
Flamingo: a Visual Language Model for Few-Shot Learning

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

Karen Simonyan^{*,‡}

* Equal contributions, ordered alphabetically, † Equal contributions, ordered alphabetically,

‡ Equal senior contributions

DeepMind

Abstract

构建能够快速适应新任务的模型，仅使用少量注释示例，是多模态机器学习研究中的一个开放挑战。我们引入了 Flamingo，一种具备此能力的视觉语言模型（VLM）系列。我们提出了关键的架构创新，来：(i) 连接强大的预训练的仅视觉模型和仅语言模型，(ii) 处理任意交错的视觉和文本数据序列，(iii) 无缝地将图像或视频作为输入。由于其灵活性，Flamingo 模型可以在包含任意交错文本和图像的大规模多模态网络语料库上进行训练，这对于赋予它们上下文中的少样本学习能力至关重要。我们对我们的模型进行了全面评估，探究和测量其快速适应各种图像和视频任务的能力。这些任务包括开放性任务，如视觉问答，其中模型需要回答一个问题；字幕任务，用于评估描述场景或事件的能力；以及闭合

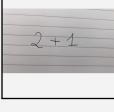
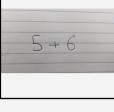
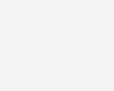
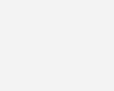
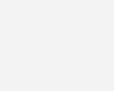
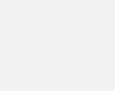
性任务，如多项选择的视觉问答。对于此类任务，仅一个 Flamingo 模型凭借任务特定示例的提示，就能够以少样本学习达到新的技术水平。在众多基准测试中，*Flamingo* 明显优于经过数千倍任务特定数据微调的模型。Building models that can be rapidly adapted to novel tasks using only a handful of annotated examples is an open challenge for multi-modal machine learning research. We introduce Flamingo, a family of Visual Language Models (VLM) with this ability. We propose key architectural innovations to: (i) bridge powerful pretrained vision-only and language-only models, (ii) handle sequences of arbitrarily interleaved visual and textual data, and (iii) seamlessly ingest images or videos as inputs. Thanks to their flexibility, Flamingo models can be trained on large-scale multimodal web corpora containing arbitrarily interleaved text and images, which is key to endow them with in-context few-shot learning capabilities. We perform a thorough evaluation of our models, exploring and measuring their ability to rapidly adapt to a variety of image and video tasks. These include open-ended tasks such as visual question-answering, where the model is prompted with a question which it has to answer; captioning tasks, which evaluate the ability to describe a scene or an event; and close-ended tasks such as multiple-choice visual question-answering. For tasks lying anywhere on this spectrum, a *single* Flamingo model can achieve a new state of the art with few-shot learning, simply by prompting the model with task-specific examples. On numerous benchmarks, *Flamingo* outperforms models fine-tuned on thousands of times more task-specific data.

1 Introduction

智能的一个关键方面是能够在简短指令下快速学会执行一项新任务 [33, 70]。虽然计算机视觉在实现类似能力方面取得了初步进展，但最广泛使用的范式仍然是“学习”、虽然计算机视觉领域在实现类似功能方面取得了初步进展，但最广泛使用的范式仍包括首先在大量有监督数据上进行预训练，然后再在感兴趣的任务上对模型进行微调 [66, 118, 143]。然而，成功的微调往往需要成千上万的注释数据点。此外，这往往需要对每个任务的超参数进行仔细调整，而且也是资源密集型的。最近，利用对比目标训练的多模态视觉语言模型（contrastive objective [50, 85]）实现了对新任务的零点适应，而无需进行微调。然而，由于这些模型仅仅提供了文本和图像之间的相似度得分，因此它们只能解决有限的使用案例，例如分类，因为在分类之前已经提供了一组有限的结果。最关键的是，这些模型缺乏生成语言的能力，因此不太适合更开放的任务，如字幕或视觉问题解答。还有一些人探索了视觉条件下的语言生成 [17, 114, 119, 124, 132]，但尚未在低数据环境下显示出良好的性能。

Input Prompt

Completion

<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>This is a chinchilla. They are mainly found in Chile.</p> </div> <div style="text-align: center;">  <p>This is a shiba. They are very popular in Japan.</p> </div> <div style="text-align: center;">  </div> </div> <p style="text-align: center;">This is</p> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>What is the title of this painting? Answer: The Hallucinogenic Toreador.</p> </div> <div style="text-align: center;">  <p>Where is this painting displayed? Answer: Louvre Museum, Paris.</p> </div> <div style="text-align: center;">  <p>What is the name of the city where this was painted? Answer:</p> </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>a flamingo. They are found in the Caribbean and South America.</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Output: "Underground"</p> </div> <div style="text-align: center;">  <p>Output: "Congress"</p> </div> <div style="text-align: center;">  <p>Output:</p> </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Output: "Underground"</p> </div> <div style="text-align: center;">  <p>Output: "Congress"</p> </div> <div style="text-align: center;">  <p>Output:</p> </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>Arles.</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>2+1=3</p> </div> <div style="text-align: center;">  <p>5+6</p> </div> <div style="text-align: center;">  <p>5+6=11</p> </div> <div style="text-align: center;">  <p>3×6</p> </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>2+1=3</p> </div> <div style="text-align: center;">  <p>5+6</p> </div> <div style="text-align: center;">  <p>5+6=11</p> </div> <div style="text-align: center;">  <p>3×6</p> </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>3×6=18</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese.</p> </div> <div style="text-align: center;">  <p>Output: A pink room with a flamingo pool float.</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese.</p> </div> <div style="text-align: center;">  <p>Output: A pink room with a flamingo pool float.</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>A portrait of Salvador Dali with a robot head.</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Les sanglots longs des violons de l'automne blessent mon cœur d'une langueur monotone.</p> </div> <div style="text-align: center;">  <p>Pour qui sont ces serpents qui sifflent sur vos têtes?</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Les sanglots longs des violons de l'automne blessent mon cœur d'une langueur monotone.</p> </div> <div style="text-align: center;">  <p>Pour qui sont ces serpents qui sifflent sur vos têtes?</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>Je suis un cur qui bat pour vous.</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>pandas: 3</p> </div> <div style="text-align: center;">  <p>dogs: 2</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>pandas: 3</p> </div> <div style="text-align: center;">  <p>dogs: 2</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>giraffes: 4</p> </div> </div>
<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>I like reading</p> </div> <div style="text-align: center;">  <p>, my favourite play is Hamlet. I also like</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>I like reading</p> </div> <div style="text-align: center;">  <p>, my favourite play is Hamlet. I also like</p> </div> <div style="text-align: center;">  </div> </div> </div>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>→</p> <div style="background-color: #fce4ec; border-radius: 10px; padding: 5px; text-align: center;"> <p>Dreams from my Father.</p> </div> </div>

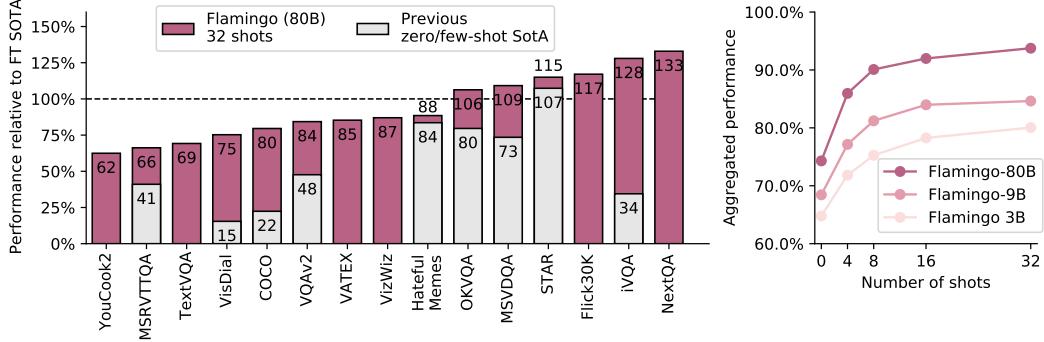


图 2: **Flamingo results overview.** *Left:* Our largest model, dubbed *Flamingo*, outperforms state-of-the-art fine-tuned models on 6 of the 16 tasks we consider with no fine-tuning. For the 9 tasks with published few-shot results, *Flamingo* sets the new few-shot state of the art. *Note:* We omit RareAct, our 16th benchmark, as it is a zero-shot benchmark with no available fine-tuned results to compare to. *Right:* *Flamingo* performance improves with model size and number of shots.

如图1所示，我们介绍了一种视觉语言模型（VLM）–largem，它在广泛的开放式视觉和语言任务中，只需通过几个输入/输出示例的提示，就能实现“少量学习”（few-shot learning）。在我们考虑的 16 项任务中，尽管使用的特定任务训练数据少了几个数量级，但 *Flamingo* 在 6 项任务上也超过了微调后的技术水平（见图 2）。为了实现这一目标，*Flamingo* 从大型语言模型（LMs）的最新研究中汲取了灵感，这些大型语言模型是很好的少量学习者 [11, 18, 42, 86]。单个大型 LM 只需使用文本界面，单个大型语言模型就能在许多任务上取得很好的性能：将任务的几个示例作为提示提供给模型，同时输入一个查询，然后模型生成一个续集，为该查询生成一个预测输出。

我们的研究表明，图像和视频理解任务（如分类、字幕或问题解答）也可以采用同样的方法：这些任务可以被视为具有视觉输入条件的文本预测问题。与 LM 不同的是，模型必须能够接收包含图片和/或视频与文本交错的多模态提示。*Flamingo* models 具备这种能力—它们是视觉条件下的自回归文本生成模型，能够摄取与图像和/或视频交错在一起的文本标记序列，并生成文本作为输出。*Flamingo* models 利用了两个互补的预训练和冻结模型：一个是可以“感知”视觉场景的视觉模型，另一个是执行基本推理形式的大型 LM。在这些模型之间添加了新的架构组件，以一种保留它们在计算密集型预训练过程中积累的知识的方式将它们连接起来。由于采用了基于感知器 *Flamingo* models (Perceiver-based /cite{jaegle2021perceiver}) 的架构，因此该 *Flamingo* models 也能够摄取高分辨率的图像或视频。

大型 LMs 性能的一个重要方面是，它们需要在大量文本数据上接受训练。这种训练提供了通用的生成能力，使这些 LM 在任务示例的提示下表现出色。同样，我们证明了我们训练 *Flamingo* 模型的最终性能至关重要。这些模型是在精心选择的互补性大规模数据新页面上训练出来的。没有使用任何以机器学习为目的的注释数据。经过这样的训练 *Flamingo* 模型可以通过简单的几次学习直接适应视觉任务，而不需要针对任何特定任务的调整。

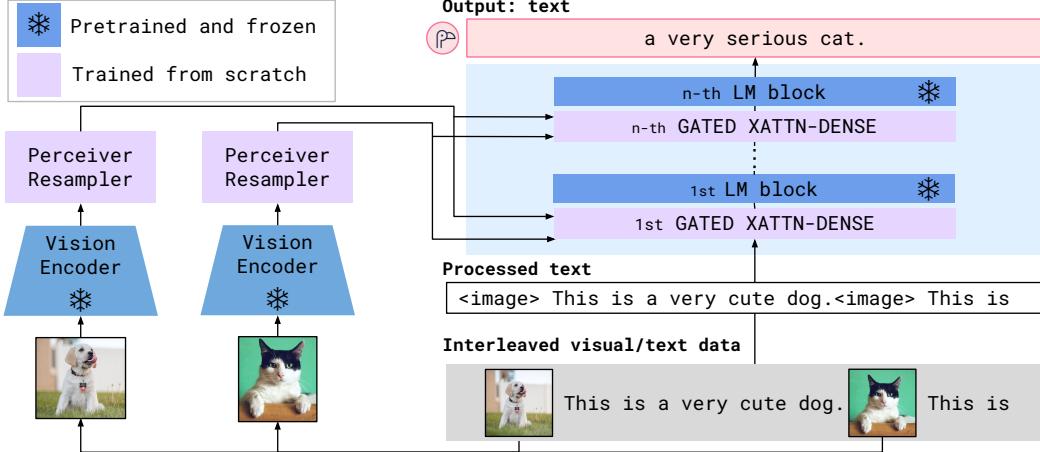


图 3: **Flamingo architecture overview.** Flamingo 是一个视觉语言模型 (VLM) 系列, 它将视觉数据与文本交错作为输入, 并将自由格式文本作为输出。Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Contributions. 总之, 我们的贡献如下: (i) 我们介绍了 VLM 的 Flamingo 系列, 它们可以仅通过几个输入/输出示例执行各种多模态任务 (如字幕、视觉对话或视觉问题解答)。得益于架构上的创新, Flamingo 模型可以高效地接受任意交错的视觉数据和文本作为输入, 并以开放的方式生成文本。(ii) 我们定量评估了模型如何通过少量学习适应各种任务。值得注意的是, 我们保留了一大批未用于验证该方法的任何设计决策或超参数的保留基准。我们用这些基准来估算无偏的少次学习性能。(iii) Flamingo 在广泛的 16 种多模态语言和图像/视频理解任务中, 为少量学习设定了新的技术水平。在这 16 项任务中的 6 项任务上, Flamingo 的表现也优于经过微调的现有技术水平, 尽管它只使用了 32 个特定任务示例, 比现有技术水平少了约 1000 倍的特定任务训练数据。有了更大的注释预算, Flamingo 还可以有效地进行微调, 从而在另外五个具有挑战性的基准上创造出新的技术水平: VQAv2、VATEX、VizWiz、MSRVTQA 和 HatefulMemes。

2 Approach

本节将介绍 Flamingo: 一种视觉语言模型, 它可以将文本与图片/视频交错作为输入, 并输出自由格式文本。图3中所示的关键架构组件可利用预训练的视觉模型和语言模型, 并将它们连接起来。选择这些组件是为了充分利用预训练的视觉模型和语言模型, 并有效地将它们连接起来。首先, 感知器重采样器 (Section 2.1) 接收来自视觉编码器 (从图像或视频中获取) 的时空特征, 并输出固定数量的视觉标记。其次, 这些视觉标记会被用于使用新初始化的交叉注意层 (Section 2.2) 来调节冻结的 LM, 这些交叉注意层会交错在预训练的 LM 层之间。这些新的层为 LM 在下一个标记预测任务中纳入视觉信息提供了一种富有表现力的方法。Flamingo 对以交错图像和视频 x 为条件的文本 y 的可能性建模如下:

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

其中, y_ℓ 是输入文本的第 ℓ 个语言标记, $y_{<\ell}$ 是前面标记的集合, $x_{\leq \ell}$ 是交错序列中标记 y_ℓ 前面的图像/视频的集合, p 由 Flamingo 模型参数化。处理交错文本和视觉序列的能力 (Section 2.3) 使得使用 Flamingo 模型进行上下文少点学习变得很自然, 这类类似于使用少点文本提示的 GPT-3。正如章节 2.4 中所述, 该模型是在不同的混合数据集上训练的。

2.1 Visual processing and the Perceiver Resampler

视觉编码器: 从像素到特征 (Vision Encoder: from pixels to features.) 我们的视觉编码器是一个经过预训练和冻结的无规范化 ResNet (NFNet) (我们使用的是 F6 模型)。我们在图像和文本对数据集上使用对比目标对视觉编码器进行预训练, 使用的是 Radford et al. [85] 的两期对比损失。我们使用的是最后阶段的输出结果, 即扁平化为一维序列的二维空间网格特征。对于视频输入, 帧以 1 FPS 的速度采样并独立编码, 以获得三维时空特征网格, 并将学习到的时间嵌入添加到网格中。然后将特征扁平化为一维, 再送入感知器重采样器。关于对比度模型训练和性能的更多详情, 分别见附录 B.1.3 和附录 B.3.2。

Perceiver Resampler: from varying-size large feature map to few visual tokens. 如图3所示, 该模块将视觉编码器与冻结语言模型连接起来。它从视觉编码器输入数量可变的图像或视频特征, 并产生固定数量的视觉输出 (64)、它降低了视觉-文本交叉关注的计算复杂度。与 Perceiver [48] 和 DETR [13] 类似, 我们学习了预定数量的潜在输入查询, 并将其输入到变换器 (Transformer) 中。并与视觉特征交叉关注。我们在消融研究 (Section refsec:ablations) 中表明, 使用这种视觉语言重置模块的效果优于普通转换器和 MLP。我们在附录 A.1.1 中提供了图解、更多架构细节和伪代码。

2.2 Conditioning frozen language models on visual representations

文本生成由变换器解码器执行, 并以感知器重采样器生成的视觉表征为条件。我们将预训练和冻结的纯文本 LM 块与从头开始训练的块交错排列, 这些块交叉关注感知器重采样器的视觉输出。

在冻结的预训练 LM 中插入新的门控交叉注意密集 gated cross attention dense 层。 我们冻结了预训练的 LM 块, 并在 (从头开始训练的) 原始层之间插入 *gated cross attention dense* 块 (Figure 4)。为了确保在初始化时, 条件化模型能产生与原始语言模型相同的结果, 我们使用了一种 \tanh 门控机制。这将新添加层的输出乘以 $\tanh(\alpha)$, 然后再将其添加到来自残差连接的输入表示中, 其中 α 是一个特定于层的可学习标量, 初始化为 0 [4])。因此, 在初始化时, 模型输出与预训练的 LM 相匹配, 从而提高了训练的稳定性和最终性能。在消融研究 (Section 3.3) 中, 我们将所提出的 GATED

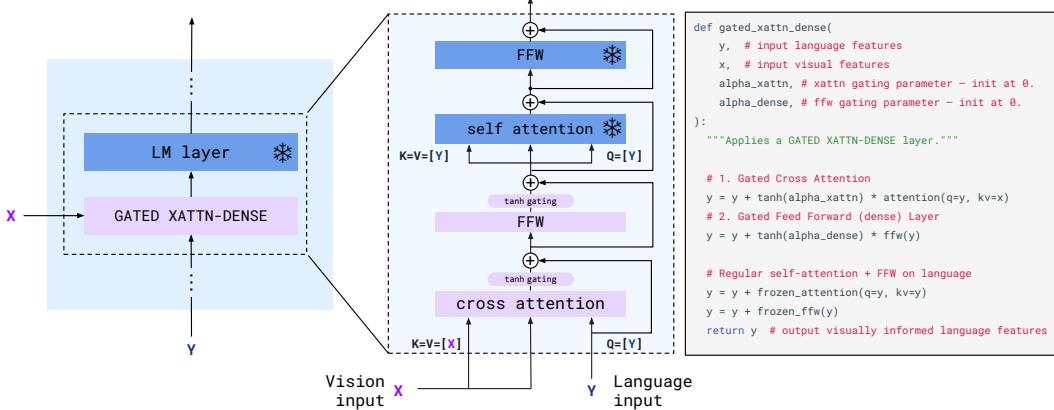


图 4: **gated xattn-dense layers.** 为了使 LM 以视觉输入为条件, 我们在现有的预训练和冻结 LM 层之间插入了新的交叉注意层。这些层中的键和值来自视觉特征, 而查询则来自语言输入。这些层之后是密集的前馈层。这些层是 *gated* 的, 因此 LM 在初始化时保持不变, 以提高稳定性和性能。

XATTN-DENSE 层与最近的替代方法进行了比较 [22, 68], 并探索了插入这些附加层的频率对效率和表现力之间权衡的影响。详见附录 A.1.2。

Varying model sizes. 我们在 1.4B、7B 和 70B 参数的 Chinchilla 模型 [42] 基础上, 对三种模型大小进行了实验; 分别称它们为 *Flamingo-3B*)、*Flamingo-9B*) 和 *Flamingo-80B*。为简洁起见, 我们在本文中将最后一个称为 *Flamingo*。在增加冷冻 LM 模块和可训练视觉文本 GATED XATTN-DENSE 模块的参数数量的同时, 我们在不同的模型中保持了固定大小的冷冻视觉编码器和可训练感知器重采样器 (相对于完整模型大小较小)。更多详情请参见附录 B.1.1。

2.3 Multi-visual input support: per-image/video attention masking

等式 (1) 中引入的图像因果建模是通过屏蔽完整的文本-图像交叉注意力矩阵来实现的、限制模型在每个文本标记处看到的视觉标记。在给定的文本标记处, 模型会关注交错序列中出现在它之前的图像的视觉标记, 而不是之前的所有图像 (在附录 A.1.3 中进行了形式化和说明)。虽然该模型每次只直接关注一幅图像, 但通过 LM 中的自我关注, 对之前所有图像的依赖性仍然存在。这种单张图像交叉关注方案的重要意义在于, 无论在训练过程中使用了多少张图像, 模型都能无缝泛化到任意数量的视觉输入。特别是, 我们在交错数据集上进行训练时, 每个序列最多只能使用 5 张图片, 但在评估过程中, 我们的模型却能从多达 32 对 (或 ‘shots’) 图片/视频和相应文本的序列中获益。我们将在第 3.3 节中证明, 这种方案比让模型直接交叉关注之前的所有图像更为有效。

2.4 Training on a mixture of vision and language datasets

我们在三种数据集的混合数据集上训练 Flamingo 模型, 所有数据集均来自网络: 来自网页的交错图像和文本数据集、图像-文本对和视频-文本对。

M3W: Interleaved image and text dataset. Flamingo 模型的少样本能力依赖于对交错文本和图像数据的训练。为此，我们收集了 *MultiModal Massive Web (M3W)* 数据集。我们从大约 4,300 万个网页的 HTML 中提取文本和图像，根据文本和图像元素在文档对象模型 (DOM) 中的相对位置，确定图像相对于文本的位置。然后，在页面上图像的位置插入纯文本中的 `<image>` 标记，并在任何图像之前和文档末尾插入一个特殊的 `<EOC>` (*end chunk*) 标记，从而构建一个示例。`end of chunk` 标记（添加到词汇表中并被学习）。我们从每篇文档中随机抽取 $L = 256$ 标记的子序列，并取样序列中包含的最多 $N = 5$ 图片。为了节省计算量，后面的图像将被舍弃。更多详情见附录 A.3。

图像/视频和文本的配对。 对于图像和文本配对，我们首先利用了 ALIGN [50] 数据集，该数据集由 18 亿张与 alt-text 配对的图像组成。作为对该数据集的补充，我们还收集了自己的图像和文本配对数据集，目标是质量更好、描述更长的图像和文本：LTIP (Long Text & Image Pairs) 由 3.12 亿张图片和文本对组成。我们还收集了一个类似的数据集，但使用的是视频而不是静态图像：VTP(Video & Text Pairs) 由 2,700 万个短视频（平均约 22 秒）和句子描述组成。我们通过在每个训练标题前添加 `<image>` 和附加 `<EOC>`，使配对数据集的语法与 M3W 的语法保持一致（详见附录 A.3.3）。

Multi-objective training and optimisation strategy. 在视觉输入的情况下，我们通过最小化每个数据集文本预期负对数似然的加权和来训练模型：

$$\sum_{m=1}^M \lambda_m \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}_m} \left[- \sum_{\ell=1}^L \log p(y_\ell | y_{<\ell}, x_{\leq \ell}) \right], \quad (2)$$

其中， \mathcal{D}_m 和 λ_m 分别是 m 数据集及其权重。调整每个数据集的权重 λ_m 是提高性能的关键。我们在所有数据集上累积梯度，发现这种方法优于“轮循”“round-robin”方法 [17]。我们在附录中提供了进一步的训练细节和消融 B.1.2。

2.5 Task adaptation with few-shot in-context learning

训练好 Flamingo 后，我们用它来完成一项视觉任务，即对多模态交错提示进行调节。我们使用文本学习 **in-context learning** 来评估我们的模型快速适应新任务的能力，类似于 GPT-3 [11]，方法是以 $(image, text)$ or $(video, text)$ 的形式交错支持示例对，然后是查询视觉输入，从而建立一个提示（详情见附录 A.2）。我们使用波束搜索进行解码，并使用我们模型的对数似然对每个可能的答案进行评分，从而执行 **open-ended** 评估。我们探索了 **zero-shot generalization**，方法是用任务中的两个纯文本示例（没有相应的图像）来提示模型。评估超参数和其他细节见附录 B.1.5。

3 Experiments

我们的目标是开发能够快速适应各种挑战性任务的模型。为此，我们考虑了 16 种流行的多模态图像/视频和语言基准。为了在项目过程中验证模型设计决策，其中 5 个基准被用作我们开发 (DEV) 集的一部分：COCO、OKVQA、VQAv2、MSVDQA 和 VATEX。由于模型选择的原因，对 DEV 基准的性能估计可能存在偏差。我们注意到，先前利用

Method	FT	Shot	OKVQA (I)	VQAv2 (I)	COCO (I)	MSVDQA (V)	VATEX (V)	VizWiz (I)	Flick30K (I)	MSRVTTQA (V)	iVQA (V)	YouCook2 (V)	STAR (V)	VisDial (I)	TextVQA (I)	NextQA (I)	HatefulMemes (I)	RareAct (V)
Zero/Few shot SOTA	\times		[34] (X) (16)	[114] (4) (0)	[124] (0) (0)	[58] (0) (0)	-	-	-	[58] (0) (0)	[135] (0) (0)	[143] (0) (0)	[79] (0) (0)	-	-	[85] (0) (0)	[85]	
<i>Flamingo</i> -3B	\times	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
<i>Flamingo</i> -3B	\times	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
<i>Flamingo</i> -9B	\times	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	47.3	30.6	26.1	56.3	-
<i>Flamingo</i> -9B	\times	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
<i>Flamingo</i> -9B	\times	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
<i>Flamingo</i>	\times	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	50.4	32.6	28.4	63.5	-
<i>Flamingo</i>	\times	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	60.8
<i>Flamingo</i>	\times	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
<i>Flamingo</i>	\times	32	57.8	67.6	113.8	52.3	65.1	49.8	75.4	31.0	45.3	86.8	42.2	55.6	37.9	33.5	70.0	-
Pretrained FT SOTA	\checkmark		54.4 (X) (10K)	80.2 (444K)	143.3 (500K)	47.9 (27K)	76.3 (500K)	57.2 (20K)	67.4 (30K)	46.8 (130K)	35.4 (6K)	138.7 (10K)	36.7 (46K)	75.2 (123K)	54.7 (20K)	25.2 (38K)	79.1 (9K)	

表 1: **Comparison to the state of the art.** A *single* Flamingo model reaches the state of the art on a wide array of image (**I**) and video (**V**) understanding tasks with few-shot learning, significantly outperforming previous best zero- and few-shot methods with as few as four examples. More importantly, using only 32 examples and without adapting any model weights, Flamingo *outperforms* the current best methods – fine-tuned on thousands of annotated examples – on seven tasks. Best few-shot numbers are in **bold**, best numbers overall are underlined.

类似基准验证和消减设计决策的工作也存在这种情况。为此，我们额外报告了 11 个基准的性能、这些基准包括字幕、视频问题解答，以及一些不太常见的功能，如视觉对话和多选问题解答任务。附录 B.1.4 中描述了评估基准。在所有基准中，我们保持所有评估超参数固定不变。根据任务的不同，我们使用了四种少量提示模板，更多详情请参见 Appendix B.1.5 中详细介绍。我们强调，我们并没有在这 11 个基准上验证任何设计决策，而只是用它们来估计我们模型的无偏少量 unbiased few-shot 学习性能。

具体来说，要估算一个模型的 few-shot 学习性能，需要用一组 *support* 样本对其进行提示，然后在一组 *query* 样本上对其进行评估。对于用于验证设计决策和超参数以及报告最终性能的 DEV 基准，我们使用了四个子集：*validation support*、*validation query*、*test support* 和 *test query*。对于其他基准，我们只需要后两个。我们将在附录 B.1.4 中报告如何形成这些子集。

我们将在第 3.1 节中报告 Flamingo 模型在 few-shot 学习上的结果。第 3.2 节给出了微调结果。消融研究见第 3.3 节。附录 B.2 提供了更多结果，包括 Flamingo 在 ImageNet 和 Kinetics700 分类任务中的表现，以及我们的对比模型的表现。附录 C 包含了更多定性结果。

3.1 Few-shot learning on vision-language tasks

Few-shot results. 结果见表 ??。Flamingo 在所考虑的 16 项基准测试中，Flamingo 远远优于之前的 *allzero-shot* 或 few-shot 测试方法。每个任务只需四个示例就能实现这一目标，这证明了视觉模型适应新任务的实用性和高效性。更重要的是，Flamingo 通常可以在多达数十万个注释示例的基础上进行微调，从而与最先进的方法相媲美。在六项任务中，Flamingo 甚至优于经过微调的 SotA，尽管它只使用了一组模型权重和 32

Method	VQAv2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA	HatefulMemes	
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	test seen
32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
Fine-tuned	82.0	82.1	138.1	84.2	65.7	65.4	47.4	61.8	59.7	118.6	57.1	54.1	86.6
SotA	81.3 [†] [133]	81.3 [†] [133]	149.6[†] [119]	81.4 [†] [153]	57.2 [†] [65]	60.6 [†] [65]	46.8 [51]	75.2 [79]	75.4[†] [123]	138.7 [132]	54.7 [137]	73.7 [84]	84.6 [†] [152]

表 2: **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* does not achieve SotA with few-shot learning. *Flamingo* sets a new SotA on five of them, outperforming methods (marked with †) that use tricks such as model ensembling or domain-specific metric optimisation (e.g., CIDEr optimisation).

一个特定任务示例。最后，尽管我们只使用了 DEV 基准来进行设计决策，但我们的结果还是很好地推广到了其他基准，这也证实了我们方法的通用性。

Scaling with respect to parameters and shots. As shown in Figure 2, the larger the model, the better the few-shot performance, similar to GPT-3 [11]. The performance also improves with the number of shots. We further find that the largest model better exploits larger numbers of shots. Interestingly, even though our Flamingo models were trained with sequences limited to only 5 images on *M3W*, they are still able to benefit from up to 32 images or videos during inference. This demonstrates the flexibility of the Flamingo architecture for processing a variable number of videos or images.

Scaling with respect to parameters and shots. 与参数和样本有关的缩放。如图 2 所示，模型越大，少镜头性能越好，这与 GPT-3 [11] 类似。性能也随着拍摄次数的增加而提高。我们进一步发现，最大的模型能更好地利用更多的镜头。有趣的是，尽管我们的 Flamingo 模型在 *M3W* 上的训练序列仅限于 5 幅图像，但它们在推理过程中仍能从多达 32 幅图像或视频中获益。这证明了 Flamingo 架构在处理不同数量的视频或图像时的灵活性。

3.2 Fine-tuning *Flamingo* as a pretrained vision-language model

虽然这并不是我们工作的重点，但我们验证了当给定更多数据时，Flamingo 模型可以通过微调权重来适应任务。在表 2 中，我们探索了在注释预算不受限制的情况下，针对给定任务微调我们最大的模型 *Flamingo*。简而言之，我们通过在较短的时间内以较小的学习率对模型进行微调，另外解冻视觉骨干网以适应更高的输入分辨率（详情见附录 in Appendix B.2.2）。我们发现，与之前介绍的在上下文中进行少量学习的结果相比，我们可以改进结果，在另外五项任务中创造了新的技术水平：VQAv2、VATEX、VizWiz、MSRVTTQA 和 HatefulMemes。

3.3 Ablation studies

在表 3 中，我们报告了使用 *Flamingo*-3B 对五个 DEV 基准的 validation subsets 进行 4 次拍摄的消融结果。请注意，与最终模型相比，我们使用了更小的批次规模和更短的训

Ablated setting	Flamingo-3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
Flamingo-3B model			3.2B	1.74s	86.5	42.1	55.8	36.3	53.4	70.7
(i) Training data	All data	w/o Video-Text pairs	3.2B	1.42s	84.2	43.0	53.9	34.5	46.0	67.3
		w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	41.6	60.9
		Image-Text pairs → LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	47.6	66.4
		w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	23.5	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	40.8
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	47.5
(iv)	Cross-attention architecture	GATED XATTN	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	50.7
		XATTN-DENSE	GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	47.8
(v)	Cross-attention frequency	Single in middle	Every	2.0B	0.87s	71.5	38.1	50.2	29.1	42.3
		Every 4th		2.3B	1.02s	82.3	42.7	55.1	34.6	50.8
		Every 2nd		2.6B	1.24s	83.7	41.0	55.8	34.5	49.7
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	44.7
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	48.3
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	44.5
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	42.9
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	50.1
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	53.9

表 3: **Ablation studies.** 每行应该与基线 Flamingo 运行 (顶层) 相比较。步数时间衡量在所有训练数据集上执行梯度更新所需的时间。表中每行的 **Overall score** 是通过除以每个基准的得分并除以其最先进的 (SotA) 性能，然后求平均值得到的。更多详情和结果见附录 B.3 和表 10。

Importance of the training data mixture. As shown in row (i), getting the right training data plays a crucial role. In fact, removing the interleaved image-text dataset *M3W* leads to a *decrease of more than 17%* in performance while removing the conventional paired image-text pairs also decreases performance (by 9.8%), demonstrating the need for different types of datasets. Moreover, removing our paired video-text dataset negatively affects performance on all video tasks. We ablate replacing our image-text pairs (ITP) by the publicly available LAION-400M dataset [96], which leads to a slight degradation in performance. We show in row (ii) the importance of our gradient accumulation strategy compared to using round-robin updates [17].

Importance of the training data mixture. 如 (i) 行所示，获取正确的训练数据起着至关重要的作用。事实上，去除交错的图像-文本数据集 *M3W* 会导致下降超过 17%，而去除传统的配对图像-文本数据对也会降低性能 (by 9.8%)，这表明需要不同类型的数据集。此外，去除我们的配对视频-文本数据集会对所有视频任务的性能产生负面影响。我们进行了替换公开可用的 LAION-400M 数据集 [96] 来代替我们的图像-文本配对 (ITP) 的实验，结果发现性能略有下降。我们在 (ii) 行展示了我们的梯度累积策略与使用轮询更新 [17] 的重要性。

Visual conditioning of the frozen LM. We ablate the use of the 0-initialized tanh gating when merging the cross-attention output to the frozen LM output in row (iii). Without it, we see a drop of 4.2% in our overall score. Moreover, we have noticed that disabling the 0-initialized tanh gating leads to training instabilities. Next, we ablate different conditioning architectures in row (iv). VANILLA XATTN, refers to the vanilla cross-attention from the original Transformer decoder [115]. In the GRAFTING approach from [68], the frozen LM is used as is with no additional layers inserted, and a stack

of interleaved self-attention and cross-attention layers that take the frozen LM output are learnt from scratch. Overall, we show that our GATED XATTN-DENSE conditioning approach works best.

Compute/Memory vs. performance trade-offs. In row (v), we ablate the frequency at which we add new GATED XATTN-DENSE blocks. Although adding them at every layer is better, it significantly increases the number of trainable parameters and time complexity of the model. Notably, inserting them every fourth block accelerates training by 66% while only decreasing the overall score by 1.9%. In light of this trade-off, we maximize the number of added layers under hardware constraints and add a GATED XATTN-DENSE every fourth layer for *Flamingo*-9B and every seventh for *Flamingo*-80B. We further compare in row (vi) the Perceiver Resampler to a MLP and a vanilla Transformer given a parameter budget. Both underperform the Perceiver Resampler while also being slower.

Vision encoder. In row (vii), we compare our NFNet-F6 vision encoder pretrained with contrastive learning (details in Appendix B.1.3) to the publicly available CLIP ViT-L/14 [85] model trained at 224 resolution. Our NFNet-F6 has a +5.8% advantage over the CLIP ViT-L/14 and +8.0% over a smaller NFNet-F0 encoder, which highlights the importance of using a strong vision backbone.

Freezing LM components prevents catastrophic forgetting. 我们在第 (viii) 行中验证了在训练时冻结 LM 层的重要性。如果从头开始训练，我们观察到性能会大幅下降 -12.9%。有趣的是，对我们预训练的 LM 进行微调也会导致性能下降 -8.0%。这表明出现了“灾难性遗忘”(catastrophic forgetting)，即模型在训练新目标时逐渐遗忘了预训练。在我们的设置中，冻结语言模型是使用混合中的预训练数据集 (MassiveText) 进行训练的更好选择。

4 Related work

Language modelling and few-shot adaptation. Language modelling has recently made substantial progress following the introduction of Transformers [115]. The paradigm of first pretraining on a vast amount of data followed by an adaptation on a downstream task has become standard [11, 23, 32, 44, 52, 75, 87, 108]. In this work, we build on the 70B Chinchilla language model [42] as the base LM for *Flamingo*. Numerous works have explored techniques to adapt language models to novel tasks using a few examples. These include adding small adapter modules [43], fine-tuning a small part of the LM [141], showing in-context examples in the prompt [11], or optimizing the prompt [56, 60] through gradient descent. In this paper, we take inspiration from the in-context [11] few-shot learning technique instead of more involved few-shot learning approaches based on metric learning [24, 103, 112, 117] or meta-learning [6, 7, 27, 31, 91, 155].

最近，基于 Transformer 的语言模型在语言建模方面取得了实质性的进展 [115]。先在大量数据上进行预训练，然后再在下游任务上进行微调的模式已经成为标准 [11, 23, 32, 44, 52, 75, 87, 108]。我们的研究基于 70B Chinchilla 语言模型 [42] 作为 *Flamingo* 的基本语言模型。许多研究探索了使用少量示例来将语言模型适应到新任务的技术。这些技术包括添加小型适配器模块 [43]，微调语言模型的一小部分 [141]，在提示中展示上下文示例 [11]，或通过梯度下降来优化提示 [56, 60]。在本文中，我们受到了基于上下文的少样本学习技术 [11] 的启发，而不是基于度量学习 [24, 103, 112, 117] 或元学习 [6, 7, 27, 31, 91, 155] 等更复杂的少样本学习方法。

When language meets vision. These LM breakthroughs have been influential for vision-language modelling. In particular, BERT [23] inspired a large body of vision-language work [16, 28, 29, 38, 59, 61, 66, 101, 106, 107, 109, 118, 121, 142, 143, 151]. We differ from these approaches as Flamingo models do not require fine-tuning on new tasks. Another family of vision-language models is based on contrastive learning [2, 5, 49, 50, 57, 74, 82, 85, 138, 140, 146]. Flamingo differs from contrastive models as it can generate text, although we build and rely upon them for our vision encoder. Similar to our work are VLMs able to generate text in an autoregressive manner [19, 25, 45, 67, 116]. Concurrent works [17, 58, 119, 124, 154] also propose to formulate numerous vision tasks as text generation problems. Building on top of powerful pretrained language models has been explored in several recent works. One recent line of work [26, 68, 78, 114, 136, 144] proposes to freeze the pretrained LM weights to prevent catastrophic forgetting [71]. We follow this idea by freezing the Chinchilla LM layers [42] and adding learnable layers within the frozen LM. We differ from prior work by introducing the first LM that can ingest arbitrarily interleaved images, videos, and text.

Web-scale vision and language training datasets. Manually annotated vision and language datasets are costly to obtain and thus relatively small (10k-100k) in scale [3, 15, 69, 122, 129, 139]. To alleviate this lack of data, numerous works [14, 50, 98, 110] automatically scrape readily available paired vision-text data. In addition to such paired data, we show the importance of also training on entire multimodal webpages containing interleaved images and text as a single sequence. Concurrent work CM3 [1] proposes to generate HTML markup from pages, while we simplify the text prediction task by only generating plain text. We emphasize few-shot learning and vision tasks while CM3 [1] primarily evaluates on language-only benchmarks in a zero-shot or fine-tuned setup.

5 Discussion

Limitations. 首先，我们的模型建立在预训练 LM 的基础上，因此直接继承了 LM 的弱点。例如，LM 先验通常是有帮助的，但偶尔也会产生幻觉和无根据的猜测。此外，LM 对于比训练序列更长的序列的泛化能力也很差。在训练过程中，它们的样本

效率也很低。解决这些问题可以加快该领域的进展，并增强 Flamingo 等 VLM 的能力。First, our models build on pretrained LMs, and as a side effect, directly inherit their weaknesses. For example, LM priors are generally helpful, but may play a role in occasional hallucinations and ungrounded guesses. Furthermore, LMs generalise poorly to sequences longer than the training ones. They also suffer from poor sample efficiency during training. Addressing these issues can accelerate progress in the field and enhance the abilities of VLMs like Flamingo. Second, the classification performance of Flamingo lags behind that of state-of-the-art contrastive models [82, 85]. These models directly optimize for text-image retrieval, of which classification is a special case. In contrast, our models handle a wider range of tasks, such as open-ended ones. A unified approach to achieve the best of both worlds is an important research direction.

Third, in-context learning has significant advantages over gradient-based few-shot learning methods, but also suffers from drawbacks depending on the characteristics of the application at hand. We demonstrate the effectiveness of in-context learning when access is limited to only a few dozen examples. In-context learning also enables simple deployment, requiring only inference, generally with no hyperparameter tuning needed. However, in-context learning is known to be highly sensitive to various aspects of the demonstrations [80, 148], and its inference compute cost and absolute performance scale poorly with the number of shots beyond this low-data regime. There may be opportunities to combine few-shot learning methods to leverage their complementary benefits. We discuss the limitations of our work in more depth in Appendix D.1.

Societal impacts. In terms of societal impacts, *Flamingo* offers a number of benefits while carrying some risks. Its ability to rapidly adapt to a broad range of tasks have the potential to enable non-expert users to obtain good performance in data-starved regimes, lowering the barriers to both beneficial and malicious applications. *Flamingo* is exposed to the same risks as large language models, such as outputting offensive language, propagating social biases and stereotypes, as well as leaking private information [42, 126]. Its ability to additionally handle visual inputs poses specific risks such as gender and racial biases relating to the contents of the input images, similar to a number of visual recognition systems [12, 21, 37, 97, 147]. We refer the reader to Appendix D.2 for a more extensive discussion of the societal impacts of our work, both positive and negative; as well as mitigation strategies and early investigations of risks relating to racial or gender bias and toxic outputs. Finally we note that, following prior work focusing on language models [72, 81, 111], the few-shot capabilities of Flamingo could be useful for mitigating such risks.

Conclusion. We proposed Flamingo, a general-purpose family of models that can be applied to image and video tasks with minimal task-specific training data. We also qualitatively explored interactive abilities of *Flamingo* such as “chatting” with the

model, demonstrating flexibility beyond traditional vision benchmarks. Our results suggest that connecting pre-trained large language models with powerful visual models is an important step towards general-purpose visual understanding.

Acknowledgments and Disclosure of Funding. This research was funded by DeepMind. We would like to thank many colleagues for useful discussions, suggestions, feedback, and advice, including: Samuel Albanie, Relja Arandjelović, Ka-reem Ayoub, Lorrayne Bennett, Adria Recasens Continente, Tom Eccles, Nando de Freitas, Sander Dieleman, Conor Durkan, Aleksa Gordi, Raia Hadsell, Will Hawkins, Lisa Anne Hendricks, Felix Hill, Jordan Hoffmann, Geoffrey Irving, Drew Jaegle, Koryay Kavukcuoglu, Agustin Dal Lago, Mateusz Malinowski, Soa Mokrá, Gaby Pearl, Toby Pohlen, Jack Rae, Laurent Sifre, Francis Song, Maria Tsimpoukelli, Gregory Wayne, and Boxi Wu.

References

- [1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A causal masked multimodal model of the internet. *arXiv:2201.07520*, 2022.
- [2] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *Conference on Neural Information Processing Systems*, 2020.
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *International Conference on Computer Vision*, 2015.
- [4] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, 2021.
- [5] Max Bain, Arsha Nagrani, Gü̈l Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *International Conference on Computer Vision*, 2021.
- [6] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. *Conference on Neural Information Processing Systems*, 2016.
- [7] Luca Bertinetto, Joao F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv:1805.08136*, 2018.

- [8] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- [9] John S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, 1990.
- [10] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv:2102.06171*, 2021.
- [11] 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. Language models are few-shot learners. In *Conference on Neural Information Processing Systems*, 2020.
- [12] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM Conference on Fairness, Accountability, and Transparency*, 2018.
- [13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.
- [14] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Computer Vision and Pattern Recognition*, 2021.
- [15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015.
- [16] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *European Conference on Computer Vision*, 2020.
- [17] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, 2021.

- [18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *arXiv:2204.02311*, 2022.
- [19] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on clip via vision-language knowledge distillation. In *ACL Findings*, 2022.
- [20] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *IEEE Computer Vision and Pattern Recognition*, 2017.
- [21] Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. Does object recognition work for everyone? In *IEEE Computer Vision and Pattern Recognition*, 2019.
- [22] Karan Desai and Justin Johnson. VirTex: Learning visual representations from textual annotations. In *IEEE Computer Vision and Pattern Recognition*, 2021.
- [23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.
- [24] Carl Doersch, Ankush Gupta, and Andrew Zisserman. CrossTransformers: spatially-aware few-shot transfer. *Conference on Neural Information Processing Systems*, 2020.
- [25] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Computer Vision and Pattern Recognition*, 2015.

- [26] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. MAGMA—multimodal augmentation of generative models through adapter-based finetuning. *arXiv:2112.05253*, 2021.
- [27] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.
- [28] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-end video-language transformers with masked visual-token modeling. *arXiv:2111.12681*, 2021.
- [29] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *Conference on Neural Information Processing Systems*, 2020.
- [30] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 2021.
- [31] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction. *arXiv:1805.09921*, 2018.
- [32] Alex Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.
- [33] Thomas L. Griffiths, Frederick Callaway, Michael B. Chang, Erin Grant, Paul M. Krueger, and Falk Lieder. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 2019.
- [34] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. KAT: A knowledge augmented transformer for vision-and-language. *arXiv:2112.08614*, 2021.
- [35] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *IEEE Computer Vision and Pattern Recognition*, 2018.
- [36] Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. *arXiv:2203.16634*, 2022.

- [37] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, 2018.
- [38] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. Decoupling the role of data, attention, and losses in multimodal transformers. *Annual Meeting of the Association for Computational Linguistics*, 2021.
- [39] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv:1606.08415*, 2016.
- [40] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL <http://github.com/deepmind/dm-haiku>.
- [41] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.
- [42] Jordan Hoffmann, Sébastien Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Eric Noland, Tom Hennigan, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv:2203.15556*, 2022.
- [43] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2019.
- [44] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv:1801.06146*, 2018.
- [45] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. *arXiv:2111.12233*, 2021.
- [46] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019.
- [47] Md Amirul Islam, Matthew Kowal, Sen Jia, Konstantinos G. Derpanis, and Neil D. B. Bruce. Global pooling, more than meets the eye: Position information is encoded channel-wise in CNNs. In *International Conference on Computer Vision*, 2021.

- [48] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.
- [49] Aashi Jain, Mandy Guo, Krishna Srinivasan, Ting Chen, Sneha Kudugunta, Chao Jia, Yinfei Yang, and Jason Baldridge. MURAL: multimodal, multitask retrieval across languages. *arXiv:2109.05125*, 2021.
- [50] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv:2102.05918*, 2021.
- [51] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv:2203.07303*, 2022.
- [52] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv:1602.02410*, 2016.
- [53] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.
- [54] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The Hateful Memes Challenge: Detecting hate speech in multimodal memes. *Conference on Neural Information Processing Systems*, 2020.
- [55] Hugo Larochelle. Few-shot classification by recycling deep learning. Invited Talk at the *S2D-OLAD Workshop, ICLR 2021*, 2021. URL <https://slideslive.com/38955350/fewshot-classification-by-recycling-deep-learning>.
- [56] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv:2104.08691*, 2021.
- [57] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Conference on Neural Information Processing Systems*, 2021.
- [58] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv:2201.12086*, 2022.

- [59] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for video+language omni-representation pre-training. *arXiv:2005.00200*, 2020.
- [60] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv:2101.00190*, 2021.
- [61] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Li-juan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 2020.
- [62] Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. A multimodal framework for the detection of hateful memes. *arXiv:2012.12871*, 2020.
- [63] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? *arXiv:2101.06804*, 2021.
- [64] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Optimization of image description metrics using policy gradient methods. In *International Conference on Computer Vision*, 2017.
- [65] Yu Liu, Lianghua Huang, Liuyihang Song, Bin Wang, Yingya Zhang, and Pan Pan. Enhancing textual cues in multi-modal transformers for VQA. *VizWiz Challenge 2021*, 2021.
- [66] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Conference on Neural Information Processing Systems*, 2019.
- [67] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. UniVL: A unified video and language pre-training model for multimodal understanding and generation. *arXiv:2002.06353*, 2020.
- [68] Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. VC-GPT: Visual conditioned GPT for end-to-end generative vision-and-language pre-training. *arXiv:2201.12723*, 2022.
- [69] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE Computer Vision and Pattern Recognition*, 2019.

- [70] Ellen M. Markman. *Categorization and naming in children: Problems of induction*. MIT Press, 1989.
- [71] Michael McCloskey and Neil J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 1989.
- [72] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes. *arXiv:2203.11147*, 2022.
- [73] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. RareAct: A video dataset of unusual interactions. *arxiv:2008.01018*, 2020.
- [74] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *IEEE Computer Vision and Pattern Recognition*, 2020.
- [75] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. *Interspeech*, 2010.
- [76] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv:2202.12837*, 2022.
- [77] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *ACM Conference on Fairness, Accountability, and Transparency*, 2019.
- [78] Ron Mokady, Amir Hertz, and Amit H. Bermano. ClipCap: CLIP prefix for image captioning. *arXiv:2111.09734*, 2021.
- [79] Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *European Conference on Computer Vision*, 2020.
- [80] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *Conference on Neural Information Processing Systems*, 2021.
- [81] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv:2202.03286*, 2022.

- [82] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *arXiv:2111.10050*, 2021.
- [83] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.
- [84] Yixuan Qiao, Hao Chen, Jun Wang, Yihao Chen, Xianbin Ye, Ziliang Li, Xianbiao Qi, Peng Gao, and Guotong Xie. Winner team Mia at TextVQA Challenge 2021: Vision-and-language representation learning with pre-trained sequence-to-sequence model. *arXiv:2106.15332*, 2021.
- [85] 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. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021.
- [86] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susanah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saftron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv:2112.11446*, 2021.
- [87] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*, 2019.

- [88] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.
- [89] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 2022.
- [90] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *IEEE Computer Vision and Pattern Recognition*, 2017.
- [91] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Conference on Neural Information Processing Systems*, 2019.
- [92] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.
- [93] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender bias in coreference resolution. *arXiv:1804.09301*, 2018.
- [94] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.
- [95] 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, Debjyoti 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. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*, 2022.
- [96] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komat-

- suzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv:2111.02114*, 2021.
- [97] Carsten Schwemmer, Carly Knight, Emily D. Bello-Pardo, Stan Oklobdzija, Martijn Schoonvelde, and Jeffrey W. Lockhart. Diagnosing gender bias in image recognition systems. *Socius*, 2020.
- [98] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018.
- [99] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv:2104.08691*, 2019.
- [100] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Computer Vision and Pattern Recognition*, 2019.
- [101] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. *arXiv:2112.04482*, 2021.
- [102] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the Kinetics-700-2020 human action dataset. *arXiv:2010.10864*, 2020.
- [103] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Conference on Neural Information Processing Systems*, 2017.
- [104] David R So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. Primer: Searching for efficient transformers for language modeling. *arXiv:2109.08668*, 2021.
- [105] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. *arXiv:1906.02243*, 2019.
- [106] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. *arXiv:1908.08530*, 2019.
- [107] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *International Conference on Computer Vision*, 2019.

- [108] Ilya Sutskever, James Martens, and Geoffrey E. Hinton. Generating text with recurrent neural networks. In *International Conference on Machine Learning*, 2011.
- [109] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformer. In *Conference on Empirical Methods in Natural Language Processing*, 2019.
- [110] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016.
- [111] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Mene-gali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. LaMDA: Language models for dialog applications. *arXiv:2201.08239*, 2022.
- [112] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, 2020.
- [113] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *Conference on Neural Information Processing Systems*, 2019.
- [114] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Conference on Neural Information Processing Systems*, 2021.
- [115] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems*, 2017.

- [116] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *International Conference on Computer Vision*, 2015.
- [117] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Conference on Neural Information Processing Systems*, 2016.
- [118] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. UFO: A unified transformer for vision-language representation learning. *arXiv:2111.10023*, 2021.
- [119] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv:2202.03052*, 2022.
- [120] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization? *arXiv:2204.05832*, 2022.
- [121] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. VLMO: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv:2111.02358*, 2021.
- [122] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. In *International Conference on Computer Vision*, 2019.
- [123] Yue Wang, Shafiq Joty, Michael Lyu, Irwin King, Caiming Xiong, and Steven Hoi. VD-BERT: A unified vision and dialog transformer with BERT. In *Conference on Empirical Methods in Natural Language Processing*, 2020.
- [124] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. *arXiv:2108.10904*, 2021.
- [125] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *arXiv:2109.01652*, 2021.
- [126] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean

- Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models. *arXiv:2112.04359*, 2021.
- [127] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *arXiv:2203.05482*, 2022.
- [128] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. STAR: A Benchmark for Situated Reasoning in Real-World Videos. In *Conference on Neural Information Processing Systems*, 2021.
- [129] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-QA: Next phase of question-answering to explaining temporal actions. In *IEEE Computer Vision and Pattern Recognition*, 2021.
- [130] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueteng Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.
- [131] Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. Zeroprompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. *arXiv:2201.06910*, 2022.
- [132] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. VLM: Task-agnostic video-language model pre-training for video understanding. *arXiv:2105.09996*, 2021.
- [133] Ming Yan, Haiyang Xu, Chenliang Li, Junfeng Tian, Bin Bi, Wei Wang, Weihua Chen, Xianzhe Xu, Fan Wang, Zheng Cao, Zhicheng Zhang, Qiyu Zhang, Ji Zhang, Songfang Huang, Fei Huang, Luo Si, and Rong Jin. Achieving human parity on visual question answering. *arXiv:2111.08896*, 2021.
- [134] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. *arXiv:2201.04288*, 2022.
- [135] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *International Conference on Computer Vision*, 2021.
- [136] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *National Conference on Artificial Intelligence (AAAI)*, 2021.

- [137] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-aware pre-training for text-VQA and text-caption. In *IEEE Computer Vision and Pattern Recognition*, 2021.
- [138] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. *arXiv:2111.07783*, 2021.
- [139] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Annual Meeting of the Association for Computational Linguistics*, 2014.
- [140] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luwei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021.
- [141] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv:2106.10199*, 2021.
- [142] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. *Conference on Neural Information Processing Systems*, 2021.
- [143] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT reserve: Neural script knowledge through vision and language and sound. In *IEEE Computer Vision and Pattern Recognition*, 2022.
- [144] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv:2204.00598*, 2022.
- [145] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv:2106.04560*, 2021.
- [146] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. *arXiv:2111.07991*, 2021.

- [147] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *IEEE Computer Vision and Pattern Recognition*, 2021.
- [148] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021.
- [149] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *National Conference on Artificial Intelligence (AAAI)*, 2018.
- [150] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *National Conference on Artificial Intelligence (AAAI)*, 2020.
- [151] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *IEEE Computer Vision and Pattern Recognition*, 2020.
- [152] Ron Zhu. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *arXiv:2012.08290*, 2020.
- [153] Xinxin Zhu, Longteng Guo, Peng Yao, Shichen Lu, Wei Liu, and Jing Liu. Vatex video captioning challenge 2020: Multi-view features and hybrid reward strategies for video captioning. *arXiv:1910.11102*, 2019.
- [154] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Xiaogang Wang, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-Perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *arXiv:2112.01522*, 2021.
- [155] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, 2019.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes] See Section 5.
 - (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5 for a brief discussion and Appendix D.2 for the full discussion.
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [N/A]
 - (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and Appendix B.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We do not observe large enough variance in our training runs to justify the computation cost incurred by multiple training runs. For the largest models, it is not feasible within our compute budget.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details can be found in Appendix B.1.2. In short, our largest run was trained on 1536 TPU chips for 15 days.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes] We properly cited the prior methods on which our work is based, as well as prior datasets when appropriate (e.g., ALIGN).
 - (b) Did you mention the license of the assets? [N/A] The assets we used are previous work for which we cited papers. We do mention the license of all visual assets we use for the figures of the paper in Appendix G.

- (c) Did you include any new assets either in the supplemental material or as a URL? [No]
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Our data was automatically scraped from million of webpages. See Datasheets [30] in Appendix F.
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Datasheets [30] in Appendix F.
5. If you used crowdsourcing or conducted research with human subjects...
- (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

We provide an overview of the Appendix below.

Method (Appendix A). We first provide additional details about our model in Appendix A.1:

- An illustration and pseudo-code for the Perceiver Resampler (described in Section 2.1) is provided in Appendix A.1.1 and Figure 5.
- A similar illustration is provided for the GATED XATTN-DENSE layer of Section 2.2 in Appendix A.1.2 and Figure 4.
- Details on our implementation of the multi-image/video attention mechanism (Section 2.3) are given in Appendix A.1.3.
- Hyperparameters for all model architectures are given in Appendix A.1.4.

We then explain how we evaluate our models using in-context few-shot learning in Appendix A.2. This includes details on how we build the few-shot prompt, how we get predictions for open- and close-ended tasks, how we obtain the zero-shot numbers, and how we leverage retrieval and ensembling to take advantage of more annotated examples.

Finally, in Appendix A.3, we provide more details on our training datasets:

- Collection of M3W in Appendix A.3.1,
- How we process M3W samples during training in Appendix A.3.2,
- Collection of LTIP and VTP in Appendix A.3.3,
- Deduplication strategy we employ to ensure that there is no leakage between our training and evaluation datasets in Appendix A.3.4.

Experiments (Appendix B). We first provide additional training and evaluation details in Appendix B.1, including:

- Details on *Flamingo*-3B, *Flamingo*-9B and *Flamingo* in Appendix B.1.1,
- The training hyperparameters in Appendix B.1.2,
- More details on the Contrastive model pretraining in Appendix B.1.3,
- Details on our evaluation benchmarks and splits in Appendix B.1.4,
- A discussion on the few-shot learning hyperparameters in Appendix B.1.5,

- The dialogue prompt used in the qualitative dialogue examples shown in Figure 1 and Figure 11 in Appendix B.1.6.

Next, we give additional results obtained by our models in Appendix B.2 including the performance of the Flamingo models on classification tasks in Appendix B.2.1, detailed fine-tuning results in Appendix B.2.2, and zero-shot results from our contrastive models (Appendix B.2.3).

Finally, we provide more ablation studies in Appendix B.3 for both the Flamingo models (Appendix B.3.1) and our contrastive pretrained Visual Encoders (Appendix B.3.2).

Qualitative results (Appendix C). More qualitative results are given in Appendix C: Figure 10 (single image sample), Figure 11 (dialogue examples), and Figure 12 (video examples).

Discussion (Appendix D). We provide a more complete discussion on our work, including limitations, failure cases, broader impacts and societal impacts of our work in Appendix D.

Model card (Appendix E). The *Flamingo* model card is provided in Appendix E.

Datasheets (Appendix F). Datasheets for M3W, LTIP and VTP are respectively given in Appendix F.1, Appendix F.2.1 and Appendix F.2.2.

Credit for visual content (Appendix G). We provide attribution for all visual illustrations used in the paper in Appendix G.

A Method

A.1 Model details

A.1.1 Perceiver Resampler

Expanding on our brief description in Section 2.1, Figure 5 provides an illustration of our Perceiver Resampler processing an example video, together with pseudo-code. Our Perceiver Resampler is similar in spirit to the Perceiver models proposed by Jaegle et al. [48]. We learn a predefined number of latent input queries, and cross-attend to the flattened visual features X_f . These visual features X_f are obtained by first adding a learnt temporal position encoding to each feature within a given video frame (an image being considered as a single-frame video). Note that we only use temporal encodings and no explicit spatial grid position encodings; we did not observe improvements from the latter. This rationale behind is likely that CNNs, such as our NFNet encoder, are known to implicitly include spatial information channel-wise [47]. The visual features are then flattened and concatenated as illustrated in Figure 5. The number of output tokens of the Perceiver Resampler is equal to the number of learnt latent queries. Unlike

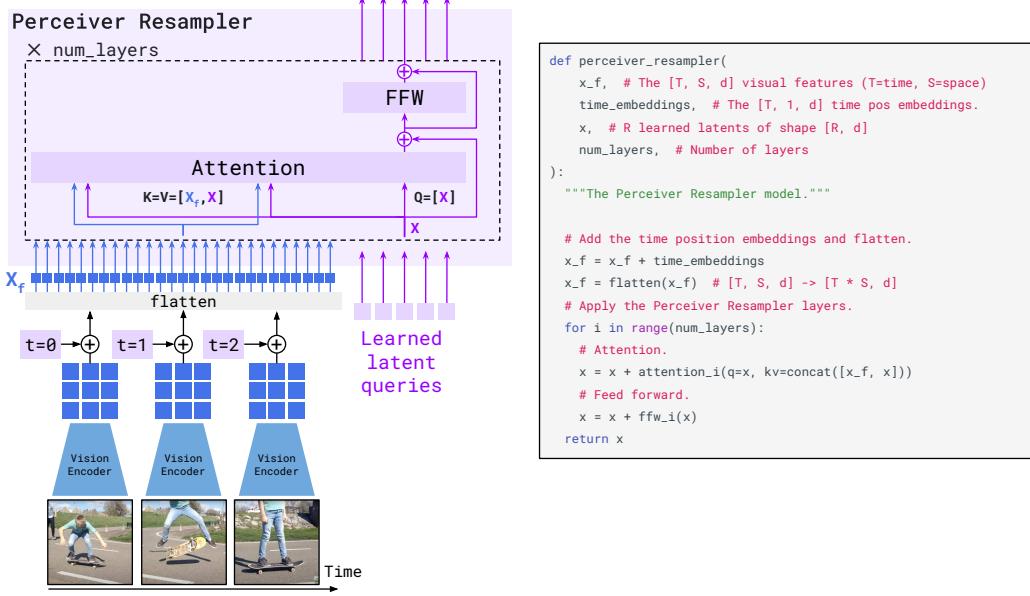


图 5: **The Perceiver Resampler** module maps a *variable* size grid of spatio-temporal visual features output by the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently from the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors.

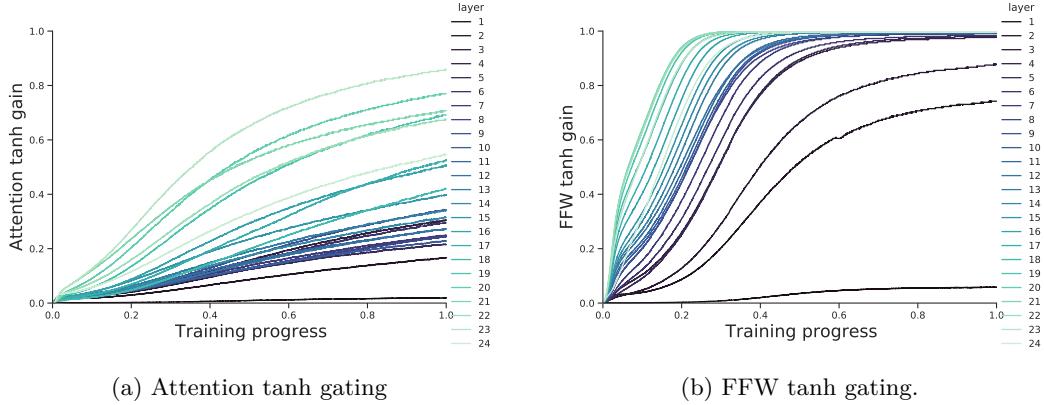


图 6: Evolution of the absolute value of the tanh gating at different layers of *Flamingo-3B*.

in DETR and Perceiver, the keys and values computed from the learnt latents are concatenated to the keys and values obtained from X_f , which we found to perform slightly better.

A.1.2 gated xattn-dense details

We provide in Figure 4 an illustration of a GATED XATTN-DENSE block and how it connects to a frozen LM block, together with pseudo-code.

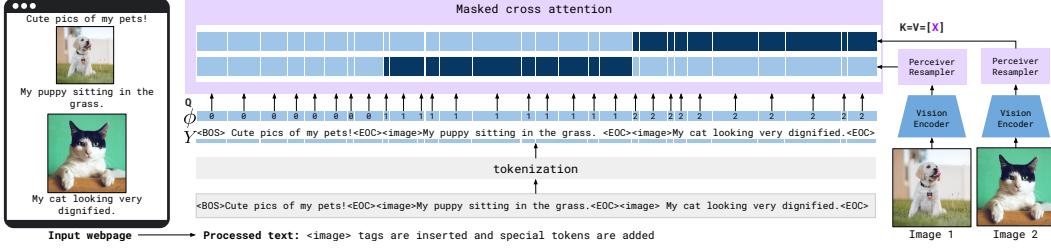


图 7: **Interleaved visual data and text support.** Given text interleaved with images/videos, e.g. coming from a webpage, we first process the text by inserting `<image>` tags at the locations of the visual data in the text as well as special tokens (`<BOS>` for “beginning of sequence” or `<EOC>` for “end of chunk”). Images are processed independently by the Vision Encoder and Perceiver Resampler to extract visual tokens. At a given text token, the model only cross-attends to the visual tokens corresponding to the last preceding image/video. ϕ indicates which image/video a text token can attend or 0 when no image/video is preceding. In practice, this selective cross-attention is achieved through masking – illustrated here with the dark blue entries (unmasked/visible) and light blue entries (masked).

We also plot in Figure 6 the evolution of the absolute value of the tanh gating values as a function of training progress (from 0% to 100%) at different layers of the LM stack for the *Flamingo-3B* model composed of 24 LM layers. All layers of the frozen LM stack seem to utilize the visual information as the tanh gating absolute values quickly grow in absolute value from their 0 initializations. We also note that the absolute values seem to grow with the depth. However, it is difficult to draw strong conclusions from this observation: the scale of the activations before gating may also vary with depth. Future work is required to better understand the effect of these added layers on the optimization dynamics and on the model itself.

A.1.3 Multi-visual input support

We illustrate in Figure 7 the masking approach we use to limit the number of visual tokens that a certain text token sees. We also formalize our notation for the interleaved sequences of images/videos and text.

Interleaved sequences of visual data and text. We consider interleaved image/video and text examples: each example holds a sequence of text y , a sequence of images/videos x , and the sequence of positions of the images in the text. Based on the visual data positions, we define a function $\phi : [1, L] \mapsto [0, N]$ that assigns to each text position the index of the last image/video appearing before this position (or 0 if no visual data appears before the position). The function ϕ defines which visual inputs we consider usable to predict token ℓ in Equation (1): the set of preceding tokens $y_{<\ell} \triangleq (y_1, \dots, y_{\ell-1})$, and the set of preceding images/videos $x_{\leq \ell} \triangleq \{x_i | i \leq \phi(\ell)\}$.

	Perceiver Resampler				GATED XATTN-DENSE				Frozen LM			
	L	D	H	Act.	L	D	H	Act.	L	D	H	Act.
Flamingo-3B	6	1536	16	Sq. ReLU	24	2048	16	Sq. ReLU	24	2048	16	GeLU
Flamingo-9B	6	1536	16	Sq. ReLU	10	4096	32	Sq. ReLU	40	4096	32	GeLU
Flamingo	6	1536	16	Sq. ReLU	12	8192	64	Sq. ReLU	80	8192	64	GeLU

表 4: Hyper-parameters for the Flamingo models’ transformers. The hidden size of each feed-forward MLP is $4D$. **L**: number of layers, **D**: transformer hidden size, **H**: number of heads, **Act.**: FFW activation, **Sq. ReLU**: Squared ReLU [104].

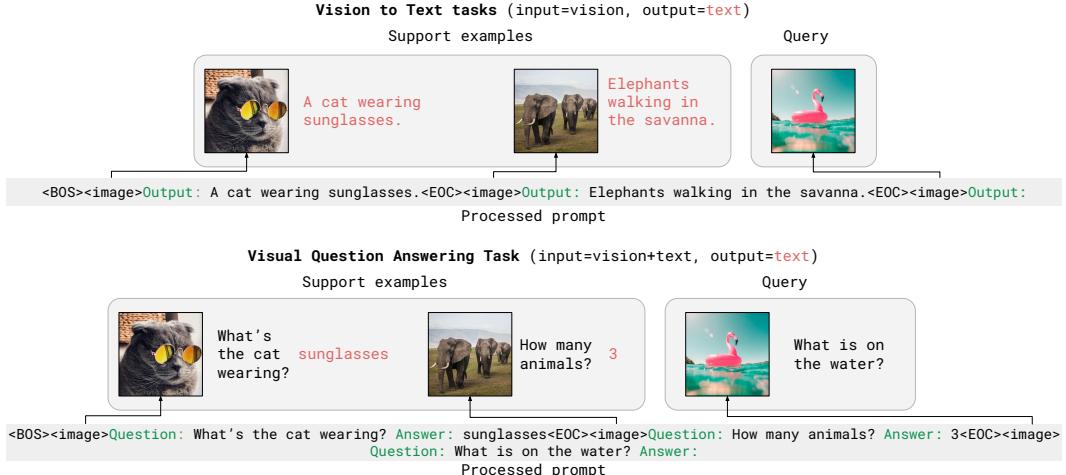


图 8: **Few-shot interleaved prompt generation.** Given some task-specific few-shot examples (a.k.a. support examples) and a query for which Flamingo should make a prediction, we build the prompt by interleaving images with their corresponding texts. We introduce some formatting to do this, prepending “**Output:**” to the expected response for all vision-to-text tasks or prompting in the format “**Question: {question} Answer: {answer}**” for visual question-answering tasks.

A.1.4 Transformer architecture

We list in Table 4 the number of layers (L), the hidden dimension (D), the number of heads (H), and the FFW activation (Act.) used for each transformer component of our Flamingo models. The dimension of keys and values in each configuration is given by D/H (96 for the Perceiver Resampler; 128 for GATED XATTN-DENSE and the frozen LM), and the hidden dimension of each feed-forward MLP is $4D$. Note that the frozen LM was trained with the GeLU activation [39], while the remaining trainable transformer layers use the Squared ReLU activation [104], which we found to outperform GeLU.

A.2 In-context few-shot evaluation details

In-context learning with Flamingo models. We evaluate the ability of our models to rapidly adapt to new tasks using in-context learning, following an analogous approach to the one used in GPT-3 [11]. In detail, we are given a set of support examples in the form of $(image, text)$ or $(video, text)$ (where the *image* or *video* is

the input visual and the *text* is the expected response and any additional task-specific information, e.g., a question) and a single visual query for which we want our model to make a prediction. Given this, we build a multimodal prompt by concatenating the support examples followed by the visual query as illustrated by Figure 8. Unless specified otherwise, we choose the concatenation order at random.

Open-ended and close-ended evaluations. In an open-ended setting, the model’s sampled text following the query image is then taken as its prediction for the image, stopping at the first <EOC> (“end of chunk”) token prediction. Unless specified otherwise, we always use beam search with a beam size of 3. In a close-ended setting, all possible outputs are independently appended to the query image, and we score each of the resulting sequences using the log-likelihood estimated by our model. These scores are then used to rank the candidate outputs in decreasing order, from most confident to least confident.

Zero-shot generalization. In the absence of few-shot examples, approaches commonly rely on prompt engineering [85] to condition the model at inference using a suitable natural language description of the task. Validation of such prompts can significantly impact performance but requires access to a number of annotated examples and cannot therefore be considered truly zero-shot. Furthermore, Perez et al. [80] have shown that such validation procedures are generally not robust with access to only a handful of samples during validation. To report zero-shot performance in our work, we instead build a prompt with *two examples* from the downstream tasks *where we remove their corresponding images or videos*. For example, for the task illustrated at the top of Figure 8, the prompt would be “<BOS>Output: This is a cat wearing sunglasses.<EOC>Output: Three elephants walking in the savanna.<EOC><image> Output:” and no support images would be fed to the model. We observed that only showing one, instead of two, text examples in the prompt is highly detrimental as the model is biased towards producing text output similar to the single provided text example. Providing more than two text examples helps but only marginally. We hence use two text examples in all zero-shot results for practicality. In practice, we believe this is not more cumbersome than finding a good natural text description for a given task. This relates to recent findings on the aspects of demonstrations that are key drivers of performance [76]. For close-ended tasks, where we use the model to score different possible answers, we observe it is not necessary to provide a single text example in the zero-shot prompt.

Retrieval-based In-Context Example Selection [136]. When the size of the support set exceeds a certain limit, it can become difficult to leverage all the examples with in-context learning: first because it becomes excessively expensive to fit all the



图 9: **Training datasets.** Mixture of training datasets of different formats. N corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets, $N = 1$. T is the number of video frames ($T = 1$ for images). H , W , and C are height, width and color channels.

examples in the prompt, and second because there is a risk of poor generalization when the prompt size exceeds the size of the sequence used during training [83]. In such situations, it is appealing to use a form of prompt selection to both limit the sequence length as well as potentially improve the prompt quality which can in turn lead to better performance [63]. In particular, we follow the Retrieval-based In-Context Example Selection (RICES) approach introduced by [136]. In detail, given a query image, we retrieve similar images in the support set by comparing the visual features extracted from our frozen pretrained visual encoder. We then build the prompt by concatenating the top- N most similar examples. Since LMs are sensitive to the ordering in the prompt due to recency bias [148], we order the examples by increasing order of similarity, such that the most similar support example appears right before the query. We notably show the effectiveness of this approach in classification settings with multiple hundreds of classes (see Appendix B.2.1) where we are given one or more images/videos per class, yielding a number of examples that would not otherwise fit in the prompt.

Prompt ensembling. We also explore ensembling the outputs of the model across multiple prompts in the close-ended setting. This can notably be combined with RICES where ensembling can be done over multiple permutations of the ranked nearest neighbors. Specifically, for a given answer, we average the log likelihoods estimated by the model over 6 random permutations of the selected few-shot examples.

A.3 Training dataset details

We train the Flamingo models on a carefully chosen mixture of datasets illustrated in Figure 9 and described next.

A.3.1 M3W collection

The selection and scraping of web pages for $M3W$ follows a similar process to the one used for collecting the *MassiveWeb* dataset [86]. We start by filtering out non-English documents. We also remove those that do not pass internal filters, which identify explicit content across images, videos, and text. We use a custom scraper to extract

salient content from the remaining documents, in the form of plain text interleaved with images, as described in Section 2.4. The text in *M3W* is collected in a similar fashion to that of *MassiveWeb*, but we also collect any images present at the same level in the HTML tree. We discard documents for which the scraping process does not yield any images.

We then apply similar text filtering heuristics, to remove low quality documents and reduce repetition, as well as some image filters to remove images that are too small (either width or height less than 64 pixels), too wide or narrow (aspect ratio greater than 3 in either direction), or unambiguously low quality (e.g. single-colour images). We discard documents that no longer contain any images following this filtering step.

A.3.2 *M3W* image-placement augmentation

During evaluation of Flamingo models, we prompt the model with an image and ask it to generate text for that image. This lends itself to a natural sequencing at inference time in which the image comes before the corresponding text output.

However, the correspondence between images and text in our interleaved M3W dataset (Section 2.4) is in general unknown (and potentially not well-defined in certain cases). As a motivating example, a simple webpage might be structured in either of the following ways:

- (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!

The text-aligned image indices (`indices`) might “ideally” be chosen such that at each point in the text, the index points to the most semantically relevant image for that text – i.e., the next image in example (a), and the previous image in example (b). In the absence of a general way to determine semantic correspondence between text and images on webpages “in the wild”, we make a simplifying assumption that the most relevant image at any given point in the text is either the last image appearing before the text token, or the image immediately following it (as in the simple examples above), and choose `indices` accordingly.

During training, for each webpage sampled, we sample with probability $p_{next} = \frac{1}{2}$ whether `indices` are chosen to map text to the previous or next image. This inevitably means we make the semantically “unnatural” choice – e.g., associating the text “This is my cat!” with the dog image in (a) above – around half of the time. We ablate this choice in Section 3.3, finding a small advantage to setting $p_{next} = \frac{1}{2}$ over either 0 (always the previous image index) or 1 (always the next image index). This suggests that there may be a beneficial “data augmentation” effect to this randomisation.

A.3.3 LTIP and VTP: Visual data paired with text

Along with our interleaved image and text dataset, we use several paired vision and text web datasets for training. One dataset is ALIGN [50], composed of 1.8 billion images paired with alt-text. ALIGN is large, but noisy and limited to images. The images are often poorly described by the corresponding alt-text annotation. For this reason, we augment it with two datasets: LTIP (Long Text & Image Pairs) consists of 312 million images, and VTP (Video & Text Pairs) consists of 27 million short videos (approximately 22 seconds on average). Both datasets are paired with more descriptive captions. For instance, the average number of tokens of an ALIGN text description is 12.4 per image, while it is 20.5 for the LTIP dataset. The LTIP and VTP datasets were collected by crawling fewer than ten websites targeting high-quality and rich image descriptions. These single-image and single-video datasets are preprocessed analogously to the *M3W* data preprocessing described previously, adding the `<image>` tag at the beginning of the sequence (immediately after `<BOS>`), and the `<EOC>` token after the text (before `<EOS>`). We deduplicated these datasets against all our benchmarks (against both the training and the evaluation sets) using image similarity, as detailed in Appendix A.3.4. Datasheets for LTIP and VTP are respectively given in Appendix F.2.1 and Appendix F.2.2.

A.3.4 Dataset deduplication against evaluation tasks

We used an internal deduplication tool to deduplicate our training datasets from our evaluation datasets. This deduplication pipeline relies on a trained visual encoder which maps embedding closer together when they are potential duplicates. Once the image embeddings have been computed, a fast approximate nearest neighbor search is performed on the training images to retrieve duplicate candidates from the validation datasets. For the paired image-text dataset, we have deduplicated our LTIP and ALIGN training images against: ImageNet (train, val), COCO (train, valid, test), OK-VQA (train, valid, test), VQAv2 (train, valid, test), Flickr30k (train, valid, test), VisDial (train, valid, test).

We did not deduplicate our image datasets against VizWiz, HatefulMemes and TextVQA as we performed these evaluations only after having trained our Flamingo models. However, we believe this had no impact on our results as the images from these datasets are unlikely to be scraped from the web; VizWiz images were obtained using a specific mobile app and only available for download, HatefulMemes memes were created by researchers instead of being scraped on the web and finally TextVQA images are from OpenImages.

Note that we did not run the deduplication on the *M3W* dataset as one training example is a full webpage of interleaved paragraph with several images, unlikely to contain

	Requires model sharding	Frozen		Trainable		Total count
		Language	Vision	GATED XATTN-DENSE	Resampler	
<i>Flamingo</i> -3B	✗	1.4B	435M	1.2B (every)	194M	3.2B
<i>Flamingo</i> -9B	✗	7.1B	435M	1.6B (every 4th)	194M	9.3B
<i>Flamingo</i>	✓	70B	435M	10B (every 7th)	194M	80B

表 5: **Parameter counts for Flamingo models.** We focus on increasing the parameter count of the frozen LM and the trainable vision-text GATED XATTN-DENSE modules while maintaining the frozen vision encoder and trainable Resampler to a fixed and small size across the different models. The frequency of the GATED XATTN-DENSE with respect to the original language model blocks is given in parentheses.

images from our benchmark suite. To verify this hypothesis, we have obtained near-duplicate statistics on the 185M individual images from *M3W* and the results are the following: in total, 1314 potential duplicates were found from the validation and test splits of ImageNet, COCO, OK-VQA, VQAv2, Flickr30k and VisDial. Out of the 1314 candidates, only 125 are exact duplicates.

For the video datasets, we did not perform any deduplication of VTP (27M videos) as none of the collected VTP videos were obtained from YouTube or Flickr, which are the sources of all of our video evaluation datasets collected on the Internet.

B Experiments

B.1 Training and evaluation details

B.1.1 Models

We perform experiments across three model sizes, where we scale the frozen language model from 1.4B to 7B and 70B; and adapt the parameter count of other components accordingly. We keep the pretrained vision encoder frozen across all experiments and use a NFNet-F6 model trained contrastively (see Appendix B.1.3), unless explicitly stated otherwise in the ablation study. We use a Perceiver Resampler with approximately 200M parameters across all three model sizes.

The decision on how many GATED XATTN-DENSE layers to interleave is mainly driven by a trade-off between memory constraints and downstream performance. We identified the optimal trade-off at small model scales, before transferring our findings to the large model architecture.

We obtain three models, *Flamingo*-3B, *Flamingo*-9B and *Flamingo*-80B, detailed below:

- The *Flamingo*-3B model builds on top of a **1.4B frozen language model** from [42]. Before each transformer block, we add a GATED XATTN-DENSE layer

attending to the visual inputs; this accounts for 1.4B additional learned parameters.

- The *Flamingo*-9B model builds on top of a **7B frozen language model** from [42]. Starting from the very first layer and before every fourth transformer blocks, we add a GATED XATTN-DENSE layer attending to the visual inputs; this accounts for 1.8B additional learned parameters.
- The *Flamingo*-80B model builds on top of the **frozen Chinchilla 70B** language model [42]. Starting from the very first layer and before every seventh transformer blocks, we add a GATED XATTN-DENSE layer attending to the visual inputs; this accounts for 10B additional learned parameters. For simplicity, we refer to this model as simply *Flamingo* throughout the paper.

In Table 5 we report the parameter count of each component of our models, as well as model sharding requirements. We provide more Transformer architecture details in Appendix A.1.4. The *Flamingo* model card [77] is also given in Appendix E.

B.1.2 Training details for the Flamingo models

Data augmentation and preprocessing. Empirically we find that it is effective to stochastically prepend the paired dataset text samples with a single space character, with probability 0.5. We attribute this to the fact that our subword tokenizer maps the beginning of various words to a different token depending on whether it is preceded by a space. This allows us to enforce invariance to this tokenizer artifact, without degrading significantly correctness of the punctuation which is already lacking in many of these samples. We observe that this leads to substantial improvement across tasks.

The visual inputs are resized to 320×320 while preserving their aspect ratios, padding the image with the mean value if required. Note that this is higher than the 288×288 resolution used for the contrastive pretraining of our Vision Encoder (see Appendix B.1.3). The increase in resolution during the final stage training was motivated by [113] showing one can obtain improved performance at a higher test-time resolution when using CNNs. This increase in resolution also comes with only a moderate computational and memory cost as no backpropagation is performed through the frozen Vision Encoder. We also employ random left/right flips and color augmentation.

For interleaved datasets (Section 2.4) we also employ augmentation by lightly randomizing the selected image indices ϕ with a hyperparameter p_{next} when sampling examples from the *M3W* dataset. This augmentation is detailed in Appendix A.3.2 and our choice of $p_{next} = \frac{1}{2}$ is ablated in Appendix B.3.1. For video training, we temporally sample a clip of 8 frames sampled at one frame per second (fps) from each training video. Although our model was trained with a fixed number of 8 frames, at inference

time, we input 30 frames at 3 FPS. This is achieved by linearly interpolating the learnt temporal position embedding of the Perceiver Resampler at inference time.

Loss and optimisation. All our models are trained using the AdamW optimizer with global norm clipping of 1, no weight decay for the Perceiver Resampler and weight decay of 0.1 for the other trainable parameters. The learning rate is increased linearly from 0 to 10^{-4} up over the first 5000 steps then held constant for the duration of training (no improvements were observed from decaying the learning rate). Unless specified otherwise we train our models for $500k$ steps. Four datasets are used for training: *M3W*, ALIGN, LTIP and VTP with weights λ_m of 1.0, 0.2, 0.2 and 0.03 respectively. These weights were obtained empirically at a small model scale and kept fixed afterwards. Batch sizes depend on the setting and are given in the next sections.

Infrastructure and implementation. Our model and associated infrastructure were implemented using JAX [8] and Haiku [40]. All training and evaluation was performed on TPUv4 instances. The largest model containing 80 billion parameters is trained on 1536 chips for 15 days and sharded across 16 devices. Megatron type sharding [99] is used to enable 16-way model parallelism for all Embedding / Self-Attention / Cross-Attention / FFW layers, while the NFNet vision layers were unsharded. ZeRO stage 1 [88] is used to shard the optimizer state. All trained parameters and optimizer accumulators are stored and updated in `float32`; all activations and gradients are computed in `bfloat16` after downcasting of parameters from `float32` to `bfloat16`. Frozen parameters are stored and applied in `bfloat16`.

B.1.3 Contrastive model details

The vision encoder is trained from scratch, together with a language encoder. Using these encoders, images and text pairs are separately encoded and projected to a shared embedding space and L2 normalized. From these embeddings, we maximize the similarity of paired embeddings and minimize the similarity of unpaired embeddings, using a multi-class cross-entropy loss, where the paired image-texts are treated as positive examples and the rest of the batch as negative examples. We use the same loss as in CLIP [85], which consists of two contrastive losses, one from text to image and the other from image to text. We use a learnable temperature parameter in the final log-softmax layer [9]. The text-to-image loss is as follows:

$$L_{\text{contrastive:txt2im}} = -\frac{1}{N} \sum_i^N \log \left(\frac{\exp(L_i^\top V_i \beta)}{\sum_j^N \exp(L_i^\top V_j \beta)} \right) \quad (3)$$

And the image-to-text loss is defined analogously:

$$L_{contrastive:im2txt} = -\frac{1}{N} \sum_i^N \log \left(\frac{\exp(V_i^\top L_i \beta)}{\sum_j^N \exp(V_i^\top L_j \beta)} \right) \quad (4)$$

The sum of the two losses is minimized. Here, V_i and L_i are, respectively, the normalized embedding of the vision and language component of the i -th element of a batch. β is a trainable inverse temperature parameter and N is the number of elements in the batch. We use the BERT [23] architecture for the language encoder. The outputs of the language and vision encoders are mean-pooled (across tokens and spatial locations, respectively) before being projected to the shared embedding space. We only use the weights from the contrastive vision encoder in the main Flamingo models.

The vision encoder is pretrained on the ALIGN and LTIP datasets. The training image resolution is 288×288 , the joint embedding space is size 1376 and the batch size is 16,384. It is trained for 1.2 million parameter update steps, each of which consist of two gradient calculation steps (more details below) on 512 TPUs v4 chips. The learning rate is decayed linearly from 10^{-3} to zero over the course of training. Images have random color augmentation and horizontal flips applied during training. We use the tokenizer employed by Jia et al. [50]. The Adam optimizer is used to optimize the network, and we apply label smoothing of 0.1. We apply 10^{-2} adaptive gradient clipping (AGC) [10] to the NFNet encoder and global norm gradient clipping of 10 for the BERT encoder.

To evaluate the pretrained model, we track zero-shot image classification and retrieval. For zero-shot image classification, we use image-text retrieval between the images and the class names. Following Radford et al. [85] we use “prompt-ensembling” in which we embed multiple texts using templates such as “A photo of a {class_name}” and average the resulting embedding.

B.1.4 Evaluation benchmarks

Our goal is to develop models that can rapidly adapt to diverse and challenging tasks in the few-shot setting. For this, we consider a wide array of popular image and video benchmarks summarized in Table 6. In total we chose 16 multimodal image/video and language benchmarks, spanning tasks that require some language understanding (visual question answering, captioning, visual dialogue) as well as two standard image and video classification benchmarks (ImageNet and Kinetics). Note that for the video datasets collected from YouTube (i.e., all video datasets except NextQA and STAR), we evaluated our model on all the publicly available video as of April 2022.

dev benchmarks. In order to validate design decisions of our model over the course of the project, we selected five benchmarks from the 16 multimodal image/video and language benchmarks as well as ImageNet and Kinetics for classification as our develop-

	Dataset	DEV	Gen.	Custom prompt	Task description	Eval set	Metric
Image	ImageNet-1k [94]	✓			Object classification	Val	Top-1 acc.
	MS-COCO [15]	✓	✓		Scene description	Test	CIDEr
	VQAv2 [3]	✓	✓		Scene understanding QA	Test-dev	VQA acc. [3]
	OKVQA [69]	✓	✓		External knowledge QA	Val	VQA acc. [3]
	Flickr30k [139]		✓		Scene description	Test (Karpathy)	CIDEr
	VizWiz [35]		✓		Scene understanding QA	Test-dev	VQA acc. [3]
	TextVQA [100]		✓		Text reading QA	Val	VQA acc. [3]
	VisDial [20]				Visual Dialogue	Val	NDCG
Video	HatefulMemes [54]			✓	Méme classification	Seen Test	ROC AUC
	Kinetics700 2020 [102]	✓			Action classification	Val	Top-1/5 avg
	VATEX [122]	✓	✓		Event description	Test	CIDEr
	MSVDQA [130]	✓	✓		Event understanding QA	Test	Top-1 acc.
	YouCook2 [149]		✓		Event description	Val	CIDEr
	MSRVTTQA [130]		✓		Event understanding QA	Test	Top-1 acc.
	iVQA [135]		✓		Event understanding QA	Test	iVQA acc. [135]
	RareAct [73]			✓	Composite action retrieval	Test	mWAP
	NextQA [129]		✓		Temporal/Causal QA	Test	WUPS
	STAR [128]				Multiple-choice QA	Test	Top-1 acc.

表 6: **Summary of the evaluation benchmarks.** DEV benchmarks were used to validate general design decision of the Flamingo models. Gen. stands for generative task where we sample text from the VLM. If a task is non-generative it means that we use the VLM to score answers among a given finite set. For most of our tasks we use a common default prompt, hence minimizing task-specific tuning (see Appendix B.1.5).

ment set (referred as DEV). To maximise its relevance, we choose the most challenging and widely studied benchmarks for captioning, visual question-answering and classification tasks on both images and videos.

Dataset splits for the dev benchmarks. Concretely, estimating few-shot learning performance of a model consists of adapting it on a set of *support* samples and evaluating it on a set of *query* samples. As a result, any evaluation set should be composed of two disjoint subsets containing respectively the support and the query samples. For the DEV benchmarks that are used both to validate design decisions and hyperparameters, as well as to report final performance, we therefore use four subsets:

- *validation support*: contains support samples for validation;
- *validation query*: contains query samples for validation;
- *test support*: contains support samples for final performance estimation;
- *test query*: contains query samples for final performance estimation.

In practice, for the *test query* subset, we use the subset that prior works report results on, for apples-to-apples comparison. While the validation set would be a natural choice for the *validation query* subset, we note that this is not possible for all benchmarks, since some benchmarks do not have an official validation set (e.g. OKVQA) and for others, the validation is commonly used to report final performance in place of the test

set (e.g. ImageNet or COCO). For simplicity, we use a subset of the original training set as the *validation query* subset. Finally, we also use additional disjoint subsets of the training set as respectively the *validation support* subset and the *test support* subset.

We now describe in more detail how we form the latter three subsets. For captioning tasks, open-ended evaluation is efficient so we evaluate on a large number of samples. Specifically, for COCO, we use the same number of samples as used in the Karpathy splits for evaluation sets (5000). For VATEX, because the training set is of limited size, we only evaluate over 1024 samples, reserving the rest for support sets. For question-answering tasks, we evaluate over 1024 samples; chosen to make both open- and close-ended evaluation reasonably fast. For image classification tasks, we evaluate over 10 images per class: 10,000 samples for ImageNet, and 7000 samples for Kinetics700. As for the support sets, for both validation and final performance estimation, we use 2048 samples across all tasks, except for classification tasks where we scale this to 32 samples per class, to better estimate expected performance for each class.

Unbiased few-shot performance estimation. Few-shot learning performance estimates on the DEV benchmarks may be biased, in the sense that over the course of this project, design decisions were made based on the performance obtained on these benchmarks. We note that this is the case for prior work which also make use of these benchmarks to validate and ablate their own design decisions. To account for this bias and provide unbiased few-shot learning performance estimates, we report performance on a remaining set of 11 benchmarks. Among those, some span the same open-ended image and video tasks as our DEV benchmarks (captioning and visual question-answering). But we also look at more specific benchmarks in order to explore less explored capabilities. These notably include: TextVQA [100] which specifically assesses OCR capabilities through question-answering; VisDial [20], a visual dialogue benchmark; Hateful-Memes [54] a vision and text classification benchmark; NextQA [129] which specially focuses on causality and temporal relation; STAR [128], a multiple-choice question answering task; and RareAct [73], a benchmark measuring compositionality in action recognition. We emphasize that *we do not validate any design decisions* on these benchmarks and use them solely to estimate unbiased few-shot learning performance after Flamingo training is done.

B.1.5 Few-shot learning evaluation hyperparameters

In few-shot learning, hyperparameter selection implicitly increases the number of shots as it requires additional validation examples. If those are not taken into account, as is often the case in practice, few-shot performance can be overestimated [80]. Similarly, cross-validation of benchmark-specific hyperparameters such as the prompt should be considered as a particularly basic few-shot learning method, where one selects the task-

specific prompt over the set of shots. But other learning approaches might be more effective in making use of these labelled examples. Given the negative results reported by [80] in terms of the robustness of cross-validation and unless mentioned otherwise, all benchmarks are run using a single set of evaluation hyperparameters, including the prompts. We optimize hyperparameters jointly across the *validation* subsets of the DEV benchmarks and do not perform any benchmark-specific cross-validation of hyperparameters, aside from a few exceptions, as we detail next.

Except for HatefulMemes and RareAct, we always use the prompt “`Output: {output}`” for all non-question-answering tasks, and “`Question: {question} Answer: {answer}`” for all question-answering / visual dialogue tasks. In particular, for VisDial [20], we use the previously described prompt to encode each questions/answers in the dialogue and the provided image caption is prepended to the dialogue history without any prompt. For HatefulMemes [54], we use a specific prompt to incorporate the OCR information provided as input which is: “`is an image with written: "{meme_text}" on it. Is it hateful? Answer: {answer}`”, where the answer is either yes or no. Note that this is the only dataset where we explicitly provide OCR text “`meme_text`” as input to Flamingo models. For TextVQA, we do not make use of the provided OCR transcripts and instead directly rely on the off-the-shelf OCR capabilities of the Flamingo models. For RareAct, a zero-shot benchmark, we change the verb names to the third person, add an article before each noun and use the prompt “`Caption: a person {verb + object}`”.

B.1.6 Dialogue prompt

For the dialogue examples in Figures 1 and 11, the following prompt is used:

This is a conversation between a human, User, and an intelligent visual AI, Flamingo. User sends images, and Flamingo describes them.

User: <a cat image>

Flamingo: That is a cat. It's a tiny kitten with really cute big ears.

User: <a dinner image>

Flamingo: This is a picture of a group of people having dinner. They are having a great time!

User: Can you guess what are they celebrating?

Flamingo: They might be celebrating the end of a successful project or maybe a birthday?

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

表 7: **Few-shot results on classification tasks.** The Flamingo models can also be used for standard classification tasks. In particular, we explore having access to support sets bigger than what our current prompt can accommodate (using up to 5000 support examples). In that regime, large gains are obtained by using the RICES method [136] as well as prompt ensembling. We also observe the same trend as with the vision-language benchmarks: bigger models do better and more shots help.

User: <a graph image>

Flamingo: This is a graph, it looks like a cumulative density function graph.

B.2 Additional performance results

B.2.1 Few-shot learning on classification tasks

We consider applying the Flamingo models to well-studied classification benchmarks like ImageNet or Kinetics700. Results are given in Table 7. We observe a similar pattern as in other experiments: larger model tend to perform better. Second, given that few-shot classification tasks often come with more training examples (e.g., 1000 for ImageNet with 1 example per class), using methods to scale to larger support sets is beneficial. RICES (Retrieval In-Context Example Selection [136] described in Appendix A.2) performs substantially better than simply selecting examples randomly for inclusion in the prompt. Indeed, *Flamingo* achieves a 9.2% improvement in ImageNet classification when selecting 16 support examples out of 5000 using RICES, compared to choosing the same number of examples randomly. Ensembling multiple prompts further boosts results. However, note that Flamingo models underperform the current dominant contrastive paradigm for classification tasks; in particular, they underperform the very contrastive model used as their vision encoder (see Appendix D.1 on Flamingo’s limitations for more details). Finally, state-of-the-art zero-shot models on ImageNet such as BASIC [82] and LiT [146] are particularly optimized on classification

tasks as they are trained on JFT-3B [145], a dataset with images and labels. Improving the performance of VLMs such as Flamingo on classification tasks is an interesting direction for future work.

B.2.2 Fine-tuning *Flamingo* as a pretrained vision-language model

To fine-tune Flamingo models on a downstream task, we train them on data batches from the task of interest in the same format as the single-image/video datasets described in Section 2.4.

Freezing and hyperparameters. When fine-tuning *Flamingo*, we keep the underlying LM layers frozen and train the same Flamingo layers as during pretraining. We also increase the resolution of the input images from 320×320 to 480×480 . Unlike in the pretraining phase, we also fine-tune the base visual encoder, finding that this typically improves results, likely due in part to the higher input resolution.

We choose certain hyperparameters on a per-task basis by grid search on a validation subset of the training set (or on the official or standard validation set where available). These hyperparameters include the learning rate (ranging from 3×10^{-8} to 1×10^{-5}) and decay schedule (exponential decay by factors of $10\times$), number of training steps, batch size (either 8 or 16), and whether visual data augmentation (color augmentation, random horizontal flips) is used.

Results. In Table 8, we present additional results for per-task *Flamingo* fine-tuning. When provided access to a large-scale task-specific dataset with many thousands of examples, we find that we can improve results over our previously presented in-context few-shot learning results, setting a new state of the art on five tasks: VQAv2, VATEX, VizWiz, MSRVTTQA, and HatefulMemes. For example, on VQAv2, we observe improved results at 82.0%, outperforming our results achieved with 32-shot in-context learning (67.3%) as well as the previous state of the art (81.3%, Yan et al. [133]).

Although these fine-tuning results come at high computational cost relative to the previously presented in-context few-shot learning results – among other challenges like hyperparameter tuning – they further demonstrate the power of VLM pretraining for visual understanding even in the presence of large amounts of task-specific training data.

In some cases our results likely trail the state of the art due in part to the fact that we simply optimise log-likelihood and do not make use of common task-specific metric optimisation tricks, such as CIDEr optimisation [64, 90] for COCO captioning, and fine-tuning on dense annotations for VisDial [79]. For example, Murahari et al. [79] report a 10% relative improvement in NDCG on VisDial from such dense annotation fine-tuning.

B.2.3 Zero-shot performance of the pretrained contrastive model

A crucial part of our approach is the Vision Encoder, pretrained separately using contrastive learning and kept frozen when training Flamingo models. We report zero-shot image classification results on ImageNet, Kinetics700 and retrieval results on Flickr30K and COCO. The classification results are presented in Table 7 while the retrieval results are given in Table 9. For the retrieval tasks, our model outperforms the current state-of-the-art contrastive dual encoder approaches CLIP [85], ALIGN [50] and Florence [140]. However, we underperform the zero-shot state-of-the-art on Kinetics700 (CLIP) and the zero-shot state-of-the-art on ImageNet (BASIC). However, as noted earlier, BASIC [82] is particularly optimized for classification: it is trained on the JFT-3B [145] dataset which has images with labels rather than captions. We have noticed training on image and short text descriptions similar to labels significantly helps for ImageNet but is detrimental for retrieval benchmarks which require capturing rich scene descriptions instead. Since our goal is to use the Vision Encoder as a feature extractor for the Flamingo models in order to capture the whole scene and not just the main object, we favor retrieval metrics over classification ones. We provide more details about the contrastive pretraining in Appendix B.1.3.

B.3 Extended ablation studies

B.3.1 Flamingo

Ablation study experimental setup. As in Table 10, we report per-task results and the Overall score (see Section 3.3) for *Flamingo*-3B on the *validation* subsets of the 5 DEV multimodal benchmarks with 4 shots in Table 10. We perform the ablation

表 8: **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* was not SotA with few-shot learning. *Flamingo* sets a new SotA on five of these tasks sometimes even beating methods that resorts to known performance optimization tricks such as model ensembling (on VQAv2, VATEX, VizWiz and HatefulMemes). Best numbers among the restricted SotA are in **bold**. Best numbers overall are underlined. Restricted SotA[†] only includes methods that use a single model (not ensembles) and do not directly optimise the test metric (no CIDEr optimisation).

Method	VQAv2		COCO	VATEX	VizWiz		MSRVTTQA	ViDial		YouCook2	TextVQA		HatefulMemes	
	test-dev	test-std			test	test		test	valid		valid	valid	test-std	
Flamingo - 32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	-	70.0
SimVLM [124]	80.0	80.3	143.3	-	-	-	-	-	-	-	-	-	-	-
OFA [119]	79.9	80.0	<u>149.6</u>	-	-	-	-	-	-	-	-	-	-	-
Florence [140]	80.2	80.4	-	-	-	-	-	-	-	-	-	-	-	-
Flamingo Fine-tuned	82.0	82.1	138.1	84.2	65.7	65.4	47.4	61.8	59.7	118.6	57.1	54.1	86.6	-
Restricted SotA [†]	80.2	80.4	143.3	76.3	-	-	46.8	<u>75.2</u>	74.5	<u>138.7</u>	54.7	73.7	79.1	[62]
[140]	[140]	[124]	[153]	-	-	-	[51]	[79]	[79]	[132]	[137]	[84]	-	-
Unrestricted SotA	81.3	81.3	<u>149.6</u>	81.4	57.2	60.6	-	-	<u>75.4</u>	-	-	-	84.6	[152]
[133]	[133]	[119]	[153]	[65]	[65]	[65]	-	-	[123]	-	-	-	-	-

	Flickr30K						COCO					
	image-to-text			text-to-image			image-to-text			text-to-image		
	R@1	R@5	R@10									
Florence [140]	90.9	99.1	-	76.7	93.6	-	64.7	85.9	-	47.2	71.4	-
ALIGN [50]	88.6	98.7	99.7	75.7	93.8	96.8	58.6	83.0	89.7	45.6	69.8	78.6
CLIP [85]	88.0	98.7	99.4	68.7	90.6	95.2	58.4	81.5	88.1	37.7	62.4	72.2
Ours	89.3	98.8	99.7	79.5	95.3	97.9	65.9	87.3	92.9	48.0	73.3	82.1

表 9: **Zero-shot contrastive pretraining evaluation.** Zero-shot image-text retrieval evaluation of our pretrained contrastive model compared to the state-of-the-art dual encoder contrastive models.

Ablated setting	Flamingo 3B value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
Flamingo 3B model (short training)										
(i) Resampler size	Medium	Small Large	3.1B 3.4B	1.58s 1.87s	86.5 84.4	42.1 42.2	55.8 54.4	36.3 35.1	53.4 51.4	70.7 69.0
(ii) Multi-Img att.	Only last	All previous	3.2B	1.74s	70.0	40.9	52.0	32.1	46.8	63.5
(iii) p_{next}	0.5	1.0	3.2B 3.2B	1.74s 1.74s	85.0 81.3	41.6 43.3	55.2 55.6	36.7 36.8	50.6 52.7	69.6 70.4
(iv) LM pretraining	MassiveText	C4	3.2B	1.74s	81.3	34.4	47.1	60.6	53.9	62.8
(v) Freezing Vision	✓	✗ (random init) ✗ (pretrained)	3.2B 3.2B	4.70s* 4.70s*	74.5 83.5	41.6 40.6	52.7 55.1	31.4 34.6	35.8 50.7	61.4 68.1
(vi) Co-train LM on MassiveText	✗	✓ (random init) ✓ (pretrained)	3.2B 3.2B	5.34s* 5.34s*	69.3 83.0	29.9 42.5	46.1 53.3	28.1 35.1	45.5 51.1	55.9 68.6
(vii) Dataset and Vision encoder	M3W+ITP+VTP and NFNetF6	LAION400M and CLIP M3W+LAION400M+VTP and CLIP	3.1B 3.1B	0.86s 1.58s	61.4 76.3	37.9 41.5	50.9 53.4	27.9 32.5	29.7 46.1	54.7 64.9

表 10: **Additional ablation studies.** Each row in this ablation study table should be compared to the baseline Flamingo run reported at the top of the table. The step time measures the time spent to perform gradient updates on all training datasets. (*): Due to higher memory usage, these models were trained using four times more TPU chips. The obtained accumulation step time was therefore multiplied by four.

using batch size of 256 for $M3W$, 512 for ALIGN, 512 for LTIP and 64 for VTP. Models are trained for 1 million gradient steps (meaning 250,000 gradient updates, for the base model as we accumulate gradients over four datasets).

Resampler size. We further investigate the architectural design of the Resampler in row (i) of Table 10. We ablate the size of our Resampler with three options: Small, Medium (default value for all Flamingo models), and Large. We see that the best performance is achieved with a medium size Resampler. Moreover, when scaled together with the frozen LM, we observed that increasing the size of the Perceiver Resampler lead to unstable training. We thus made a conservative choice to keep the same medium Resampler size for all our Flamingo models.

Effect of how many images are cross-attended to. In the interleaved image-text scenario, we ablate whether the model can only attend to the single most recent previous image, or to all the previous images (row (ii) of Table 10). We can see that the single image case leads to significantly better results (7.2% better in the overall score). One potential explanation is that when attending to all previous images, there is no explicit way of disambiguating between different images in the cross-attention

inputs. Nonetheless, recent work has shown that such disambiguation is still possible implicitly through the causal attention mechanism [36]. We also explored more explicit ways to enable this while attending to all previous images by modifying the image tags to include an index (<image 1>, <image 2>, etc.) and/or learning absolute index embeddings added to the cross-attention features for each image. These strategies were not as robust as our method when the number of images per sequence changes between training and test time. Such a property is desirable to reduce the number of images per sequence during training for better efficiency (we use $N = 5$ at training time) while still generalizing to many images for few-shot evaluation (we go up to $N = 32$ at test time). For these reasons, we keep the single image cross-attention strategy for the Flamingo models. Note that while the model cannot explicitly attend to all previous images due to this masking strategy, it can still implicitly attend to them from the language-only self-attention that propagates all previous images’ features via the previous text tokens.

***M3W* image placement data augmentation.** Given a webpage, we don’t know in advance if the text of the page will mention the previous or the next image in the two-dimensional layout of the page DOM. For this reason, we explore a data augmentation on *M3W* controlled by p_{next} which indicates whether a given text token attends to the previous or the next image (see more details in Appendix A.3.2). The default value $p_{next} = \frac{1}{2}$ means that for each webpage sampled, we decide uniformly at random whether the model attends to the previous or next image. $p_{next} = 0$ means the model always attends to the previous image while $p_{next} = 1$ means the model always attends to the following image. The results (row (iii) of Table 10) show that using this randomization is beneficial.

Language model pretraining. To measure the importance of text pretraining, we compare the performance of using a frozen decoder-only Transformer either pretrained on MassiveText (our main model) or pretrained on the C4 dataset [87] (row (iv) of Table 10). Using the C4 dataset (which is smaller and less filtered than MassiveText) for training leads to a significant loss in performance (-7.9% overall). We note that the performance notably decreases for tasks that involve more language understanding such as visual question-answering tasks (OKVQA, VQAv2 and MSVDQA) while it remains on par for tasks that do not require as much language understanding (COCO, VATEX). This highlights the importance of pretraining the LM on a high-quality text-only dataset.

Freezing the vision encoder. During Flamingo training, we freeze the pretrained components (Vision Encoder and LM layers) while training newly added components from scratch. We ablate in (v) of Table 10 this freezing decision by training the Vision Encoder weights either from scratch or initialized with the contrastive vision-language

task. If trained from scratch, we observe that the performance decreases by a large margin of -9.3% . Starting from pretrained weights still leads to a drop in performance of -2.6% while also increasing the compute cost of the training.

Alternative to freezing the LM by co-training on MassiveText. Another approach for preventing catastrophic forgetting is to co-train on MassiveText [86], the dataset that was used to pretrain the language model. Specifically, we add MassiveText to the training mixture, with a weight λ_m of 1.0 (best performing after a small grid search), using a sequence length of 2048 and the exact same setting as the pretraining of Chinchilla [42] for computing the text-only training loss. In order to co-train on MassiveText, we need to unfreeze the language model but we keep the vision encoder frozen. We perform two ablations in row **(vi)** of Table 10: starting from a pretrained language model (with a learning rate multiplier of 0.1 of the LM weights) versus initializing from scratch (with the same learning rate everywhere). In both cases, the overall scores are worse than our baseline which starts from the language model, pretrained on MassiveText, and is kept frozen throughout training. This indicates that the strategy of freezing the language model to avoid catastrophic forgetting is beneficial. Even more importantly, freezing the LM is computationally cheaper as no gradient updates of the LM weights are required and we do not need to train on an additional dataset. This computational argument is even more relevant for our largest model, *Flamingo-80B*, where we freeze almost 90% of the overall weights.

Additional experiments using the LAION400M dataset. In order to provide reference numbers that are more easily reproducible using publicly available datasets and network weights we also provide two additional ablations using the CLIP ViT L-14 weights [85] and the LAION400M dataset [96] in rows **(vii)** of Table 10.

B.3.2 Dataset mixing strategies for the contrastive pretraining

Dataset	Combination strategy	ImageNet accuracy top-1	COCO					
			image-to-text			text-to-image		
			R@1	R@5	R@10	R@1	R@5	R@10
LTIP	None	40.8	38.6	66.4	76.4	31.1	57.4	68.4
ALIGN	None	35.2	32.2	58.9	70.6	23.7	47.7	59.4
LTIP + ALIGN	Accumulation	45.6	42.3	68.3	78.4	31.5	58.3	69.0
LTIP + ALIGN	Data merged	38.6	36.9	65.8	76.5	15.2	40.8	55.7
LTIP + ALIGN	Round-robin	41.2	40.1	66.7	77.6	29.2	55.1	66.6

表 11: **Effect of contrastive pretraining datasets and combination strategies.** The first two rows show the effect of training a small model on LTIP and ALIGN only; the final three show the results of a small model trained on combinations of these datasets, comparing different combination strategies.

One key to achieving strong results was the inclusion of our new dataset LTIP alongside ALIGN for training. Despite being a smaller dataset ALIGN by a factor of 6, a contrastive model trained on only LTIP outperforms one trained only on ALIGN on our evaluation metrics, suggesting that dataset quality may be more important than scale in the regimes in which we operate. We also find that a model trained on both ALIGN and LTIP outperforms those trained on the two datasets individually and that how the datasets are combined is important.

To demonstrate this, we train a small model with an NFNet-F0 vision encoder, BERT-mini language encoder and batch size 2048 for 1 million gradient-calculation steps on ALIGN, LTIP and a mixture of the two. The results are presented in Table 11. It shows the results of training models on the combined datasets using three different merging regimes:

- Data merged: Batches are constructed by merging examples from each dataset into one batch.
- Round-robin: We alternate batches of each dataset, updating the parameters on each batch.
- Accumulation: We compute a gradient on a batch from each dataset. These gradients are then weighted and summed and used to update the parameters.

Across all evaluation metrics, we find that the Accumulation method outperforms other methods of combining the datasets. Although the LTIP dataset is $5 \times$ smaller than the ALIGN dataset, this ablation study suggests that the quality of the training data can be more important than its abundance.

C Qualitative results

In addition to the samples in Figure 1, in this section we provide selected samples covering different interaction modalities in Figures 10, 11, and 12. Unlike the quantitative benchmark results which use beam search with a beam width of 3 for decoding, all qualitative results presented in this section use greedy decoding for faster sampling.

Figure 10 shows the simplest form of interaction where a single image is provided followed by a text prompt either in the form of a question or the start of a caption. Even though the model is not trained specifically for the question and answer format, the capabilities of the pretrained language model allows this adaptation. In many of these examples, *Flamingo* can do at least one step of implicit inference. Some of the objects are not named in the prompt but their properties are queried directly. Based on its visual input, the model manages to recall the knowledge relevant to the referred object and thus produces the correct answer. Vision networks trained contrastively have

been shown to learn character recognition capabilities [85]. We observe that *Flamingo* preserves this capability in the full model, in some cases for text that is rather small with respect to the size of the image.

Since our model can accept inputs in the form of arbitrary sequences of visuals and language, we test its abilities to hold an extended dialogue with interleaved images and text. Figure 11 shows some samples which are generated by prompting the model with a brief dialogue (Appendix B.1.6) followed by user interaction including image insertions. Even after several rounds of interaction *Flamingo* can still successfully attend to the image and reply to questions that can not be guessed by language alone. We observe that multiple images can be separately attended: simple comparisons and inferences are handled properly.

Lastly, we investigated similar capabilities with video inputs as they present some extra challenges compared to images. Figure 12 shows some selected samples. As seen in the figure, in some cases *Flamingo* can successfully integrate information from multiple frames (e.g., videos scanning through a scene or text) and answer questions involving temporal understanding (e.g., in the last example, with the word “after”).

D Discussion

D.1 Limitations, failure cases and opportunities

Here, we describe some limitations and failure cases of our models, as well as opportunities for further improving our models and extending their abilities.

Classification performance. Although our visual language models have important advantages over contrastive models (e.g., few-shot learning and open-ended generation capabilities), their performance lags behind that of contrastive models on classification tasks. We believe this is because the contrastive training objective directly optimizes for text-image retrieval, and in practice, the evaluation procedure for classification can be thought of as a special case of image-to-text retrieval [85]. This is not the case for the language modeling objective we use to train our visual language models and this may contribute to the observed performance gap on classification tasks. In particular, Zhao et al. [148] have shown that language models suffer from various biases arising from the training data distribution, the set of samples used in the prompt, and their order. They also show that such issues can be mitigated with calibration techniques, provided one can assume a certain prior distribution (e.g., uniform) over the label space. This assumption doesn’t hold in general, and further research is needed to develop techniques to address these issues in the few-shot setting. More generally, seeking objectives, architectures, or evaluation procedures that could bridge the gap between these two classes of models is a promising research direction.

<p>Input Prompt</p>  <p>Question: What do you think the capacities of these are? Answer:</p>	 <p>Question: What is odd about this image? Explain why it is unusual. Answer:</p>	 <p>Question: What country is this? Why do you think so? Answer:</p>
<p>Completion</p> <p>The floppy disk is 1.44MB and the CD is 700MB.</p>	<p>The image is odd because the elephant is in the back of the truck. It is unusual because elephants are not usually transported in the back of a truck.</p>	<p>It is Canada. I think so because the flag is the Canadian flag.</p>
<p>Input Prompt</p>  <p>Question: What nutrients is the fruit in the image rich in? Answer:</p>	 <p>Question: What is the purpose of this? Answer:</p>	 <p>Question: What season is this? Explain why. Answer:</p>
<p>Completion</p> <p>Vitamin C, potassium, and fiber.</p>	<p>To keep the mosquitoes out of the bed.</p>	<p>This is winter season. The Christmas tree and the presents are the main reasons why it is winter.</p>
<p>Input Prompt</p>  <p>Question: What is the title of the book? Answer:</p>	 <p>The ingredients of this dish are</p>	 <p>The soundtrack includes</p>
<p>Completion</p> <p>The House Book.</p>	<p>: beef, potatoes, carrots, broccoli, and lemon.</p>	<p>the songs "Let It Go" and "For the First Time in Forever" by Tony Award winner Idina Menzel, who plays Elsa [...]</p>

图 10: Selected single image samples. Gray boxes are user input and the pink boxes are *Flamingo* output.



This is a dog. It's a white fluffy dog.

What season is this?

It looks like a summer day.

Why do you think so?

The grass is green and the dog is not wearing a sweater.

What breed is the dog?

It's a Samoyed.

Is the dog running?

No, it's sitting.

Can you describe the pose of its back legs?

The back legs are bent and the dog is sitting on its haunches.



Which city is this?

This is a street sign in New York City.

Which street corner is it?

It is the intersection of Madison Avenue and East 42nd Street.

Which subway station is nearby?

The closest subway station is Grand Central Station.

How about the nearest park?

The nearest park is Bryant Park.

There is another sign. What does it say?

It says "One way".

I would like you to read the color and then say the color it is written in. For example:

PURPLE

Color is "Purple" and it is written in red.

GREEN

Color is "Green" and it is written in blue.

YELLOW

Color is "Yellow" and it is written in green.

BLACK

Color is "Black" and it is written in yellow.

Well done! Do you know the name of the test these images come from?

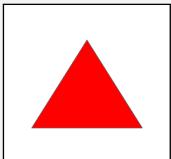
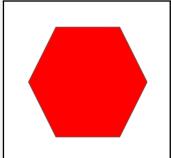
I think it is called the Stroop test.

Can you explain how humans perform in this test?

Humans are slower when the color of the word and the color of the word are different.

How about you?

I am not affected by this difference.

What is the difference between these two images?

The first one is a triangle and the second one is a hexagon.

Which one has larger internal angles?

The hexagon has larger internal angles.

What color are they?

They are both red.



What is in this picture?

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

What is the monster made out of?

It's made out of vegetables.

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

It's made out of a woolen fabric.



What happened here?

It looks like the dog is destroying the sofa.

How does the person thinking about it?

The person is angry.

What is the person holding?

It looks like a cup of coffee.

How is the weather outside?

It looks like a sunny day.



What is the name of the company on this card?

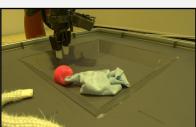
It's Homebase.

What is its web url?

It's homebase.co.uk.

What are they doing?

They are a home improvement retailer.



What do you think the robot is doing here?

The robot is picking up a ball.

What color is the ball?

It's red.

I want the ball to be hidden. Is there something the robot can use for that?

The robot can use a cloth to hide the ball.

What color is the cloth?

The cloth is blue.

图 11: Selected dialogue samples. Gray boxes are user input and the pink boxes are *Flamingo* output. For dialogue, *Flamingo* is provided with a custom prompt (hidden from the visualization but shown in Appendix B.1.6) containing a dialogue with 3



图 12: Selected video samples. These are all of the frames the model sees. (Best viewed with zoom.)

Legacies of language models. Our models build on powerful pretrained causal language models, and as a side effect, directly inherit their weaknesses. For instance, causal modeling of the conditioning inputs is strictly less expressive than bidirectional modeling. In this direction, recent work has shown that non-causal masked language modeling adaptation [120] followed by multitask fine-tuning [95, 125, 131] can efficiently improve the zero-shot performance of causal decoder-only language models. Furthermore, transformer-based language models tend to generalize poorly to test sequences significantly longer than the training ones [83]. In settings where the expected text output is too long, the ability of the models to leverage enough shots for few-shot learning can be affected. For instance, for the VisDial dataset [20], a single shot consists of an image followed by a long dialogue composed of 21 different sentences. A sequence of 32 VisDial shots is thus composed of at least $32 \times 21 = 672$ sentences, which in practice means that the prompt length ranges from 4096 to 8192 tokens. This is significantly longer than the maximum sequence length (2048) our LMs have been trained on [42]. To this end, we have capped our reported results on VisDial at 16 shots. On another note, while our ablations demonstrate the importance of the language model priors in-

Input Prompt  <p>Question: What is on the phone screen? Answer:</p>	 <p>Question: What can you see out the window? Answer:</p>	 <p>Question: Whom is the person texting? Answer:</p>
Output  <p>A text message from a friend.</p>	<p>A parking lot.</p>	<p>The driver.</p>

图 13: **Hallucinations and ungrounded guesses in open-ended visual question answering.** *Left:* The model occasionally hallucinates by producing answers that seem likely given the text only, but are wrong given the image as additional input. *Middle:* Similar hallucinations can be provoked by adversarially prompting the model with an irrelevant question. *Right:* A more common pitfall arises when the model makes ungrounded guesses when the answer cannot be determined based on the inputs. Few-shot examples and more sophisticated prompt design may be used to mitigate these issues. More broadly, addressing these issues is an important research direction towards improving our models’ applications in open-ended visual dialogue settings.

herited from frozen language models, we suspect that they may play a role in occasional hallucinations and ungrounded guesses observed in open-ended dialogue settings. We provide and analyze examples of such behaviours in Figure 13. Finally, language modeling suffers from poor sample efficiency during pretraining [11]. Mitigating this issue has the potential to greatly accelerate progress in the field, by improving turnaround of large-scale training runs and in turn increasing the feasibility of more systematic exploration of design decisions at larger scales. Further discussion on typical weaknesses observed for large LMs can be found in [11, 86].

Trade-offs of few-shot learning methods. In the paper, we use in-context learning as our “go-to” few-shot learning method (see Section 2.5). This method has notable advantages over gradient-based approaches such as fine-tuning. Indeed, in-context learning requires almost no hyperparameter tuning, works reasonably well in the very low data regime (dozens of examples), and only requires inference, simplifying deployment. In contrast, gradient-based approaches require carefully tuned design choices to avoid overfitting (either by proper learning rate schedule or architecture design [43]) and often need more data (thousands) to work well. This motivated our focus on in-context learning; however, this approach also has drawbacks we discuss next.

Inference compute cost. The compute cost of in-context learning with transformer models scales linearly with the number of shots if one can reuse the few-shot prompt for multiple query samples (by caching the keys and values) and quadratically otherwise.

In contrast, gradient-based few-shot learning approaches [43] have constant complexity with respect to the number of shots during inference.

Prompt sensitivity. In-context learning has also been shown to be disconcertingly sensitive to various aspects of the demonstrations, such as the order of the samples [148] or their format.

Leveraging more shots. When using in-context learning, performance plateaus rapidly as the number of few-shot samples increases beyond 32. This proves a striking contrast with typical gradient-based methods, for which the amount of correctly paired training data is a critical factor for performance. We note that RICES (Retrieval In-Context Example Selection [136] described in Appendix A.2) effectively mitigates this issue for classification tasks (Appendix B.2.1), but still faces similar issues beyond a small number of example per class.

Task location. Recent work on understanding what makes in-context learning effective sheds some light on a possible explanation for why more shots do not always help [76, 92]. In more detail, Brown et al. [11] raise the question of whether in-context learning actually “learns” new tasks at inference time based on the provided input-output mappings, or simply recognizes and identifies tasks learned during training. On this question, the findings of Reynolds and McDonell [92] suggest that the latter is the key driver of performance across diverse settings, and refer it as *task location*. Similarly, Min et al. [76] show that the mapping from input to output generally has limited impact on few-shot performance, as opposed to specifying the overall format of the examples. In line with these findings, we also observe non-trivial zero-shot performance using prompt without any images, hence also highlighting that the format of the task matters significantly. Intuitively, a handful of samples may often be enough to perform task location well, but the model may generally not be able to leverage further samples at inference time to refine its behaviour.

In summary, there is no “golden” few-shot method that would work well in all scenarios. In particular, the best choice of few-shot learning approach strongly depends on characteristics of the application, an important one being the number of annotated samples. On this point, in our work, we demonstrate that in-context learning is highly effective in the data-starved regime (32 samples or fewer). There may be opportunities to combine different methods to leverage their complementary benefits, in particular when targeting less data-constrained data regimes (e.g., hundreds of samples).

Extending the visual and text interface. Natural language is a powerful and versatile input/output interface to provide descriptions of visual tasks to the model and generate outputs or estimate conditional likelihoods over possible outputs. However, it may be a cumbersome interface for tasks that involve conditioning on or predicting

more structured outputs such as bounding boxes (or their temporal and spatio-temporal counterparts); as well as making spatially (or temporally and spatio-temporally) dense predictions. Furthermore, some vision tasks, such as predicting optical flow, involve predicting in continuous space, which is not something our model is designed to handle out of the box. Finally, one may consider additional modalities besides vision that may be complementary, such as audio. All of these directions have the potential to extend the range of tasks that our models can handle; and even improve performance on the ones we focus on, thanks to synergies between the corresponding abilities.

Scaling laws for vision-language models. In this work, we scale Flamingo models up to 80B parameters and provide some initial insights on their scaling behaviour across evaluation benchmarks, summarized in Figure 2. In the language space, an important line of work has focused on establishing scaling laws for language models [42, 53]. In the vision domain, Zhai et al. [145] take a step in this direction. Similar efforts have yet to be made for vision-language models, including contrastive models, as well as visual language models such as the ones we propose. While language modeling scaling law research has focused on perplexity as the golden metric, we speculate that it may be more directly useful for our purposes to establish such trends in terms of aggregate downstream evaluation task performance.

D.2 Benefits, risks and mitigation strategies

D.2.1 Benefits

Accessibility. A system like Flamingo offers a number of potential societal benefits, some of which we will discuss in this section. Broadly, the fact that Flamingo is capable of task generalisation makes it suitable for use cases that have not been the focus of vision research historically. Typical vision systems are trained to solve a particular problem by training on large databases of manually annotated task-specific examples, making them poorly suited for applications outside of the narrow use cases for which they were deliberately trained. On the other hand, Flamingo is trained in a minimally constrained setting, endowing it with strong few-shot task induction capabilities. As we've shown in our qualitative examples (Appendix C), Flamingo can also be used through a "chat"-like interface for open-ended dialogue. Such capabilities could enable non-expert end users to apply models like Flamingo even to low-resource problems for which little to no task-specific training data has been collected, and where queries might be posed in a variety of formats and writing styles. In this direction, we have shown that *Flamingo* achieves strong performance on the VizWiz challenge¹, which promotes visual recognition technologies to assist visually impaired people. A dialogue interface could also promote better understanding and interpretability of visual language models.

¹<https://vizwiz.org/>

It could help highlight issues with bias, fairness, and toxicity the model may pick up on from the training data. Overall, we believe that Flamingo represents an important step towards making state-of-the-art visual recognition technology more broadly accessible and useful for many diverse applications.

Model recycling. From a modeling perspective, although Flamingo is computationally expensive to train, it importantly leverages pretrained frozen language models and visual encoders. We demonstrated that new modalities can be introduced into frozen models, thereby avoiding expensive retraining. As such models continue to grow in size and computational demands, “recycling” them will become increasingly important from an environmental perspective (as well as a practical one), as described in Larochelle [55] and explored in Strubell et al. [105] for language models. We hope such results may inspire further research into how existing models can be repurposed efficiently rather than trained from scratch.

D.2.2 Risks and mitigation strategies

This section provides some early investigations of the potential risks of models like Flamingo. This study is preliminary and we foresee that further research efforts should be undertaken to better assess those risks. We also discuss potential mitigation strategies towards safely deploying these models. Note that as explained in our Model Card [77] in Appendix E, this model was developed for research purposes only and should not be used in specific applications before proper risk analyses are conducted and mitigation strategies are explored.

By construction, *Flamingo* inherits the risks of Large LMs. Recall that a large part of our model is obtained by freezing the weights of an existing language model [42]. In particular, if provided with no images *Flamingo* falls back to language model behavior. As such *Flamingo* is exposed to the same risks of large language models: it can output potentially offensive language, propagate social biases and stereotypes, as well as leaking private information [126]. In particular, we refer to the analysis presented in the Chinchilla paper (Hoffmann et al. [42], Section 4.2.7) in terms of gender bias on the Winogender dataset [93] which demonstrate that even though this model is less biased towards gender than previous models [86], gender biases are still present. In terms of unprompted toxicity, we also refer to the analysis from Chinchilla [42] which highlights that overall the propensity of the model to produce toxic outputs when not prompted to do so is rather low, as measured by computing the *PerspectiveAPI* toxicity score on 25,000 samples. Weidinger et al. [126] detail possible long-term mitigation strategies for these risks. They include social or public policy interventions, such as the creation of regulatory frameworks and guidelines; careful product design, for instance relating to user interface decisions; and research at the intersection between AI Ethics

	female - male = Δ	CIDEr difference darker - lighter = Δ	CIDEr overall
AoANet [46]	-	+0.0019	1.198
Oscar [61]	-	+0.0030	1.278
<i>Flamingo</i> , 0 shot	$0.899 - 0.870 = +0.029$ ($p = 0.52$)	$0.955 - 0.864 = +0.091$ ($p = 0.25$)	0.843
<i>Flamingo</i> , 32 shots	$1.172 - 1.142 = +0.030$ ($p = 0.54$)	$1.128 - 1.152 = -0.025$ ($p = 0.76$)	1.138

表 12: **Bias evaluation of *Flamingo* for COCO captioning.** We report results on the COCO dataset splits over gender and skin tone provided by Zhao et al. [147].

and NLP, such as building better benchmarks and improving mitigation strategies. In the short term, effective approaches include relying on prompting to mitigate any biases and harmful outputs [86]. Next, we explore the additional risks incurred by Flamingo’s additional visual input capabilities.

Gender and racial biases when prompted with images. Previous work has studied biases that exist in captioning systems [37, 147]. Such modeling biases can result in real-world harms if deployed without care. For AI systems to be useful to society as a whole, their performance should not depend on the perceived skin tone or gender of the subjects – they should work equally well for all populations. However, current automatic vision system performance has been reported to vary with race, gender or when applied across different demographics and geographic regions [12, 21, 97]. As a preliminary study assessing how Flamingo’s performance varies between populations, we follow the study proposed in Zhao et al. [147] and report how the captioning performance of our model varies on COCO as a function of gender and race. Note that we use a different evaluation protocol from the one proposed by Zhao et al. [147]; in that work, they measure results across 5 pretrained models and compute confidence intervals across aggregated per-model scores. Here, we have just one copy of our model (due to its high training cost), and we instead perform statistical tests on the per-sample CIDEr scores across the splits from Zhao et al. [147]. We report the results in Table 12.

Overall, when comparing the CIDEr scores aggregated among images labeled as *female* versus *male*, as well as when comparing *darker skin* versus *lighter skin*, we find there are no statistically significant differences in the per-sample CIDEr scores. To compare the two sets of samples, we use a two-tailed *t*-test with unequal variance, and among the four comparisons considered, the lowest *p*-value we find is $p = 0.25$, well above typical statistical significance thresholds (e.g. a common rejection threshold might be $p < \alpha = 0.05$). This implies that the differences in scores are indistinguishable from random variation under the null hypothesis that the mean scores are equal. We note that a failure to reject the null hypothesis and demonstrate a significant difference does not imply that there are no significant differences; it is possible that a difference exists that could be demonstrated with larger sample sizes, for example. However, these preliminary results are nonetheless encouraging.

Toxicity when prompted with images. We also evaluate the toxicity of *Flamingo* using the *Perspective API*² to evaluate the toxicity of the model’s generated captions when prompted with images from the COCO test set. We observe that some captions are labelled as potentially toxic by the classifier; however, when examining them manually, we do not observe any clear toxicity – output captions are appropriate for the images provided. Overall, based on our own experiences interacting with the system throughout the course of the project, we have not observed toxic outputs when given “safe-for-work” imagery. However this does not mean the model is incapable of producing toxic outputs, especially if probed with “not-safe-for-work” images and/or toxic text. A more thorough exploration and study would be needed if such a model were put in production.

Applying Flamingo for mitigation strategies. Thanks to its ability to rapidly adapt in low-resource settings, *Flamingo* could itself be applied in addressing some of the issues described above. For instance, following Thoppilan et al. [111], adequately conditioned or fine-tuned Flamingo models could be used for filtering purposes of toxic or harmful samples in the training data. In their work, they observe significant improvements relating to safety and quality when fine-tuning on the resulting data. Furthermore, during evaluation, such adapted models could be used to down-rank or exclude outputs that might be classified as offensive, promoting social biases and stereotypes or leaking private information, thus accelerating progress in this direction even for low-resource tasks. Our results on the HatefulMemes benchmark represent a promising step in this direction. Recent work in the language modeling space has also shown success in training an LM to play the role of a “red team” and generate test cases, so as to automatically find cases where another target LM behaves in a harmful way [81]. A similar approach could be derived for our setting. Enabling the model to support outputs with reference to particular locations within the visual inputs, or to external verified quotes is also an interesting direction [72, 111]. Finally, in Figure 11, we provide qualitative examples demonstrating that Flamingo can explain its own outputs, suggesting avenues to explainability and interpretability using the model’s text interface.

E *Flamingo* Model Card

We present a model card for Flamingo in Table 13, following the framework presented by Mitchell et al. [77].

Model Details

²<https://perspectiveapi.com/>

Model Date	March 2022
Model Type	Transformer-based autoregressive language model, conditioned on visual features from a convnet-based encoder. Additional transformer-based cross-attention layers incorporate vision features into the language model’s text predictions. (See Section 2 for details.)

Intended Uses

Primary Intended Uses	The primary use is research on visual language models (VLM), including: research on VLM applications like classification, captioning or visual question answering, understanding how strong VLMs can contribute to AGI, advancing fairness and safety research in the area of multimodal research, and understanding limitations of current large VLMs.
Out-of-Scope Uses	Uses of the model for visually conditioned language generation in harmful or deceitful settings. Broadly speaking, the model should not be used for downstream applications without further safety and fairness mitigations specific to each application.

Factors

Card Prompts – Relevant Factor	Relevant factors include which language is used. Our model is trained on English data. Our model is designed for research. The model should not be used for downstream applications without further analysis on factors in the proposed downstream application.
Card Prompts – Evaluation Factors	<i>Flamingo</i> is based on Chinchilla (a large proportion of the weights of Chinchilla are used as this) and we refer to the analysis provided in [42, 86] for the language only component of this work. We refer to our study presented in Appendix D.2.2 for a toxicity analysis when the model is conditioned on an image.

Metrics

Model Performance Measures	We principally focus on the model’s ability to predict relevant language when given an image. For that we used a total of 18 different benchmarks described in Appendix B.1.4 spanning various vision and language tasks such as classification (ImageNet, Kinetics700, HatefulMemes), image and video captioning (COCO, VATEX, Flickr30K, YouCook2, RareAct), visual question answering (OKVQA, VizWiz, TextVQA, VQAv2, MSRT-TQA, MSVDQA, iVQA, STAR, NextQA) and visual dialog (VisDiag). This was tested either in an open ended setting where <i>Flamingo</i> generate language and we compare the outputs with the ground truth or in a close ended setting where we directly score various outcomes using the likelihood of the model.
Decision thresholds	N/A
Approaches to Uncertainty and Variability	Due to the costs of training <i>Flamingo</i> , we cannot train it multiple times. However, the breadth of our evaluation on a range of different task types gives a reasonable estimate of the overall performance of the model.

Evaluation Data

Datasets	See Table 6 for a detailed list.
Motivation	We chose our evaluation datasets to span an important range of vision and language tasks to correctly assess the ability of <i>Flamingo</i> to produce relevant text given an image.
Preprocessing	Input text is tokenized using a SentencePiece tokenizer with a vocabulary size of 32,000. Images are processed so that their mean and variance are 0 and 1 respectively.

Training Data

See [50], the Datasheet in Appendix F.1, Appendix F.2.1, Appendix F.2.2

Quantitative Analyses

Unitary Results	<p><i>Flamingo</i> sets a new state of the art in few-shot learning on a wide range of open-ended vision and language tasks. On the 16 tasks we consider, <i>Flamingo</i> also surpasses the fine-tuned state-of-art in 6 of the cases despite using orders of magnitude less task-specific training data. We refer to Section 3 for the full details of our quantitative study.</p>
Intersectional Results	We did not investigate intersectional biases.

Ethical Considerations

Data	The data is sourced from a variety of sources, some of it from web content. Sexually explicit content is filtered out, but the dataset does include racist, sexist or otherwise harmful content.
Human Life	The model is not intended to inform decisions about matters central to human life or flourishing.
Mitigations	Apart from removing sexual explicit content we did not filter out toxic content, following the rationale of Rae et al. [86]. More work is needed on mitigation approaches to toxic content and other types of risks associated with language models, such as those discussed in Weidinger et al. [126].
Risks and Harms	The data is collected from the internet, and thus undoubtedly toxic and biased content is included in our training dataset. Furthermore, it is likely that personal information is also in the dataset that has been used to train our models. We defer to the more detailed discussion in Weidinger et al. [126].
Use Cases	Especially fraught use cases include the generation of factually incorrect information with the intent of distributing it or using the model to generate racist, sexist or otherwise toxic text with harmful intent. Many more use cases that could cause harm exist. Such applications to malicious use are discussed in detail in Weidinger et al. [126].

表 13: **Flamingo Model Card.** We follow the framework presented in Mitchell et al. [77].

F Datasheets

F.1 M3W dataset

We follow the framework defined by Gebru et al. [30] and provide the datasheet for *M3W* in Table 14.

Motivation	
For what purpose was the dataset created? Who created the dataset? Who funded the creation of the dataset?	The dataset was created for pre-training vision-language models and was created by researchers and engineers.
Any other comments?	None.
Composition	
What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?	All instances of the dataset are documents from the web containing interleaved text and images.
How many instances are there in total (of each type, if appropriate)?	There are 43.3M instances (documents) in total, with a total of 185M images and 182 GB of text.
Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?	The dataset is a sample from a larger set.
What data does each instance consist of?	Each instance is made up of a sequence of UTF-8 bytes encoding the document's text, as well as a sequence of integers indicating the positions of images in the text, and the images themselves in compressed format (see Section 2.4).

Is there a label or target associated with each instance?	No, there are no labels associated with each instance.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	There are no relationships between the different instances in the dataset.
Are there recommended data splits?	We use random splits for the training and development sets.
Are there any errors, sources of noise, or redundancies in the dataset?	There is significant redundancy at the sub-document level.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	The dataset likely contains some data that might be considered offensive, insulting or threatening, as such data is prevalent on the web. We do not try to filter out such content, with the exception of explicit content, which we identify using dedicated filter.

Collection Process

How was the data associated with each instance acquired?	The data is available publicly on the web.
What mechanisms or procedures were used to collect the data?	The data was collected using a variety of software programs to extract and clean the raw text and images.
If the dataset is a sample from a larger set, what was the sampling strategy?	We randomly subsample documents.

Over what timeframe was the data collected?	The dataset was collected over a period of several months in 2021. We do not filter the sources based on creation date.
---	---

Were any ethical review processes conducted?	No.
--	-----

Preprocessing/cleaning/labeling

Was any preprocessing/Cleaning/Labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?	Yes —the pre-processing details are discussed in Appendix A.3.1.
Is the software used to preprocess/clean/label the instances available?	No.

Uses

Has the dataset been used for any tasks already?	Yes, we use the dataset for pre-training multimodal language and vision models.
Is there a repository that links to any or all papers or systems that use the dataset?	No, the dataset has only been used to train the models in this paper.
What (other) tasks could the dataset be used for?	We do not foresee other usages of the dataset at this stage.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labelled that might impact future uses?	The dataset is static and thus will become progressively more “stale” . For example, it will not reflect new language and norms that evolve over time. However, due to the nature of the dataset it is relatively cheap to collect an up-to-date version.
Are there tasks for which the dataset should not be used?	The dataset described in this paper contains English language text almost exclusively and therefore should not be used for training models intended to have multilingual capabilities.

Distribution	
Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?	No.

表 14: ***M3W Datasheet***. We follow the framework as presented by Gebru et al. [30].

F.2 Image and video text pair datasets

F.2.1 Datasheet for LTIP

Motivation	
For what purpose was the dataset created? Who created the dataset? Who funded the creation of the dataset?	The dataset was created for pre-training vision-language models and was created by researchers and engineers.
Any other comments?	None.
Composition	
What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?	All instances of the dataset are image-text pairs.
How many instances are there in total (of each type, if appropriate)?	The dataset contains 312M image-text pairs.
Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?	The dataset is a sample from a larger set.
What data does each instance consist of?	Each instance is made up of a sequence of UTF-8 bytes encoding the document's text, and an image in compressed format (see Appendix A.3.3).
Is there a label or target associated with each instance?	No, there are no labels associated with each instance.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	There are no relationships between the different instances in the dataset.
Are there recommended data splits?	We use random splits for the training and development sets.

Are there any errors, sources of noise, or redundancies in the dataset?	The data is relatively high quality but there is a chance that some instances are repeated multiple times.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	The websites that were used for this dataset were carefully selected to avoid such content. However given the scale of the data it is possible that some data could be considered offensive or insulting.

Collection Process

How was the data associated with each instance acquired?	The data is available publicly on the web.
What mechanisms or procedures were used to collect the data?	The data was collected using a variety of software programs to extract and clean the raw text and images.
If the dataset is a sample from a larger set, what was the sampling strategy?	N.A.
Over what timeframe was the data collected?	The dataset was collected over a period of several months in 2021. We do not filter the sources based on creation date.
Were any ethical review processes conducted?	No.

Preprocessing/cleaning/labeling

Was any preprocessing/Cleaning/Labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?	Some automatic text formatting was applied to remove from the captions dates and locations that were not relevant to the training objective.
Is the software used to preprocess/clean/label the instances available?	No.

Uses

Has the dataset been used for any tasks already?	Yes, we use the dataset for pre-training multimodal language and vision models.
Is there a repository that links to any or all papers or systems that use the dataset?	No, the dataset has only been used to train the models in this paper.
What (other) tasks could the dataset be used for?	We do not foresee other usages of the dataset at this stage.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labelled that might impact future uses?	The dataset is static and thus will become progressively more “stale”. For example, it will not reflect new language and norms that evolve over time. However, due to the nature of the dataset it is relatively cheap to collect an up-to-date version.
Are there tasks for which the dataset should not be used?	The dataset described in this paper contains English language text almost exclusively and therefore should not be used for training models intended to have multilingual capabilities.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?	No.
---	-----

表 15: **LTIP Datasheet.** We follow the framework as presented by Gebru et al. [30].

F.2.2 Datasheet for VTP

Motivation	
For what purpose was the dataset created? Who created the dataset? Who funded the creation of the dataset?	The dataset was created for pre-training vision-language models and was created by researchers and engineers.
Any other comments?	None.
Composition	
What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?	All instances of the dataset are video-text pairs.
How many instances are there in total (of each type, if appropriate)?	The dataset contains 27M video-text pairs.
Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?	The dataset is a sample from a larger set.
What data does each instance consist of?	Each instance is made up of a sequence of UTF-8 bytes encoding the document's text, and a video in compressed format (see Appendix A.3.3).
Is there a label or target associated with each instance?	No, there are no labels associated with each instance.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	There are no relationships between the different instances in the dataset.
Are there recommended data splits?	We use random splits for the training and development sets.

Are there any errors, sources of noise, or redundancies in the dataset?	The data is relatively high quality but there is a chance that some instances are repeated multiple times.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	The websites that were used for this dataset were carefully selected to avoid such content. However given the scale of the data it is possible that some data could be considered offensive or insulting.

Collection Process

How was the data associated with each instance acquired?	The data is available publicly on the web.
What mechanisms or procedures were used to collect the data?	The data was collected using a variety of software programs to extract and clean the raw text and videos.
If the dataset is a sample from a larger set, what was the sampling strategy?	N.A.
Over what timeframe was the data collected?	The dataset was collected over a period of several months in 2021. We do not filter the sources based on creation date.
Were any ethical review processes conducted?	No.

Preprocessing/cleaning/labeling

Was any preprocessing/Cleaning/Labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?	Some automatic text formatting was applied to remove from the captions dates and locations that were not relevant to the training objective.
Is the software used to preprocess/clean/label the instances available?	No.

Uses

Has the dataset been used for any tasks already?	Yes, we use the dataset for pre-training multimodal language and vision models.
Is there a repository that links to any or all papers or systems that use the dataset?	No, the dataset has only been used to train the models in this paper.
What (other) tasks could the dataset be used for?	We do not foresee other usages of the dataset at this stage.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labelled that might impact future uses?	The dataset is static and thus will become progressively more “stale”. For example, it will not reflect new language and norms that evolve over time. However, due to the nature of the dataset it is relatively cheap to collect an up-to-date version.
Are there tasks for which the dataset should not be used?	The dataset described in this paper contains English language text almost exclusively and therefore should not be used for training models intended to have multilingual capabilities.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?	No.
---	-----

表 16: **VTP Datasheet**. We follow the framework as presented by Gebru et al. [30].

G Credit for visual content

- Figure 1:
 - Row 1: All images are provided under license by Unsplash.
 - Row 2: All images are under the public domain.
 - Row 3: First two images are provided under license by Unsplash.
 - Row 5: Available from DALL • E 2 [89].
 - Row 6: First two are provided under license by Unsplash, the third one is provided by Wikimedia Commons, licensed under CC BY-ND 2.0.
 - Row 7: The images are provided by Wikimedia Commons, licensed under CC BY-ND 2.0.
 - Row 8: The images are provided by Wikimedia Commons, licensed under CC BY-ND 2.0.
 - Row 9: This video is from YFCC100M, licensed under CC BY-ND 2.0.
 - Dialogue 1: Available from DALL • E 2 [89].
 - Dialogue 2: The first icon is provided under license by Flaticon, the second image is provided under license by Unsplash, the third one is provided under license by Sketchfab.
 - Dialogue 3: Available from CLIP [85].
 - Dialogue 4: Chicago and Tokyo pictures obtained from Unsplash.
- Model Figures 3, 7, 9 and 8: All images are provided under license by Unsplash.
- Qualitative Figures 10, 11, 12, and 13: All visuals are sourced from various sources including the COCO dataset, Wikimedia Commons, licensed under CC BY-ND 2.0 or available from DALL • E 2 [89].