

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std			test-dev	test-std		valid	test-std		valid	test-std	
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 [†]	81.3 [†]	149.6[†]	81.4 [†]	57.2 [†]	60.6 [†]	46.8	75.2	75.4[†]	138.7	54.7	73.7	84.6 [†]
	[133]	[133]	[119]	[153]	[65]	[65]	[51]	[79]	[123]	[132]	[137]	[84]	[152]

Table 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).

Ablated setting	<i>Flamingo</i> -3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
<i>Flamingo</i> -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	62.9
(iii) Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	47.5	66.5
(iv) Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	50.7	66.9
		GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	47.8	63.1
(v) Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	42.3	59.8
		Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	50.8	68.8
		Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	49.7	68.2
(vi) Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	44.7	66.6
		Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	48.3	66.7
(vii) Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	44.5	64.9
		NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	42.9	62.7
(viii) Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	50.1	57.8
		✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	53.9	62.7

Table 3: **Ablation studies.** Each row should be compared to the baseline *Flamingo* run (top row). Step time measures the time spent to perform gradient updates on all training datasets.

Finally, despite having only used the DEV benchmarks for design decisions, our results generalize well to the other benchmarks, confirming the generality of our approach.

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.

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

While not the main focus of our work, we verify that when given more data, *Flamingo* models can be adapted to a task by fine-tuning their weights. In Table 2, we explore fine-tuning our largest model, *Flamingo*, for a given task with no limit on the annotation budget. In short, we do so by fine-tuning the model on a short schedule with a small learning rate by additionally unfreezing the vision backbone to accommodate a higher input resolution (details in Appendix B.2.2). 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 additional tasks: VQAV2, VATEX, VizWiz, MSRVTTQA, and HatefulMemes.

3.3 Ablation studies

In Table 3, we report our ablation results using *Flamingo*-3B on the *validation* subsets of the five DEV benchmarks with 4 shots. Note that we use smaller batch sizes and a shorter training schedule compared to the final models. The **Overall score** is obtained by dividing each benchmark score by its state-of-the-art (SotA) performance from Table 1 and averaging the results. More details and results are given in Appendix B.3 and Table 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].

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. We verify the importance of freezing the LM layers at training in row (viii). If trained from scratch, we observe a large performance decrease of -12.9% . Interestingly, fine-tuning our pretrained LM also leads to a drop in performance of -8.0% . This indicates an instance of “catastrophic forgetting” [71], in which the model progressively forgets its pretraining while training on a new objective. In our setting, freezing the language model is a better alternative to training with the pre-training dataset (MassiveText) in the mixture.

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].

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. 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, Kareem Ayoub, Lorrayne Bennett, Adria Recasens Contente, Tom Eccles, Nando de Freitas, Sander Dieleman, Conor Durkan, Aleksa Gordić, Raia Hadsell, Will Hawkins, Lisa Anne Hendricks, Felix Hill, Jordan Hoffmann, Geoffrey Irving, Drew Jaegle, Koray Kavukcuoglu, Agustin Dal Lago, Mateusz Malinowski, Soňa Mokrá, Gaby Pearl, Toby Pohlen, Jack Rae, Laurent Sifre, Francis Song, Maria Tsimpoukelli, Gregory Wayne, and Boxi Wu.