MobiLlama: Towards Accurate and Lightweight Fully Transparent GPT

Omkar Thawakar[1*], **Ashmal Vayani**[1*], **Salman Khan**[1,2], **Hisham Cholakal**[1], **Rao M. Anwer**[1,3], **Michael Felsberg**[5], **Tim Baldwin**[1,4], **Eric P. Xing**[1], **Fahad Shahbaz Khan**[1,5]

[1]Mohamed bin Zayed University of AI, [2]Australian National University, [3]Aalto University
[4]The University of Melbourne, [5]Linköping University

Abstract

'*Bigger the better*' has been the predominant trend in recent Large Language Models (LLMs) development. However, LLMs do not suit well for scenarios that require on-device processing, energy efficiency, low memory footprint, and response efficiency. These requisites are crucial for privacy, security, and sustainable deployment. This paper explores the '*less is more*' paradigm by addressing the challenge of designing accurate yet efficient Small Language Models (SLMs) for resource constrained devices. Our primary contribution is the introduction of an accurate and fully transparent open-source 0.5 billion (0.5B) parameter SLM, named *MobiLlama*, catering to the specific needs of resource-constrained computing with an emphasis on enhanced performance with reduced resource demands. *MobiLlama* is a SLM design that initiates from a larger model and applies a careful parameter sharing scheme to reduce both the pre-training and the deployment cost. Our work strives to not only bridge the gap in open-source SLMs but also ensures full transparency, where complete training data pipeline, training code, model weights, and over 300 checkpoints along with evaluation codes is available at : https://github.com/mbzuai-oryx/MobiLlama.

1 Introduction

Recent years have witnessed a tremendous surge in the development of Large Language Models (LLMs) with the emergence of prominent closed-source commercial models such as ChatGPT, Bard, and Claude. These LLMs exhibit surprising capabilities, typically called emergent abilities, towards solving complex tasks. Most existing popular LLMs follow a similar trend that bigger is always better, where scaling model size or data size typically provides improved model capacity and performance on downstream tasks. For instance, the recent Llama-2 70 billion (70B) model (Touvron et al., 2023) is considered more favorable in different chat applications due to its effectiveness towards handling dialogues, logical reasoning, coding, compared to its 7B counterpart which is typically better suited for basic tasks such as categorization or summaries. While these LLMs demonstrate impressive performance in handling complex language tasks, a key limitation is their size and computational requirements. For instance, the large-scale Falcon (Almazrouei et al., 2023) 180B model was trained using 4096 A100 GPUs and requires large memory and compute for deployment with dedicated high-performance servers and scalable storage systems.

Recently, Small Language Models (SLMs) have shown potential in terms of providing decent performance with emergent abilities achieved at a significantly smaller scale compared to their large-scale LLM counterparts. Modern SLMs like Microsoft's Phi-2 2.7 billion (Li et al., 2023b) highlight the growing focus in the community on achieving more with less. SLMs offer advantages in terms of efficiency, cost, flexibility, and customizability. With fewer parameters, SLMs offer significant computational efficiency in terms of fast pre-training and inference with reduced memory and storage requirements. This is critical in real-world applications where efficient resource utilization is highly desired. It particularly opens up possibilities in resource-constrained computing, where the models are required to be memory efficient to operate on low-powered devices (e.g., edge). SLMs support on-device processing that enhances privacy, security, response time, and personalization. Such an integration can lead to advanced personal assistants, cloud-independent applications, and improved energy efficiency with a reduced carbon footprint.

The landscape of language models, especially SLMs, is currently marked by a notable lack of open-source availability. While LLMs have gar-

[*]Equal contribution.

nered significant attention, the proprietary nature of most models has led to limited transparency and accessibility, particularly in the realm of SLMs. This gap hinders the scientific and technological exploration of these more efficient, compact and performant models. Recognizing this, there's a growing need in the community for fully transparent open-source SLMs, which would facilitate a deeper understanding of their capabilities and limitations and spur innovation by allowing broader community access to their architecture and reproducible training methodologies. We argue that bridging this gap is crucial for democratizing access to collaborative advancement for SLMs. Therefore, we investigate the problem of designing accurate yet efficient SLMs from scratch with the intention to provide full transparency in the form of access to entire training data pipeline and code, model weights, more than 300 checkpoints along with evaluation codes.

When designing a SLM from scratch it is desired that the resulting model is accurate, while maintaining efficiency in terms of pre-training and deployment. A straightforward way is to scale-down a larger LLM design to the desired model size (e.g., 0.5B) by reducing either the size of the hidden dimension layers or the number of layers. We empirically observe both these design strategies to provide inferior performance. This motivates us to look into an alternative way of designing a SLM from scratch that is accurate yet maintains the efficiency, while offering full transparency.

Contributions:

We introduce a SLM framework, named *MobiLlama*, with an aim to develop accurate SLMs by alleviating the redundancy in the transformer blocks. Different to the conventional SLM design where dedicated feed forward layers (FFN) are typically allocated to each transformer block, we propose to employ a shared FFN design for all the transformer blocks within SLM. Our *MobiLlama* leveraging a shared FFN-based SLM design is accurate and maintains efficiency, while offering full transparency in terms of data pipeline, training code, model weights and extensive intermediate checkpoints along with evaluation codes.

We empirically show that our *MobiLlama* performs favorably compared to conventional SLMs design schemes when performing pre-training from scratch. Our *MobiLlama* 0.5B model outperforms existing SLMs of similar size on nine different benchmarks. *MobiLlama* 0.5B achieves a gain of 2.4% in terms of average performance on nine

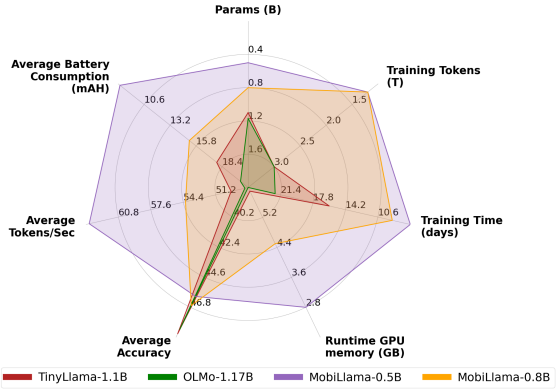

Figure 1: Comparison of our *MobiLlama* 0.5B and 0.8B models with recent OLMo-1.17B (Groeneveld et al., 2024) and TinyLlama-1.1B (Zhang et al., 2024) in terms of pre-training tokens, pre-training time and memory, model parameters, overall accuracy across nine benchmarks and on-device efficiency (average battery consumption and average token/second on a PC with RTX2080Ti). Our *MobiLlama* achieves comparable accuracy while requiring significantly fewer pre-training data (1.2T tokens vs. 3T tokens), lesser pre-training time and GPU memory along with being efficient in terms of deployment on a resource constrained device.

benchmarks, compared to the best existing 0.5B SLM in the literature. We further develop a 0.8B SLM that originates from our 0.5B model by utilizing a wider shared-FFN scheme in transformer blocks, achieving top performance among existing SLMs falling under less than 1B parameters category. Lastly, we build multimodal models on top of our SLM to showcase visual perception and reasoning capabilities. Fig. 1 shows a comparison of our *MobiLlama* with recent fully transparent relatively larger SLMs in terms of accuracy, pre-training complexity and on-board deployment cost.

2 Related Work

While LLMs have gained tremendous popularity (Zhao et al., 2023), one of their key limitations is the size and computational requirements both during pre-training and deployment. Another issue is limited availability of fully transparent open-source LLMs that provide complete access to data pipeline, training code along with checkpoints and evaluation protocols. Prior works explore making several components of LLM framework efficient such as, attention mechanism (Dao, 2023) and optimization strategies (Loshchilov and Hutter, 2017). Further, existing efforts also include exploring post-training sparsification schemes (Ashkboos et al., 2024) or quantization (Hoefler et al., 2021; Zhu et al., 2023; Xiao et al., 2023) of computationally

Model	#Params	Training Time	GPU Hours	GPU memory	No. of layers	Hidden dim size
baseline1	0.54B	7.5 days	28.8K	3.2 GB	22	1024
baseline2	0.52B	7 days	26.9K	3 GB	8	2048
large-base	1.2B	12 days	46.1K	6 GB	22	2048
MobiLlama	0.52B	7 days	26.6K	3 GB	22	2048

Table 1: Comparison of our *MobiLlama* with the two baselines and the large-base model. We show the comparison in terms of total number of parameters, training time, total GPU hours, GPU memory, number of transformer layers and the hidden dimension size in each layer. The numbers are computed on A100 GPUs with 80 GB memory each. Compared to *large-base*, our *MobiLlama* reduces the GPU training hours by 42% along with a significant reduction in GPU memory with the same design configuration (number of layers and hidden dimension size etc.). Further, our *MobiLlama* possesses increased model capacity in terms of number of layers and hidden dimension size while maintaining comparable training cost and parameters, compared to *baseline1* and *baseline2*.

expensive LLM. In several cases, such a post-hoc sparsification can reduce the performance of LLMs with more on-device memory consumption, compared to a SLM trained from scratch. Further, these techniques typically employ LLMs with limited transparency and accessibility.

Recently, designing SLMs from scratch have gained attention (Biderman et al., 2023; Wu et al., 2023; Zhang et al., 2024; Li et al., 2023a; Lin et al., 2021b; Shoeybi et al., 2019; Zhang et al., 2022). SLMs have shown potential as an alternative especially in case of limited pre-training compute as well as deployment in resource-constrained environments (e.g., edge devices). Further, SLMs can support on-device processing which in turn can enhance security, privacy, response efficiency, and personalization. Here, we strive to construct fully transparent accurate yet computationally efficient SLMs by maintaining the model's capacity to capture complex patterns and relationships in data while reducing the redundancy often present in the parameters of SLMs. Prior works (Frantar et al., 2022; Gholami et al., 2022; Pires et al., 2023; Pan et al., 2023; Bhojanapalli et al., 2021) exploring alleviating redundancy in transformer design either focusing on the attention mechanism or on the single feed-forward layer in BERT style architectures. Different from these approaches, we explore alleviating the redundancy in the SLM architectures with an LLM objective function by focusing on the sharing mechanism of MLP blocks having multiple feed-forward network (FFN) layers.

3 Method

3.1 Baseline SLM Design

We first describe our baseline 0.5B SLM architecture that is adapted from recent TinyLlama (Zhang et al., 2024) and Llama-2 (Touvron et al., 2023). The baseline architecture comprises N layers, where each layer consists of hidden dimensions of M and intermediate size (MLPs) of 5632. The vocabulary size is $32K$ and max. context length is C. We consider two different design choices when constructing a 0.5B model from scratch. In first design choice, named $baseline1$, the number of layer is set to $N = 22$ and hidden size of each layer is set to $M = 1024$. In second design choice, named $baseline2$, we set the number of layer to $N = 8$ and hidden size of each layer is set to $M = 2048$.

We note that both the aforementioned baseline designs struggle to strike an optimal balance between accuracy and efficiency. While a reduced size of hidden dimensions (1024) in case of $baseline1$ aids in computational efficiency, it can likely hamper the model's capacity to capture complex patterns within the data. Such a reduction in dimension can potentially lead to a bottleneck effect, where the model's ability to represent intricate relationships and nuances in the data is constrained, thereby affecting the overall accuracy. On the other hand, reducing the number of hidden layers (22 to 8), as in the $baseline2$, affects the model's depth that in turn hampers its ability to learn hierarchical representations of the language. Achieving superior performance on tasks requiring deeper linguistic comprehension and contextual analysis likely requires combining the advantages of the two aforementioned baselines. However, increasing the model capacity of $baseline1$ and $baseline2$ into a single model (22 layers and hidden dimension size of 2048) results in a significantly larger parameterized model of 1.2B with increased training cost (see Tab. 1). We name this larger model as *large-base*. Next, we present our proposed *MobiLlama* 0.5B model design that does not reduce hidden dimension size in each layer ($baseline1$) or the total number of layers ($baseline2$), while maintaining a comparable training efficiency (see Tab. 1).

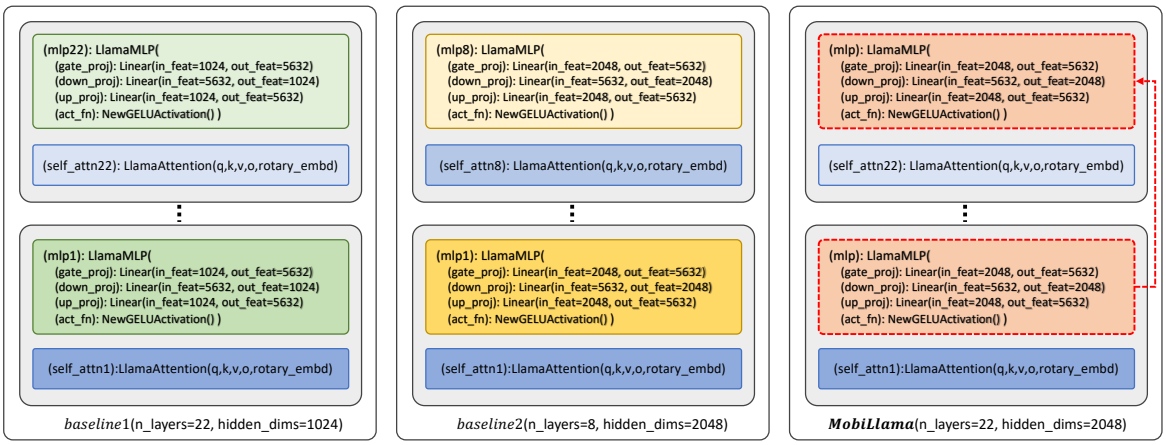

Figure 2: Illustrative comparison of our *MobiLlama* with the two baselines. For each case, we show two transformer blocks denoted by different self-attention layers. In the case of both *baseline1* and *baseline2*, a dedicated MLP block comprising three FFN layers is utilized for each transformer layer. In contrast, our *MobiLlama* utilizes a single MLP block (highlighted by the same color) that is shared across different transformer layers. This enables to increase the capacity of the network in terms of layers and hidden dimension size without any significant increase in the total number of trainable parameters.

3.2 Proposed SLM Design: MobiLlama

The proposed approach, *MobiLlama*, constructs a SLM of desired sizes (e.g., 0.5B model) by first initiating from a larger model size design, *large-base*. Then, we employ a careful parameter sharing scheme to reduce the model size to a pre-defined model configuration, thereby significantly reducing the training cost. Generally, both SLMs and LLMs typically utilize a dedicated multilayer perceptron (MLP) block comprising multiple feed forward network (FFN) layers within each transformer block. In such a configuration (e.g., *large-base*), the FFN layers account for a substantial 65% of the total trainable parameters, with attention mechanisms and heads contributing 30% and 5%, respectively. As a consequence, a significant number of parameters are concentrated within the FFN layers, thereby posing challenges during pre-training with respect to computational cost and the model's ability to achieve faster convergence. To address these issues, we propose to use a sharing scheme where the FFN parameters are shared across all transformer layers within the SLM. This enables us to significantly reduce the overall trainable parameters by 60% in our *MobiLlama*, compared to the *large-base*. Such a significant parameter reduction also enables us to increase the model capacity in terms of number of layers and hidden dimension size without any substantial increase in the training cost (see Tab. 1).

Fig. 2 compares our architecture design with two baselines. In case of both baselines, a dedicated MLP block that consists of multiple FFN layers is used in each transformer layer. Instead, our efficient *MobiLlama* design utilizes a single MLP block which is shared across different layers of transformer within the SLM. This helps in increasing the model capacity without any increase in the total number of trainable parameters in the model.

Subset	Tokens (Billion)
Arxiv	30.00
Book	28.86
C4	197.67
Refined-Web	665.01
StarCoder	291.92
StackExchange	21.75
Wikipedia	23.90
Total	1259.13

Table 2: Data mix in Amber-Dataset.

Hyperparameter	Value
Number Parameters	$0.5B$
Hidden Size	2048
Intermediate Size (in MLPs)	5632
Number of Attention Heads	32
Number of Hidden Layers	22
RMSNorm ϵ	$1e^{-6}$
Max Seq Length	2048
Vocab Size	32000

Table 3: *MobiLlama* architecture & hyperparameters.

3.3 Towards Fully Transparent MobiLlama

As discussed earlier, fully transparent open-source SLM development is desired to foster a more inclusive, data/model provenance, and reproducible collaborative SLM research development environment. To this end, we present here pre-training dataset

Model Name	#Params	HellaSwag	Truthfulqa	MMLU	Arc_C	CrowsPairs	piqa	race	siqa	winogrande	Average
gpt-neo-125m	0.15B	30.26	45.58	25.97	22.95	61.55	62.46	27.56	40.33	51.78	40.93
tiny-starcoder	0.17B	28.17	47.68	26.79	20.99	49.68	52.55	25.45	38.28	51.22	37.86
cerebras-gpt-256m	0.26B	28.99	45.98	26.83	22.01	60.52	61.42	27.46	40.53	52.49	40.69
opt-350m	0.35b	36.73	40.83	26.02	23.55	64.12	64.74	29.85	41.55	52.64	42.22
megatron-gpt2-345m	0.38B	39.18	41.51	24.32	24.23	64.82	66.87	31.19	40.28	52.96	42.81
LiteLlama	0.46B	38.47	41.59	26.17	24.91	62.90	67.73	28.42	40.27	49.88	42.26
gpt-sw3-356m	0.47B	37.05	42.55	25.93	23.63	61.59	64.85	32.15	41.56	53.04	42.48
pythia-410m	0.51B	40.85	41.22	27.25	26.19	64.20	67.19	30.71	41.40	53.12	43.57
xglm-564m	0.56B	34.64	40.43	25.18	24.57	62.25	64.85	29.28	42.68	53.03	41.87
Lamini-GPT-LM	0.59B	31.55	40.72	25.53	24.23	63.09	63.87	29.95	40.78	47.75	40.83
MobiLlama (Ours)	0.5B	52.52	38.05	26.45	29.52	64.03	72.03	33.68	40.22	57.53	**46.00**
Lamini-GPT-LM	0.77B	43.83	40.25	26.24	27.55	66.12	69.31	37.12	42.47	56.59	45.49
MobiLlama (Ours)	0.8B	54.09	38.48	26.92	30.20	64.82	73.17	33.37	41.60	57.45	**46.67**

Table 4: State-of-the-art comparisons with existing < *1B params models* on *nine* benchmarks. In case of around 0.5B model series, our *MobiLlama* achieves a substantial gain of 2.4% in terms of average performance on nine benchmarks. Further, our *MobiLlama* 0.8B model achieves an average score of 46.67.

and processing details, architecture design configuration with training details, evaluation benchmarks and metrics. In addition, we will publicly release complete training and evaluation codes along with intermediate model checkpoints.

Pre-training Dataset and Processing: For pre-training, we use 1.2T tokens from LLM360 Amber dataset (Liu et al., 2023b). The Amber dataset provides a rich and varied linguistic landscape having different text types, topics, and styles. Tab. 2 shows the data mix from Amber dataset gathered from various sources.

Arxiv (30 Billion Tokens) subset is drawn from the repository of scientific papers, provides complex, domain-specific language and technical terminology, enriching the understanding of academic prose. *Book (28.9 Billion Tokens)* subset comprises tokens from a broad range of literature with diverse narrative styles, cultural contexts, and rich vocabulary, deepening the grasp of storytelling and language nuances. *C4 (197.7 Billion Tokens)* is the Colossal Clean Crawled Corpus (C4) that offers a vast and cleaned selection of web text, providing a broad linguistic foundation that includes various registers, styles, and topics. *Refined-Web (665 Billion Tokens)* subset is a curated web crawl and offers the model exposure to contemporary, informal, and varied internet language, enhancing the relevance and applicability to modern communication. *StarCoder (291.9 Billion Tokens)* subset is a vast collection used for code understanding featuring 783GB of code across 86 programming languages. It includes GitHub issues, Jupyter notebooks, and commits, totaling approximately 250 billion tokens. These are meticulously cleaned and de-duplicated for training efficiency. *StackExchange (21.8 Billion Tokens)* is from the network of Q&A websites, this subset aids the model in learning question-answering formats and technical discussions across diverse topics. *Wikipedia (23.9 Billion Tokens)* is an encyclopedia collection, it offers well-structured and factual content that helps the model to learn encyclopedic knowledge and formal writing styles.

From the above-mentioned subsets, Arxiv, Book, C4, StackExchange and Wikipedia are sourced from RedPajama-v1 (Computer, 2023). The Amber dataset uses RefinedWeb (Penedo et al., 2023) data to replace common_crawl subset of RedPajama-v1. These subsets amount to 1259.13 billion tokens.

Initially, raw data sourced from the above sources is tokenized using Huggingface LLaMA tokenizer (Touvron et al., 2023). Subsequently, these tokens are organized into sequences with each containing 2048 tokens. To manage data, these sequences are merged to the token sequences and divided the amalgamated dataset into 360 distinct segments. Each data segment, structured as a jsonl file, carries an array of token IDs along with a source identifier that denotes the originating dataset. Each data sample is designed to have 2049 tokens.

Architecture Design: Our *MobiLlama* 0.5B comprises a hidden size of 2048, an intermediate size of 5632 in its MLPs, and operates with 32 attention heads across 22 hidden layers. It is designed to handle sequences up to 2048 tokens long, supported by a vocabulary size of 32,000. The precision in normalization is ensured by an RMSNorm epsilon of $1e^{-6}$ to obtain a more stable training. We utilize RoPE (Rotary Positional Embedding) (Su et al., 2024) to encode positional information in our *MobiLlama*. Similar to (Zhang et al., 2024), we employ a combination of Swish and Gated Lin-

Model	HellaSwag	Truthfulqa	MMLU	Arc_C	Average
baseline1	42.44	38.46	25.08	26.18	33.04
baseline2	42.15	38.70	25.73	26.10	33.17
MobiLlama	44.47	40.12	26.48	26.53	**34.40**

Table 5: Baseline comparison on four benchmarks. Here, both the baselines and our *MobiLlama* comprise the same parameters (0.5B) and are pre-trained on 100B tokens from Amber. Our *MobiLlama* achieves favorable performance compared to the two baselines, while operating on a similar training budget.

ear Units together as activation functions. Tab. 3 presents details of our model configuration. We also derive a 0.8B version from our *MobiLlama* by widening the shared FFN design. Compared to the 0.5B model, our 0.8B design increases the hidden dimension size to 2532 and the intermediate size to 11,080 while the rest of the configuration is same.

For pre-training of our *MobiLlama*, we use a public cluster having 20 GPU nodes each equipped with 8 NVIDIA A100 GPUs with 80 GB memory each and 800 Gbps interconnect for model training. Each GPU is interconnected through 8 NVLink links, complemented by a cross-node connection configuration of 2 port 200 Gb/sec (4× HDR) InfiniBand, optimizing the model's training process. To further enhance the training efficiency, we employ flash-attention mechanism and follow the pre-training hyper-parameters established by the LLaMA (Touvron et al., 2023) model. Our *MobiLlama* model's training is performed using the AdamW optimizer, leveraging hyperparameters $\beta_1 = 0.9, \beta_2 = 0.95$, with an initial learning rate of $\eta = 3e^{-4}$. This rate follows a cosine learning rate schedule, tapering to a final rate of $\eta = 3e^{-5}$. We further incorporate a weight decay of 0.1 and apply gradient clipping at 1.0 with a warm-up period over $2,000$ steps. Adapting to our hardware configuration of 20 GPU nodes, we optimize the pre-training batch size to 800 (160×5), achieving a throughput of approximately $14k$-$15k$ tokens per second on a single GPU. During our model pre-training, we save intermediate checkpoints after every 3.3B tokens which will be publicly released.

Evaluation Benchmarks and Metrics:

For a comprehensive performance evaluation, we use nine different benchmarks from the Open LLM Leaderboard[1].

HellaSwag (Zellers et al., 2019) assesses the model's ability to predict the correct ending to a scenario from a set of possible continuations, thereby testing common sense reasoning. TruthfulQA (Lin et al., 2021a) evaluates the model to provide truthful answers, focusing on its understanding of facts and its ability to avoid deception. MMLU (Hendrycks et al., 2020) measures the model's broad knowledge across numerous subjects such as, humanities, science, technology, engineering and management. ARC_Challenge (Clark et al., 2018) tests complex reasoning with science questions. CrowsPairs (Nangia et al., 2020) evaluates the model's biases by comparing sentences that differ only by the demographic group mentioned, aiming for fairness. PIQA (Bisk et al., 2020) evaluates the model's physical commonsense knowledge, requiring understanding of everyday physical processes. Race (Lai et al., 2017) assesses reading comprehension through multiple-choice questions based on passages. SIQA (Sap et al., 2019) focuses on the model's social commonsense reasoning and its understanding of social dynamics. Winogrande (Sakaguchi et al., 2021) evaluates the model's ability to resolve ambiguities in text, testing its commonsense reasoning.

Following the Analysis-360 framework (Liu et al., 2023b) that is built on llm-harness (Gao et al., 2023), we conduct extensive evaluations under the standard settings with varying shots for detailed assessments, validating the model's robustness and adaptability across diverse linguistic tasks. Following the standard evaluation protocol, our evaluation setting consists of 10, 25, 5 and 5 shot evaluation for Hellaswag, ARC_Challenge, Winogrande and MMLU, while zero-shot for rest of the benchmarks.

4 Results

Baseline Comparison: We first present a comparison with the two baselines in Tab. 5) for 0.5B model series. For the baseline evaluation, we pre-train all the models on the same 100B tokens from the Amber dataset and report the results on four benchmarks: HellaSwag, TruthfulQA, MMLU, and Arc_C. Our *MobiLlama* achieves favourable performance compared to the two baselines by achieving an average score of 34.4 over the four benchmarks. We note that this performance improvement is achieved without any significant increase in the training cost (see Tab. 1), highlighting the merits of the proposed SLM design.

State-of-the-art Comparison: We compare our *MobiLlama* 0.5B and 0.8B with existing SLMs having comparable (less than 1B) parameters: gpt-

[1] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard