Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora

Michael Y. Hu[1] **Aaron Mueller**[2,3] **Candace Ross**[4]
Adina Williams[4,7] **Tal Linzen**[1] **Chengxu Zhuang**[6] **Ryan Cotterell**[8]
Leshem Choshen[5,6] **Alex Warstadt**[8] **Ethan Gotlieb Wilcox**[9]
[1]New York University [2]Northeastern University [3]Technion [4]Meta AI (FAIR)
[5]IBM Research [6]MIT [7]ML Commons
[8]ETH Zürich [9]Georgetown University
michael.hu@nyu.edu

Abstract

The BabyLM Challenge is a community effort to close the data-efficiency gap between human and computational language learners. Participants compete to optimize language model training on a fixed language data budget of 100 million words or less. This year, we released improved text corpora, as well as a vision-and-language corpus to facilitate research into cognitively plausible *vision* language models. Submissions were compared on evaluation tasks targeting grammatical ability, (visual) question answering, pragmatic abilities, and grounding, among other abilities. Participants could submit to a 10M-word text-only track, a 100M-word text-only track, and/or a 100M-word and image multimodal track. From 31 submissions employing diverse methods, a hybrid causal-masked language model architecture outperformed other approaches. No submissions outperformed the baselines in the multimodal track. In follow-up analyses, we found a strong relationship between training FLOPs and average performance across tasks, and that the best-performing submissions proposed changes to the training data, training objective, and model architecture. This year's BabyLM Challenge shows that there is still significant room for innovation in this setting, in particular for image-text modeling, but community-driven research can yield actionable insights about effective strategies for small-scale language modeling.

1 Introduction

This paper describes the second BabyLM Challenge and its findings. The broader goals and motivation of the challenge have remained constant since the first iteration last year. At the heart of both this year's and last year's challenge is the observation that children are incredibly data-efficient language learners, whereas artificial neural-network-based language models are not. On the one hand, children are exposed to less than 100 million word tokens by the age of 13 (Gilkerson et al., 2017), at which point they have mastered their native language(s). On the other hand, today's ANN-based language models are trained on trillions of words—five to six orders of magnitude more than the typical human language learner. For a more in-depth discussion on the issue of data efficiency, see the findings of last year's challenge (Warstadt et al., 2023) as well as Wilcox et al. (2024), a position piece written by many of the challenge organizers.

The learning discrepancy between humans and models raises two important questions: First, how is it that humans are able to learn language so efficiently? And second, what insights from human language learning can be used to improve language models? It is our hope that by creating a platform for interested parties to experiment with data-limited and cognitively inspired language modeling, we can continue to make progress on these interrelated questions. In particular, our goal with BabyLM is to contribute to:

1. Building more cognitively and developmentally plausible models of human language acquisition and processing, which can be used for the scientific study of language.

2. Optimizing training pipelines prior to scaling, allowing for faster iteration on architectures and hyperparameters.

3. Enabling research on language model training to a wider group of interested researchers, beyond highly-funded industry labs.

The main difference between this year's and last year's challenge is twofold: First, this year we allowed participants to bring their own datasets, as long as they stayed within the 100 million word limit for our *Strict* track, or the 10 million word limit for our *Strict-Small* track. The motivation behind this decision is that pretraining data quality has been linked to large improvement gains in at-scale language models (Gunasekar et al., 2023),

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 1–21
November 15-16, 2024

so this year we allowed participants to improve the training data beyond the provided dataset, which was effectively a dataset baseline. Second, this year included a *Multimodal* track, in which participants trained on aligned text-image data, and tested their models in a novel text-image evaluation pipeline. Non-linguistic information, such as visual input, potentially plays a large role in child language acquisition. While visual input is not inherently necessary for successful language acquisition (for example, blind children learn language largely without issue), visual grounding has been linked to faster language learning (Pérez-Pereira and Castro, 1992; Campbell et al., 2024). Furthermore, visual grounding has long been hypothesized to aid word learning: children learn nouns more easily than verbs (Gentner, 1982; McDonough et al., 2011), arguably because the former are more easily linked to visual stimuli than the latter. Additionally, children learn concrete nouns easier than abstract nouns (Bergelson and Swingley, 2013). However, visual grounding also presents several challenges: Words may be time-delayed with respect to their referents, or one word may be uttered in a context with multiple competing possible referents. With this in mind, our hope was that the *Multimodal* track would help to explore the space of possible computational models for visual grounding during language acquisition.

Findings and takeaways. This year, we received 31 submissions from 17 different countries making diverse contributions. Examples included submissions proposing novel architectures, new training objectives, innovating on knowledge distillation methods, and proposing curriculum learning methods, among others. We conduct a meta-analysis of the results, which yields several concrete recommendations. The best-performing submissions constructed their own training datasets, proposed new model architecture, or new training objectives. Performance on the BabyLM evaluations also correlated strongly with training FLOPs, suggesting that high-compute training regimes still tend to reliably perform better, even in low-data settings. The BabyLM research community also showed growing attention to tokenization and multilingual language modeling, while maintaining interest in curriculum learning and applying linguistic biases to language models.

Our data (pretraining corpora and evaluation data; [link]), preprocessing code [link], baselines [link] and evaluation pipeline [link] are all publicly available. We also release the submitted models of those who agreed to release them, along with their hyperparameters and results [link]. The leaderboard may be found here [link].

2 Competition Details

Tracks. The second BabyLM Challenge included three competition tracks: ***Strict***, ***Strict-Small***, and ***Multimodal***. Additionally, we opened a standalone ***Paper*** track, accepting research related to cognitive modeling with language models or small-scale pretraining, similar to a workshop.

The *Strict* and *Strict-Small* tracks required that submissions be trained on 100M words or less and 10M words or less, respectively. These tracks no longer required that participants use the fixed dataset from last year's challenge, although we still provided an updated version of this dataset, described in Section 3. Models in these tracks were evaluated on language-only evaluation tasks.

In the *Multimodal* track, participants trained multimodal image-text models. Participants were allowed to use any model and training procedure they desired, as long as the model could assign (pseudo) log-likelihoods to strings of text, conditioned on an image. Again, participants were free to construct their own datasets, including unlimited visual inputs, as long as the text data was within a 100M word budget. To facilitate easier participation in this track, we released a suggested multimodal dataset that consisted of 50% text-only and 50% paired image-text data. Submissions to this track were evaluated on both language-only and additional multimodal tasks.

3 Pretraining Corpus

This year, we updated the text-only dataset from the previous competition and provided a novel image-text dataset for the *Multimodal* track. Data for both the text-only and multimodal datasets can be downloaded from `https://osf.io/ad7qg/`.

For the text-only dataset updates, we increased the proportion of child-oriented data (counting both transcribed speech and written data) to 70% up from 39% last year, and we increased transcribed speech data to 58% up from 55% last year. We have eliminated the Wikipedia portion of the data (except for Simple English Wikipedia) due to being the only non-spoken and non-child-level data, and we have eliminated the QED portion due to qual-

Dataset	Description	# Words (multimodal)	# Words (strict)	# Images
Localized Narratives[a]	Image Caption	27M	–	0.6M
Conceptual Captions 3M[b]	Image Caption	23M	–	2.3M
CHILDES[c]	Child-directed speech	14.5M	29M	–
British National Corpus (BNC), dialogue portion[d]	Dialogue	4M	8M	–
Project Gutenberg (children's stories)[e]	Written English	13M	26M	–
OpenSubtitles[f]	Movie subtitles	10M	20M	–
Simple English Wikipedia[g]	Written Simple English	7.5M	15M	–
Switchboard Dialog Act Corpus[h]	Dialogue	0.5M	1M	–
Total	–	100M	100M	2.9M

Table 1: Datasets for the multimodal and strict tracks of the 2nd BabyLM Challenge. Word counts are approximate and subject to slight changes. [a]Pont-Tuset et al. (2020a) [b]Sharma et al. (2018a) [c]MacWhinney (2000) [d]Consortium (2007) [e]Gerlach and Font-Clos (2018) [f]Lison and Tiedemann (2016a) [g]`https://dumps.wikimedia.org/simplewiki/` [h]Stolcke et al. (2000)

ity issues. We have also reduced our reliance on OpenSubtitles, which can include scripted speech, which is arguably less ecologically valid than other spoken sources. CHILDES now comprises a significantly larger portion of the new dataset. We use the entire available English portion of CHILDES including both caregiver and child utterances, increasing the proportion of child-oriented discourse from 5% last year to 29%.[1] We also replaced last year's children's stories and Project Gutenberg data with a custom children's stories dataset sourced entirely from Project Gutenberg. We select child-appropriate books using the provided `subject` metadata, and then select the 1000 most common books, giving us a combined corpus of 26M words. For more details about other data sources, see (Warstadt et al., 2023).

In addition, we provide a novel image-text dataset to facilitate easier participation in the *Multimodal* track. This dataset has two components: First, we provide 50M words of text-only data, drawn from the 100M BabyLM corpus via stratified sampling (that is, we preserve the relative distribution from the different data sources). Second, we provide paired text-image data that includes 50M words of text. This paired data comes from two sources: 27M words from the Localized Narratives dataset (Pont-Tuset et al., 2020b) and 23M words from the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018b). For the Localized Narratives dataset, we used the text captions and the images from the MS-COCO (Lin et al., 2014) and Open Images (Kuznetsova et al., 2020) subsets. For the CC3M dataset, we used the image-caption pairs whose images were still valid in January 2024. In the OSF directory at the above link, we provided scripts to download the images. Table 1 gives an overview of the datasets comprising the BabyLM pretraining set, and descriptions of each data source are provided in Appendix A.

3.1 Preprocessing

We released train, validation, and test splits for each of the ten data sources in *Strict* and *Strict-Small* in proportions 83.3%/8.3%/8.3%, respectively. The 10M word *Strict-Small* training set is sampled randomly from the *Strict* training set: after preprocessing, we downsampled and split each source by randomly sampling chunks of 2000 lines or longer. The code and instructions for downloading and preprocessing the raw data are publicly available.[2]

We performed minimal preprocessing in terms of filtering and reformatting text. Notably, we preserved newlines, meaning newlines do not consistently delimit documents, paragraphs, or sentences, as in some pretraining datasets. We used WikiExtractor (Attardi, 2015) to extract text from the `xml` Simple English Wikipedia dump dated 2022-12-01. We removed `<doc>` tags in Simple English Wikipedia and selected the spoken subset of the BNC by taking only lines from the `xml` containing the `<stext>` tag and extracting the text from the `xml`. We used code by Gerlach and Font-Clos (2020) to download and preprocess data from Project Gutenberg, which we additionally filtered to contain only English texts by authors born after 1850. The OpenSubtitles and Wikipedia portions of the pretraining corpus were shared with us in preprocessed form, having had duplicate documents

[1]We thank Brian MacWhinney (personal correspondence) for alerting us to the existence of this additional CHILDES data.

[2]`https://github.com/babylm/babylm_data_preprocessing`

removed from OpenSubtitles and preprocessing steps performed to Wikipedia similar to our Simple English Wikipedia procedure.[3] We used regular expressions to remove speaker and dialog act annotations from the Switchboard Dialog Act Corpus and annotations from the CHILDES data. We preserved speaker annotations and scene descriptions from CHILDES. We performed no preprocessing on the remaining datasets.

4 Evaluation and Submission

As in last year, we distributed a shared evaluation pipeline based on the LM Evaluation Harness (Gao et al., 2021). For the *Strict* and *Strict-Small* tracks, evaluation tasks were largely the same as the previous year: we used BLiMP (Warstadt et al., 2020), the BLiMP Supplement (Warstadt et al., 2023), and a subset of (Super)GLUE tasks (Wang et al., 2019, 2018a) as the public evaluation set. BLiMP measures whether LMs prefer grammatical to minimally-differing ungrammatical sentences (i.e., minimal pairs) and spans a range of grammatical phenomena including subject-verb agreement, binding, and control/raising constructions. The BLiMP supplement is a disjoint subset of minimal pairs designed specifically for last year's BabyLM Challenge to test linguistic knowledge not covered by BLiMP, such as dialogue and pragmatics. (Super)GLUE is designed to measure natural language understanding across a diverse array of subtasks; its tasks include question answering and natural language inference, among others.

For the *Multimodal* track, participants were required to evaluate on the evaluation benchmarks from the text tracks; this was to establish whether training on image data facilitated sample-efficient language modeling. In addition, we included a suite of multimodal evaluation tasks. The public evaluation datasets included Visual Question Answering (VQA; Antol et al., 2015; Goyal et al., 2017) and Winoground (Thrush et al., 2022). VQA measures whether vision-language models (VLMs) prefer correct answers to questions about visual inputs, and Winoground measures whether LMs prefer accurate descriptions of images among minimally differing options (e.g., given an image of dirt on top of a light bulb, does the VLM prefer "a lightbulb on top of dirt", or "dirt on top of a lightbulb", and vice versa given another image where the lightbulb is on top of dirt).

This year, we used the Elements of World Knowledge (EWoK) dataset (Ivanova et al., 2024) as the hidden task for the text tracks. This task measures pragmatic, commonsense, and discourse knowledge. For the *Multimodal* track, the hidden task was DevBench (Tan et al., 2024); this benchmark contains subtasks targeted at evaluating visual and linguistic abilities that emerge at different stages of children's development, including subtasks where (i) the model must pick the correct image associated with a given word; (ii) the model must pick the correct image corresponding to a sentence; and (iii) the model must assign appropriately higher or lower similarity scores to more or less similar images. The data for these tasks was released two weeks before the model submission deadline. We selected these tasks based on whether they capture distinct phenomena from the public evaluation tasks, such that optimizing only for individual tasks or narrow subsets of linguistic competencies would not be overly rewarded.

Most of the evaluation tasks were zero-shot. Zero-shot evaluation entails comparing the probabilities of different sequences of text. Thus, all submitted models were required to assign a (pseudo) log-likelihood to a sequence of tokens. Additionally, the (Super)GLUE tasks required fine-tuning a classification head appended to the model. Models did not need to *generate* sequences for any evaluation task; thus, both autoregressive and masked language modeling architectures could be used.

4.1 Evaluation Pipeline

We provided code to unify the evaluation setup across submissions. This was released as a public repository on GitHub.[4] The evaluation pipeline supports models implemented in HuggingFace, including Transformer-based architectures, structured state space models (e.g., Mamba; Gu and Dao, 2024), and recurrent neural networks (Peng et al., 2023), among other architectures. Note, however, that we did not restrict the model submissions to HuggingFace-based models; participants were allowed to use their own evaluation setup if desired, so long as they were able to produce predictions

[3] We thank Haau-Sing Li for allowing us to use this preprocessed data.

[4] https://github.com/babylm/evaluation-pipeline-2024

in the expected format.[5] For model and result submissions, users were required to (i) upload a link to their model (on any file-hosting service), and (ii) provide model predictions for each example of each task; we provided a template specifying the format of the predictions file in the evaluation pipeline repository.

Data preprocessing. NLP tasks in our evaluation pipeline often contained vocabulary that is not contained in the BabyLM pretraining corpora. To address this mismatch, we filtered each evaluation task according to its lexical content. We first computed two vocabularies by collecting all words that appear at least twice in the *Strict-Small* corpus and collecting all words that appear at least twice in the *Multimodal* corpus. Then, we took the intersection of these two vocabularies to obtain the final vocabulary. Finally, we iterated through each example in each evaluation task; if an example contained any words that appeared less than twice in the final vocabulary, we filtered the example. Otherwise, each dataset is presented in its original format. See Table 4 in Appendix B for details on the size of the filtered datasets.

4.1.1 Evaluation Paradigms

Zero-shot evaluation. For zero-shot tasks—all of them except (Super)GLUE—we modified the `lm-eval-harness` repository, originally by EleutherAI (Gao et al., 2021). This provides functionality for scoring autoregressive decoder-only LMs and encoder-decoder LMs. For encoder-only LMs, we modified the repository to support masked language model scoring as described in Salazar et al. (2020), and as updated by Kauf and Ivanova (2023).[6] We also modified the pipeline to support multimodal models and tasks.

Finetuning. Prior to the challenge, we experimented with zero-shot learning and few-shot in-context learning for (Super)GLUE. However, this often resulted in random-chance accuracies from our baselines; we therefore employed finetuning. While finetuning technically adds to the training set size, we consider this acceptable, as finetuning on a single GLUE or MSGS task does not meaningfully add to the domain-general linguistic abilities of language models. For tasks requiring finetuning—namely, (Super)GLUE (Wang et al., 2018b, 2019)—we base our scripts on HuggingFace's example finetuning scripts for text classification.[7] We modified the script from last year's pipeline to work with more recent versions of HuggingFace transformers. We provided a default set of hyperparameters that we found to work well across our baseline models, though participants were allowed to modify hyperparameters if they wished. We also provided support for fine-tuning models via low-rank adapters (LoRA; Hu et al., 2022). This enabled the possibility of faster and more compute-efficient model adaptation for our tasks.

4.2 Submission process

Submission format. The submission form was hosted via OpenReview. We required a link to the models, as well as a link to the predictions of these models for all examples for all tasks. The predictions file was formatted as a JSON; each example had an entry with an example ID as its key, and the the prediction of the model as its value. For classification tasks, a prediction was a label ID integer. For zero-shot tasks, predictions were the string that received the highest probability according to the model. The submission process for the competition consisted of three components, which are outlined below:

Paper submission. Each participant submitted a paper detailing their research, methodology, experimental design, and key findings. This was required for all participants, even if they did not submit a model to compete in the challenge.

Artifact submission. In addition to the paper, participants who opted to compete and adhere to the competition rules were required to provide supplementary materials, including model outputs, checkpoints, and pretraining data (unless the default pretraining dataset was used). Participants were also required to upload their predictions for all evaluation tasks.

Submission form. To facilitate comparability and reproducibility, participants were asked to fill in a standardized form that captured model metadata, including hyperparameters, submission de-

[5] Upon release of the evaluation pipeline, we announced that we would provide support as needed to teams training LMs not based in HuggingFace.

[6] We used the implementation of Misra (2022) in the `minicons` library.

[7] https://github.com/huggingface/transformers/blob/211f93aab95d1c683494e61c3cf8ff10e1f5d6b7/examples/pytorch/text-classification/run_glue.py

scriptions, and links to custom data if the standard corpus was not used.

4.3 Baselines

As opposed to last year's baselines, which were selected and trained relatively naively, this year's baselines were based on the architectures of winning submissions from last year's challenge. For the *Strict* and *Strict-Small* tracks, we released the following baselines: LTG-BERT (encoder-only; Samuel et al., 2023) and BabyLlama (decoder-only; Timiryasov and Tastet, 2023a). Although a variant of LTG-BERT (called ELC-BERT) won last year's challenge (Charpentier and Samuel, 2023), Wilcox et al. (2024) showed that similar performance on BabyLM evaluations can be achieved without the additional modifications of ELC-BERT. Thus, we chose LTG-BERT as the baseline, as it is a simpler model. BabyLlama is architecturally similar to Llama (albeit with far fewer parameters), and is additionally trained using knowledge distillation. For the *Multimodal* track, we released vision language models based on GIT (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) architectures, both of which are autoregressive.

Implementation details. For LTG-BERT, we initially used the code provided in the repository linked in Samuel et al. (2023), but we encountered unstable training due to loss spikes with this setup. We therefore used the LTG-BERT model released on HuggingFace, and trained it using the HuggingFace trainer. While training was still relatively unstable compared to other architectures, this procedure yielded performance in the expected range relative to other baselines. For BabyLlama, we use the code from the repository linked in Timiryasov and Tastet (2023a), with small changes for compatibility with this year's BabyLM corpus. For the GIT and Flamingo baselines, we adapt the implementation of Zhuang et al. (2024). Note that these baselines are not necessarily meant to achieve high scores on our evaluation tasks; rather, they are meant to encourage participants to innovate and improve beyond naive applications of existing methods.

5 Competition Results

In this section, we discuss the overall results of the competition (§5.1), track winners (§5.2), and this year's Outstanding Papers (§5.3).

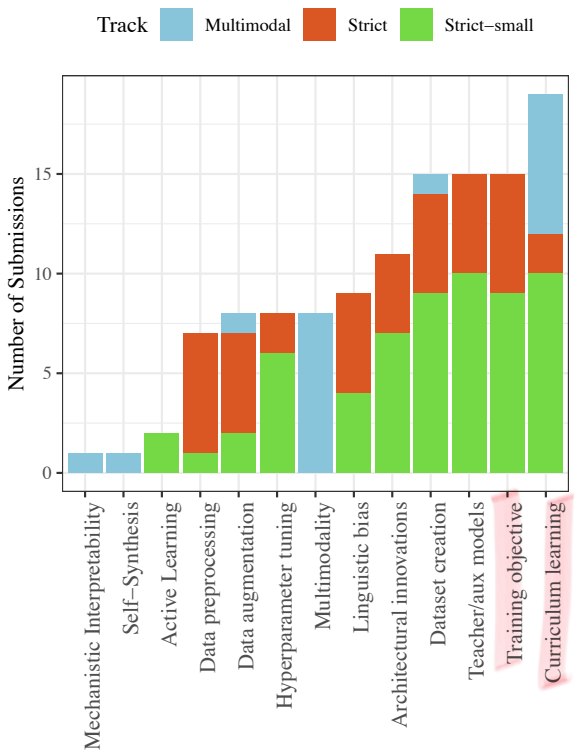

Figure 1: A breakdown of the various approaches used in the 2024 BabyLM challenge, organized by category and track. Curriculum learning again takes the top spot as the most popular approach, followed by training objective innovations.

We received 31 papers and 64 models in total, with two models submitted to the paper track. Table 2 shows the submission counts for each track. Despite efforts to make text–vision pretraining as accessible as possible, only three teams submitted to the *Multimodal* track, for a total of 8 model submissions. As none of these submissions outperformed our baselines, we decided not to award a winner in this track. Despite this disappointment, we hope that our datasets and evaluation resources serve as a basis for further exploration of text-image models in the years to come.

We found that many submissions focused their efforts on similar techniques. To better quantify this, we devised, in Figure 1, a typology of the most common approaches and assigned each submitted model one or more labels. §6.3 provides more detailed descriptions of each approach, as well as results indicating which ones were most effective.

All participants are affiliated with universities or independent research institutions. Participants' home institutions are located in 16 different countries. The number of participants by country is

	# Models	# Participants
Multimodal	8	3
Strict-Small	35	18
Strict	19	11
Total	*64*	*31*

Table 2: Total number of models and participants per track. Participants who submitted to multiple tracks are counted once in the total. Two models were submitted to the *Paper* track only.

as follows (multinational submissions are counted more than once): Germany (8), United States (6), Netherlands (4), Italy (2), UK (2), Canada (1), China (1), Greece (1), Hungary (1), Iran (1), Israel (1), Japan (1), Norway (1), Singapore (1), Sweden (1), Switzerland (1), and Taiwan (1).

5.1 Overall Results & Track Winners

The results from all submissions are shown in Figure 2, with the scores of the top-performing models in each track detailed in Table 3. In the figure, dashed gray lines show the performance of non-competition models (either baselines or skylines), and solid green lines show human performance on evaluation metrics. For GLUE, we use the human scores reported in Nangia and Bowman (2019) and for BLiMP we use the *individual* human agreement scores reported in Warstadt et al. (2020). For Winoground, we plot the human *group* score reported in Thrush et al. (2022), which is slightly more stringent than our model evaluation setup as it requires humans to make the correct judgments over a set of several comparisons. For VQA, we report the *Question + Image* score on *real* images reported in Antol et al. (2015). Again, the human task is arguably more difficult than our own evaluation as it assesses correctness in open-ended responses, rather than by comparing ground-truth captions to distractors. Therefore, the difference between the human and model scores on the vision tasks is likely an underestimate of the true difference between their respective visual capabilities.

We start our discussion by noting several high-level trends, before turning to the winning models. First, as with last year, we notice the same overall pattern of scores between our three different tracks—models in the *Strict* track tend to perform better than those in the *Strict-Small* (although the variance is higher), and models in the *Multimodal* track perform worse. *Ceteris paribus*, more data indeed helps models learn, and learning from multimodal data remains challenging. Within text evaluations, models also perform slightly better on BLiMP compared to GLUE, which is a trend we observed last year as well.

Did model performance improve over last year? At the upper end of the distribution, the answer is *yes*. This year, one model in the *Strict-Small* track beats our Llama skyline on BLiMP, and the best model in the *Strict* track is within just 2.5 percentage points shy of the human score on this task. In addition to these few high-performing models, we also observed a small upward shift in the distribution of model scores compared to last year. For example, last year only 5 models in the *Strict-Small* track achieved a GLUE score of higher than 70; this year that increased to 7 models. For the *Strict* track, this number was 7 last year and 8 this year. One explanation for this small upward shift is that this year we allowed contestants to bring their own data for the *Strict* and *Strict-Small* tracks, provided they stayed within the data limits for each track. Many contestants modified our provided data by procuring new sources, generating data from auxiliary language models, or filtering the existing data. As we shall see in section 6.3, dataset creation was an effective method, and we hypothesize that performance increases on our benchmark tasks over last year can be partially attributed to such data-related improvements.

The introduction of EWoK as our hidden evaluation allowed us to observe that current systems do not learn world knowledge within 100M words. Most submissions perform near chance, at 50% (where dots are colored purple); the maximum score was 58.4%.[8] This observation highlights a potential area for future research. It may be that the current BabyLM corpus—used by many of the submitting teams—simply does not contain the world knowledge that EWoK is designed to test. One other possibility is that existing architectures have a bias towards learning linguistic phenomena more

[8]Many masked language model submissions initially reported EWoK scores around 60–70%. This was likely due to a default behavior of the LM evaluation harness, which assigns a label of 0 when the probability of both sequences is the same. When changing this behavior to instead uniformly sample a label when the sequence probabilities are the same, most models get closer to 50–60% accuracy. We confirmed these scores using a scoring script not based in the LM evaluation harness. This only affected EWoK: we were able to closely reproduce the participant-submitted scores for all other zero-shot tasks, with or without uniform sampling.

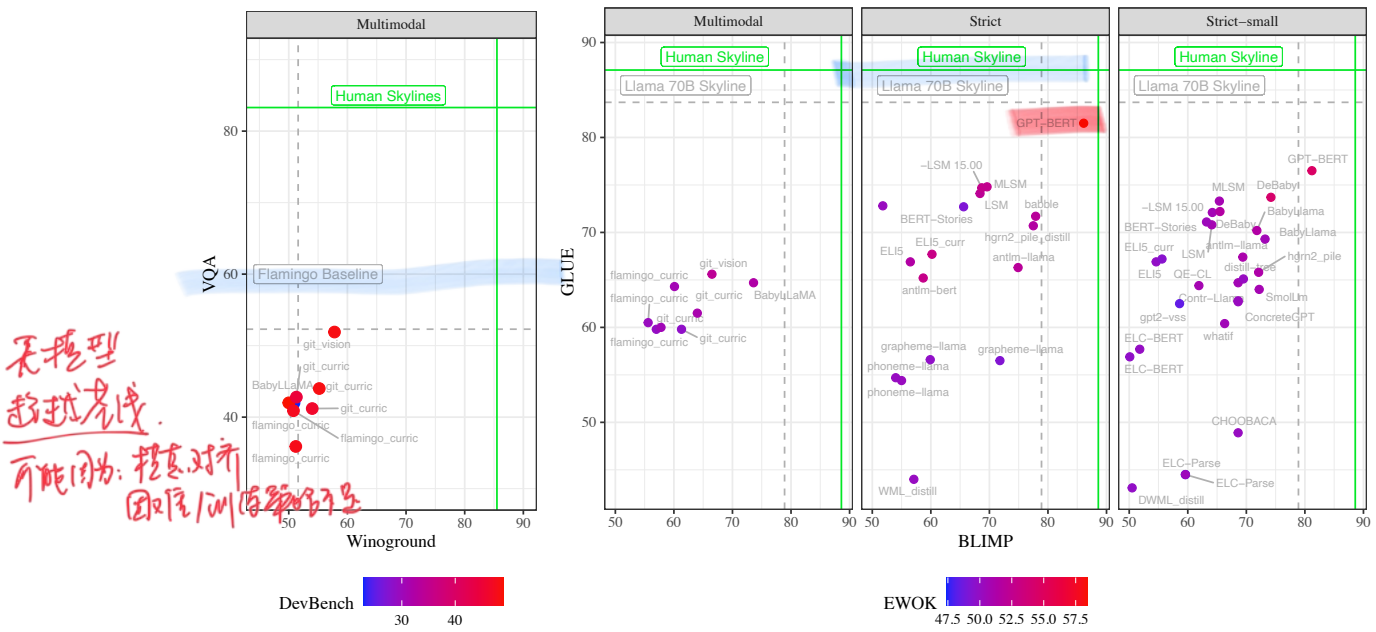

Figure 2: Overall results: At left, multimodal models on multimodal tasks; at right, all models on text tasks. N.B. Human scores for multimodal evals differ somewhat from how we evaluate our models.

easily than relationships between concepts, physical properties, and other topics covered by EWoK. Further work on data (perhaps including data attribution methods) and algorithms will help elucidate why EWoK is so challenging for BabyLM models.

Finally, the *Multimodal* track proved challenging, and no submission beat the baselines we released. We discuss this further in Section 5.2.

5.2 Winning Submissions

***Strict* and *Strict-Small* tracks.** The winner of both the *Strict* and *Strict-Small* tracks is GPT-BERT, submitted by (Charpentier and Samuel, 2024). GPT-BERT merges the causal (CLM) and masked language modeling (MLM) objectives from GPT and BERT, respectively, using the following key insight: by shifting MLM predictions one position to the right, the MLM predictions become aligned with next-token predictions from CLM. The authors use this insight to combine both objectives and seamlessly mix between MLM and CLM.

To train on MLM and CLM simultaneously, the authors duplicate the training data, masking and processing each copy differently for causal and masked language modeling. For each training batch, the authors choose to draw data from the CLM dataset copy with probability p and from the MLM dataset with probability $1 - p$. The authors explore a range of values for p, finding that a 1:7 causal-to-masked ratio tends to give good performance across a variety of tasks. GPT-BERT modifies the LTG-BERT architecture by adding gates on attention heads, as well as the residual connection reweighting proposed in ELC-BERT (Charpentier and Samuel, 2023), the winner of *Strict* and *Strict-Small* from last year.

A different submission to this year's competition, AntLM (Yu et al., 2024), also explored combining CLM and MLM by alternating between the two objectives on a per-epoch basis. The authors found that the best schedule for training LTG-BERT was 6 epochs of CLM, followed by 60 epochs of MLM, followed by 6 more epochs of CLM. While AntLM gets lower scores than GPT-BERT, it performs well overall, also beating our baselines. We conclude that 1) the LTG-BERT architecture remains a strong backbone for small language models, provided one can train it effectively, and 2) combining causal and masked language modeling objectives clearly improves performance over single objective baselines.

***Multimodal* track.** We did not award a winner for the *Multimodal* track this year. We received three submissions, and none outperformed the baselines we released. This speaks to the difficulty of multimodal learning in general. Leveraging both the text and vision modalities is challenging because the model can often learn unimodal shortcuts to

	Model	BLiMP	BLiMP Supplement	(Super)GLUE	EWoK	*Text Average*	VQA	Winoground	DevBench	*Vision Average*
Strict	GPT-BERT	**86.1**	**76.8**	**81.5**	**58.4**	**75.7**	–	–	–	–
	BabbleGPT	77.9	69.5	71.7	52.0	67.8				
	MLSM	69.6	65.4	74.8	52.6	65.6	–	–	–	–
	Best baseline: LTG-BERT	69.2	66.5	68.4	51.9	64.8	–	–	–	–
Strict-small	GPT-BERT	81.2	69.4	76.5	54.6	70.4	–	–	–	–
	DeBaby	74.2	63.7	73.7	54.3	66.5	–	–	–	–
	BabyLlama-2	71.8	63.4	70.2	51.5	64.2	–	–	–	–
	Best baseline: BabyLlama	69.8	59.5	63.3	50.7	61.6	–	–	–	–
Multimodal	GIT-1vd125	66.5	60.9	65.6	52.2	61.3	51.9	**57.8**	48.1	52.6
	Wake/Sleep	73.6	55.6	64.7	51.4	61.3	42.0	50.9	22.8	38.6
	Flamingo$_{CL}$	60.1	53.3	64.3	50.7	57.1	40.9	50.8	47.3	46.3
	Best baseline: Flamingo	70.9	65.0	69.5	52.7	65.2	**52.3**	51.6	**59.5**	**54.5**

Table 3: Macro averages for each benchmark across the top-performing systems (by overall score), best baseline, and skylines.

solve tasks (Dancette et al., 2021), or the information provided by different modalities may not be aggregated properly (Gadzicki et al., 2020). Furthermore, even if there are synergistic effects from multimodal or paired inputs, such as gains in learning sample efficiency, these gains can be ephemeral given more training time (Zhuang et al., 2024).

While this year's *Multimodal* track presents what is essentially a negative result, we hope that our multimodal resources lower the barrier to entry for future research in this area. Effective methods in this space remain an unsolved challenge.

5.3 Outstanding Paper Awards

We presented Outstanding Paper awards to "From Babble to Words: Pre-Training Language Models on Continuous Streams of Phonemes" (Goriely et al., 2024) and "Exploring the effect of variation sets on language model training efficiency" (Haga et al., 2024).

We selected Goriely et al. (2024) for its exploration of phonology, the study of sound or sign patterns in language, to inform tokenization. The authors incorporated phonemes into tokenization by converting raw text into phonemic transcriptions using the `phonemizer` package (Bernard and Titeux, 2021). They carefully ablate character-based, whitespace, and phoneme-aware tokenization schemes, ultimately arriving at a negative result: the standard BPE tokenization algorithm (Sennrich et al., 2016) outperforms other tokenization schemes on BabyLM's text benchmarks. However, as one might expect, phoneme-aware tokenization allows models to perform better at tasks that require phonological knowledge, such as the recognition of plausible pseudowords, or transcriptions of words that are slightly mispronounced.

Haga et al. (2024) tackle the observation from prior work that child-directed speech improves the efficiency of training language models for certain downstream tasks, such as semantic extraction (You et al., 2021) and learning of syntactic structure (Mueller and Linzen, 2023). They hypothesize that the benefits from training on child-directed speech could be due to the existence of variation sets—consecutive rephrasings of the same sentence—which are common in child-directed speech. They construct synthetic variation sets by prompting GPT-4 for paraphrases of sentences selected from CHILDES. Haga et al. find that changing the proportion of synthetic variation sets in the training data can indeed improve the performance of language models on BabyLM's evaluation tasks, although the exact characterization of this relationship remains unclear. We selected Haga et al. (2024) for the novel connections it makes

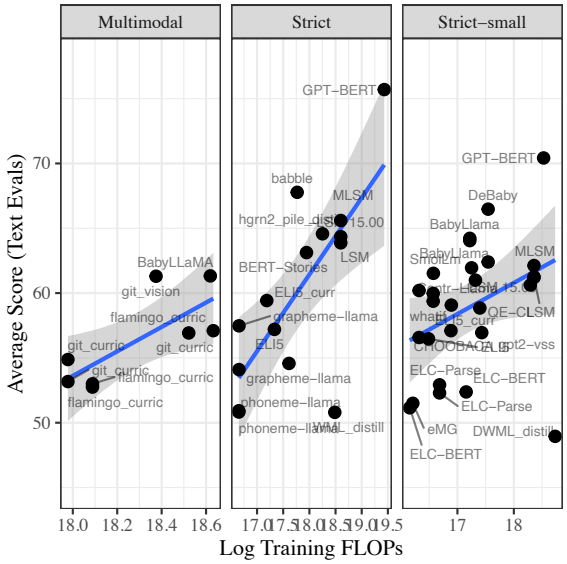

Figure 3: The relationship between training FLOPs and final score.

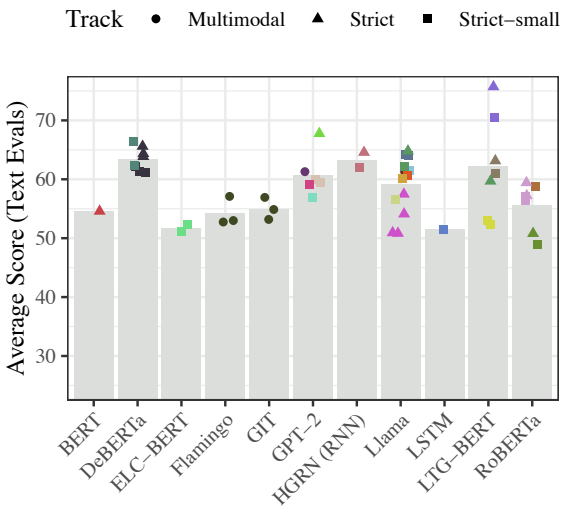

Figure 4: Scores aggregated by backbone architecture. Colors indicate different submissions.

between language modeling and specific theories from cognitive science.

6 Discussion

In this section, we discuss several trends in this year's submissions (§6.1–6.3) and spotlight approaches (§6.4) which we believe point the way towards novel and interesting work in this area.

6.1 Compute Budget

Although we did not collect systematic metadata about last year's models, we observed that our top-performing submissions tended to be more resource-intensive, particularly in the sense that winning models were trained on a large number of epochs. This raised questions about whether their high performance was due to architectural innovations or a large compute budget. We investigate this issue further in Figure 3, by visualizing the relationship between models' performance on our text-only evaluations, and their total training FLOPs. We observe a positive relationship across all three tracks. To test this relationship, statistically, we fit a linear mixed-effects regression model using the `lmer4` package in `R`, with the average score on the text evaluations as our response variable, and log training FLOPs, backbone architecture and track as covariates. We included random slopes corresponding with the model's submission ID number, which indicates the research group that submitted it. We did not include interactions between the fixed effects or random slopes due to convergence issues with the model. Inspecting the fitted model, we find that more training FLOPs leads to better performance ($\beta = 2.7, p < 0.01$), as expected.

6.2 Backbone Architecture

In Figure 4, we visualize the averaged text evaluation score broken down by each submission's backbone architecture. Relative to last year, we received more submissions using Llama. DeBERTa and HGRN (a type of RNN) lead to the highest average scores, while the highest-scoring individual models were all based on LTG-BERT, similar to last year. To test the impact of the backbone model, we inspected the fixed effects associated with model architecture from the linear regression model described above. We found that no level of backbone architecture leads to statistically significant effects for $\alpha = 0.05$, however, we did find large coefficients and smaller p values for several model architectures including DeBERTa ($\beta = 9.1, p = 0.06$), GPT-2 ($\beta = 8.5, p = 0.07$), Llama ($\beta = 7.7, p = 0.07$), and LTG-BERT ($\beta = 8.5, p = 0.06$).

Our interpretation of this result is that there are likely benefits from certain backbone architectures, but that these effects might not be strong enough to be picked up in a statistical analysis of 64 models. Interestingly, recent work has noted that different architectures and training setups often tend to converge to neural representations with similar properties and capabilities (Huh et al., 2024), and we

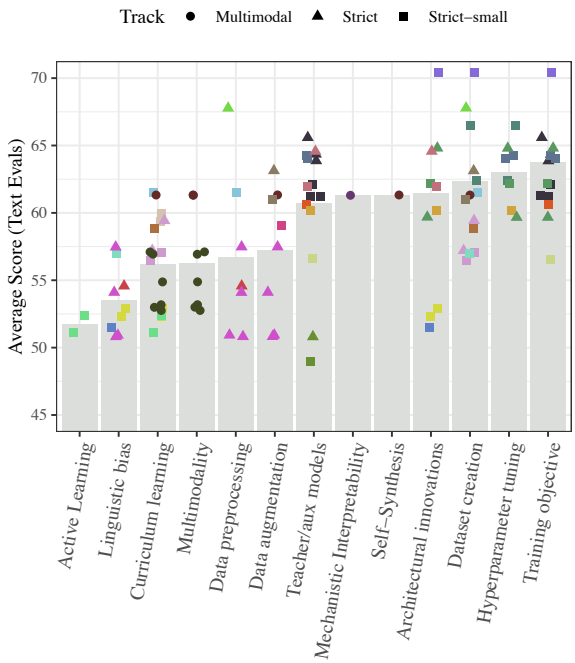

Figure 5: Scores on the BabyLM challenge, aggregated by approach. Colors indicate different submissions, which are plotted twice if they use more than one approach. Axes are zoomed to show variation in the 45-60 range more clearly.

speculate that a similar property might hold for the best models in this year's competition.

Furthermore, different backbone architectures clearly have different variances in average text evaluation score (see Figure 4). This exposes another axis of architecture quality: robustness in training. For example, in this year's competition, DeBERTa (He et al., 2021) had high average scores, compared to other architectures, and low variance between scores in submissions. The winning architecture this year was based on LTG-BERT, but LTG-BERT also had the highest variance among all backbone architectures. This suggests that picking the "best" architecture might involve trading off between architectures that can achieve high scores and architectures that are straightforward to optimize and result in lower variance.

6.3 Common Methods

In Figure 5 we visualize the models based on the approaches they employed. Each participant selected the categories that best fit their model, and categories were largely based on the typology of approaches we designed for analyzing the results of last year's challenge, however, we also let participants write-in approaches that we did not list.[9] Note that models are counted twice if they use more than one approach.

We find that modifications with the training objective, dataset creation, hyperparameter tuning, and architectural innovations lead to the highest average scores, although the latter also leads to a lot of variance across models. As with last year, curriculum learning, while popular, did not lead to high scores, on average. To investigate these trends more rigorously, we fit a mixed effects linear regression model in `lme4`. Our response variable was the average score for text-based evaluations, our covariates were dummy-coded variables indicating the approaches used for each model. We also included random intercepts associated with each submission ID number, corresponding to the research group that created the model. We did not include the interactions between the dummy variables due to convergence issues with the model. We found effects to be significant at $\alpha = 0.05$ for four approaches: training objective innovations ($\beta = 4.5, p < 0.001$), dataset creation ($\beta = 4.8, p < 0.05$), architectural innovations ($\beta = 3.5, p < 0.05$), and linguistic bias ($\beta = -7.3, p < 0.001$). Note that all coefficients are positive except for linguistic bias, meaning that this approach lead to *lower* scores. We also found a negative effect for curriculum learning ($\beta = -3.6, p = 0.055$), although the effect is not significant at the $\alpha = 0.05$ level. That being said, Figure 5 suggests that curriculum learning is not an effective strategy for improving language models, at least in the BabyLM setting.

6.4 Spotlighted Approaches 新光方向

In this section, we highlight trends and new approaches used in this year's submissions.

Recurrent Neural Networks (RNNs) RNNs (Elman, 1990) made their debut in the BabyLM competition this year. The most effective RNN approach used the HGRN architecture (Qin et al., 2023), an RNN that adds complex forget gates on top of the Gated Recurrent Unit (GRU) architecture (Cho et al., 2014). As we noted in §6.2, the backbone architecture, including both RNNs and Transformers, did not have a statistically signifi-

HGRN 架构 表现接近 Transformer

[9] Although some participants wrote "controlled experiments" and "evaluation methods," we removed these from our visualization, as every team that submitted a model technically used these approaches.

cant impact on the models' performance on downstream evaluations, which is to say that the average performances across the best architectures were close. Nevertheless, RNNs and Transformers do have many differences, including their ability to express complex functions and the cost of performing inferences (Merrill et al., 2020; Merrill and Sabharwal, 2024). Because RNNs may be better equipped to model human language at an algorithmic level and may be more compute effective in certain settings, it was a notable finding from this year's challenge that their performance is roughly equivalent to that of many Transformers.

Synthetic Data Several contestants explored using LLMs to create synthetic training data with simple vocabularies and sentences. For example, Haga et al. (2024), used GPT-4 to create variation sets—synthetic data that was inspired by rephrases in child-directed speech. Theodoropoulos et al. (2024) extended the TinyStories approach (Eldan and Li, 2023), sampling a dataset of stories using the vocabulary of a three to four-year-old child by prompting GPT-4.

Corpus Construction Since we allowed contestants to construct their own datasets, many submissions made adjustments to the baseline BabyLM corpus. Common approaches included adding data with simpler sentences and shorter words (Ghanizadeh and Dousti, 2024) or data better suited to certain downstream evaluations (Charpentier and Samuel, 2024). Edman et al. (2024) viewed training corpus construction from the perspective of second language learning, skewing the training data towards sources that explain the rules of a language.

Auxiliary Models Explorations of auxiliary models and knowledge distillation were largely based on the BabyLlama approach introduced in last year's BabyLM challenge (Tastet and Timiryasov, 2024; Yam and Paek, 2024). BabyLlama (Timiryasov and Tastet, 2023b) trains an ensemble of causal language models on a dataset and then distills the ensemble into one final model via knowledge distillation (Hinton et al., 2015). Experiments revealed that BabyLlama's two-step training approach definitively outperforms simply training one causal language model (Tastet and Timiryasov, 2024). Berend (2024) used an extra training phase before pretraining, where the model learned to recover the sparsely encoded latent representation of an auxiliary model.

Tokenization Along with RNNs, a new trend this year was linguistically inspired tokenization (Goriely et al., 2024; Bunzeck et al., 2024). Teams explored how graphemes and phonemes could be incorporated into the language model tokenization pipeline. The primary benefit of adding graphemes and phonemes is to allow language models to perform tasks related to morphology or phonology (how words look and sound): areas where language models previously were limited (Lavechin et al., 2023). Grapheme and phoneme-aware tokenization schemes did not seem to help language models on the base BabyLM evaluation tasks.

Multi-objective training A highly successful approach across several submissions was using multiple objectives during training. GPT-BERT and AntLM, discussed in §5.2, used different methods to combine the masked and causal language modeling objectives, and both were highly successful compared to other submissions.

Training Objective Curricula Finally, a promising variant of curriculum learning this year involved creating curricula over training objectives. Salhan et al. (2024) selectively masked different parts of speech for masked language modeling over the course of training. This approach goes beyond changing the data order, which was the approach used in most curriculum learning submissions we received. We encourage participants for next year's challenge interested in curriculum learning to think beyond data order.

7 Conclusion

The second BabyLM Challenge has demonstrated that significant progress can be made in data-efficient language modeling through community-driven research efforts. With 31 submissions from 17 countries, the challenge revealed several key insights: innovations in model architecture, training objectives, and dataset construction proved particularly effective, with GPT-BERT, a hybrid causal-masked language model architecture, emerging as the strongest approach for the *Strict* and *Strict-Small* tracks. However, the strong correlation between training FLOPs and performance suggests that computational resources remain a crucial factor even in low-data settings.

While this year's challenge added a multimodal track, in an attempt to model grounded language learning environments, no submissions outper-

A Text Only Datasets

CHILDES. The Child Language Data Exchange System (CHILDES; MacWhinney, 2000) is a multilingual database compiling transcriptions from numerous researchers of adult–child interactions in a range of environments, from structured laboratory activities to the home. Huebner and Willits (2021) further process CHILDES, selecting only interactions with American English-speaking children ages 0–6, removing all child utterances, and tokenizing the data. The resulting dataset[10] contains about 5M words.

British National Corpus. The BNC (Consortium, 2007) is a 100M word multi-domain corpus of British English from the second half of the 20th century. We select only the dialogue portion of the corpus, totaling about 10M words.

Children's Book Test. CBT is a compilation of over a hundred children's books from Project Gutenberg by Hill et al. (2016). The dataset was originally released with a set of questions for testing named entity prediction, which we do not include in the pretraining data.

Children's Stories Text Corpus. This dataset consists of manually selected children's stories from Project Gutenberg. It was compiled by Bensaid et al. (2021) for the development of a story generation system.

Project Gutenberg. The Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2020) is a curated and preprocessed selection of over 50k literary books in the public domain from Project Gutenberg totaling over 3B tokens.[11] This distribution comes with extensive metadata that allows us to filter texts by language and date.

OpenSubtitles. This dataset (Lison and Tiedemann, 2016b) is a compilation of publicly available subtitles from TV and movies on a third-party website.[12] We use only the English portion.

Wikipedia. Wikipedia is a volunteer-authored encyclopedia hosted by the Wikimedia Foundation. We use only the English portion.

Simple English Wikipedia. Simple English is classified as a separate language in Wikipedia, thus the texts here are disjoint from those in English Wikipedia. The texts use shorter sentences and high-frequency vocabulary and avoid idioms.

Switchboard Corpus. The Switchboard Corpus (Godfrey et al., 1992) is a collection of transcribed telephone conversations between pairs of strangers. We accessed the text through the Switchboard Dialog Act Corpus (Stolcke et al., 2000).

A.1 Text–Image Datasets

The corpus for the *Multimodal* track consisted of 50M words from the above datasets, as well as 50M more from image-caption datasets. These include the following:

Localized Narratives. Localized Narratives (Pont-Tuset et al., 2020a) is an image-caption dataset. Images are labeled by human annotators; the annotators were asked to describe an image with their voice while hovering their mouse over the region being described. We use the MS-COCO and Open Images subsets.

Conceptual Captions. Conceptual Captions (Sharma et al., 2018b) is an image-capture dataset consisting of automatically scraped and filtered images and captions/annotations from billions of web pages.

[10] https://github.com/phueb/BabyBERTa/blob/master/data/corpora/aochildes.txt
[11] https://gutenberg.org/
[12] http://opensubtitles.org/

B Evaluation Data Details

As described in Section 4.1, we filtered out evaluation examples containing words that did not appear at least twice in both the *Strict-Small* and *Multimodal* pretraining corpora. Here, we present the number of training and test examples for each evaluation task after filtering.

Note that we only control for lexical content: other factors, such as sentence length, syntactic complexity, and overall linguistic style, remain distinct between our corpus and these tasks. In the future, it would be helpful for researchers to focus on designing tasks on which both children *and* language models can be reasonably evaluated.

Note, too, that this filtering step implies that we cannot directly compare results obtained from the BabyLM Challenge to prior evaluations using the full datasets. We also cannot directly compare to results from last year's challenge, though we believe the overlap between the evaluation sets across the BabyLM Challenges is likely high.

Task	Subtask	|Train|	|Test|
BLiMP	–	–	59875
BLiMP Supplement	Hypernym	–	842
	Question-Answer Congruence (easy)	–	64
	Question-Answer Congruence (tricky)	–	165
	Subject-Auxiliary Inversion	–	3867
	Turn-taking	–	280
(Super)GLUE	CoLA	8551	522
	SST-2	67349	436
	MRPC	3668	204
	QQP	363846	20215
	MNLI	392702	4908
	MNLI-mismatched	–	4916
	QNLI	104743	2732
	RTE	2490	139
	BoolQ	9427	1635
	MultiRC	27243	2424
	WSC	554	52
EWoK	Agent Properties	–	2210
	Material Dynamics	–	770
	Material Properties	–	170
	Physical Dynamics	–	120
	Physical Interactions	–	556
	Physical Relations	–	818
	Quantitative Properties	–	314
	Social Interactions	–	294
	Social Properties	–	328
	Social Relations	–	1548
	Spatial Relations	–	490

Task	Subtask	|Train|	|Test|
VQA	–	–	25230
Winoground	–	–	746
DevBench	Visual Vocabulary	–	433
	Test of Receptive Grammar (TROG)	–	79
	THINGS	–	12340

Table 4: Number of training and test examples for each BabyLM evaluation task. We present the number of examples for the text-only tasks (left) and the multimodal tasks (right). We show the number of examples *after* filtering based on the pre-training corpus vocabulary (Section 4.1). Note that only the (Super)GLUE has training examples; the rest of the tasks are zero-shot.

C Subtask Results

Here, we present a more detailed breakdown of results by subtask. Each task has a subsection containing a table where results are described, as well as a textual description containing and overview of the main takeaways for each task.

C.1 BLiMP and BLiMP Supplement

GPT-BERT was the best-performing model on the BLiMP tasks in both the *Strict* and *Strict-Small* tracks. The only subtask where it did not perform best among all models was for Hypernym, where the LTG-BERT baseline was best. BabbleGPT and AntLM were the runners-up in the *Strict* track, whereas DeBaby and BabyLlama-2 were the runners-up in the *Strict-Small* track. In general, submissions to the *Multimodal*

track did not consistently outperform the baseline models; Wake/Sleep outperformed the best baseline (Flamingo) on BLiMP, but no submission outperformed Flamingo on the BLiMP Supplement.

In general, the average BLiMP score across subtasks was effective in distinguishing between high- and low-performing systems: there is high variance across submissions, and those that perform best on BLiMP also tend to perform comparatively well on other tasks.

Similarly to last year, we observe that the HYPERNYM test suite is beyond the ability of language models of this scale. All models (including last year's skylines) perform very close to chance, suggesting either that their preferences are virtually random guessing, or they show systematic biases that essentially cancel out due to counterbalancing in the test data. However, we hesitate to conclude that these models have no knowledge of lexical entailment relations for two reasons: First, these test sentences are somewhat unnatural logical statements that are out-of-domain for the models; and second, there is less reason *a priori* to believe that logically invalid statements have lower probabilities than valid statements.

Among the QUESTION–ANSWER CONGRUENCE test suites, we find that the "tricky" set is still highly discriminative, probably due in part to its adversarial nature. This tells us that most models are easily fooled by locally coherent distractor answers and pay too little attention to cross-sentential long-distance dependency between a *wh*-word and a congruent answer. Only the top-performing models in the *Strict* track score better than chance, and the RoBERTa skyline outperforms all models by a wide margin.

The tests for SUBJECT–AUXILIARY INVERSION are relatively easy: the best models reach near-perfect accuracy, and all models score relatively high compared to other test suites.

Finally, TURN TAKING is highly discriminative, with some models performing at or near chance, while the best model achieves accuracy over 90%.

		BLiMP	BLiMP Supplement					
	Model	Macro average	Macro average	Hypernym	Q–A congruence (easy)	Q–A congruence (tricky)	Subject–aux inversion	Turn taking
Strict	GPT-BERT	**86.1**	**76.8**	48.8	**90.6**	**59.4**	**96.3**	**88.9**
	BabbleGPT	77.8	69.5	47.9	81.2	52.1	81.9	84.3
	AntLM	74.9	66.0	49.3	79.7	43.6	78.3	79.3
	Base baseline: LTG-BERT	69.2	66.5	**55.0**	75.0	53.3	87.5	61.4
Strict-small	GPT-BERT	81.2	69.4	47.1	73.4	54.5	86.3	85.7
	DeBaby	74.2	63.7	53.3	79.7	49.1	84.1	52.1
	BabyLlama-2	73.2	63.1	49.8	59.4	41.2	90.3	75.0
	Best baseline: BabyLlama	69.8	59.5	49.6	54.7	41.2	86.0	66.1
Multimodal	Wake/Sleep	73.6	55.6	49.5	50.0	30.9	85.3	62.1
	GIT-1vd125	66.5	60.9	48.2	57.8	44.2	86.5	67.9
	GIT_{CL}	64.0	51.2	48.9	50.0	20.0	83.7	53.6
	Best baseline: Flamingo	70.9	65.0	48.8	75.0	43.6	86.2	71.4

Table 5: BLiMP Supplement accuracies for each subtask for the top performing systems (by overall score), best baseline, and skylines. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

C.2 GLUE/SuperGLUE

Scores on (Super)GLUE tasks (Table 6) show that GPT-BERT is the best-performing system in both the *Strict* and *Strict-Small* tracks. Notably, its performance in the *Strict-Small* track is better than the runners-up in the *Strict* track, suggesting that this approach is highly data-efficient and/or well-tuned for small-scale language modeling. BabbleGPT and AntLM were again the runners-up for (Super)GLUE in the *Strict* track, and DeBaby was again the runner-up for the *Strict-Small* track. MLSM is now second

runner-up in the *Strict-Small* track. Once again, no submissions outperformed the best baseline (Flamingo) in the *Multimodal* track. This largely confirms findings from the BLiMP and BLiMP Supplement tasks.

	Model	Macro average	CoLA	SST-2	MRPC	QQP	MNLI	MNLI-mm	QNLI	RTE	BoolQ	MultiRC	WSC
Strict	GPT-BERT	**81.5**	**62.4**	**94.0**	**94.4**	**89.1**	**85.2**	**85.3**	**90.8**	**69.1**	**78.4**	**73.3**	**75.0**
	Babble-GPT	71.7	37.8	89.4	83.8	84.0	75.3	76.4	82.9	66.2	63.7	65.1	63.5
	AntLM	66.3	22.2	89.4	84.9	84.2	74.8	74.4	83.2	55.4	65.8	59.9	34.6
	Best baseline: LTG-BERT	68.4	34.6	91.5	83.1	86.7	77.7	78.1	78.2	46.8	61.7	52.6	61.5
Strict-small	GPT-BERT	76.5	48.9	92.2	91.5	87.1	80.2	80.5	86.4	64.0	72.5	69.3	69.2
	DeBaby	73.7	41.8	89.2	91.2	86.6	78.1	77.6	85.5	69.8	71.1	64.2	55.8
	MLSM	73.3	45.2	90.6	82.2	86.6	76.4	77.4	84.7	60.4	69.4	67.6	65.4
	Best baseline: BabyLlama	63.3	2.2	86.2	82.0	83.6	72.4	74.2	82.8	49.6	65.0	60.1	38.5
Multimodal	GIT-1vd125	65.6	30.7	89.7	81.5	83.3	72.7	72.6	78.4	51.8	64.2	54.7	42.3
	Wake/Sleep	64.7	12.2	79.8	78.4	80.5	69.4	70.6	79.8	52.5	63.1	65.8	59.6
	$\text{Flamingo}_{\text{CL}}$	64.3	31.8	88.3	82.4	81.9	70.4	71.4	69.9	46.0	66.5	56.2	42.3
	Best baseline: Flamingo	69.5	36.7	90.4	84.2	85.1	75.8	76.4	83.8	60.4	69.1	60.5	42.3

Table 6: (Super)GLUE results for each subtask for the top performing systems (by overall score), best baseline, and skylines. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

C.3 Multimodal Tasks

Model	Macro average	VQA	Winoground
GIT-1vd125	**54.9**	51.9	**57.8**
GIT_{CL}	49.6	44.0	55.2
Wake/Sleep	46.5	42.0	50.0
Best baseline: GIT	54.8	**54.1**	55.5

Table 7: Results for the public multimodal tasks for the top performing systems (by average score), and the best baseline. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

Model	Macro average	Visual Vocabulary	TROG	THINGS
$\text{Flamingo}_{\text{CC}}$	49.0	66.4	34.2	**46.5**
GIT_{CL}	48.2	73.1	**39.5**	32.1
GIT-1vd125	48.1	**84.9**	35.5	23.8
Best baseline: Flamingo	**59.5**	80.7	38.2	32.6

Table 8: Results for the DevBench tasks for the top performing systems (by average score), and the best baseline. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.