SLlama: A Small Language Model for Extremely Low Resource Domains

Anonymous ACL submission

Abstract

Efficient language modeling is essential for low-resource languages and computationally constrained environments (Warstadt et al., 2023b). We introduce SLlama, a parameter-efficient LLama variant, leveraging RRHP, PWA, SPMLP, and Layer Weight Sharing to reduce model size while maintaining performance. These modifications reduce a model's parameter count from $vh + n11h^2$ to $vh/4 + n_g(3h^2 + 6h)$ where $n_g \leq n$ while preserving linguistic competence. SLlama outperforms the state-of-the-art Baby Llama by 18.88%, specifically having a 31.72% gain in linguistic knowledge acquisition without distillation while retaining 63.3% of its GLUE performance at lower computational cost. Evaluations on GLUE, BLiMP, BLiMP Supplement, and Ewok confirm SLlama's robustness, particularly in syntactic and semantic generalization. Its efficiency in low-resource settings highlights potential for on-device NLP and multilingual modeling, demonstrating that extreme model compression can preserve linguistic capabilities.

1 Introduction

Despite the remarkable progress of language models (LMs), state-of-the-art models such as GPT, Llama, and DeepSeek require substantial computational resources, limiting their deployment on edge devices and in low-resource settings where access to high-performance hardware is constrained. This makes efficient small-scale language modeling an essential research area, particularly for applications requiring on-device inference.

This work *investigates parameter-efficient Transformer models capable of learning language from minimal data, a critical challenge for resource-constrained language modeling*. Inspired by the methodology of Warstadt et al. (2023a), we constrain training data to 10 million tokens, hypothesizing that strategic architectural modifications can enable efficient language acquisition under such conditions. Our central research question is:

How can we train a language-proficient model on a small corpus (a few million tokens) while ensuring the feasibility of resource-constrained edge deployment?

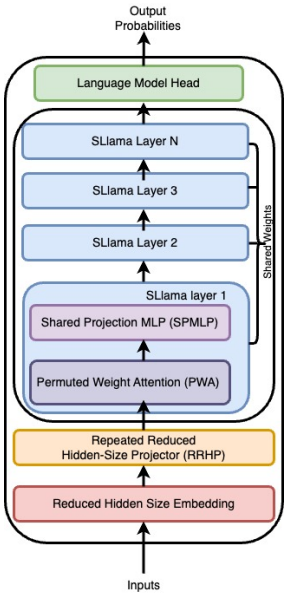

Figure 1: SLlama - Llama Architecture with Reduced Embedding, Repeated Projection, Permuted Weight Attention, Shared Projection MLP and Weight Sharing

Prior research has explored data-efficient training techniques, including approximate attention, mixed-precision training, model/data parallelism, importance sampling, pruning, and quantization (Bai et al., 2024). However, the impact of architectural modifications on small-data LMs remains underexplored. To address this gap, we systematically investigate model size reduction strategies applied to the Llama architecture, holding dataset size and training strategy constant.

Using the BabyLM Challenge dataset (Warstadt et al., 2023a), our proposed SLlama (Small Llama) achieves performance improvements of 18.88%

and 14.32% over baselines trained on 10M and 100M tokens, respectively. While weight sharing, a common compression technique, has a minimal effect on GLUE scores, we find that it adversely affects linguistic knowledge acquisition in small-data models. To address these limitations, we introduce alternative embedding weight reduction schemes, alongside novel attention mechanisms and reassessed layer-sharing techniques.

1.1 Contributions

Our key contributions are:

1. A systematic study of the relationship between hyperparameters (hidden size, number of layers) and their impact on linguistic, conceptual, and world knowledge.
2. The development and implementation of novel weight reduction techniques, designed specifically for small-data Transformer models.
3. Empirical evidence that models trained on just 10M tokens, with carefully optimized architectures, can achieve competitive linguistic proficiency.

To ensure transparency and reproducibility, we release code, trained models, and evaluation scripts on GitHub and Hugging Face.

2 Preliminaries

Architectural Backbone. Transformer models, underpinned by self-attention, have become the backbone of modern natural language processing. Self-attention mechanisms differ across architectures, with decoder-based models gaining prominence due to their autoregressive nature, which makes them well-suited for open-ended text generation tasks (Lu et al., 2024). Among these, Meta's Llama models (Touvron et al., 2023a,b; Grattafiori et al., 2024) have seen widespread adoption across academia and industry, owing to their efficient training pipeline, optimized Transformer block implementations, state-of-the-art performance, and broad availability.

Architectural advancements in Transformer models typically focus on:

- Attention mechanisms, including efficiency optimizations and memory reduction techniques (Zhang et al., 2024a; Kitaev et al., 2020; Ainslie et al., 2023).
- Positional encoding, crucial for representing token order in sequence modeling (Su et al., 2023).
- Feed Forward Network (FFN) implementations, which affect model capacity and efficiency (Liu et al., 2021).
- Normalization strategies, such as RMSNorm and layer normalization, which improve stability and convergence (Grattafiori et al., 2024; Radford et al., 2019).

While Llama shares many similarities with other decoder-based Transformers, it stands out due to its fine-tuned architectural refinements, high-quality data curation, and superior pretraining pipeline. Given these advantages, we selected the Llama architecture as the foundation for our model, ensuring comparability with existing research. For our experiments, we use Hugging Face's implementation of Llama, retaining its default configurations except for three key hyperparameters: (i) Hidden size (h) (ii) Intermediate layer size (iii) Number of layers (n). Following the recommendations of Tang et al. (2024), we tie the embedding layer and language model head, a widely used strategy to improve parameter efficiency in small-scale language models.

Small Data Training. Our experiments utilized the BabyLM challenge dataset Choshen et al. (2024), with a complete data description available in Warstadt et al. (2023a). After initial hyperparameter search, all pretraining employed cosine learning rate decay with minimum and maximum rates of 4×10^{-5} and 4×10^{-4}, respectively. We set the gradient accumulation to 2, batch size to 128, and sequence length to 256. Training runs were conducted for 3,000 iterations based on the observation that optimal evaluation loss was typically achieved by the 1,500th iteration (approximately the 10th epoch) and a slight improvement at the 2,500th iteration.

The Baby Llama model (Timiryasov and Tastet, 2023), which was among the leading solutions in the original BabyLM challenge and serves as the state-of-the-art baseline for the second BabyLM challenge[1], was trained using knowledge distillation from two larger teacher models (Llama and GPT2), with the student model reportedly outperforming the teachers. To isolate the effect of distil-

[1] https://github.com/babylm/evaluation-pipeline-2024?tab=readme-ov-file

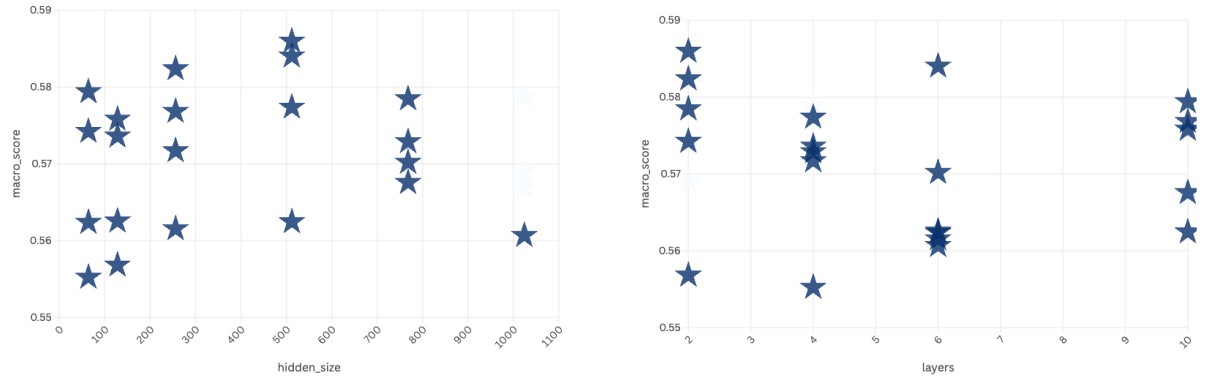

Figure 2: Correlation of hidden size and number of layers to macro score (average of a model's BLiMP, BLiMP Supplement, GLUE, and Ewok scores) with Spearman correlations of 0.88 and 0.38, respectively.

Reduction Technique	Model Size (M)	BLiMP Sup. (%)	Ewok (%)	GLUE (%)	Avg. (%)
Baby Llama	58.0000	69.80 (59.50)	50.70	63.30	60.80
EWT	2.3683	56.00 (50.47)	57.75	63.45	56.92
LHRP	2.8814	60.47 (49.22)	57.58	63.26	57.63
AHRP	2.8820	59.02 (52.80)	56.58	62.41	57.70
RRPH	**2.8803**	91.94 (77.61)	57.91	63.57	72.76
Llama Untied	4.4163	91.94 (77.61)	57.83	63.94	**72.83**

Table 1: Measuring the impact of parameter reduction of the embedding layer relative to Baby Llama (Timiryasov and Tastet, 2023) and Embedding Weight Tying (EWT). The weights of the embedding layer and the language model head were not tied for LHRP, AHRP, RRPH and Llama Untied.

lation, we conducted initial experiments to characterize the inherent capabilities of the Llama architecture and to establish the relationship between its key configuration parameters (hidden size, intermediate size, and number of layers) and performance on the aforementioned evaluation tasks. Starting with a hidden size of 64 (to minimize resource consumption), we varied the number of layers from 2 to 12. We observed that the macro-average scores for models with six and eight layers were similar, as were those for models with ten and twelve layers. Based on this, we focused subsequent experiments on layer counts of 2, 4, 6, and 10 while logarithmically increasing the hidden size from 64 to 1,024. The model with a hidden size of 512 and 2 layers achieved the best average macro score.

Evaluation and Analysis Evaluation was performed using the pipeline provided by Choshen et al. (2024); Gao et al. (2023), encompassing four tasks: BLiMP, BLiMP supplement (Warstadt et al., 2023c), GLUE (Wang et al., 2019), and Ewok (Ivanova et al., 2024). These tasks assess linguistic competence (BLiMP), conceptual understanding (GLUE), and general world knowledge (Ewok).

Further analysis, presented in Figure 2 , explored the correlation between model size parameters (hidden size and number of layers) and the model's performance across the different evaluation dimensions (linguistic competence, world knowledge, and conceptual understanding). While statistical significance was generally weak, several trends emerged: 1) a weak but consistent positive correlation between hidden size and BLiMP score (linguistic knowledge); 2) an inconsistent positive relationship between hidden size and GLUE score; 3) a strong and consistent negative correlation between hidden size and world knowledge; 4) an inconsistent positive trend between the number of layers and linguistic competence; 5) a weak positive trend between the number of layers and conceptual understanding; and 6) a noticeable weak negative trend between the number of layers and linguistic competence. These observations suggest the need to carefully balance horizontal (hidden size) and vertical (number of layers) scaling, particularly with limited data. However, the positive impact of increasing layer count for smaller hidden sizes was evident, supporting previous findings (Liu et al., 2024). Based on these preliminary experiments, we observed that a hidden size of 64 and 6 lay-

ers serves as a suitable configuration for exploring the impact of architectural modifications in subsequent experiments, minimizing computational cost, memory usage, and experimental time.

3 Model Reduction

Having established a more computationally efficient baseline compared to Baby Llama, we proceeded with systematic model size reduction. While architectural innovations in Transformer models often target complexity reduction, very few emphasize decreasing parameter count (Liu et al., 2024). Consequently, research has focused on minimizing the memory footprint of these models by reducing parameters within the embedding layer, language model head, and MLP units (Tang et al., 2024; Liu et al., 2024; Zhang et al., 2024b). Although vocabulary size (v) reduction is a common practice (Tang et al., 2024), we chose to maintain the vocabulary size in the Hugging Face Llama3 implementation (Grattafiori et al., 2024). We argue that while reducing the vocabulary size offers immediate gains through a smaller prediction space, it may harm the representation of out-of-vocabulary (OOV) words due to increased sub-word tokenization. Therefore, our investigation of parameter reduction schemes, detailed below, focuses on the embedding layer, Feed Forward Network, and the self-attention blocks of a Transformer model.

3.1 Embedding Parameter Reduction

Embedding Weight Tying (EWT) is a widely used technique for reducing language model size by sharing the weights of the embedding layer with those of the language model head (Liu et al., 2020). This reduces the model's parameter count by vh, where h is the hidden size and v is the vocabulary size. Mnih and Teh (2012) hypothesized that rows corresponding to semantically similar words should exhibit near-identical representations—such that the input embedding encodes synonyms in a comparable manner, while the output embedding assigns similar score distributions to interchangeable words. Expanding on this, Press and Wolf (2017) empirically demonstrated that tying input and output embeddings produces a joint representation more closely aligned with the output embedding of an untied model, leading to improved perplexity both with and without dropout. However, their findings also suggest that untied embeddings evolve into distinct representations.

Our study extends this distinction to linguistic knowledge acquisition, revealing that embedding sharing adversely affects a model's linguistic competence. Specifically, in an untied model, the output embedding retains less fundamental linguistic knowledge, whereas the input embedding preserves richer linguistic representations, as shown in Table 1. These findings highlight the necessity of maintaining layer-specific representational nuances when reducing model size. To address this, we propose alternative parameter-reduction strategies that optimize efficiency while preserving the linguistic integrity of intermediate representations.

Inspired by the Mixed Dimension Embeddings (MDE) approach proposed by Pansare et al. (2022) and Ginart et al. (2021), we explored reducing the dimensionality of the embedding layer. Specifically, we reduced the hidden size (h) of the embedding layer by a factor of four (h_r) . Given that the hidden layers of the decoder are initialized with h, a projection scheme is required to map the reduced embedding dimension to the original hidden size h. We investigated three such projection methods: Linear Hidden-Size Reduction and Projection (LHRP), Attention Hidden-Size Reduction and Projection (AHRP), and Repeated Reduced Hidden-Size and Projection (RRHP). LHRP employs a linear layer as described in Equation 1, effectively reducing the parameters from vh to vh_r. This method technically projects the embedding vector into a larger dimensional space, effectively assuming the relationship between the small and large representations is linear.

$$\text{Linear}(x, A) = xA^T + b \quad (1)$$

where:

$x \in \mathbb{R}^{m \times h_r}$

$A \in \mathbb{R}^{h_r \times h}$

AHRP leverages the conventional attention mechanism described in Equation 2. AHRP utilises $vh_r + 2h_r + h^2/r$ parameters instead of vh. Conceptually, AHRP magnifies the cogent dimensions of the smaller representations. Finally, RRHP initializes the embedding layer with the reduced hidden size h_r and repeats the resulting representation r times before feeding it to the decoder layers, effectively repeating the information encoded in the smaller representation r times. This method reduces the parameter count by $3vh_r$.

Following the training configurations described

previously, we trained and evaluated models incorporating these reduction schemes. The performance of each technique is presented in Table 1. The unexpected performance increase observed with RRHP led us to investigate the lower scores obtained with the other methods. Through the Llama Untied model, we discovered that weight-tying was the primary cause. The significant performance improvement observed specifically on the BLiMP task suggests that architectural choice is paramount to performance, with weight tying adversely affecting the linguistic knowledge encoded within the embedding layer. While we acknowledge the limitations of generalizing from our specific experimental setup, our findings support the observation by Eldan and Li (2023) that deeper layers are primarily responsible for conceptual understanding, as reflected in the relatively stable GLUE scores. Based on our findings, we recommend RRHP as a preferred parameter reduction technique over embedding weight tying for small language models trained on limited data. However, we emphasize that the optimal choice of parameter reduction technique ultimately depends on the specific application requirements.

$$\text{Attn\_weight}(Q, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2)$$

$$\text{Attn}(Q, K, V) = \text{Attn\_weight}(Q, K)V \quad (3)$$

where:

$Q \in \mathbb{R}^{h_r \times h_r}$

$K \in \mathbb{R}^{h_r \times h_r}$

$V \in \mathbb{R}^{h_r \times h}$

3.2 Self-Attention Parameter Reduction

While the multi-head attention mechanism has been instrumental in the success of language models, its computational and memory demands remain a concern. Consequently, optimized attention implementations with reduced complexity have been proposed (Zhang et al., 2024a; Kitaev et al., 2020), often demonstrating comparable performance to standard multi-head attention (MHA). Although the inference-time memory consumption associated with the key–value (KV) cache is a compelling challenge, this work focuses on reducing the parameter count required for self-attention within small language models, thereby effectively reducing the memory demand of a model both at training and inference time especially in resource constrained environments. Drawing inspiration from the embedding parameter reduction strategies discussed previously, we introduce three novel attention mechanisms aimed at reducing parameter count: Shared Key Query Attention (SKQA), Repeat-Reduced-Attention (RRA), and Permutated Weight Attention (PWA).

The design of SKQA stems from the interpretation of the attention mechanism as a similarity selection process, which is particularly relevant in language modeling. The attention weights are computed according to Equation (2), and the attention output is derived using Equation (3). Equation (2) can be viewed as computing a probability distribution of inter-token similarity when K and Q are equivalent. We investigated the feasibility of this similarity-based attention by equating the weights of K and Q; effectively reducing parameter count by h^2.

RRA, in contrast, was inspired by the Repeated Reduced Hidden-Size and Projection reduction technique described earlier, that is, $Q, K, V \in \mathbb{R}^{h \times h_r}$ and are subsequently repeated. Finally, PWA was motivated by the embedding layer reduction strategy presented by Li et al. (2017); Algorithm 1 illustrates its implementation. PWA effectively reduces memory demand from $4h^2$ to $6h$. The average absolute difference in perplexity from the baseline MHA implementation is 0.077645 (a negligble value), indicating similar convergence behaviour but with a reduced parameter count. The performance of each intra-layer reduction technique on downstream tasks is presented in Table 2. PWA demonstrates the best balance between model size and overall performance, closely followed by SKQA, as shown in Table 2. Relative to SKQA, PWA reduces parameter count by a larger factor. While RRA achieved convergence and maintained competitive GLUE and Ewok scores, the model's linguistic competence suffered.

3.3 MLP Block Parameter Reduction

The Multi-Layer Perceptron (MLP) or Feed-Forward Network (FFN) within a Transformer architecture constitutes a significant portion of the model's parameters. Typically, the MLP consists of two fully connected layers: an expansion layer that increases the dimensionality of the input from hidden size h to intermediate size nh ($3h$ in our case), and a projection layer that reduces the dimensionality back to h. This results in a substantial number

Reduction Technique	Model Size (M)	BLiMP Sup. (%)	Ewok (%)	GLUE (%)	Avg. (%)
MHA (Baseline)	4.42	91.94 (77.61)	57.71	63.72	72.75
MHA^{R}	2.88	91.94 (77.61)	58.08	63.64	**72.82**
PWA	4.32	91.94 (77.61)	57.52	63.47	72.64
PWA^{R}	**2.78**	91.94 (77.61)	57.76	63.02	72.58
RRA	4.34	59.20 (52.51)	57.88	63.33	58.23
RRA^{R}	2.81	62.28 (51.65)	57.87	62.83	58.66
SKQA	4.42	91.94 (77.61)	58.25	63.18	72.75
SKQA^{R}	2.88	91.94 (77.61)	57.71	63.75	72.75

Table 2: Performance of different attention parameter reduction methods. Model^{R} variants utilize Repeat Embedding Reduction. MHA is equivalent to Llama Untied in Table 1

Algorithm 1 Permutated Weight Attention

Require: $h, n, m > 0$
Ensure: $\texttt{permutation}(n, m) > 3h$
  $\texttt{permutes} \leftarrow$ list of $\texttt{permutation}(n, m)$
  $\theta \leftarrow \text{Embedding}(n, h)$
  $\texttt{q\_idx} \leftarrow \texttt{permutes[0:h]}$
  $\texttt{k\_idx} \leftarrow \texttt{permutes[h:2h]}$
  $\texttt{v\_idx} \leftarrow \texttt{permutes[2h:3h]}$
  $Q = \text{Linear}(x, \theta[q\_idx])$
  $K = \text{Linear}(x, \theta[k\_idx])$
  $V = \text{Linear}(x, \theta[v\_idx])$
  $\texttt{attn} = \text{Attn}(Q, K, V)$

of parameters: $6h^2$ for both expansion and downward projection layers. Furthermore, the Llama architecture incorporates a gate projection layer, introducing an additional h^2 parameters totaling $7h^2$. Given this parameter count, the MLP block in transformer models is a prime candidate of overparametrization, making it a key target for parameter reduction strategies. To reduce this parameter overhead, we introduce a novel modification to the MLP block: Shared Projection MLP (SPMLP). In SPMLP, we share the weights between the expansion and projection layers, effectively reducing the parameter count by $3h^2$ parameters. This weight sharing strategy not only reduces the model's memory footprint but also encourages a more symmetrical and potentially more efficient information flow within the MLP block while having a direct implementation. Although we observe a minor decline in overall performance, the balance between model efficiency and parameter reduction remains compelling. We show the impact of SPMLP on model performance in Table 3.

3.4 Inter-Layer Weight Reduction Strategies

To further reduce model size, we explored two common inter-layer weight reduction techniques: layer reuse and weight sharing. Layer reuse (Liu et al., 2024) passes the hidden state through a layer multiple times (in our case, twice). Thus, if layer reuse $r = 2$, the model is intialized with n/r layers where n is the number of layers, effectively reducing model size by $11nh^2/2$ parameters provided no reduction scheme was introduced. On the other hand, Weight sharing (Lan et al., 2020) ties the weights of multiple layers, significantly reducing the number of parameters to $11n_gh^2$ where n_g is the number of groups the layers are divided into . We implemented both techniques, sharing weights across all layers in the model for the weight-sharing approach. Table 3 presents the performance of models employing these reduction strategies. While the macro-average scores across the three models show minimal variation, the substantial parameter reduction achieved through weight-sharing presents a compelling trade-off. However, the observed performance decline suggests a potential loss of fine-grained information, warranting further investigation. Notably, while weight sharing in conjunction with SPMLP results in a slight performance degradation, the substantial reduction in model size justifies its consideration, particularly within the context of this study, which prioritizes memory efficiency.

4 SLlama Architecture and Discussions

Based on our experimental findings, we introduce SLlama (Small Llama), a parameter-efficient variant of Llama incorporating Repeated Reduced Hidden Size and Projection (RRHP), Permutated Weight Attention (PWA), Shared Projection Multi-Layer Perceptron (SPMLP), and Layer Weight

Reduction Technique	Model Size (M)	BLiMP Sup. (%)	Ewok (%)	GLUE (%)	Avg. (%)
Reuse	2.67	91.94 (77.61)	57.84	63.83	**72.81**
Reuse^{S}	2.67	91.94 (77.61)	57.63	62.40	72.41
Share	2.63	91.94 (77.61)	57.76	63.14	72.62
Share^{S}	**2.61**	91.94 (77.61)	57.22	62.33	72.28

Table 3: Impact of inter-layer and SPMLP weight reductions techniques. Technique^{S} utilizes Shared Projection MLP (SPMLP) .

Model	Hidden Size	Macro Score (%)
Llama	128	72.45
SLlama	128	71.62
Llama	192	72.01
SLlama	192	71.85
Llama	256	72.34
SLlama	256	71.31

Table 4: Scaling Llama and SLlama Models by increasing the hidden size while maintaining the number of layers at 6. SLlama mirroring the nuances of Llama Architecture.

Sharing across key components. Compared to Baby Llama (Timiryasov and Tastet, 2023), SLlama has around 20× fewer parameters and improves linguistic knowledge acquisition by 31.72% without any knowledge distillation while maintaining a comparable GLUE score with significantly fewer resources.

SLlama's strong performance on linguistic tasks with minimal data highlights its potential for highly resource-constrained language modeling. Unlike existing compression methods, SLlama demonstrates superior robustness in rigorous linguistic evaluations, making it a viable candidate for efficient language modeling. As shown in Table 4, our reduction strategies preserve key performance characteristics of larger models while significantly lowering computational costs. These findings reinforce SLlama's promise as a resource-efficient yet high-performing model for NLP applications.

4.1 Language Model Evaluation Metrics

Our findings align with prior research showing that embedding weight tying does not significantly affect GLUE scores. However, we reveal a critical limitation: while conceptual understanding remains stable, weight tying severely impairs linguistic competence, particularly in low-resource settings. This highlights the need for evaluation metrics that capture both fundamental and advanced language skills, as standard benchmarks may overlook linguistic and cultural knowledge essential for real-world applications.

Given this impact, we support calls for more comprehensive evaluation frameworks that assess language structure, semantic generalization, and cultural representation (Tao et al., 2024; Bhatt and Diaz, 2024). Future NLP evaluations should incorporate metrics for cultural acquisition, conceptual transfer, and linguistic diversity, ensuring that compression techniques do not compromise essential language understanding. As research advances in small-scale, resource-efficient NLP, it is crucial to develop evaluation methodologies that balance efficiency with linguistic and cultural fidelity.

4.2 Parameter Budgeting

Repeated Reduced Hidden Size and Projection (RRHP) and Permutated Weight Attention (PWA) preserve linguistic knowledge and conceptual understanding despite aggressive reductions in embedding and attention parameters, challenging conventional assumptions about scaling laws in Transformer models. This finding underscores the importance of efficient parametrization over sheer model size, suggesting that many current models may be over-parametrized. Given the central role of self-attention in Transformers, our results indicate that extreme parameter reduction within attention mechanisms does not necessarily degrade performance, provided architectural adaptations are implemented to maintain expressivity.

5 Related Work

The pursuit of powerful yet efficient language models has driven significant research. While scaling models through increased data and parameters has yielded impressive results, e.g., PALM (Chowdhery et al., 2022) and GPT-3 (Brown et al., 2020), the associated computational costs are prohibitive for many applications. This has spurred research on data-efficient training methods, architectural innovations, and model compression techniques.

Data-Efficient Language Models. Research on data efficiency has explored dataset reduction via k-means clustering (Kaddour, 2023), deduplication (Lee et al., 2022), and selective high-quality data curation (Gunasekar et al., 2023; Mueller and Linzen, 2023; Eldan and Li, 2023; Huebner et al., 2021). These studies highlight the importance of data diversity in model performance (Lu et al., 2024; Mekala et al., 2024). Aligning with this work, we train SLlama on a constrained 10M-token dataset, inspired by the BabyLM challenge (Warstadt et al., 2023b,a; Choshen et al., 2024), to advance efficient language modeling with limited data.

Model Compression Techniques and Small Language Model Design. Prior work has tackled the memory demands of large embedding tables in recommender systems using techniques like ROBE (Desai et al., 2022), MEmCom (Pansare et al., 2022), Mixed Dimension Embeddings (Ginart et al., 2021), and Slim Embeddings (Li et al., 2017). Beyond embeddings, inter-layer weight sharing and factorized embedding parameterization (Lan et al., 2020) have reduced model size in BERT (Devlin et al., 2019). Building on these efforts, we propose novel embedding weight reduction schemes and alternative attention mechanisms to minimize model size while preserving linguistic capabilities.

While model compression reduces memory footprint, small model design optimizes architectures for edge deployment. The rise of large models such as GPT-3 (Brown et al., 2020) has fueled interest in efficient alternatives such as OPT (Zhang et al., 2022), Phi (Gunasekar et al., 2023), and PanGu-π (Tang et al., 2024), which achieve strong performance with fewer parameters and innovations that challenge the assumption that architecture has minimal impact given a fixed resource budget (Kaplan et al., 2020). Our work extends this research by introducing SLlama, a parameter-efficient architecture optimized for high-quality language modeling on limited data.

Weight Sharing and Efficient Attention Mechanisms. Weight sharing is a common compression technique (Tang et al., 2024; Lan et al., 2020; Ainslie et al., 2023), but its effectiveness varies across model components. While prior work (Liu et al., 2020) suggests normalizing embedding weights to mitigate degradation, our study systematically assesses its impact. We find that sharing weights between key and query modules in self-attention preserves performance while reducing parameters. However, sharing input and output embeddings degrades linguistic competence, highlighting the need for selective weight-sharing strategies to maintain representation quality and expressiveness.

Recent efforts to optimize multi-head attention have focused on reducing computational complexity and memory consumption, particularly by refining the KV cache (Zhang et al., 2024a; Kitaev et al., 2020). While techniques like GQAm (Ainslie et al., 2023) enhance inference efficiency, they primarily target runtime performance rather than structural efficiency. In contrast, our work aims to explicitly minimize the parameter count within the attention mechanism, reducing the overhead required for computing attention weights and outputs while maintaining model effectiveness.

6 Conclusion

This study demonstrates the feasibility of training effective language models with limited data and resources. By leveraging architectural innovations such as RRHP, PWA, SPMLP, and Layer Weight Sharing, we enhance the linguistic capabilities of small models trained on just 10M tokens. Our findings show that careful design can mitigate performance degradation, enabling compact yet powerful models. This work advances accessible AI by supporting deployment on personal devices and improving resource-constrained language modeling. We anticipate ultra-compact models pushing PWA to its limits, redefining trade-offs between parameter count, computational cost, and capability.

Future research should explore whether adaptive architectures can dynamically allocate resources rather than statically distributing parameters across layers. This introduces parameter budgeting as a complementary paradigm to FLOP-based efficiency metrics, offering a more nuanced framework for scaling in resource-constrained NLP applications. A deeper understanding of parameter efficiency could enable models to achieve state-of-the-art performance with significantly reduced computational footprints, fostering adaptive architectures that allocate resources based on task complexity. This shift from static, over-parametrized models to dynamically efficient architectures has profound implications for low-resource language modeling, edge deployment, and sustainable AI development.

Limitations

While this study demonstrates promising results, several limitations must be considered. Our findings are primarily based on the LLaMA architecture, and while certain trends may generalize, further research is needed to assess the applicability of our techniques across diverse model architectures. Additionally, the BabyLM dataset, while useful for studying small-data training, lacks linguistic diversity, limiting the evaluation of our models to English. Future work should explore performance on more diverse datasets, including low-resource languages, and assess the models' ability to acquire commonsense and factual knowledge.

Moreover, real-world deployment challenges remain, particularly regarding performance on edge devices, where quantization-related degradation has yet to be fully examined. The scalability of our compression techniques to larger models and datasets also requires further investigation. Ultimately, striking an optimal balance between model efficiency and linguistic richness is an ongoing challenge, and future research should focus on refining model reduction strategies to ensure robust language representation while maintaining computational efficiency.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, Xinyuan Song, Carl Yang, Yue Cheng, and Liang Zhao. 2024. Beyond efficiency: A systematic survey of resource-efficient large language models. *Preprint*, arXiv:2401.00625.

Shaily Bhatt and Fernando Diaz. 2024. Extrinsic evaluation of cultural competence in large language models. *Preprint*, arXiv:2406.11565.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Aditya Desai, Li Chou, and Anshumali Shrivastava. 2022. Random offset block embedding array (robe) for criteotb benchmark mlperf dlrm model : 1000× compression and 3.1× faster inference. *Preprint*, arXiv:2108.02191.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

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

A.A. Ginart, Maxim Naumov, Dheevatsa Mudigere, Jiyan Yang, and James Zou. 2021. Mixed dimension embeddings with application to memory-efficient recommendation systems. In *2021 IEEE International Symposium on Information Theory (ISIT)*, page 2786–2791. IEEE Press.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov,

Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

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

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, 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. *Preprint*, arXiv:2405.09605.

Jean Kaddour. 2023. The minipile challenge for data-efficient language models. *Preprint*, arXiv:2304.08442.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *Preprint*, arXiv:2001.04451.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *Preprint*, arXiv:1909.11942.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. *Preprint*, arXiv:2107.06499.

Zhongliang Li, Raymond Kulhanek, Shaojun Wang, Yunxin Zhao, and Shuang Wu. 2017. Slim embedding layers for recurrent neural language models. *Preprint*, arXiv:1711.09873.

Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. 2021. Pay attention to mlps. *Preprint*, arXiv:2105.08050.

Jinyang Liu, Yujia Zhai, and Zizhong Chen. 2020. Normalization of input-output shared embeddings in text generation models. *Preprint*, arXiv:2001.07885.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *Preprint*, arXiv:2402.14905.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *Preprint*, arXiv:2409.15790.

Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. 2024. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *Preprint*, arXiv:2402.10430.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. *Preprint*, arXiv:1206.6426.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. *Preprint*, arXiv:2305.19905.

Niketan Pansare, Jay Katukuri, Aditya Arora, Frank Cipollone, Riyaaz Shaik, Noyan Tokgozoglu, and Chandru Venkataraman. 2022. Learning compressed embeddings for on-device inference. *Preprint*, arXiv:2203.10135.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. *Preprint*, arXiv:1608.05859.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Yehui Tang, Fangcheng Liu, Yunsheng Ni, Yuchuan Tian, Zheyuan Bai, Yi-Qi Hu, Sichao Liu, Shangling Jui, Kai Han, and Yunhe Wang. 2024. Rethinking optimization and architecture for tiny language models. *Preprint*, arXiv:2402.02791.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS Nexus*, 3(9):pgae346.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023b. 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. 2023c. Blimp: The benchmark of linguistic minimal pairs for english. *Preprint*, arXiv:1912.00582.

Jiale Zhang, Yulun Zhang, Jinjin Gu, Jiahua Dong, Linghe Kong, and Xiaokang Yang. 2024a. Xformer: Hybrid x-shaped transformer for image denoising. *Preprint*, arXiv:2303.06440.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.