Exploring Curriculum Learning for Vision-Language Tasks: A Study on Small-Scale Multimodal Training

Rohan Saha[1], Abrar Fahim[1], Alona Fyshe[1,2], Alex Murphy[1]
[1]Department of Computing Science, [2]Department of Psychology
University of Alberta
{rsaha, afahim2, alona, amurphy3}@ualberta.ca

Abstract

For specialized domains, there is often not a wealth of data with which to train large machine learning models. In such limited data / compute settings, various methods exist aiming to *do more with less*, such as finetuning from a pretrained model, modulating difficulty levels as data are presented to a model (curriculum learning), and considering the role of model type / size. Approaches to efficient *machine* learning also take inspiration from *human* learning by considering use cases where machine learning systems have access to approximately the same number of words experienced by a 13 year old child (100M words). We investigate the role of 3 primary variables in a limited data regime as part of the multimodal track of the BabyLM challenge. We contrast: (i) curriculum learning, (ii), pretraining (with text-only data), (iii) model type. We modulate these variables and assess them on two types of tasks: (a) multimodal (text+image), and (b) unimodal (text-only) tasks. We find that curriculum learning benefits multimodal evaluations over non-curriclum learning models, particularly when combining text-only pretraining. On text-only tasks, curriculum learning appears to help models with smaller trainable parameter counts. We suggest possible reasons based on architectural differences and training designs as to why one might observe such results.

1 Introduction

Recent vision-language models (VLMs) have achieved superior performance on numerous benchmark datasets (such as the Llama[1] and Gemini models[2]), and continue advancing rapidly as models are scaled up. The number of parameters of such models is often on the order of billions. These models require multiple days of compute, and hundreds of GPUs (e.g., Radford et al. (2021)), resulting in massive energy consumption (Luccioni et al., 2024). Furthermore, to train such large models, we require massive amounts of pretraining data. For example, 70M image-text pairs were used to train the Flava foundation model (Singh et al., 2022). Pretraining VLMs on such large scale data is often infeasible for independent researchers and university research labs with limited compute.

In contrast to *machine* learning, *human* learning is much more efficient, a finding which has led researchers to consider which methods might promote more *human-like* learning in artificial neural networks. This was originally argued for in early work on curriculum learning (Bengio et al., 2009), citing the fact that humans do not learn from randomly sampled data, but benefit from learning over structured chunks, typically increasing in difficulty (a curriculum).

To this end, we explore the application of curriculum learning to VLMs with limited input data as part of the BabyLM challenge (Choshen et al., 2024). For the multimodal track, which contains a dataset of image-caption pairs, we take inspiration from phase-based curriculum methodology used in Ayyubi et al. (2023). We use Part-of-Speech (PoS) linguistic features from the captions to categorize samples into different phases, to generate a learning curriculum. However, instead of training the model only one phase at a time (as used in Ayyubi et al. (2023)), we train the model on the current and previous phases such that the pool of data which can be sampled increases at each phase.

From our experiments, we observe that:

- In a limited data setting, curriculum learning can improve the performance of VLMs on certain multimodal and text-only evaluation benchmarks.
- Pretraining VLMs on developmentally plausible text-only data prior to adapting to multimodal data may help improve performance on some evaluation tasks, but not others.

[1] https://llama.meta.com/
[2] https://deepmind.google/technologies/gemini/

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 65–81
November 15-16, 2024

2 Background

2.1 Curriculum Learning

Curriculum Learning (CL) takes inspiration from the learning process in humans by presenting data to a machine learning model in an easy-to-difficulty manner (Elman, 1993; Bengio et al., 2009). CL consists of two parts: (1) a scoring function to rank data samples based on difficulty, and (2) a pacing function, which controls the distribution of data samples presented to the model. In the standard CL implementation, the pacing function introduces can be samples in ascending order of difficulty (or decreasing difficulty in the case of anti-curriculum learning (Hacohen and Weinshall, 2019; Wu et al., 2021)).

While extensive research has shown that in certain cases, curriculum learning can provide performance gains in vision (Hacohen and Weinshall, 2019; Wang et al., 2019b; Soviany, 2020) and Natural Language Processing (NLP) tasks (Nagatsuka et al., 2021; Maharana and Bansal, 2022; Sun et al., 2023), in other cases, the benefit is unclear (Campos, 2021; Martinez et al., 2023; Chobey et al., 2023; Edman and Bylinina, 2023a). Importantly, with the prevalence of vision-language models, it is crucial to understand how the application of CL modulates VLMs to work in the domain of limited data and compute.

2.2 Curriculum Learning for Vision Language Models

Some previous work has applied CL to multimodal models where the data modality consists of images and texts. Srinivasan et al. (2023) showed that CL applied to a transformer model helps improve performance on zero-shot image and text retrieval tasks over a baseline CLIP model (Radford et al., 2021). CL has also shown benefit in other multimodal domains, such as medical report generation (Liu et al., 2023), image-captioning (Ayyubi et al., 2023), and visual question answering (Li et al., 2020). However, these works either rely on non vision-transformer based image encoders (such as an R-CNN), or conduct evaluation on a small set of evaluation tasks. It is also unclear whether: (i) training VLMs on image-caption data improves model performance on text-only benchmarks; (ii) how CL affects downstream performance in models with additional text pretraining compared to randomly initialized models.

In this work, we present a study where we apply CL to VLMs trained on *small data*. We hope to provide the research community with a better understanding of the effects of CL on popular VLMs such as the Generative Image Transformer (GIT) (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) models. Furthermore, we also explore the effect of CL on downstream model performance on various zero-shot multimodal and text-based benchmarks.

3 Methods

3.1 Data

We use the dataset provided as part of the BabyLM multimodal track (Choshen et al., 2024). The data consist of 100M words in total: 50M words from varied text corpora (described in Choshen et al. (2024)) and the other 50M words are text captions taken from the Conceptual Captions (Sharma et al., 2018) and Localized Narratives (Pont-Tuset et al., 2020) image-caption datasets. In total, the multimodal data consists of $\sim$ 2.9M image-caption pairs.

One of the key experimental variables we examine is the impact of text pretraining. For multimodal models, we compare the performance of models trained on image-caption data (consisting of 50M words), starting either from a randomly initialized model or from a model pretrained on the text-only corpora mentioned above (50M words). Model variants not pretrained on the text-only corpora only use the words in the captions of the associated training images (i.e., models are trained on only 50M words and the corresponding images).

3.2 Models

We train two VLMs: (1) GIT Wang et al. (2022) and (2) Flamingo (Alayrac et al., 2022). We chose these models because they were selected as reference baselines provided by the BabyLM challenge (Choshen et al., 2024). Both GIT and $Flamingo$ models consist of vision encoders to encode image inputs, and text decoders to generate free-form text.

We use the default configurations for the GIT[3] and $Flamingo$[4] models provided in the BabyLM challenge to compare the performance of our models to the baselines reported by the challenge. Following the default configurations, we use pretrained vision encoders[5] for both the GIT and

[3]https://huggingface.co/babylm/git-2024
[4]https://huggingface.co/babylm/flamingo-2024
[5]https://huggingface.co/facebook/dino-vitb16

Flamingo models. Furthermore, according to default model configurations, we update all model parameters for the *GIT* model, but for *Flamingo*, we keep the vision encoder frozen, and update all other parameters. *GIT* has a total of 198 million parameters (198 million trainable parameters), and *Flamingo* has 255 million total parameters (169 million trainable parameters because of the frozen vision encoder).

Tokenizer: Pretrained tokenizers are trained on data that exceed the limit imposed by the challenge. Thus, we train a new *WordPiece* tokenizer (using a `bert-base-uncased` model configuration) from scratch on the text-only and caption data (100M words total). We use the same tokenizer for both *GIT* and *Flamingo* to avoid confounding model performance differences with the tokenizer choice.

3.3 Curriculum Framework

We discuss the respective implementations of the scoring and pacing functions for the curriculum learning framework below.

Scoring function: A scoring function assigns a *difficulty score* $k \in \mathbb{R}$ to each sample in the dataset, where a sample x_i is easier than a sample x_{i+1}, if $k_{x_i} < k_{x_{i+1}}$.

Previous works have used a variety of scoring functions to measure sample difficulty, such as the loss scoring function in image classification (Hacohen and Weinshall, 2019) and text classification settings (Xu et al., 2020; Maharana and Bansal, 2022). Relatedly, in sample-efficient pretraining of language models, average sentence rarity (Borazjanizadeh, 2023), sentence length (DeBenedetto, 2023) or other combinations of individual text statistics (Edman and Bylinina, 2023b) have been used to rank data samples (for a comprehensive survey, see Soviany et al. (2021)). More recently, in multimodal settings, cross-modal similarity (Zhang et al., 2022) has been used to rank examples to improve model performance in image-captioning tasks. All in all, it must be noted that determining the difficulty of image-caption pairs is non-trivial and an active research problem.

For our experiments, we explored the applicability of linguistic information such as Part-of-Speech (PoS) tags to determine difficulty of samples. We took inspiration from the scoring function used by Ayyubi et al. (2023), where a PoS tagger was used to count the number of nouns in the caption, as an indirect measure of the number of concepts present in the image. The number of concepts, in turn, determined the difficulty of the image-caption pair.

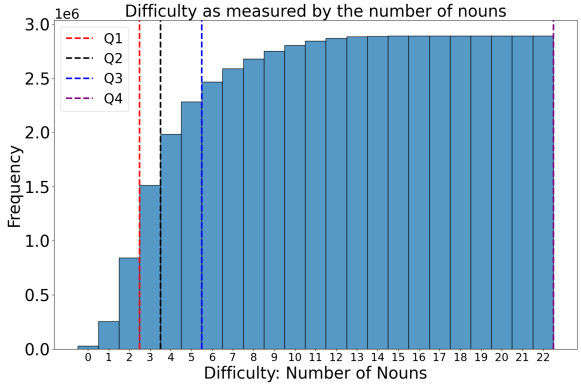

Figure 1: Cumulative distribution of scores for all the image-caption pairs. The dashed vertical lines determine each of the four quartiles, where each quartile contains the samples that belong to a specific curriculum phase.

As the BabyLM challenge has limits on the number of words that can be used to train systems, we trained our own PoS tagger to tag the image captions. To train the tagger, we first created a training dataset by annotating the provided text-only and caption data, with POS symbols[6], using an off-the-shelf PoS tagger from NLTK [7]. Then we used this newly created annotated training dataset to train a custom PoS tagger on the permissible limited text words. We implemented the PoS tagger using a token classification model using $BERT_{BASE}$ as the backbone model architecture. We trained the tagger for 5 epochs[8], using a batch size of 512 and half-precision (FP16) training.

Distribution of difficulty scores: We show the cumulative distribution of the scores assigned by the PoS scoring function in Figure 1. For images having multiple captions, we consider the maximum value of the difficulty (maximum number of nouns) amongst all the captions for that image. We use maximum difficulty to account for the most complex interpretation of the image and avoid underestimation of the difficulty value.

Ordering: In our experiments, we order the samples in ascending order of difficulty, to explore the

[6]These are non-word elements such as NN for noun, or JJ for determiner

[7]https://www.nltk.org/api/nltk.tag.pos_tag.html

[8]We observed that 5 epochs were sufficient to achieve ∼97.42% accuracy on a 10% held out validation dataset. We then trained the tagger on all the data (train+validation).

performance improvement of unimodal and multimodal models when they are trained in a manner similar to how humans acquire novel information. Although previous work has shown that a descending ordering of difficulty can be beneficial for model performance for certain tasks (e.g., Maharana and Bansal (2022)), we leave this for future research given limited compute.

Pacing function: A pacing function controls the rate at which samples of different training curriculum phases are presented to the model. Multiple different pacing strategies exist, such as fixed exponential pacing, step pacing for image classification (Hacohen and Weinshall, 2019), competence function (Platanios et al., 2019) for machine translation, to name a few.

For our experiments, we design a simple pacing function inspired by the phase-level pacing function (Ayyubi et al., 2023) and competence-based pacing function (Platanios et al., 2019). We use the quartiles from the cumulative distribution of the sample difficulty scores (Figure 1), giving us four *blocks* of difficulty levels. For simplicity, we also train our model in four phases, where in each training phase p, we train the model on samples that have difficulty levels in the p^{th} quartile. For example, in Figure 1, the first phase (p_1) contains samples with difficulty level $k \leq 2$, the second phase contains samples with difficulty level $k \leq 3$, the third phase contains samples with $k \leq 5$, while the fourth phase contains all the samples in the dataset. For each training phase, we randomly sample training batches from the set of data available up to the corresponding training quartile. It must be noted with each new block, the number of available data points increases, which has an effect during training, where earlier epochs are faster (because of fewer samples) compared to later epochs.

This approach contrasts the phase-level curriculum learning introduced by Ayyubi et al. (2023), where the model is trained only on samples from a specific block, which may cause the model to focus more on samples in that specific block, while not retaining previously learned information from earlier phases. Furthermore, our pacing function has the added advantage of not requiring extensive hyperparameter tuning, such as the exponential pacing function used by Hacohen and Weinshall (2019), and is thus suitable for scenarios with limited computational resources.

3.4 Models Variants

For both GIT and $Flamingo$, we train four model variants, two of which are baseline models and two are trained using curriculum learning. In each pair, we train one model only on the image-caption data starting from random initialization (except the vision encoder which is pretrained), while we first pretrain the other variant on the text-only corpus, before training on image-caption data.

Baselines: For the first baseline variant, we train the model on the image-caption dataset (50M words) using standard i.i.d training. We refer to this variant with **C** (denoting that the model is trained on the image-caption data only) for both $GIT_{Baseline}$ and $Flamingo_{Baseline}$. For the second baseline variant, we first train the model on the text-only dataset (containing 50M words) using standard i.i.d training. We then continue the training procedure on image-caption dataset (containing another 50M words) using standard i.i.d training. We refer to this variant as **T+C**, for both $GIT_{Baseline}$ and $Flamingo_{Baseline}$.

Our choice to also train the **T+C** model variant stems from previous work showing that exposing the model to developmentally plausible data, such as child-directed speech, before exposing it to complex data, can benefit model performance (Huebner et al., 2021). Thus, we explore the difference in model performance, when we first train the model on the text-only dataset, before continuing the training procedure on the image-caption data.

Curriculum models: For curriculum variants, we use CL on the image-caption pairs because we hypothesize that applying CL on multimodal data will improve model performance. We refer to these variants trained only on the image-caption pairs as **C** under GIT_{CL} and $Flamingo_{CL}$. We also train **T+C** variants of CL models, where we first pretrain the model on the text-only dataset using standard i.i.d training, and then use curriculum learning to continue the training procedure on the image-caption pairs.

To summarize, we trained four variants for each model, two of which were trained using standard training (no curriculum), and the other two were trained using curriculum learning. For GIT and $Flamingo$ baseline variants, we train the model on the image-caption only (**C**) data, and both text + image-caption (**T+C**) data. Similarly, for the curriculum variants, we train each model on, image-

caption data only (**C**) data, and both text + image (**T+C**) data.

4 Training and Evaluation Details

Training Details: For the curriculum variants, we train the model for two epochs per each *difficulty phase* (of which there are four). We used a learning rate of $1e^{-5}$, maximum token length of 50, and 32 samples per batch [9], and Adam optimizer[10] (Kingma and Ba, 2017).

When training the **T+C** variants of our baseline and curriculum models, we first trained the model on the text-only dataset for twenty epochs (instead of eight epochs for image-caption data) and use the same hyperparameter values. We used an NVIDIA A5000 GPU with 24GB vRAM, with half-precision (FP16) to train the models. We provide the total time required to train each model variant in Appendix A. For all experiments, we set the random seed to 0 to remove variation in the results due to different random sampling and initialization. We also hold out 5% of the full image-caption dataset to validate the model. We show the validation loss curves in Appendix B.

Evaluation: To evaluate the performance of our models, we use the evaluation pipeline provided by challenge (Gao et al., 2023; Choshen et al., 2024). We report the performance of all the variants of the GIT and $Flamingo$ models on the multimodal, and text-based evaluation tasks.

4.1 Multimodal evaluation datasets

Winoground : The Winoground dataset (Thrush et al., 2022) evaluates a model's ability to perform visio-linguistic compositional reasoning. Specifically, given two image-caption pairs, the goal is to match the image to the corresponding caption, where both captions contain an identical set of words, but in a different order (e.g. *It's a fire truck* vs *it's a truck fire*). The dataset consists of 400 examples with 800 unique images and captions. To assess model performance, we use the unpaired text-score metric as provided in the BabyLM evaluation pipeline.

VQAv2: The VQAv2 dataset (Goyal et al., 2017) is a large-scale visual question answering dataset. It contains open-ended questions about images, requiring models to understand the visual content and generate appropriate answers. We use accuracy as the choice of metric as reported in the BabyLM evaluation pipeline. For this task the model has to select the best answer for a given image and question, in the presence of 7 distractors.

DevBench: The DevBench dataset (Tan et al., 2024) is a multimodal benchmark for developmental evaluation that evaluates how closely a model's outputs align with human responses. It includes tasks such as object recognition, action recognition, and visual question answering, using data from both children and adults. The BabyLM pipeline uses three tasks from the DevBench dataset: (1) The (Lexical) Visual Vocabulary (lex-viz_vocab) task involves selecting the correct image from several image options based on a given word. (2) The (Grammatical) Test of Reception of Grammar (gram-trog) task involves choosing the correct image based on a sentence, testing grammatical understanding using distractor images that correspond to sentences with different word orderings (e.g. "a white cat sitting on a brown couch" vs. "a brown cat sitting on a white couch"). Finally, (3) the (Semantic) THINGS Similarity (sem-things) task uses Representational Similarity Analysis (RSA) to compare the model's image similarity judgments with human responses.

4.2 Text-only evaluation datasets

BLIMP (and BLIMP Supplement): The `BLIMP` dataset (Warstadt et al., 2020) is a benchmark for evaluating syntactic and semantic knowledge in language models. It consists of sentences with systematic variations in syntax and semantics. The `BLIMP Supplement` extends the original dataset with additional challenging examples.

(Super)GLUE: The `(Super)Glue` benchmark (Wang et al., 2018, 2019a) is a collection of diverse natural language understanding tasks designed to evaluate a model's ability to perform well across multiple domains and evaluates generalized linguistic ability. The BabyLM challenge includes tasks, `COLA, SST2, MRPC, QQP, MNLI, MNLI-MM, QNLI, RTE` from the `GLUE` benchmark, and the tasks `BoolQ, RTE and WSC` from `SuperGLUE` benchmark. To fine tune all our model variants, we use a train batch size of 128, validation batch size of 16,

[9] We use a batch size of 32 when training on the image-caption data, but we use a value of 256 when pretraining the model (T+C variant) on the text-only dataset as memory requirements are lower.

[10] We use default hyperparameters for Adam: β_1 = 0.9, β_2 = 0.999, eps=$1e^{-08}$, weight_decay=0.

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
Winoground	54.02	55.50	51.34	55.23	50.00	51.21	51.21	50.80
VQAv2	41.22	41.72	42.84	43.98	41.99	43.00	35.93	40.85

Table 1: Results for baseline and curriculum models on the Winoground and VQAv2 evaluation datasets. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
lex-viz_vocab	72.27	75.63	78.15	73.11	66.39	52.94	58.82	54.62
gram-trog	32.89	38.16	32.29	39.47	34.21	34.21	34.21	35.53
sem-things	33.39	25.79	22.83	32.08	46.46	47.99	50.21	51.66
Avg: $devbench_{acc}$	46.18	46.52	44.63	48.22	49.02	45.05	47.75	47.27

Table 2: Accuracy results for baseline and curriculum models on the DevBench dataset. RSA scores are used for sem-things **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants..

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
lex-viz_vocab	68.25	68.59	70.19	70.66	64.47	57.63	63.08	57.46
gram-trog	44.46	46.51	44.77	45.79	43.59	42.77	42.54	43.29
sem-things	33.39	25.79	22.83	32.08	46.46	47.99	50.21	51.66
Avg: $devbench_{hs}$	48.70	46.96	45.93	49.51	51.51	49.46	51.94	50.80

Table 3: Human similarity scores for baseline and curriculum models on the DevBench dataset. RSA scores are used for sem-things. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

learning rate of $5e^{-5}$, early stopping patience of 3, maximum sequence length of 50, and maximum number of epochs=10. We used default values for all other hyperparameters provided in the BabyLM evaluation pipeline.

EWOK: The `EWOK` dataset (Ivanova et al., 2024) is a zero-shot dataset for evaluating compositional generalization in language models. It consists of sentences with compositional structures that require models to generalize to unseen combinations of words and syntactic patterns.

5 Results

As unimodal and multimodal tasks are qualitatively different, we analyze the three experimental variables of interest (curriculum, pretraining & model type) in the context of each task type. Namely, we report the results for all variants of GIT and $Flamingo$ models across two main task types that differ with respect to their data inputs: (i) multimodal (image+captions), and (i) unimodal (text-only).

5.1 Multimodal (image+captions)

We show the multimodal evaluations results in Table 1 for `Winoground` and `VQAv2`, and in Tables 2 (accuracy) and 3 (human similarity) for `DevBench`.

5.1.1 Curriculum Learning

The GIT_{CL} model performs better than $GIT_{Baseline}$ on `VQAv2` and `DevBench` datasets,

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BliMP Supp	44.29	52.89	48.61	51.24	44.24	52.59	45.71	53.28
BLiMP filtered	57.85	62.90	61.34	64.05	57.03	59.82	55.64	60.13
(Super)$GLUE_{avg}$	59.96	61.12	59.79	61.46	59.82	62.79	60.53	64.29
$EWOK_{avg}$	50.62	51.55	49.82	50.98	50.03	50.67	50.16	50.71

Table 4: Average results for the text-only evaluation datasets.**C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

with and without pretraining on separate text data. This is not the case for Winoground, which we note has quite unique properties, such as specifically probing model representations for compositionality (see Section 4.1).

We find that $Flamingo_{CL}$ only performs better than its associated baseline ($Flamingo_{Baseline}$) on the DevBench dataset when using accuracy, and when evaluating using human response scores. This result indicates that curriculum training may benefit multimodal model performance when evaluated on benchmark datasets that focus on developmentally plausible evaluation of language models.

5.1.2 Text Pretraining

Compared to training on just image-caption data, pretraining with the text-only data (variant **T+C**) produces higher scores across both $GIT_{Baseline}$ and GIT_{CL} models on Winoground and DevBench, while the results are more mixed for $Flamingo$ models. However, in $Flamingo_{CL}$ on the VQAv2 dataset, we see the largest gain in performance due to text pretraining (from 35.93 to 40.85, a gain of $\sim$ 5% in Table 1). On the DevBench evaluation for GIT_{CL}, we also see the 2nd largest gain in performance due to text pretraining (from 44.63 to 48.22 for accuracy, and from 45.93 to 49.51 when using reference human similarity scores; a gain of $\sim$ 4%). Interestingly, the highest result of all models on the Winoground dataset are the GIT models with text pretraining, suggesting that text-only pretraining is a big contributor to the properties of the Winoground evaluation benchmark (compositionality). However, one must be cautious about generalizing this finding as the performance increase could simply result from the model being trained on more data.

As we only use a single seed to report these results, we wanted to confirm that our observation is not simply due to random chance. Thus, we conduct more experiments where we train all GIT variants using two more seeds, and observe a similar pattern in our findings (text pretraining aids model performance). We provide these results in Appendix C.

5.1.3 Model Type

The two models differ in their application of attention mechanism and model size, measured by the number of trainable parameters (See Section 3.2). $Flamingo$ has a frozen image encoder (unlike GIT) and cross-attention is applied prior to each LM block in the Transformer stack (which internally contains the standard self-attention mechanism). In contrast, GIT uses a projection module to bring image embeddings into the same space as the text embeddings and applies successive self-attention on these vectors. We see multiple variants of GIT outperform $Flamingo$ (especially for Winoground, VQAv2, and lex-viz_vocab, gram-trog subsets for DevBench). In the multimodal evaluation context, we believe this could be due to the ability for GIT to update the parameters of its vision encoder, perhaps additionally by making use of the fact that image tokens can self-attend to one another (unlike the cross-attention in $Flamingo$, which does not have this property).

5.2 Unimodal (text-only)

We summarize the results for the unimodal (text-only) evaluation in Table 4. This table contains summary results for the three text-only evaluation benchmarks (see Section 4.2). Table 9 contains detailed results on the (Super)GLUE and EWOK benchmarks. We also provide a detailed breakdown of model performance for each text-based task in Appendix D.

5.2.1 Curriculum Learning

Closely related to the observations for multimodal benchmarks, we see that curriculum learning variants outperform corresponding baselines variants on the unimodal (text-only) benchmarks. Although both GIT_{CL} and $Flamingo_{CL}$ outperformed their corresponding baselines (Tables 4 and 9), the effect was greater in $Flamingo_{CL}$.

5.2.2 Text Pretraining

We outline the averaged results in Table 4 and show that for both $Flamingo$ and GIT, text pretraining leads to a gain in performance. In fact, all **T+C** variants (curriculum and baseline) for both models showe better performance compared to **C** variants. Coupled with curriculum learning, we observe performance benefits on all text-based evaluation datasets. These results suggest that text pretraining conveys a clear advantage for multimodal models when they are evaluated on certain text-based benchmarks.

5.2.3 Model Type

Unlike the multimodal results, considering the average results in Table 4, there was no consistent pattern where one model type outperformed the other. For example, on `(Super)GLUE`, both baseline and CL **T+C** variants of $Flamingo$ outperformed respective GIT variants. However, this was not the case for `BLIMP filtered`, where we observed the opposite pattern - all variants of GIT outperformed all variants of $Flamingo$. Such a result could result from the fact that both GIT and $Flamingo$ become more similar in their architecture in the text-only evaluation setting. This can stem relaxed requirement to incorporate image information, making both models resemble standard autoregressive Transformer decoders (the trainable parameter count changes in this context because GIT's vision encoder was trainable in the multimodal case, while $Flamingo$'s was frozen). This results in the trainable parameter count for GIT being 198M and 169M for $Flamingo$ (Section 3.2).

5.3 Brief Summary of Results

For the multimodal evaluation, we observe that text pretraining before image-caption training boosts model performance compared to no text pretraing. However, these observations must be cautiously generalized across model types; text pretraining largely conveys a benefit in all GIT models, but this benefit is inconsistent for $Flamingo$. For instance, the $Flamingo_{CL}$ variant benefits from additional text-only pretraining over just image-caption training (for `VQAv2`, `gram-trog`, and `sem-things`), but this effect is unclear for the $Flamingo_{Baseline}$. For GIT model variants, curriculum learning (combined with pretraining) resulted in the best overall model scores on `VQAv2` and `DevBench` (considering average scores in Tables 2 and 3).

For the text-only evaluation, removing the image component from both the GIT and $Flamingo$ models effectively reduces them to text-only transformer architectures with differing number of parameters. This likely explains why the models show similar performance across tasks despite their original multimodal design. Nonetheless, we see that in Table 4, the $Flamingo_{CL}$ **T+C** variant can be more suited to learning representations leading to better scores across the `SuperGLUE` benchmark, and `BLiMP supplement` dataset. But on `BLiMP filtered` (and less pronounced for `EWOK`), the **T+C** variant of GIT_{CL} outperforms the **T+C** variant of $Flamingo_{CL}$.

Conclusion

In this study, we explore the application of a curriculum learning (CL) approach to training vision-language models (VLMs) in a limited data setting. We use a custom trained Part-of-Speech (PoS) tagger to determine the complexity of image-caption pairs. We train two variants for each of the GIT and $Flamingo$ models using curriculum learning and compare their performance against variants trained using standard i.i.d training. We find that while CL training shows potential, its benefits are not universally applicable across all GIT and $Flamingo$ variants. However, for certain model configurations, CL enhances performance on a range of downstream, multimodal and text-based tasks (zero-shot and finetuning). Importantly, pretraining VLMs on developmentally plausible text data prior to multimodal training can contribute to performance gains. Nonetheless, generalizing this result requires careful consideration, as factors such as model architecture, training data composition, and the nature of evaluation tasks can significantly affect model performance.

Code and Data Availability

We release our code, model predictions, and model checkpoints.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Hammad A Ayyubi, Rahul Lokesh, Alireza Zareian, Bo Wu, and Shih-Fu Chang. 2023. Learning from children: Improving image-caption pretraining via curriculum. *arXiv preprint arXiv:2305.17540*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Nasim Borazjanizadeh. 2023. Optimizing GPT-2 pretraining on BabyLM corpus with difficulty-based sentence reordering. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 356–365, Singapore. Association for Computational Linguistics.

Daniel Campos. 2021. Curriculum learning for language modeling. *arXiv preprint arXiv:2108.02170*.

Aryaman Chobey, Oliver Smith, Anzi Wang, and Grusha Prasad. 2023. Can training neural language models on a curriculum with developmentally plausible data improve alignment with human reading behavior? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 98–111, Singapore. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Computing Research Repository*, arXiv:2404.06214.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for babylm strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206.

Lukas Edman and Lisa Bylinina. 2023a. Too much information: Keeping training simple for babylms. *arXiv preprint arXiv:2311.01955*.

Lukas Edman and Lisa Bylinina. 2023b. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Preprint*, arXiv:1612.00837.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. 2020. A competence-aware curriculum for visual concepts learning via question answering. In *European Conference on Computer Vision*, pages 141–157. Springer.

Fenglin Liu, Shen Ge, Yuexian Zou, and Xian Wu. 2023. Competence-based multimodal curriculum learning for medical report generation. *Preprint*, arXiv:2206.14579.

Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and*

Transparency, FAccT '24, page 85–99, New York, NY, USA. Association for Computing Machinery.

Adyasha Maharana and Mohit Bansal. 2022. On curriculum learning for commonsense reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–992, Seattle, United States. Association for Computational Linguistics.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – Curriculum Learning for Infant-inspired Model Building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 84–99, Singapore. Association for Computational Linguistics.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 989–996, Held Online. INCOMA Ltd.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *ECCV*.

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. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. *Preprint*, arXiv:2112.04482.

Petru Soviany. 2020. Curriculum Learning with Diversity for Supervised Computer Vision Tasks. *Preprint*, arXiv:2009.10625.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum Learning: A Survey. *arXiv preprint*.

Tejas Srinivasan, Xiang Ren, and Jesse Thomason. 2023. Curriculum learning for data-efficient vision-language alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5619–5624.

Shichao Sun, Ruifeng Yuan, Jianfei He, Ziqiang Cao, Wenjie Li, and Xiaohua Jia. 2023. Data selection curriculum for abstractive text summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7990–7995.

Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. Devbench: A multimodal developmental benchmark for language learning. *Preprint*, arXiv:2406.10215.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *Preprint*, arXiv:2205.14100.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019b. Dynamic curriculum learning for imbalanced data classification. *Preprint*, arXiv:1901.06783.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When do curricula work? *Preprint*, arXiv:2012.03107.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Hongkuan Zhang, Saku Sugawara, Akiko Aizawa, Lei Zhou, Ryohei Sasano, and Koichi Takeda. 2022. Cross-modal similarity-based curriculum learning for image captioning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7599–7606, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Comparison of Training Times

We show the comparison of the training times for different baseline and curriculum variants in Table 5.

Model	Variant	Hours
$GIT_{Baseline}$	C	$\sim$ 80
	T+C	$\sim$ 109
GIT_{CL}	C	$\sim$ 50
	T+C	$\sim$ 79
$Flamingo_{Baseline}$	C	$\sim$ 79
	T+C	$\sim$ 105
$Flamingo_{CL}$	C	$\sim$ 46
	T+C	$\sim$ 72

Table 5: Comparison of training times amongst all model variants. These training times include validation loss calculation after every epoch. The pretraining on the text-only dataset (for the T+C variants) accounted for about 29 hours for the GIT model and around 26 hours for the $Flamingo$ model. Curriculum models take fewer hours to train because of the dynamic nature of the training data size that grows during training.

B Validation loss curves

We show the validation loss curves on a held out 5% of the image-caption data in Figure 2.

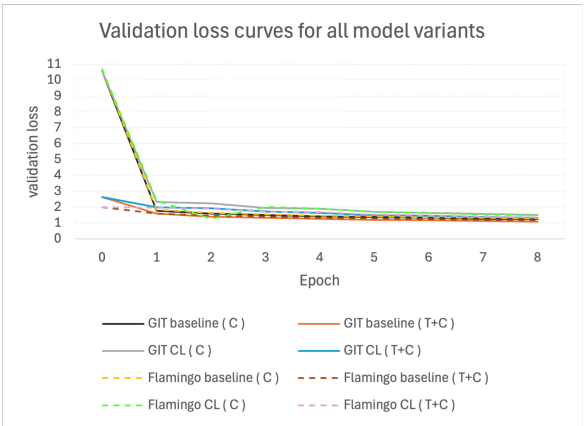

Figure 2: Validation loss curves for all the model variants. GIT variants are shown in solid lines and $Flamingo$ variants are shown in dashed lines. The x-axis denotes the epochs, and the value at the 0th epoch denotes the validation loss of the model before being trained on the image-caption pairs (i.e., before training on the first epoch). For the **T+C** variants, since the model is pretrained on the text-only dataset before being trained on the image-caption pairs, the loss starts at a lower value compared to the model variants on image-caption data only (**C**) that were randomly initialized.

C GIT model multimodal results across 3 seeds

We show the multimodal evaluation results for the different GIT model variants in Tables 6 for Winoground and VQAv2, 7 for accuracy on DevBench, and 8 human similarity scores on DevBench.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
Winoground	54.02	53.71	51.52	54.65
VQAv2	38.80	41.90	42.28	42.60

Table 6: Results for GIT baseline and GIT curriculum models on the multimodal evaluation datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
lex-viz_vocab	72.93	72.55	75.91	71.71
gram-trog	38.16	36.84	32.26	41.67
sem-things	30.88	25.61	17.34	30.78
$Average_{acc}$	47.32	45.00	41.84	48.05

Table 7: Accuracy results for GIT baseline and GIT curriculum models on the devbench datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
lex-viz_vocab	68.64	68.07	68.65	68.71
gram-trog	44.90	44.61	43.72	45.71
sem-things	30.88	25.61	17.34	30.78
$Average_{hs}$	48.14	46.10	43.24	48.40

Table 8: Human similarity results for GIT baseline and GIT curriculum models on the devbench datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

D Evaluation results on (Super)GLUE, EWOK, and BLiMP

We show the results for all models and corresponding variants on each individual subtask for the text-only evaluation tasks in Tables 9 for (Super)GLUE and EWOK, 10 for BLiMP Supplement, and 11 , 12, 13, 14 for BLiMP.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
$SuperGLUE_{ft}$	boolq	64.04	65.2	64.04	70.21	67.77	66.91	68.07	66.54
	cola (mcc)	6.68	6.68	0.0	6.68	0.0	17.7	0.0	31.75
	mnli	68.7	69.74	69.34	69.93	66.24	70.03	67.07	70.37
	mnli-mm	69.43	70.22	69.26	70.77	66.9	70.2	66.35	71.42
	mrpc (f1)	82.12	82.13	81.23	81.35	81.05	82.51	79.87	82.39
	multirc	55.57	57.43	57.55	56.97	60.81	53.55	58.21	56.23
	qnli	63.14	64.42	67.5	65.59	65.81	68.92	67.86	69.91
	qqp (f1)	80.92	81.7	80.12	81.53	79.83	82.05	79.91	81.88
	rte	46.04	48.92	46.04	46.04	46.04	52.52	56.12	46.04
	sst2	84.40	87.39	84.17	88.53	85.09	87.84	83.94	88.30
	wsc	38.46	38.46	38.46	38.46	38.46	38.46	38.46	42.31
EWOK	agent prop	50.05	50.14	50.09	49.59	49.46	50.32	49.91	49.68
	mat-dynam	51.56	52.21	51.30	50.65	49.48	52.21	50.52	54.42
	mat-prop	50.59	52.35	47.06	49.41	46.47	53.53	51.76	51.18
	phy-dynam	49.17	55.83	48.33	58.33	54.17	48.33	50.0	51.67
	phy-inter	49.64	50.0	50.18	50.18	50.18	49.1	48.74	49.1
	phy-relation	50.24	49.88	50.61	49.51	52.57	50.12	49.27	50.86
	quant-prop	51.91	50.96	49.68	50.96	49.36	53.5	50.64	50.0
	social-interac	50.34	50.34	50.34	49.66	49.32	49.32	49.66	50.0
	social-prop	50.3	50.91	50.91	50.0	50.0	49.09	50.61	50.0
	social-relation	50.32	51.42	49.94	50.0	49.29	50.0	50.45	50.06
	spatial-relation	52.65	53.06	49.59	52.45	50.0	51.84	50.20	50.82

Table 9: Breakdown of model performance on each subtask for the `(Super)Glue` and `EWOK` datasets. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP Supplement	hypernym	47.86	48.81	49.76	48.93	49.17	48.93	48.1	51.19
	qa_congruence_easy	29.69	51.56	35.94	50.0	32.81	51.56	37.5	53.12
	qa_congruence_tricky	27.88	24.24	27.27	20.0	20.0	30.91	27.27	28.48
	subject_aux_inversion	66.02	83.76	80.06	83.68	68.53	81.54	71.4	82.91
	turn_taking	50.0	56.07	50.0	53.57	50.71	50.0	44.29	50.71
	Average	44.29	52.89	48.61	51.24	44.24	52.59	45.71	53.28

Table 10: Breakdown of model performance on each subtask for the `BLiMP Supplement` dataset. Cells highlighted in green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	determiner_noun _agreement_with_adj _irregular_1	64.62	74.51	71.87	76.32	50.56	62.53	49.86	67.69
	principle_A_domain_3	51.75	51.97	48.67	51.22	48.46	48.57	49.31	45.59
	sentential_negation _npi_scope	47.65	61.31	57.52	55.57	56.83	54.54	55.57	50.86
	complex_NP_island	41.13	51.89	41.61	54.37	58.87	43.5	62.17	41.13
	irregular_plural _subject_verb _agree-ment_1	55.35	64.68	63.06	64.18	49.5	57.71	51.87	60.45
	distractor_agreement _relational_noun	41.62	46.7	47.21	51.27	52.03	46.83	49.37	47.46
	matrix_question _npi_licensor_present	3.98	44.78	4.2	33.05	84.5	59.85	35.84	39.5
	passive_2	70.65	70.32	72.54	72.2	70.32	70.1	72.09	64.12
	adjunct_island	78.45	64.12	48.38	66.38	55.6	59.81	63.25	56.03
	wh_vs_that_with_gap	16.1	26.55	8.05	25.9	12.19	14.47	35.8	17.74
	irregular_past _partici-ple_adjectives	59.63	66.6	79.19	63.68	46.51	48.8	45.37	67.01
	drop_argument	71.96	74.02	73.8	76.41	70.87	70.0	70.11	68.91
	principle_A_domain_2	49.62	57.7	57.16	59.02	46.34	50.93	50.82	56.28
	anaphor_gender _agreement	45.21	46.04	36.77	47.79	74.46	47.79	42.33	39.55
	wh_questions_subject _gap_long_distance	93.0	85.53	97.9	89.5	81.68	88.8	61.38	89.96
	only_npi_licensor _present	61.68	74.72	93.99	52.72	72.22	92.52	97.05	58.28
	intransitive	54.84	60.02	53.57	61.98	57.49	59.1	57.6	60.14
	ellipsis_n_bar_1	43.64	49.88	52.37	59.6	38.4	61.97	51.0	52.12
	regular_plural_subject _verb_agreement_1	44.16	58.54	53.15	58.76	49.44	55.84	45.96	61.12
	principle_A_domain_1	84.57	93.0	96.83	91.79	57.99	93.0	93.22	80.42
	irregular_past _partici-ple_verbs	63.8	65.39	58.6	59.45	61.04	66.56	49.26	68.05
	sentential_subject _is-land	54.63	62.12	67.33	56.71	53.69	51.93	49.84	63.89

Table 11: BLIMP - individual task results. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	wh_vs_that_with_gap _long_distance	13.52	10.0	5.49	10.22	15.6	8.46	40.77	12.2
	principle_A_recons-truction	54.6	50.36	53.05	35.26	56.05	53.26	50.47	55.43
	regular_plural_subject_verb _agreement_2	55.03	66.88	64.44	68.25	48.99	51.43	51.22	61.9
	ellipsis_n_bar_2	29.59	51.93	31.76	53.26	37.92	45.41	33.57	55.68
	determiner_noun _agreement_with _adj_irregular_2	65.36	75.71	70.12	77.5	60.0	65.12	57.26	68.33
	passive_1	78.1	71.55	80.48	76.19	70.36	75.83	77.02	71.9
	irregular_plural _subject_verb _agree-ment_2	59.64	68.61	71.86	67.94	48.88	60.09	55.83	69.06
	existential_there _sub-ject_raising	54.11	75.65	56.06	77.81	59.74	67.42	55.3	71.21
	left_branch_island _echo_question	52.69	18.69	61.14	18.27	22.39	23.34	6.65	33.37
	expletive_it_object _raising	63.9	63.77	62.32	62.45	63.37	64.16	61.92	63.77
	coordinate_structure _constraint_object _extraction	36.14	33.4	51.74	53.74	40.99	50.26	46.36	61.54
	causative	58.07	67.48	56.48	70.17	52.57	60.15	50.12	59.78
	npi_present_2	38.4	61.38	45.19	58.64	46.28	61.6	26.15	44.64

Table 12: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	animate_subject_trans	46.05	44.53	22.64	38.68	30.55	49.84	64.46	66.31
	transitive	69.93	73.04	71.08	75.23	52.65	63.59	60.25	58.99
	determiner_noun _agreement_with_adj_2	65.99	78.53	65.57	81.62	50.05	60.04	56.11	70.24
	determiner_noun _agreement_irregular_2	75.12	81.34	72.2	84.88	63.17	73.78	61.71	77.56
	left_branch_island _simple_question	46.37	36.8	62.78	35.44	39.54	45.53	33.96	37.64
	wh_vs_that_no_gap	85.13	91.17	94.19	94.89	90.36	93.26	64.0	93.26
	tough_vs_raising_2	67.72	69.24	74.57	72.5	51.74	63.7	56.85	72.07
	principle_A_case_1	99.78	100.0	99.78	100.0	93.31	98.79	98.25	98.03
	wh_questions_subject_gap	81.51	85.41	91.43	88.86	82.63	89.2	72.16	87.53
	only_npi_scope	35.72	50.3	69.3	46.12	79.81	61.05	75.03	39.67
	distractor_agreement _relative_clause	43.51	46.73	40.07	44.78	54.31	48.91	53.16	48.56
	existential_there _quantifiers_2	58.29	17.34	38.31	30.63	19.87	34.03	21.08	18.33
	determiner_noun _agreement_1	74.27	81.92	71.69	84.39	56.51	70.72	58.56	75.03
	superlative_quantifiers_1	61.08	71.71	48.52	85.39	51.17	39.43	57.3	37.59
	determiner_noun _agreement_with_adjective_1	64.84	80.49	69.77	81.89	56.81	63.88	57.56	71.28
	sentential_negation _npi_licensor_present	90.64	99.35	99.56	92.49	91.95	99.56	72.91	98.91
	wh_questions_object_gap	55.65	49.71	73.69	57.97	73.11	64.96	72.53	60.3
	determiner_noun _agreement_2	69.92	80.88	71.21	82.92	52.52	66.38	57.14	75.94
	existential_there _quantifiers_1	78.06	92.15	77.96	94.52	75.48	66.77	74.73	68.6
	inchoative	43.04	50.53	40.12	52.16	43.63	49.01	44.91	50.76
	coordinate_structure _constraint_complex_left _branch	40.07	30.13	55.08	27.37	35.76	38.41	33.11	30.13
	superlative_quantifiers_2	86.51	75.56	88.03	79.11	78.19	48.68	76.27	46.96
	npi_present_1	40.48	52.59	53.14	57.43	48.4	57.98	50.72	57.87
	wh_island	17.71	27.92	32.08	51.88	61.25	18.12	48.75	40.42
	existential_there_object _raising	70.44	66.13	67.73	60.96	68.23	70.94	66.26	67.98

Table 13: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	wh_vs_that_no_gap_long_distance	86.4	94.4	94.97	96.57	89.6	94.29	61.37	93.37
	principle_A_c_command	69.13	71.88	66.07	75.58	57.61	75.69	66.17	78.12
	animate_subject_passive	61.45	70.28	73.85	72.18	63.13	65.14	60.67	72.51
	anaphor_number_agreement	73.15	80.34	62.41	86.14	71.0	72.82	49.41	74.22
	determiner_noun _agreement_irregular_1	64.61	70.63	67.25	75.18	59.47	62.56	54.63	73.57
	tough_vs_raising_1	33.12	49.89	28.69	46.62	51.9	46.41	47.36	39.45
	principle_A_case_2	62.84	77.27	72.35	79.23	54.97	62.95	48.96	62.62

Table 14: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.