5eeb05268#:~:text=match%20at%20L436%20Vision%20Encoder,Language%20Model%20and%20generate%20text)

. In practice, Flamingo interleaves the language model’s layers with special multimodal layers that allow the model to attend to images. This design **bridges powerful pretrained unimodal models** – the vision backbone (e.g. EfficientNet) processes images, and the text backbone (e.g. a large Transformer like Chinchilla) processes language – by connecting them with trainable cross-attention blocks​

[openreview.net](https://openreview.net/forum?id=EbMuimAbPbs#:~:text=%28VLM%29%20with%20this%20ability,answering)

. Flamingo is trained on a vast collection of interleaved image-text data, enabling it to perform **in-context learning**: given just a few example QA pairs (as text) and a new image and question, Flamingo can generate a high-quality answer **without finetuning**. It achieved state-of-the-art few-shot performance on VQA and other tasks, even outperforming models fine-tuned on thousands of examples​

[openreview.net](https://openreview.net/forum?id=EbMuimAbPbs#:~:text=video%20tasks.%20These%20include%20open,specific%20data)

. In summary, Flamingo’s architecture fuses modalities by injecting visual information into a language model, making it extremely flexible for zero/few-shot VQA and captioning.

**Dataset:** The Flamingo paper evaluates the model on general VQA benchmarks like **VQAv2** and **OK-VQA** to demonstrate its few-shot capabilities​

[openreview.net](https://openreview.net/pdf?id=mWVoBz4W0u#:~:text=VQAv2%20OKVQA%20TextVQA%20VizWiz,)

​

[openreview.net](https://openreview.net/pdf?id=mWVoBz4W0u#:~:text=accuracy,suggest%20that%20leveraging%20external%20knowledge)

. Notably, Flamingo-80B achieved about 82% accuracy on VQAv2, which is on par with state-of-the-art models that were fully fine-tuned​

[openreview.net](https://openreview.net/pdf?id=mWVoBz4W0u#:~:text=Flamingo%20%2880B%29%2082,)

. The **VQAv2 dataset** (described above) is publicly available​

[visualqa.org](https://visualqa.org/download.html#:~:text=%40InProceedings,2017%7D%2C)

. OK-VQA is a challenging VQA dataset requiring external knowledge, also openly available (hosted by the authors of OK-VQA). For instance, OK-VQA’s images/questions (from MS COCO, with questions about commonsense or world knowledge) can be downloaded from its project page or via the AllenAI repository. These public datasets allow testing Flamingo’s zero-shot performance by providing a few QA exemplars from the training set as context and then evaluating on the validation/test questions. *(Flamingo was evaluated in a 32-shot setting on OK-VQA, where it set a new state-of-the-art​*

[*openreview.net*](https://openreview.net/pdf?id=mWVoBz4W0u#:~:text=Table%203%3A%20VQA%20Accuracy%20results,generation%20setting%2C%20and%20still%20outperform)

*.)* Both datasets are free to use for research.

**Implementation Tips:** Re-implementing Flamingo from scratch is non-trivial due to its size, but one can create a **mini Flamingo** using standard components. The recipe is: (1) take a **pretrained vision encoder** (e.g. a ResNet or ViT) and a **pretrained language model** (e.g. GPT-2 or T5). Keep their weights fixed to start (as Flamingo did for stability). (2) Add a **Perceiver Resampler** module: this is a small transformer that accepts the many image feature vectors from the vision encoder and outputs a fixed number of learned visual tokens​

[medium.com](https://medium.com/@paluchasz/understanding-flamingo-visual-language-models-bea5eeb05268#:~:text=Vision%20Encoder%20,Language%20Model%20and%20generate%20text)

. This gives a compact representation of the image. (3) Integrate these visual tokens into the language model by inserting **cross-attention layers** in the language model’s stack​

[medium.com](https://medium.com/@paluchasz/understanding-flamingo-visual-language-models-bea5eeb05268#:~:text=representations%20produced%20by%20the%20Perceiver,architecture%20can%20be%20visualised%20below)

. For example, after every few transformer layers of the language model, add a cross-attention that uses the visual tokens as key/value and the text representations as query. This way, when the language model generates an answer, it can attend to visual content at multiple points. (4) Train this combined model on image-question-answer data. You can simulate Flamingo’s training by using image-caption pairs and QA pairs – present the model with a sequence like “[Image] Caption. [Image] Q: question? A:” and have it continue the answer. This in-context format teaches the model to handle interleaved image-text. (5) During inference on VQA, provide a few QA examples (text only) plus the new image and question, and sample the model’s answer. In practice, one might use frameworks like Hugging Face Transformers to implement custom cross-attention layers. Focus on **fusing the modalities** by the gated cross-attention approach: the image features influence the text generation but the language model remains mostly unchanged. Even if training a full Flamingo clone is infeasible without enormous data and compute, these steps allow a smaller-scale implementation that captures the essence of Flamingo’s multimodal fusion approach. Fine-tuning such a model on VQA v2 (or using few-shot prompts as Flamingo does) would then allow it to answer visual questions.

**Decouple Before Interact: Multi-Modal Prompt Learning for Continual VQA (*TRIPLET framework*) – ICCV 2023**

**Summary:** This paper addresses **continual learning** for VQA (learning on a stream of tasks without forgetting) with a novel multi-modal prompt tuning strategy​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=modalities%20will%20lead%20to%20poor,modeling%20interactions%20between%20inputs%20and)

​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=novel%20approach%20that%20builds%20on,VQA)

. The proposed *TRIPLET* approach uses a pre-trained vision-language model (like CLIP or ViLT) and extends it with **decoupled prompts** – essentially learnable embeddings prepended to the visual and textual inputs, respectively​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=modalities%20will%20lead%20to%20poor,modeling%20interactions%20between%20inputs%20and)

. Unlike naive fine-tuning, these prompts are **modality-specific** (separate prompts for image and for question) and are learned in a way that they capture task-specific knowledge without interfering with each other (*“decouple before interact”*). During inference, the model first processes the image and question with their respective prompts (which helps retain past knowledge for each modality) and then applies a prompt interaction module to fuse the modalities​

[openaccess.thecvf.com](https://openaccess.thecvf.com/content/ICCV2023/html/Qian_Decouple_Before_Interact_Multi-Modal_Prompt_Learning_for_Continual_Visual_Question_ICCV_2023_paper.html#:~:text=modal%20vision,demonstrate%20that%20our%20TRIPLET%20outperforms)

​

[openaccess.thecvf.com](https://openaccess.thecvf.com/content/ICCV2023/html/Qian_Decouple_Before_Interact_Multi-Modal_Prompt_Learning_for_Continual_Visual_Question_ICCV_2023_paper.html#:~:text=novel%20approach%20that%20builds%20on,VQA)

. This two-stage fusion (first within each modality via prompts, then cross-modal) improves continual learning performance by minimizing interference between vision and language streams. TRIPLET demonstrated superior accuracy on continual VQA benchmarks, outperforming prior methods in both image-only and question-only incremental scenarios​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=interactions%20between%20modalities,VQA)

.  
**Dataset:** Evaluated on synthetic continual learning benchmarks built from existing data (e.g. splits derived from **VQA v2** and **Visual Genome** questions)​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=TDIUC%20Visual%20Genome%20Visual%20Question,0)

​

[paperswithcode.com](https://paperswithcode.com/paper/decouple-before-interact-multi-modal-prompt#:~:text=Visual%20Genome%20%20%20,30%20%20%20TDIUC)

. *VQA v2* is publicly accessible on the official VQA website (VisualQA.org) for training and evaluation.  
**Implementation:** To reproduce TRIPLET:

* Start with a pretrained multi-modal model (for example, a transformer that accepts image region features and question tokens). Freeze the original model weights to preserve initial capabilities.
* Introduce **learnable prompt vectors**: e.g. a small set of learned tokens prepended to the text input sequence, and a set of learned vectors added to the image feature inputs. These serve as memory slots that can adapt to new tasks.
* For each new training task or data batch, **optimize only the prompt vectors** (and perhaps a small interaction network) while keeping the backbone fixed. The prompts should be decoupled by modality: you maintain separate prompt parameters for the vision stream and the language stream​

[openaccess.thecvf.com](https://openaccess.thecvf.com/content/ICCV2023/html/Qian_Decouple_Before_Interact_Multi-Modal_Prompt_Learning_for_Continual_Visual_Question_ICCV_2023_paper.html#:~:text=modal%20vision,demonstrate%20that%20our%20TRIPLET%20outperforms)

.

* After learning modality-specific prompts, implement an **interaction module** that combines the prompted image and question features. This could be an attention layer that takes the output of the vision encoder and language encoder (after inserting prompts) and fuses them (for example, cross-attention from question to image). The prompts themselves can also interact here if needed.
* Use continual learning strategies: as new tasks arrive, update the prompts incrementally. The decoupling helps reduce forgetting, but you might also employ replay or regularization if necessary (the paper introduced two new CL-VQA benchmarks to test this).
* By following this prompt-based tuning instead of full-model fine-tuning, the model should retain performance on older VQA tasks while adapting to new ones. Standard training protocols for VQA (cross-entropy on answers) apply at each incremental step.

**Generate then Select: Open-ended VQA Guided by World Knowledge (*RASO pipeline*) – ACL 2023 (Findings)**

**Summary:** This work presents a two-stage VQA approach called **RASO** (Retrieval-Augmented Solver for Open-ended VQA) that first generates many possible answers using a language model and then selects the best answer using a vision-language verifier​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=The%20open,strategy%20guided%20by%20world%20knowledge)

​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=methods%20suffer%20from%20low%20knowledge,As%20proved%20in)

. The motivation is to inject **world knowledge** into VQA without training a huge multimodal model end-to-end. In stage 1, a large *pre-trained language model* (like GPT-3) is prompted with the question (and some context) to **produce a list of candidate answers**​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=methods%20suffer%20from%20low%20knowledge,As%20proved%20in)

​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=pipeline%20that%20deploys%20a%20generate,art%20by%204.1%25%20on%20OK)

. These answers may be partially or fully correct (the LM has broad knowledge but no image). In stage 2, a smaller multimodal model evaluates each candidate by checking it against the image. This is essentially a **multi-modal fusion** step: the image and question are fed into an answer-scoring network along with each candidate answer (for example, as a ternary input: image+question+answer)​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=dependency%20on%20the%20PLM%20quality,domain%20training%20data)

. The network (which can be a classifier that outputs whether the answer is correct) uses visual-text co-attention to fuse the image and answer, ensuring the selected answer is consistent with the visual content. By generating diverse guesses and using the image to validate them, this method expands coverage of world knowledge while still grounding in the image. RASO improved accuracy on OK-VQA (a knowledge-based VQA benchmark) by 4.1% over prior state-of-the-art​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=pipeline%20that%20deploys%20a%20generate,art%20by%204.1%25%20on%20OK)

, showing that the generate-and-select strategy can effectively leverage external knowledge.  
**Dataset:** Evaluated on **OK-VQA** (requires outside knowledge; 14K questions)​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=The%20open,strategy%20guided%20by%20world%20knowledge)

​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=multi,art%20by%204.1%25%20on%20OK)

. *OK-VQA* is openly available – see the project page with downloads for questions and images

[okvqa.allenai.org](https://okvqa.allenai.org/#:~:text=Summary)

​

[okvqa.allenai.org](https://okvqa.allenai.org/download.html#:~:text=OK)

.  
**Implementation:** To implement a similar pipeline:

* **Answer generation:** Use a pre-trained large language model (e.g. GPT-3 via API, or an open model like LLaMA 2) and prompt it with the question. You might add a prefix such as *“Q: [question]? A:”* and ask it to generate a list of possible answers (maybe by instructing it to think of multiple possible answers). Because LMs can incorporate world knowledge (e.g. facts about objects, events, etc.), this step brings in external knowledge. Aim to generate *N* answer candidates, where *N* could be, say, 5 or 10.
* **Answer selection:** Design a multimodal model that takes an image, the question, and a proposed answer as input and outputs a score or likelihood that the answer is correct for that image-question pair. This could be implemented by concatenating the question and answer text, encoding them with a language encoder, combining with image features via cross-attention. For example, use a pretrained vision-language model (like CLIP or ViLBERT) and fine-tune it to predict true vs. false answer given the image-question. Train this model on VQA data (for which ground-truth answer is known; treat wrong answers as negatives).
* During inference, run each candidate answer from stage 1 through the selection model to get a confidence score. Choose the answer with the highest score as the final prediction​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=methods%20suffer%20from%20low%20knowledge,As%20proved%20in)

. This model will prefer answers that not only *sound* plausible but also actually match the image (resolving the ambiguity when the LM produces knowledge-based answers that might not apply to the specific image).

* Ensure that the answer generation stage is fairly broad (to cover needed knowledge) but constrained enough to not produce irrelevant nonsense. You can prompt the LM to output answers in a format (like a numbered list) and then parse it. The selection model can be a lightweight binary classifier or a ranking model.
* This modular approach requires two components to be trained/tuned separately: the answer verifier (needs VQA training data with correct/incorrect labels) and possibly some prompt tuning for the LM. Because the code for the exact implementation was not released, one would experiment with different LMs and multimodal classifiers. The end result is a system that integrates visual grounding with language-based world knowledge through a generate-and-filter paradigm​

[aclanthology.org](https://aclanthology.org/2023.findings-acl.147/#:~:text=pipeline%20that%20deploys%20a%20generate,art%20by%204.1%25%20on%20OK)

.