

Developmentally Plausible Multimodal Language Models Are Highly Modular

Alina Klerings¹

Christian Bartelt¹

Aaron Mueller^{2,3}

¹ University of Mannheim ² Northeastern University ³ Technion – IIT
alina.klerings@uni-mannheim.de aa.mueller@northeastern.edu

Abstract

Large language models demonstrate emergent modularity, where functionally specialized components and circuits arise to handle specific tasks or task formats. If similar modules arise in models trained on more cognitively plausible datasets, it could inform debates surrounding what kinds of mechanisms would be learnable given more human-like language learning signals. In this paper, we describe a multimodal vision-language model submitted to the BabyLM Challenge. Our model achieves similar performance to the best-performing architectures from last year, though visual information does not improve performance on text-only tasks over text-only models (in accordance with prior findings). To better understand how the model processes the evaluation tasks of the BabyLM Challenge, we leverage causal interpretability methods to locate the neurons that contribute to the model’s final decisions. We find that the models we train are highly modular: distinct components arise to process related tasks. Furthermore, on text-and-image tasks, adding or removing visual inputs causes the model to use distinct components to process the same textual inputs. This suggests that modal and task-specific specialization is efficiently learned, and that a high degree of functional specialization arises in even small-scale language models.

1 Introduction

Despite impressive capabilities across a wide range of tasks, language models (LMs) remain highly data-inefficient: LMs typically require orders of magnitude more data during pretraining than humans encounter over their entire lifetime (Gilker et al., 2017). This inefficiency has driven interest in alternative approaches to language learning that leverage more human-like language learning scenarios. One such effort is the BabyLM Challenge (Warstadt et al., 2023), which promotes the development of language models trained on the

quantity of linguistic input that children receive when learning language. To create a more developmentally plausible training setup, the 2024 iteration of the challenge (Choshen et al., 2024) provides aligned image and text data.

Evaluating these more cognitively plausible models requires a focused analysis not only of how models behave, but also of the mechanisms¹ underlying their behaviors. Conventional benchmarks are finite and often deploy identically distributed train/test splits, causing us to overlook key aspects of how models generalize. To address this, mechanistic interpretability has emerged as a framework for obtaining a more algorithmic understanding of how neural networks perform particular behaviors. This typically entails causally attributing model behavior to specific components, or causal graphs composed thereof.

We conduct a study around one of the baseline architectures from the BabyLM Challenge that incorporates both language and vision: the generative image transformer (GIT; Wang et al., 2022). We train and evaluate a suite of language-only and multimodal models with this architecture to investigate the role of visual inputs in language learning. Specifically, we first examine how different weighting schemes for text and image-text loss signals affect model performance and assess whether visual input offers any benefit for language learning. As expected, visual data leads to enhanced performance on multimodal benchmarks compared to text-only models. However, we find no significant benefit of visual data for performance on text-only benchmarks. This supports prior findings of a multimodal submission from last year’s BabyLM Challenge (Amariuca and Warstadt, 2023), as well as findings of Zhuang et al. (2024).

Then, using attribution patching (Syed et al., 2023), we identify the most causally important neu-

¹At a high level, a mechanism can be defined as a causal graph describing how inputs are transformed into outputs.

rons in GIT’s text decoder across tasks. This analysis reveals a high level of modularity,² with separate internal mechanisms being deployed even for slightly different subtasks of the same task. Most surprisingly, the same textual input is processed differently in the text decoder depending on whether visual inputs are present. This suggests that visual inputs do not merely add to textual information, but rather activate distinct mechanisms in the model’s language processing components. These findings suggest that modal and task-specific specialization is efficiently learnable in human-like learning scenarios, even in the absence of human-like learning biases.³ These findings extend prior work on emergent modularity in pre-trained language models (e.g., Zhang et al., 2023; Csordás et al., 2021; Agarwala et al., 2021) to a more cognitively plausible training scenario, thus allowing us to make more convincing claims as to what kinds of linguistic functional specializations can arise from human-like language learning signals.

Our main contributions are as follows:

- An analysis of what small-scale language models gain from visual inputs over pure text.
- A causal analysis of which text decoder neurons perform each BabyLM evaluation task, and how the addition of vision data changes these component sets.
- A suite of minimally differing autoregressive text-only and text-and-image models for future analyses.⁴

2 Related Work

Small-scale multimodal language modeling

Many believe that grounding text data in some symbolic representation or alternate modality is necessary for robust language understanding (Bender and Koller, 2020; Bisk et al., 2020, *inter alia*). Thus, assuming the training corpus is no more than what a human could realistically be exposed to when learning language, the addition of aligned visual data may provide an even better test ground for understanding what kinds of structures are learnable from data alone (without a human-like inductive bias).

²In this context, “modularity” refers to function-based neuron grouping (Zhang et al., 2023), where particular neuron clusters have specific functions.

³This degree of modularity is not necessarily desirable nor undesirable; see §5.

⁴Our code and models are publicly available:
https://github.com/klerings/babylm_analysis

Recent related work has investigated whether visual inputs can aid in word learning, finding largely negative results—but crucially, visual inputs *are* helpful in the kinds of low-resource scenarios we investigate (Zhuang et al., 2024). The 2023 BabyLM Challenge received many multimodal submissions; most relevant to ours is the text-and-vision submission of Amariucaí and Warstadt (2023).

Mechanistic interpretability Mechanistic interpretability methods allow us to more deeply understand where and how particular tasks are accomplished in a neural network. This paper focuses more on *localizing* than qualitatively *explaining* model behavior—but localization can itself reveal whether certain behaviors are performed using the same underlying mechanisms. For example, one line of work aims to causally quantify whether the most important neurons for a particular task overlap with those from highly related tasks in language models (e.g., Finlayson et al., 2021; Sankaranarayanan et al., 2024). There also exist investigations of the mechanisms underlying how vision-language models accomplish particular tasks (e.g., Palit et al., 2023; Salin et al., 2022). Past work has used other (not always causal) methods to discover that language models are highly modular; this includes work with small-scale CNN and LSTM-based models (Csordás et al., 2021; Agarwala et al., 2021), as well as large Transformer-based models (Zhang et al., 2023).

Our work extends this literature through analyses of developmentally plausible multimodal language models. We investigate whether these models use similar mechanisms to perform diverse natural language processing (NLP) tasks, and whether they use the same mechanisms to perform the same tasks with and without image data. While our models are not directly comparable to human learners due to differing inductive biases and a relatively small quantity of visual inputs, they nonetheless provide evidence as to the kinds of mechanisms that are learnable from a realistic language learning dataset.

3 Methods

3.1 Model Training

We closely replicate the challenge baseline setup as a foundation for our causal analysis, with the goal of mechanistic insights rather than model optimization. Specifically, we train a series of generative image transformer (GIT; Wang et al., 2022)

models on the official training data for the multimodal track of the BabyLM Challenge (Choshen et al., 2024). The corpus is composed of two parts: one half consists of text-only data—primarily transcribed speech and child-directed language—while the other half is composed of paired image-caption data from sources such as Localized Narratives (Pont-Tuset et al., 2020) and Conceptual Captions (Sharma et al., 2018).

GIT Architecture The GIT architecture consists of two main components: an image encoder and a text decoder. For the image encoder, we use DINOv2 (Oquab et al., 2024), a Vision Transformer (ViT; Dosovitskiy et al., 2021), which is pretrained independently in a self-supervised manner using only image data, thus not counting towards the word budget imposed by the challenge. The text decoder is then jointly pretrained with the image encoder on image-text pairs, following a causal language modeling objective.

GIT also offers the advantage that it can function as a decoder-only language model when image input is absent, enabling additional training on text-only data and facilitating evaluation on both unimodal and multimodal tasks.

Multimodal Loss GIT uses a standard cross-entropy loss for language modeling, which is computed over two types of training data: (1) samples containing both images and text (from Localized Narratives and Conceptual Captions) and (2) text-only samples (from the BabyLM corpus). These two types of data are handled separately during training, with distinct loss terms for each.

For samples that include both images and text, the model computes a loss by predicting the caption tokens, conditioned on the preceding text tokens and the projected image encoding. This loss is denoted as $\mathcal{L}_{\text{multi}}$. Notably, the image input from this corpus can be disabled to simulate a language-only model.

For text-only samples (from the BabyLM corpus), the model computes a unimodal loss, \mathcal{L}_{uni} , where each token is predicted based solely on the preceding text tokens.

The total loss during training is a weighted sum of these two components:

$$\mathcal{L} = w_1 \mathcal{L}_{\text{multi}} + w_2 \mathcal{L}_{\text{uni}} \quad (1)$$

We investigate the impact of varying weight con-

figurations⁵. A configuration denoted as 1/1 implies equal weighting ($w_1 = w_2$), while 1/0.5 refers to $w_1 = 1$ and $w_2 = 0.5$.

When we include images in the captions corpus, the weights w_1 and w_2 not only determine the degree of emphasis placed on child-directed language in the BabyLM corpus, but also adjust the contributions of multimodal and unimodal loss signals during training. For more information on implementation and hyperparameters, see App. A.

3.2 Benchmarks

We evaluate our models on the official benchmarks of the BabyLM Challenge to verify their competitiveness with the challenge baselines and ensure relevance of any conclusions drawn from the subsequent analysis. For language understanding this includes BLiMP (Warstadt et al., 2020), BLiMP Supplement (Warstadt et al., 2023), EWoK (Ivanova et al., 2024) and GLUE (Wang et al., 2018, 2020), see Table 7 in App. C.2 for examples. BLiMP and its supplement consist of sentence pairs with one grammatically correct and one incorrect sentence. EWoK tests logical entailment requiring world knowledge and reasoning, where the model must choose the more semantically likely of two continuations given prior context. Accuracy on BLiMP and EWoK is measured by how often the model assigns a higher probability to the correct sentence. Meanwhile, GLUE tests natural language understanding after task-specific finetuning.

To assess combined textual and visual understanding, the BabyLM Challenge evaluates on the visual question answering benchmark VQAv2 (Goyal et al., 2019) using 7 distractor answers, as well as on Winoground (Thrush et al., 2022) and DevBench (Tan et al., 2024). Winoground includes images paired with two sentences: one accurately describing the image, and another minimally differing sentence that reflects a contrasting scenario. For samples in DevBench, the model must instead select one of multiple images given a textual concept or scenario. These are each evaluated in a zero-shot manner. In addition to the BabyLM evaluation tasks, we evaluate on the visual question answering benchmark MMStar (Chen et al., 2024), which has been manually curated to exclude questions that could be answered via linguistic information alone.

⁵Since increasing the relative importance of one loss component is equivalent to decreasing the importance of the other, we only experiment with varying w_1 .

3.3 Baselines

We compare our baseline replication against the released baselines from the BabyLM competition. For text-and-vision tasks, this includes Flamingo (Alayrac et al., 2024) and GIT, which are trained on the multimodal BabyLM training corpus. For text-only tasks, this also includes last year’s winning architectures, BabyLlama (Timiryasov and Tastet, 2023) and LTG-BERT (Georges Gabriel Charpentier and Samuel, 2023), both trained on the official training data from the Strict track, comprising the same number of words as the multimodal corpus.⁶

3.4 Attribution Patching

We causally attribute model behaviors to specific neurons to determine whether the most important components are shared across task settings. A key technique for this purpose is **attribution patching** (Syed et al., 2023) with integrated gradients (AP-IG; (Hanna et al., 2024; Marks et al., 2024)), a linear approximation of the computationally more expensive activation patching (Vig et al., 2020; Finlayson et al., 2021; Geiger et al., 2021). Activation patching entails intervening on the activation of a model component during a forward pass; the extent to which this intervention changes the model behavior is measured as the **indirect effect** (IE). Activation patching is often used with contrastive input pairs, where activations from one prompt are transferred into a forward pass on a minimally different prompt. It also supports interventions like setting the activation to zero⁷ or replacing the activation with its mean across some dataset.

In attribution patching, rather than directly patching neuron activations, the indirect effect is linearly approximated by multiplying the gradient of the target metric m with respect to the neuron’s activation x by the difference between the original activation x and the counterfactual activation x' :

$$\hat{\text{IE}} = \frac{\delta m}{\delta x} \cdot (x' - x) \quad (2)$$

The gradient can be viewed as a local approximation of how much changing the neuron’s activation would affect m , so multiplying this by how much x changes gives us an estimate of how much

⁶But from a different distribution. The 50M words of image-caption data are replaced by data more closely resembling the text-only corpus’s distribution.

⁷This is not entirely principled and may even be out-of-distribution for the network, as a neuron’s baseline value will not necessarily be 0.

m will change. Typically, m is the logit difference between a correct token completion and minimally different incorrect token completion. High-magnitude $\hat{\text{IE}}$ values indicate that a neuron significantly influences a particular model behavior.⁸

Benchmark-specific prompts and metrics For BLiMP, we select a subset of subtasks consistent with the “one-prefix-method” (Linzen et al., 2016) which ensures that both sentences of a pair share an initial phrase but diverge at a critical word that determines grammaticality. This format generalizes well to VQA, where the logit difference is computed between the target answer and the first distractor that consists of a single token.

Attribution patching is primarily suited to cases where the correct and counterfactual answers can be distinguished by a single token. This is not the case for the other tasks of the BabyLM challenge. Therefore, we adapt the prompt structure and target metric to suit the specific nature of each benchmark, as illustrated in Table 7 in App. C.2.

While MMStar has a multiple-choice structure similar to VQA, the answer choices often exceed a single token in length, rendering the single-token logit difference metric unsuitable. For EWOK and Winoground, the tasks are not formulated as question-answer pairs; instead, the objective is to select the more plausible sentence given a preceding sentence or image. Accordingly, we employ an alternative metric that compares the sum of logits for the entire correct sentence S_1 against the sum for the entire incorrect sentence S_2 , given a textual or visual context. In other words, $m = \sum_{s_1 \in S_1} p(s_1) - \sum_{s_2 \in S_2} p(s_2)$. For EWOK, we repeat the context sentence following the first context and continuation; these are separated by a newline, allowing the model to process the full text input for each comparison (and thus allowing us to backpropagate after comparing $p(s_1)$ and $p(s_2)$). In Winoground, the context consists of the image representation, and both possible description sentences separated by newlines. Similarly, for MMStar, the prompt is made up of the image and both question-answer pairs (where the question is repeated), separated by newlines.

This design presents a challenge: prior work has shown that language models can be semantically

⁸This includes positive as well as negative $\hat{\text{IE}}$ values. An example of components that *negatively* and significantly impact performance are Negative Name Mover Heads in the Indirect Object Identification task (Wang et al., 2023).

	BLiMP	BLiMP-Supp.	EWoK	GLUE	Avg.	Avg. w/o GLUE
Baseline Models						
BabyLlama (100M)	73.1	60.6	52.1	69.0	63.7	61.9
LTG-BERT (100M)	69.2	66.5	51.9	68.4	64.0	62.5
Flamingo	70.9	65.0	52.7	69.5	64.5	62.9
GIT	65.2	62.7	52.4	68.3	62.2	65.1
Multimodal Models						
GIT 1/1	70.0 (2.03)	65.8 (2.26)	51.9 (0.75)	-	-	62.6
GIT 1/0.5	68.9 (1.41)	64.1 (1.96)	52.7 (0.40)	-	-	61.9
GIT 1/0.25	71.2 (1.34)	64.6 (2.29)	52.5 (0.20)	-	-	62.8
GIT 1/0.125	66.3 (1.88)	61.7 (1.44)	52.3 (0.91)	65.6	61.5	60.1
Language-only Models						
GIT 1/1	72.0 (1.54)	65.6 (1.89)	51.9 (0.39)	66.5	64.0	63.2
GIT 1/0.25	71.6 (1.22)	64.0 (2.32)	52.6 (0.38)	-	-	62.7

Table 1: Results for text-only benchmarks averaged across 3 random seeds. Avg. columns refer to macroaverage over the respective tasks. For GIT, we show the corpus weightings as w_1/w_2 .

and syntactically primed (Meyer and Schvaneveldt, 1971; Neely, 1977; Bock, 1986) to favor text more similar to prior text that has already been seen in the same context (van Schijndel and Linzen, 2018; Prasad et al., 2019). Therefore, we randomly alternate the order of correct and incorrect continuations to account for priming effects on average across examples. While this will not yield accurate *behaviors* per se, we care more about the relative probability *change* between $p(S_1)$ and $p(S_2)$ when a component is ablated, rather than their actual values; this design will still allow us to measure this quantity when averaging across inputs.

For each benchmark, we retrieve the 100 most important MLP neurons in the text decoder by $\hat{\mathbf{I}}\mathbf{E}$ over all layers. We obtain the top neurons for each subtask within a benchmark. For some tasks that do not have subtasks such as VQA and Winoground, we automatically generate subcategorizations of examples. For more information on subtask definitions and the automatic subcategorization procedure, see App. C.1. We exclude DevBench from this analysis because its samples consist of multiple images, each requiring a separate forward pass, rendering attribution patching unfeasible.

4 Results

We train and evaluate four weighting configurations for the multimodal model and two for the text-only model; for each configuration, we average

across three random seeds. Detailed information on the learning progress of each model is provided in App. B.

4.1 Benchmarking Results

We use the challenge benchmarks to validate that our models perform sufficiently well for meaningful neuron analysis. To explore the impact of visual information on language-only and multimodal learning, we evaluate all models on both text-only^{9,10} and text-vision benchmarks.

Furthermore, we test the multimodal model’s performance on vision tasks without image input, simulating its behavior as a language-only model. The average and standard deviation across all random seeds are presented in Tables 1 and 2.

Text-only Results For the text-only benchmarks, our models are on par with or slightly below the performance of the baseline models, except for GLUE, which we exclude from our causal attribution study.

There is no single weighting configuration that consistently performs best across all datasets, but

⁹Due to computational constraints, only the best model per modality and random seed is reported for GLUE. The best unimodal model is selected from 1/1 to ensure a fair comparison with other language-only models that similarly balance loss signals across all samples.

¹⁰The GLUE metric is an unweighted mean of each subtask accuracy, except QQP and MRPC (where we use F1 scores), and CoLA (where we use the Matthews correlation coefficient).

Input	VQA		Winoground		DevBench	MMStar		Avg.
	multimodal	text-only	multimodal	text-only	multimodal	multimodal	text-only	multimodal
Baseline Models								
Flamingo	52.3	45.0	51.6	50.0	60.1	24.1	22.6	47.0
GIT	54.1	48.4	55.5	50.0	50.5	25.9	22.4	46.5
Multimodal Models								
GIT 1/1	51.5 (3.52)	49.2 (1.01)	55.4 (0.13)	50.0 (0.0)	48.7 (1.22)	25.1 (0.35)	23.0 (0.57)	45.1
GIT 1/0.5	53.1 (1.40)	47.5 (1.09)	55.9 (2.46)	50.0 (0.0)	50.2 (1.50)	24.3 (0.57)	21.8 (0.98)	45.7
GIT 1/0.25	52.2 (1.12)	47.4 (0.81)	56.2 (0.79)	50.0 (0.0)	47.6 (0.75)	25.8 (0.18)	22.5 (0.73)	45.3
GIT 1/0.125	52.6 (1.40)	48.6 (0.68)	57.0 (0.66)	50.0 (0.0)	47.8 (2.52)	26.7 (0.52)	22.6 (1.41)	45.9
Language-only Models								
GIT 1/0.1	-	49.4 (0.72)	-	50.0 (0.0)	-	-	22.9 (1.33)	-
GIT 1/0.25	-	48.0 (0.60)	-	50.0 (0.0)	-	-	24.0 (1.21)	-

Table 2: Results for multimodal benchmarks with (multimodal) and without (text-only) visual input averaged across 3 random seeds. “Avg.” is a macroaverage over multimodal tasks. For GIT, we show loss weightings as w_1/w_2 .

the models achieving the highest average performance are 1/1 for the language-only setup and 1/0.25 in the multimodal case. This is contrary to observations regarding the evaluation loss (App. B), where lower weightings on BabyLM data samples (w_2) correlated with performance improvement.

No significant performance differences are observed between models trained on textual data alone and those incorporating both text and image inputs, when comparing the same weightings. This suggests that the addition of multimodal data does not yield measurable improvements in this specific context. This aligns with findings from [Zhuang et al. \(2024\)](#).

Multimodal Results In multimodal tasks, our models exceed baseline performance on Winoground and MMStar but show a slight underperformance on VQA and a more significant drop on DevBench.

Results from both language-only and multimodal models without visual input provide validation and confirm that performance decreases substantially when image inputs are excluded. As on the text-only tasks, there is no single multimodal weighting configuration that consistently outperforms across all benchmarks. However, for tasks such as Winoground and MMStar, which require visual input for an above chance performance, the 1/0.125 weighting configuration proves most effective, as it places significantly more emphasis on the visual loss signal during training.

We present learning curves for the best-

performing models in each modality across the BabyLM evaluation tasks in Figure 5 in App. B. For the multimodal model, we observe an order in which phenomena are acquired: BLiMP performance peaks early, whereas EWoK performance gradually improves later in training. In App. B, we discuss this order of acquisition further, and discuss how learning curves differ between multimodal and text-only models.

4.2 Causal Neuron Analysis

To explore whether neuron activation patterns are shared across tasks or modalities, we compute the average indirect effect for each MLP neuron in the text decoder of the strongest multimodal GIT model (1/0.125) per subtask. Then, we select the top 100 neurons by indirect effect and analyze their overlap across subtasks from all benchmarks.

Modularity within benchmarks For text-only benchmarks, the results (Figure 1) indicate a significant degree of neuron sharing in GIT among subtasks within each benchmark. Specifically, for EWoK, over 70% of the top neurons are pairwise shared between subtasks. However, given the low performance on the EWoK benchmark, it is possible that these neurons are not responsible for task solving, but rather pick up on spurious heuristics; we therefore focus on BLiMP and VQA¹¹. Here, we observe a similar though less pronounced trend of intra-benchmark neuron sharing. For BLiMP,

¹¹Note that VQA questions have seven distractor answers, so random chance performance is 12.5%

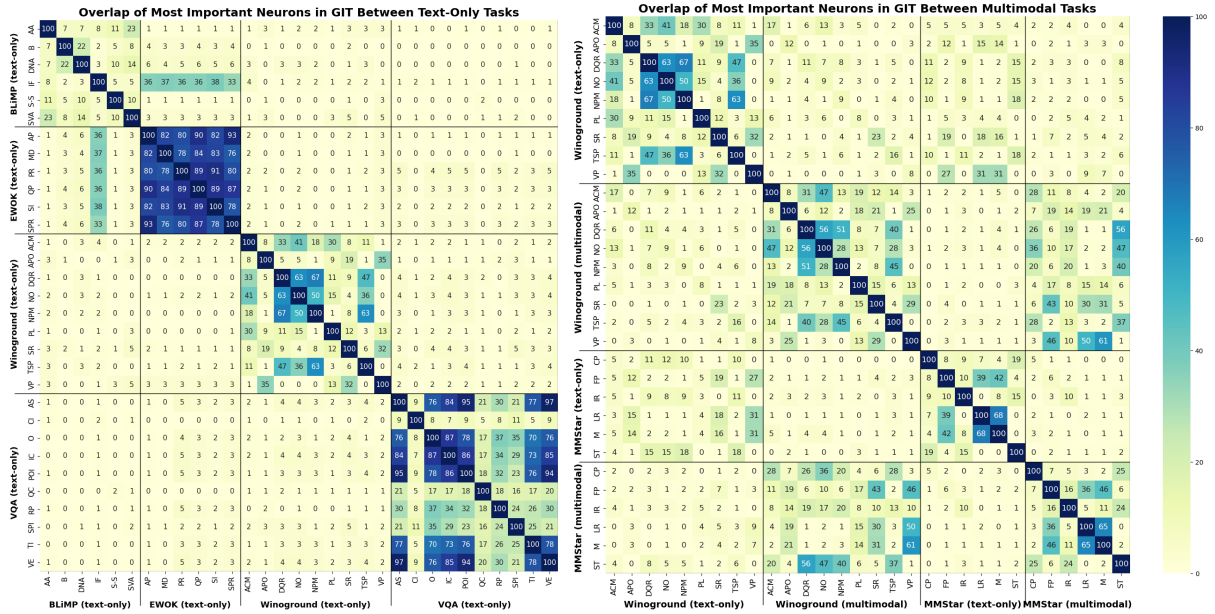


Figure 1: Overlap between top 100 neurons by ÎE per subtask for text-only (*left*) and multimodal benchmarks with and without visual inputs (*right*). Subtask names are abbreviated; see App. C.1 for full names and example counts.

there is an overlap of about 20% between two sub-task pairs. In VQA, many subtask pairs even share over 70% of their task-relevant neurons.

Component sharing across benchmarks The primary factor in determining neuron overlap appears to be task similarity: subtasks within the same benchmark are more similar and display a stronger overlap, whereas tasks across different benchmarks are very distinct and share little neurons. The 30% overlap between Irregular Forms (IF) in BLiMP and all EWoK subtasks is an exception, but models did not score well on these tasks; these could therefore be encoding spurious heuristics or irrelevant information.

Distinct processing of multimodal input A shift in neuron overlap is observed when comparing the same subtasks with and without visual input (Figure 1; full results in Figure 6 in App. C.3). The addition of vision leads to greater overlap of important neurons between all pairs of tasks: for example, the overlap between subtasks in MMStar and Winoground is 30% or less without images, but rises to 40-60% for certain subtasks when visual input is introduced.¹² This increase in shared components is also observed between VQA and Winoground, as well as between VQA and MM-

Star. Interestingly, this increase in shared top components does not extend to intra-benchmark subtasks. Here, we find a mixture of subtask pairs that increase their overlap, mostly in VQA, and subtask pairs that decrease their overlap as in Winoground.

Furthermore, we find the overlap between the same task with and without vision to be minimal for both Winoground and MMStar. This suggests that the presence of visual input significantly changes the mechanisms employed by the language decoder to solve these tasks.

Neuron Sharing in Flamingo To evaluate how well our findings generalize to other multimodal architectures, we conduct a causal neuron analysis on the BabyLM Flamingo baseline model. Unlike GIT, which relies on self-attention, Flamingo integrates vision and text using cross-attention between a frozen image encoder and text decoder.

Flamingo exhibits a similar degree of intra- and inter-task neuron overlap as GIT, with overlap increasing when visual input is added (Figure 7 in App. C.3). However, in contrast to GIT, EWoK displays only selective subtask overlap, aligning more closely with patterns observed in other datasets.

Notably, there is a significant amount of shared neurons between text-only and image-text variants of VQA. This was not observed with GIT (see Figure 2). While adding image inputs in other multimodal cases alters the salient features in the text, Flamingo’s processing of VQA suggests the image

¹²Note that this is the overlap between MLP neurons in the *text decoder*, not in the image encoder. It is not necessarily intuitive that adding visual information should change the text processing mechanisms to this degree.

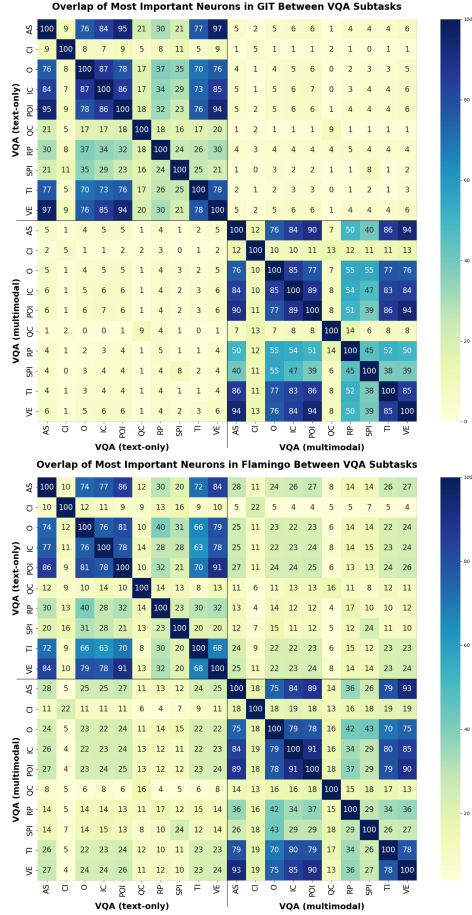


Figure 2: Overlap between top 100 neurons of GIT (*top*) and Flamingo (*bottom*) for subtasks of VQA with and without visual inputs.

supports rather than *redirects* the text decoder.

4.3 Neuron Ablation

To verify the causal influence of the identified top components, we perform a neuron ablation study on VQA. We mean ablate the most influential neurons for each subtask and measure the resulting performance changes, quantifying the effect of the removed information. We consider the top neurons of two settings: (1) text-only, where the multimodal GIT model processes just text, and (2) multimodal, integrating both text and visual inputs. We then mean ablate these distinct neuron sets in the multimodal model. We measure accuracy by the sign of the logit difference between correct answer and first distractor of token length one.

We witness an expected drop in GIT’s performance for eight of the ten VQA subtasks (Figure 3; see Figure 8 in App. C.4 for all subtasks), confirming the task-relevance of the identified top neurons.

When measuring performance after ablations,

we note four patterns. (1) Performance sometimes drops comparably when ablating only text-only neurons, or only text-image neurons. This could indicate that there are more task-relevant neurons shared than the overlap matrix of top 100 neurons implies, or simply that these two sets redundantly encode similar mechanisms. (2) Ablating text-image neurons sometimes results in a greater drop in performance. This suggests that the most important neurons are the ones processing the task multimodally, which could be indicative of successful fusion of vision and text data. (3) Some tasks experience a larger performance drop when ablating the text-only neurons, which means for these tasks, much of the model’s performance can be attributed to question-answer likelihoods rather than visual reasoning. (4) There are two cases where performance *increases* after ablations: Color Identification and Quantity & Counting. Our models achieve comparatively low accuracies on these tasks before ablations; it is thus unclear whether these ablations improve scores because (i) the top neurons encode actively unhelpful spurious information, or (ii) ablating them causes the model to rely on some other heuristic that happens to be more successful (or both).

Similarly, in the pretrained Flamingo model, seven out of ten VQA subtasks show a performance decrease when either text-only or text-image neurons are ablated (Figure 9 in App. C.4). However, the drop is relatively small, indicating the model’s robustness to MLP neuron ablations. This suggests that either more than 100 neurons are involved in task-relevant processes, or that critical processing takes place in other components of the architecture, such as the cross-attention mechanism.

5 Discussion

We find little neuron overlap between vision-and-text and text-only variants of the same tasks. This suggests a significant degree of modularity in small-scale multimodal language models.¹³ This raises important questions: is component sharing between unimodal and multimodal processing mechanisms of the same task desirable? Can it serve as a signal of effective merging of information across modali-

¹³However, Flamingo’s processing of VQA is an exception. This may be due to the training pipeline: text and image encoders are first trained separately, and then cross-attention between these frozen modules is learned using multimodal data. This contrasts with GIT, where text decoder and text-image associations are jointly learned.

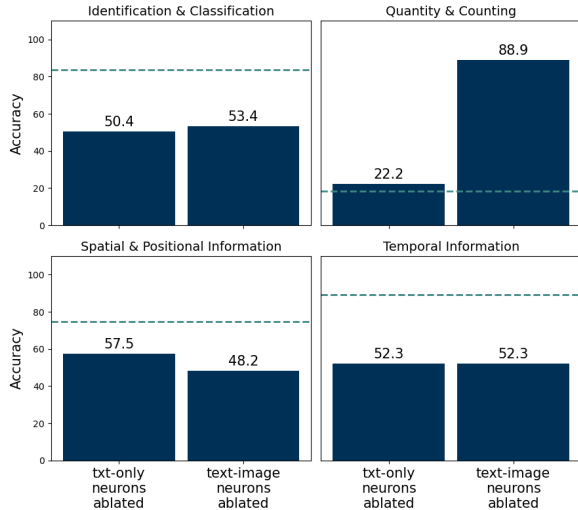


Figure 3: Multimodal accuracy for VQA subtasks when mean ablating GIT’s top neurons. The dashed line indicates accuracy before ablations. The left and right bars show model performance given text and vision inputs when ablating either the top neurons from the text-only version of the task, or the top neurons from the text-and-image version of the task.

ties? To investigate, future research could explore the relationship between neuron overlap and task performance, ideally across diverse architectures.

Many causes could explain the minimal overlap across similar benchmarks. First, the conceptual space in the model representations could be such that there are few features or skills in common across tasks; thus, to the model, these tasks have little in common. Investigating this possibility would involve a more thorough qualitative analysis of the features implicated in performing each task. Second, different tasks may share features, but the model might learn domain-specific versions of qualitatively identical features. Follow-up research could vary task formats—for example, by paraphrasing all examples—and analyze whether this changes the top neurons. That said, there is some correlation between task similarity and component overlap within a benchmark. This serves as a sanity check, and also indicates that even small models tend to share processing mechanisms across closely related tasks with similar formats. This is a more parameter-efficient strategy compared to representing similar tasks in a fully modular fashion.

Is a high degree of task modularity desirable? Some argue that emergent modularity can be harnessed for better generalization in language models (e.g., Qiu et al., 2024); it could also enable more fine-grained mechanistic understanding and con-

trol. However, modularity will generally result in reduced parameter-efficiency. It could also be a signal that a model is not efficiently compressing information in a generalizable way, such that it must relearn similar phenomena for distinct task settings. We speculate that there exist more or less desirable types and extents of modularity in neural language models, and that classifying these types of modularity could be especially helpful in assessing parameter-(in)efficiency.

Relatedly, when speaking of modularity, it is essential to distinguish between two types of neural modules: (i) skill-related neural groups that share general abilities *independent of specific tasks*, and (ii) task-related neural groups that are specialized for particular *task formats*. In our experiments, we predominantly observe the latter. From an engineering perspective, there is no clear indication whether this would enhance performance or efficiency. However, if one’s goal is to model human language processing, perhaps modularity could be a useful signal. Certain regions of the brain specialize toward particular tasks, even in the presence of similar visual stimuli across tasks (Dupont et al., 1993); different specialized regions for the same task can also arise given sufficiently distinct stimuli (Müller et al., 2024). Our findings agree with both. Whether emergent task modules in developmentally plausible language models correspond to comparable regions in the human brain remains an interesting open question.

6 Conclusion

Developmentally plausible multimodal language models exhibit a high degree of modularity. Furthermore, adding visual inputs changes how the text decoder processes a task, and increases the amount of shared components between tasks. Our findings highlight the types of functional specialization that can arise in language models trained on developmentally plausible data, and raise questions about trade-offs between sample-efficiency, parameter-efficiency, and cognitive plausibility.

Acknowledgments

This research is supported by the Ministry of Economic Affairs, Labor and Tourism of Baden-Württemberg and the bwHPC resources of Baden-Württemberg. A.M. is supported by a postdoctoral fellowship under the Zuckerman STEM Leadership Program.

Limitations

Our study focuses on two multimodal architectures. Other models such as CLIP combine visual and language data differently, and therefore, the influence of image data on the model’s behaviors and mechanisms may be qualitatively different. Despite this, our current findings suggest that visual information does not significantly aid in language learning, highlighting the need for novel fusion strategies between the two modalities.

Additionally, there is room for improvement in the scope of the analyzed components during attribution patching. While we primarily examined MLP neurons, which are crucial for language generation, the role of attention layers impacts a model’s decoding ability equally. Future work could investigate the influence of visual data on the emergence of task-specific attention heads, building on prior studies in mechanistic interpretability.

References

- Atish Agarwala, Abhimanyu Das, Brendan Juba, Rina Panigrahy, Vatsal Sharan, Xin Wang, and Qiuyi Zhang. 2021. [One network fits all? modular versus monolithic task formulations in neural networks](#). In *International Conference on Learning Representations*.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, 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. 2024. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Theodor Amariuca and Alexander Scott Warstadt. 2023. [Acquiring linguistic knowledge from multimodal input](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 128–141, Singapore. Association for Computational Linguistics.
- Emily M. Bender and Alexander Koller. 2020. [Climbing towards NLU: On meaning, form, and understanding in the age of data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. [Experience grounds language](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.
- J. Kathryn Bock. 1986. [Syntactic persistence in language production](#). *Cognitive Psychology*, 18(3):355–387.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.
- 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](#).
- Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. [Are neural nets modular? inspecting functional modularity through differentiable weight masks](#). In *International Conference on Learning Representations*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Patrick Dupont, Guy A. Orban, Rufin Vogels, Guy Bormans, Johan Nuyts, Christiaan Schiepers, Michael De Roo, and Luc Mortelmans. 1993. [Different perceptual tasks performed with the same visual stimulus attribute activate different regions of the human brain: A positron emission tomography study](#). *Proceedings of the National Academy of Sciences of the United States of America*, 90(23):10927–10931.
- Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. [Causal analysis of syntactic agreement mechanisms in neural language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.
- Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam

- Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. 2024. [NNsight and NDIF: Democratizing access to foundation model internals](#).
- Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. [Causal abstractions of neural networks](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.
- Lucas Georges Gabriel Charpentier and David Samuel. 2023. [Not all layers are equally as important: Every layer counts BERT](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.
- Jill Gilkerson, Jeffrey A Richards, Steven F Warren, Judith K Montgomery, Charles R Greenwood, D Kimbrough Oller, John HL Hansen, and Terrance D Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American journal of speech-language pathology*, 26(2):248–265.
- Lila Gleitman. 1990. The structural sources of verb meanings. *Language acquisition*, 1(1):3–55.
- Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2019. [Making the v in vqa matter: Elevating the role of image understanding in visual question answering](#). *Int. J. Comput. Vision*, 127(4):398–414.
- Jane Grimshaw. 1979. [Complement selection and the lexicon](#). *Linguistic Inquiry*, 10(2):279–326.
- Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. [Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms](#).
- 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*.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. [Assessing the ability of LSTMs to learn syntax-sensitive dependencies](#). *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. [Dissociating language and thought in large language models](#). *Trends in Cognitive Sciences*, 28(6):517–540.
- Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. [Sparse feature circuits: Discovering and editing interpretable causal graphs in language models](#).
- David E Meyer and Roger W Schvaneveldt. 1971. Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of experimental psychology*, 90(2):227.
- Veronika I. Müller, Edna C. Cieslik, Linda Ficco, Sandra Tyralla, Amir Ali Sepehry, Taraneh Aziz-Safaie, Chunliang Feng, Simon B. Eickhoff, and Robert Langner. 2024. [Not all stroop-type tasks are alike: Assessing the impact of stimulus material, task design, and cognitive demand via meta-analyses across neuroimaging studies](#). *Neuropsychology Review*.
- James H Neely. 1977. Semantic priming and retrieval from lexical memory: Roles of inhibitionless spreading activation and limited-capacity attention. *Journal of experimental psychology: general*, 106(3):226.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. [Dinov2: Learning robust visual features without supervision](#).
- V. Palit, R. Pandey, A. Arora, and P. Liang. 2023. [Towards vision-language mechanistic interpretability: A causal tracing tool for blip](#). In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2848–2853, Los Alamitos, CA, USA. IEEE Computer Society.
- Steven Pinker. 1984. *Language Learning and Language Development*. Harvard University Press.
- Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision – ECCV 2020*, pages 647–664, Cham. Springer International Publishing.
- Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. [Using priming to uncover the organization of syntactic representations in neural language models](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.
- Zihan Qiu, Zeyu Huang, and Jie Fu. 2024. [Unlocking emergent modularity in large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2638–2660, Mexico City, Mexico. Association for Computational Linguistics.

- Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. 2022. [Are vision-language transformers learning multimodal representations? a probing perspective](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11248–11257.
- Aruna Sankaranarayanan, Dylan Hadfield-Menell, and Aaron Mueller. 2024. [Disjoint processing mechanisms of hierarchical and linear grammars in large language models](#). In *ICML 2024 Workshop on LLMs and Cognition*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual captions: A cleaned, hypervised, image alt-text dataset for automatic image captioning](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.
- Aaquib Syed, Can Rager, and Arthur Conmy. 2023. [Attribution patching outperforms automated circuit discovery](#). In *NeurIPS Workshop on Attributing Model Behavior at Scale*.
- Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wan-jing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. [Devbench: A multimodal developmental benchmark for language learning](#).
- 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5238–5248.
- Inar Timiryasov and Jean-Loup Tastet. 2023. [Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.
- Marten van Schijndel and Tal Linzen. 2018. [A neural model of adaptation in reading](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. [Investigating gender bias in language models using causal mediation analysis](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. [Superglue: A stickier benchmark for general-purpose language understanding systems](#).
- 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](#).
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.
- 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.
- Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, Zhiyuan Liu, Ruobing Xie, Maosong Sun, and Jie Zhou. 2023. [Emergent modularity in pre-trained transformers](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4066–4083, Toronto, Canada. Association for Computational Linguistics.
- Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. [Visual grounding helps learn word meanings in low-data regimes](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1311–1329, Mexico City, Mexico. Association for Computational Linguistics.

A Model Training

A.1 Hyperparameters

We train all models for a maximum of 30 epochs, using a learning rate of 1e-4 with a weight decay

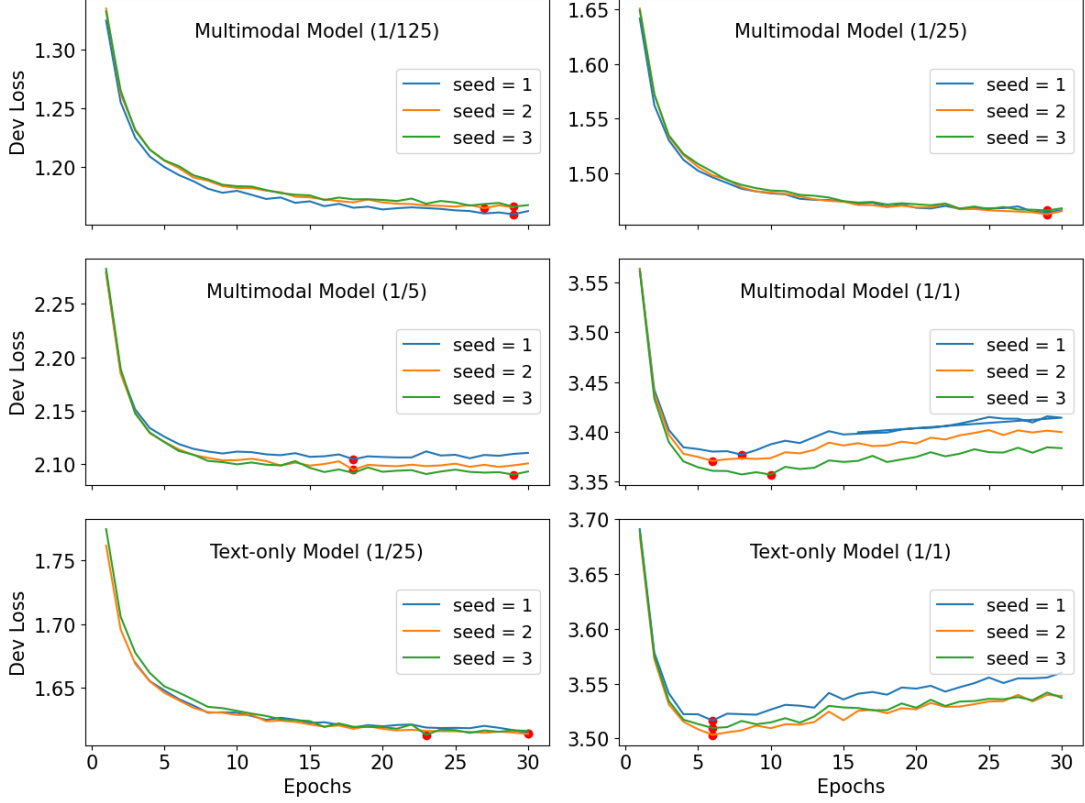


Figure 4: Evaluation loss of GIT per epoch for each of the weighting configurations across three random seeds. Red dot marks the best epoch.

of 0.1. The AdamW optimizer is employed with a batch size of 128, and early stopping is applied to prevent overfitting.

A.2 Tokenization

A single tokenizer is utilized across both unimodal and multimodal models to enhance comparability between the different settings. The tokenizer is trained on the BabyLM corpus as well as the image captions from the Localized Narratives and Conceptual Captions datasets, with a vocabulary size of 32,778 tokens.

B Learning Curves

After every epoch, we compute the validation loss on the unimodal or multimodal development set from the BabyLM Challenge, depending on the model we are working with. We provide learning curves for all weighting configuration in Figure 4.

In the 1/0.125 and 1/0.25 weighting configurations, the loss consistently decreases across seeds and modalities, indicating potential for further improvement with additional epochs. In contrast, the 1/0.5 multimodal models show convergence within the 30-epoch limit. For the 1/1 configura-

tion, where train losses are evenly weighted, overfitting occurs after six to ten epochs in both unimodal and multimodal setups. We conclude that the fusion of language and vision is only learned reliably with a strong multimodal loss signal. For the language-only model, the setting with lower w_2 value exhibits the better convergence, suggesting that language skills and decoding abilities may be more effectively learned from non-child-directed language present in image captions.

We also present learning curves for each benchmarking task. Learning curves for the best-performing models in each modality across benchmarks are visualized in Figure 5. For the multimodal model, there appears to be an order in which phenomena are acquired: task performance on vision benchmarks and the EWoK dataset increases steadily. In contrast, performance on the BLiMP and BLiMP Supplement datasets peaks early in training and subsequently fluctuates or declines. We discuss this in more detail below. The language-only model shows minimal performance change over time on BLiMP, VQA, Winoground, and MM-Star benchmarks, with performance remaining at initial levels. For the EWoK dataset, performance

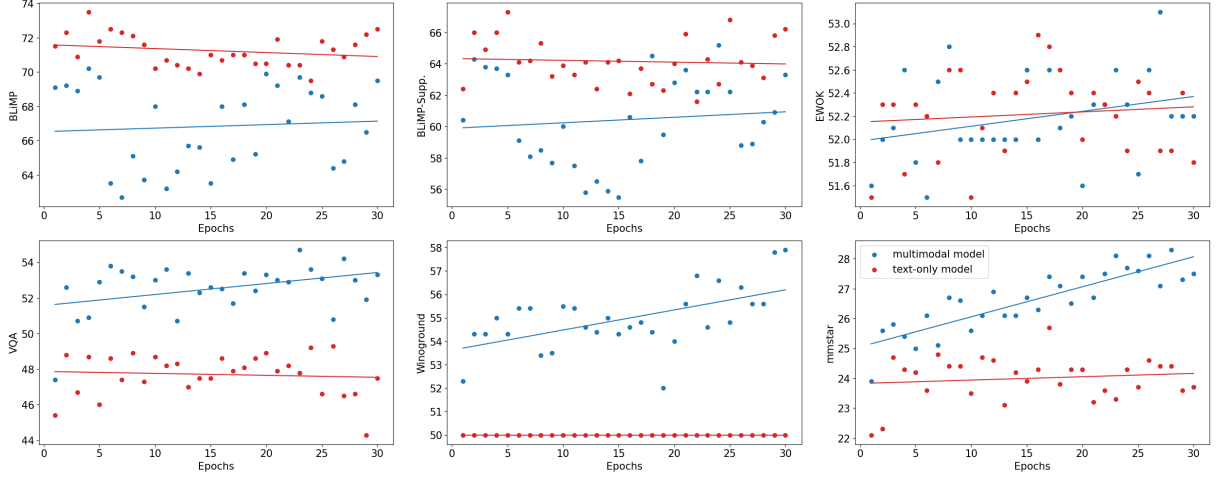


Figure 5: Learning progress of best GIT model per modality on each benchmark. Moving average smoothing is applied with window size 3.

peaks around 15 epochs before declining, whereas on the smaller BLiMP Supplement task, performance fluctuation occurs almost immediately.

These findings align with observations from evaluation loss curves, where the 1/0.125 multimodal model exhibits continued learning, while the 1/1 language-only model reaches an early local minimum.

Acquiring linguistic abilities in order. We observe a distinct order of acquisition in language models: learning curves across benchmarks indicate an almost immediate proficiency in distinguishing between valid and invalid formal linguistic structures, primarily with respect to morphosyntactic rules. This is reflected in the high scores achieved on both BLiMP and BLiMP Supplement early in training. In contrast, performance on EWoK, a benchmark that assesses more functional (semantic and pragmatic) linguistic abilities in context, improves gradually and slowly over time—and peaks at significantly lower scores. This phased “order of acquisition” deviates somewhat from human language development, where syntactic and semantic signals can assist in learning the other throughout language acquisition (Gleitman, 1990; Grimshaw, 1979; Pinker, 1984). This finding could support the existence of a clear distinction between effective representations of the formal structure of language, and representations of how language should be interpreted and deployed in context (Machowald et al., 2024); nonetheless, this finding is preliminary and should be investigated in more depth and in a greater variety of architectures and learning scenarios.

C Causal Neuron Analysis

C.1 Subtask Categories

The subtask categories are either provided explicitly in the dataset (e.g. EWoK, BLiMP, MM-Star) or automatically aggregated using a large language model. Since the subtask labels in VQA and Winoground are too fine-grained, we leverage ChatGPT-4o to automatically merge them to broader categories. This is achieved in a two-stage process, where we first ask for the generation of superclass labels and then for the assignment of these labels to the fine-grained categories. This process is done in two steps to ensure that each fine-grained label is assigned exactly one superclass, see Table 3

Label Creation

The following is a list of VQA/Winoground question types. It is too fine-grained, merge the categories to 10 combined categories that are reasonable to group together, and give the merged categories a new name...

Assignment

Classify each of these following question types with exactly one of these super categories...

Table 3: ChatGPT-4o prompts used to generate new subtask labels.

In Tables 4 and 5 we provide a mapping between original and superclass label per benchmark and in Table 6 we report the number of samples per supercategory, alongside an abbreviation used in heatmap plots.

Person and Object Identification			
are these	are they	is he	is it
is the man	is the person	is the woman	is this
is this person	what is the man	what is the person	what is the woman
what is this	who is		
Other (General Queries and Miscellaneous)			
are the	what does the	what is	do you
what is the	how	none of the above	what
what are	what are the		
Action and State			
can you	has	could	
Color Identification			
what color	what color is the	what color are the	what color is
what is the color of the			
Verification and Existence			
are	does the	is the	are there
does this	is there a	are there any	is
is there	do	is there	
Identification and Classification			
is that a	what animal is	what is the name	is this a
what kind of	what sport is	is this an	what type of
which	what brand		
Temporal Information			
was	what time		
Spatial and Positional Information			
how many people are in	what room is	where are the	what is in the
where is the	what is on the		
Reason and Purpose			
why	why is the		
Quantity and Counting			
how many	what number is	how many people are	

Table 4: VQA subtask categories with their original question types.

Adjectival Comparisons and Modifications		
Adjective-Age	Adjective-Size	Adjective-Manner
Adjective-Color	Adjective-Color (3-way swap)	Adjective-Shape
Adjective-Texture	Adjective-Animate	Adjective-Weight
Adjective-Temperature	Adjective-Speed	Adjective-Height
Adjective-Manner Phrase	Adjective-Speed Phrase, Verb-Intransitive	Adverb-Animate
Verb Phrases (Intransitive and Transitive)		
Verb-Intransitive	Verb-Transitive	Verb-Transitive Phrase, Verb-Intransitive, Preposition Phrase
Verb-Transitive Phrase	Verb-Intransitive, Noun	Verb-Intransitive Phrase
Verb-Intransitive, Determiner-Numeral	Verb-Intransitive, Adjective-Manner	Verb-Intransitive, Verb-Transitive Phrase
Verb-Intransitive Phrase, Adverb-Animate	Verb-Intransitive Phrase, Preposition	Verb-Transitive, Noun
Noun Phrases and Modifiers		
Noun, Adjective-Color	Noun Phrase, Adjective-Animate	Noun
Noun Phrase	Noun Phrase, Adjective-Color	Noun Phrase, Determiner-Possessive
Noun Phrase, Determiner-Numeral	Noun, Verb-Intransitive	Noun, Preposition Phrase, Scope
Noun, Adjective-Size		
Altered POS		
Sentence	Altered POS	Altered POS, Determiner-Numeral
Preposition and Locations		
Preposition Phrase, Scope	Preposition Phrase	Preposition
Determiner and Quantifier Relationships		
Determiner-Numeral	Determiner-Possessive	Determiner-Numeral Phrase
Determiner-Numeral, Noun Phrase		
Scope and Relations		
Scope	Scope, Preposition, Verb-Intransitive	Scope, Preposition Phrase
Scope, Adjective-Manner	Scope, Adjective-Texture	Scope, Conjunction Phrase
Scope, Relative Clause	Scope, Conjunction	Scope, Verb-Transitive
Scope, Preposition	Relative Clause, Scope	Scope, Preposition Phrase, Adjective-Color
Scope, Altered POS, Verb-Intransitive, Verb-Transitive	Scope, Noun, Preposition	
Negation and Opposites		
Negation, Scope	Negation, Noun Phrase, Preposition Phrase	
Temporal and Spatial Phrases		
Adjective-Temporal	Adjective-Spatial	Adverb-Temporal
Adverb-Spatial Phrase	Adverb-Spatial	

Table 5: Winoground subtask categories with their original question types.

BLIMP (linguistics_term)			MMStar (category)		
Subtask Name	Abb.	Num.	Subtask Name	Abb.	Num.
Subject Verb Agreement	SVA	34	Fine-grained Perception	FP	247
S-Selection	S-S	417	Instance Reasoning	IR	243
Anaphor Agreement	AA	688	Science and Technology	ST	174
Binding	B	1056	Coarse Perception	CP	245
Determiner Noun Agreement	DNA	1710	Math	M	112
Irregular Forms	IF	67	Logical Reasoning	LR	204
VQA			Winoground		
Subtask Name	Abb.	Num.	Subtask Name	Abb.	Num.
Person and Object Identification	POI	3208	Adjectival Comparisons and Modifications	ACM	184
General Queries and Miscellaneous (renamed: Other)	O	4648	Verb Phrases (Intransitive and Transitive)	VP	52
Action and State	AS	286	Noun Phrases and Modifiers	NPM	268
Color Identification	CI	2343	Altered POS	APO	46
Verification and Existence	VE	4894	Preposition and Locations	PL	68
Identification and Classification	IC	2393	Determiner and Quantifier Relationships	DQR	50
Temporal Information	TI	176	Scope and Relations	SR	42
Spatial and Positional Information	SPI	708	Negation and Opposites	NO	18
Reason and Purpose	RP	100	Temporal and Spatial Phrases	TSP	12
Quantity and Counting	QC	27			
EWoK (Domain)					
Subtask Name	Abb.	Num.			
Physical Relations	PR	818			
Spatial Relations	SPR	476			
Physical Interactions	PI	556			
Agent Properties	AP	2056			
Material Dynamics	MD	770			
Social Properties	SP	325			
Social Relations	SOR	1548			
Quantitative Properties	QP	310			
Social Interactions	SI	294			
Physical Dynamics	PD	120			
Material Properties	MP	170			

Table 6: (Aggregated) subtask categories per benchmark with their abbreviation and number of contained samples.

C.2 Prompt Format and Metrics

An example for each prompting format used in attribution patching is given in Table 7, alongside the metric used to compute the patching effect.

C.3 Heatmap for all subtasks

We provide an extensive heatmap for the neuron overlap between subtasks of all benchmarks in Figure 6 for GIT and in Figure 7 for Flamingo.

C.4 Neuron Ablation

We provide the ablation effect for all subtasks of VQA (in their multimodal variant) when ablating the top neurons with their mean activation in Figure 8 for GIT and in Figure 9 for Flamingo.

C.5 Library

To perform attribution patching and neuron ablations, we use `nnsight` (Fiotto-Kaufman et al., 2024).



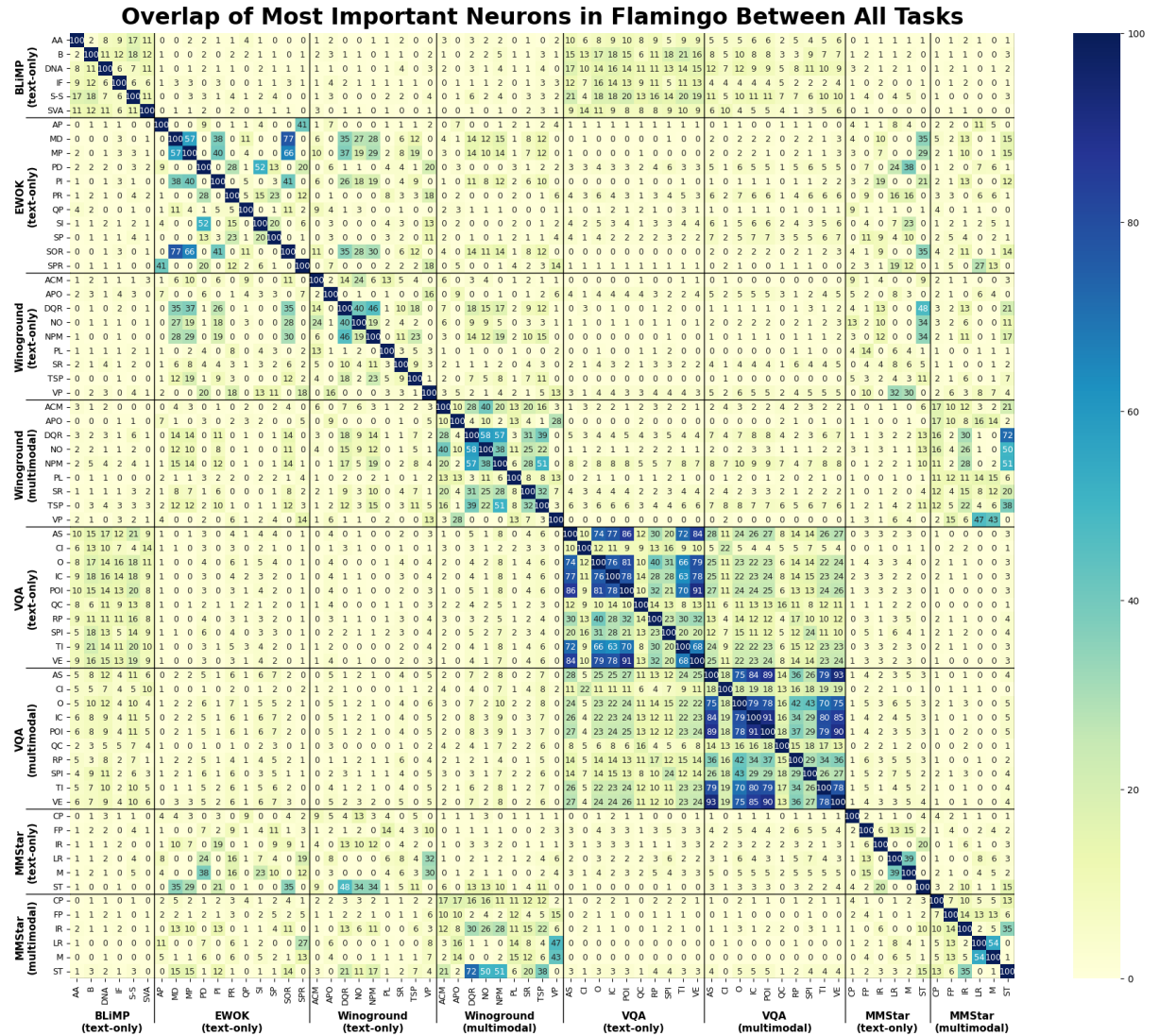


Figure 7: Overlap in Flamingo between the top 100 neurons by indirect effect per subtask for all benchmarks.




Benchmark	Prompt		Metric
BLiMP	The books about Galileo		$\text{logit diff} = \text{final logit}[\text{token}=\text{"are"}] - \text{final logit}[\text{token}=\text{"is"}]$
VQA		Is this photo in color?	$\text{logit diff} = \text{final logit}[\text{token}=\text{"no"}] - \text{final logit}[\text{token}=\text{"yes"}]$
EWoK	Chao is making Yan's job easier. Chao is helping Yan. \n Chao is making Yan's job easier. Chao is hindering Yan.		$\text{logit diff} = \text{logit sum}[\text{"Chao is helping Yan"}] - \text{log sum}[\text{"Chao is hindering Yan."}]$
Winoground		some plants surrounding a lightbulb \n a lightbulb surrounding some plants	$\text{logit diff} = \text{logit sum}[\text{a lightbulb surrounding some plants}] - \text{logit sum}[\text{some plants surrounding a lightbulb}]$
MMStar		What is the main theme of the image? Transportation \n What is the main theme of the image? Outdoor recreation	$\text{logit diff} = \text{logit sum}[\text{What is the main theme of the image? Transportation}] - \text{logit sum}[\text{What is the main theme of the image? Outdoor recreation}]$

Table 7: Example prompts and their respective performance metric per benchmark used for attribution patching.

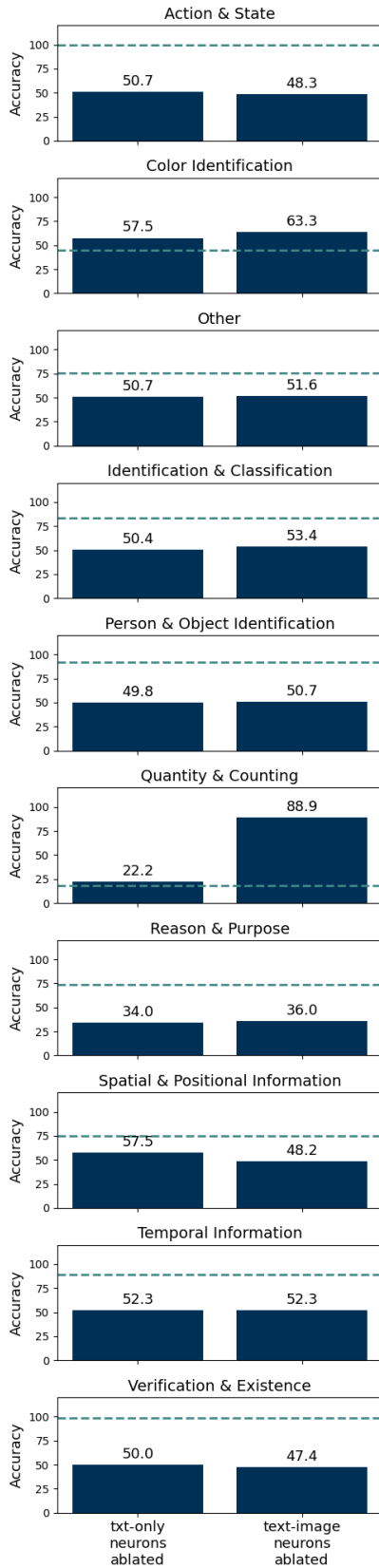


Figure 8: Clean and ablated GIT accuracy on VQA. Dashed line marks clean accuracy. The left and right bars show model performance without vision and with vision respectively.

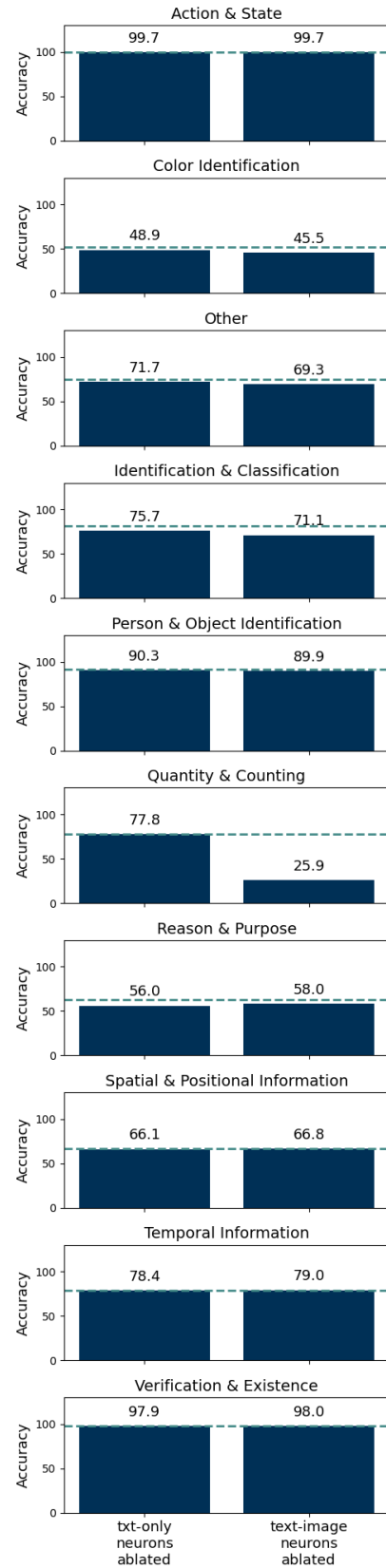


Figure 9: Clean and ablated Flamingo accuracy on VQA. Dashed line marks clean accuracy. The left and right bars show model performance without vision and with vision respectively.