MobiLlama: Towards Accurate and Lightweight Fully Transparent GPT

Anonymous ACL submission

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 will be publicly released.

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 garnered 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 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.

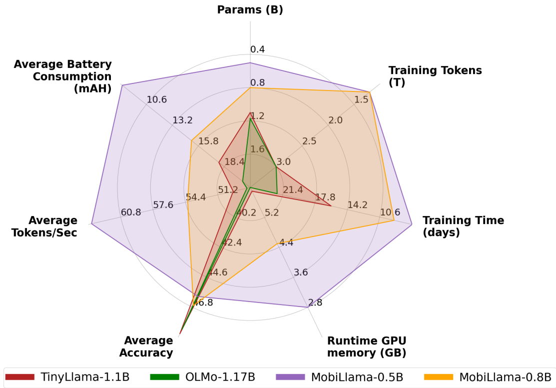

Figure 1: Comparison of our *MobiLlama* 0.5B and 0.8B models with recent OLMo-1.17B (Groeneveld et al., 2024) and TinyLlama-1.1B (Zhang et al., 2024a) in terms of pre-training tokens, pre-training time and memory, model parameters, overall accuracy across nine benchmarks and on-device efficiency (average 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.

2 Related Work

LLMs have become immensely popular but face challenges with size and computational demands during training and deployment, with limited availability of fully open-source models that provide complete transparency (Zhao et al., 2023). Modern efforts to enhance efficiency focus on optimizing components like the attention mechanism (Dao, 2023) and employing strategies such as sparsification (Ashkboos et al., 2024) and quantization (Hoefler et al., 2021; Zhu et al., 2023; Xiao et al., 2023). Recently, the development of Small Language Models (SLMs) has been emphasized for use in resource-limited environments, aiming for on-device processing to improve security and efficiency (Biderman et al., 2023; Wu et al., 2023; Zhang et al., 2024a; Li et al., 2023a; Lin et al., 2021b; Shoeybi et al., 2019; Zhang et al., 2022). Our work extends these efforts by focusing on re-

Model	#Params	Training Time	GPU Hours	GPU memory	No. of layers	Hidden dim size	Avg Accuracy
baseline1	0.54B	7.5 days	28.8K	3.2 GB	22	1024	32.97
baseline2	0.52B	7 days	26.9K	3 GB	8	2048	32.57
baseline3	0.22B	4.1 days	11.8K	3 GB	22	2048	21.78
large-base	1.2B	12 days	46.1K	1 GB	22	2048	36.08
MobiLlama	0.52B	7 days	26.6K	3 GB	22	2048	35.55

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*. Additionally, *MobiLlama* demonstrates superior performance to *baseline3* with similar capacity and enhanced efficiency due to its full transformer block sharing.

ducing redundancy in SLMs through sharing mechanisms in MLP blocks (Frantar et al., 2022; Gholami et al., 2022; Pires et al., 2023; Pan et al., 2023; Bhojanapalli et al., 2021).

Recent studies have investigated the effects of reducing layers in neural architectures on capturing complex linguistic structures, with specific focus on the efficiency and necessity of attention mechanisms (Voita et al., 2019; Michel et al., 2019). (Michel et al., 2019) highlights the critical roles of intermediate layers, while (Ma et al., 2023) explores structural pruning to maintain performance post-training. Building upon these insights, our MobiLlama model enhances the interaction between layers through a shared FeedForward Network (FFN), optimizing both the structural efficiency and the processing capability of the model.

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., 2024a) and Llama-2 (Touvron et al., 2023). 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 $M = 2048$. Additionally, We also experiment similar to the one in (Gao et al., 2022) by considering full transformer block sharing across all layers of the LLM referred to as $baseline3$ with $N = 22$ and $M = 2048$.

We note that 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. Furthermore, by sharing the entire transformer block across all layers as in the $baseline3$, limits the model capacity by constraining all layers to learn identical features, thereby reducing the model's expressiveness and flexibility to capture complex data variations. This design can lead to overfitting specific training patterns and impair learning dynamics due to shared gradients. Moreover, such sharing hinders scalability in larger models as diverse layer-specific needs cannot be effectively addressed (Raganato and Tiedemann, 2018).

Achieving superior performance on tasks requiring deeper linguistic comprehension and contextual analysis likely requires combining the advantages of the two aforementioned designes ($baseline1$ and $baseline2$). 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 with the capacity of both the models $baseline1$ and $baseline2$, while maintaining a comparable training efficiency (see Tab. 1).

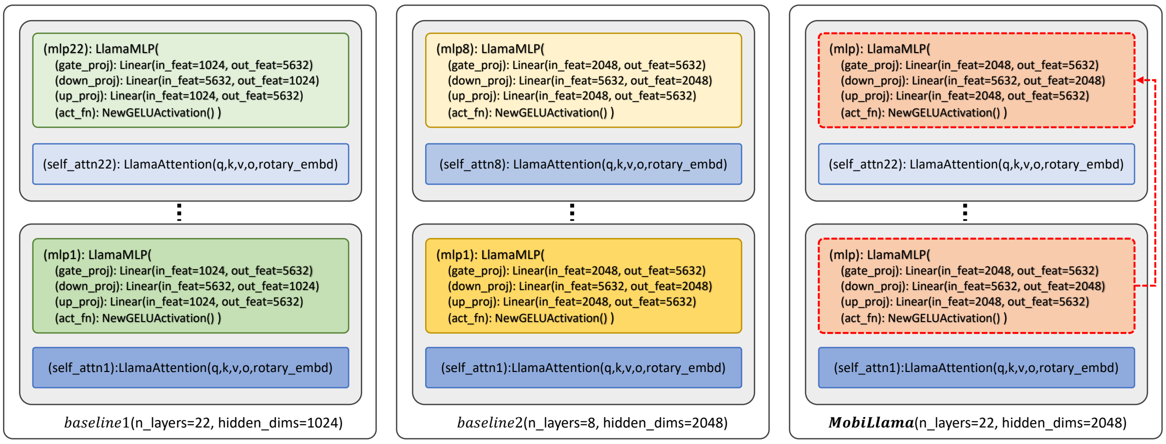

Figure 2: Illustrative comparison of our *MobiLlama* with the two baselines. For each case, we show two transformer blocks denoted by different self-attention layers. In the case of both *baseline1* and *baseline2*, a dedicated MLP block comprising three FFN layers is utilized for each transformer layer. 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.

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 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.

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.

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 3: 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.

Table 2 represents the data mix from Amber Dataset gathered from various sources. The dataset's comprehensive nature supports the model's ability to grasp subtle distinction of language, enhancing its performance on a variety of tasks, from language understanding to content generation. 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: Tab. 4 presents details of our model configuration. We utilize RoPE (Rotary Positional Embedding) (Su et al., 2024) to encode positional information in our *MobiLlama*. Similar to (Zhang et al., 2024a), we employ a combination of Swish and Gated Linear Units together as activation functions. 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 maintained same.

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 4: *MobiLlama* architecture & hyperparameters.

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. 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 the baseline evaluation, we pre-train all the models on the same 120B 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 $baseline1$ and $baseline2$ and significantly outperform $baseline3$ by achieving an average score of 35.55 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.

Model	HellaSwag	Truthfulqa	MMLU	Arc_C	Average
baseline1	42.44	38.16	25.12	26.18	32.97
baseline2	43.66	38.54	25.76	26.32	33.57
baseline3	28.41	25.02	16.68	17.01	21.78
MobiLlama	48.42	39.36	26.56	27.88	**35.55**

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 120B tokens from Amber. Our *MobiLlama* achieves favorable performance compared to the three baselines, while operating on a similar training budget.

State-of-the-art Comparison: We compare our *MobiLlama* 0.5B and 0.8B with existing SLMs having comparable (less than 1B) parameters: gpt-neo (Black et al., 2021), tiny-starcoder (Li et al., 2023a), cerebras-gpt (Dey et al., 2023), opt (Zhang et al., 2022), megatron-gpt-2 (Shoeybi et al., 2019), LiteLlama, gpt-sw3, pythia (Biderman et al., 2023), xglm (Lin et al., 2021b), Lamini-LM (Wu et al., 2023). Among existing methods falling around 0.5B model series category, pythia-410m achieves an average score of 43.57. Our *MobiLlama* 0.5B model achieves superior performance with an average score of 46.0, outperforming pythia-410m by 2.4% in terms of average performance on nine benchmarks. Notably, *MobiLlama* achieves superior performance on the HellaSwag benchmark which is designed to evaluate the model's capabilities in the NLP text completion task. Further, *MobiLlama* also performs favorably on commonsense reasoning tasks with superior results on piqa and winogrande benchmarks. Further, our *MobiLlama* 0.8B model achieves an average score of 49.06.

Efficiency Comparison: We present the comparison of our model in terms of efficiency and resource consumption on various low-end hardware platforms: a PC with RTX-2080Ti GPU, a laptop with i7 CPU, and a smartphone with Snapdragon-685 processor. Tab. 6 shows the comparison of our *MobiLlama* 0.5B with *large-base* 1.2B, Llama2-7B (Touvron et al., 2023) and Phi2-2.7B (Li et al., 2023b) model, in terms of the average processing speed in tokens per second (Average Tokens/Sec), average memory consumption (Avg Memory Consumption) in megabytes (MB), and the average battery consumption (Average Battery Consumption/1000 Tokens) in milliampere-hours (mAH). Our *MobiLlama* performs favorably in terms of efficiency across different hardware platforms.

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

Platform	Model	#Params (↓)	Precision	Avg Tokens/Sec (↑)	Avg Memory Consumption (↓)	Avg Battery Consumption /1k Tokens (↓)	CPU Utilization (↓)
RTX2080Ti	Llama2	7B	bf16	14.85	27793 MB	135.51 mAH	31.62%
	Phi2	2.7B	bf16	32.19	12071 MB	59.13 mAH	24.73%
	large-base	1.2B	bf16	50.61	6254 MB	18.91 mAH	18.25%
	MobiLlama	0.5B	bf16	**63.38**	**3046** MB	**8.19** mAH	**14.79**%
CPU-i7	Llama2	7B	4bit	5.96	4188 MB	73.5 mAH	49.16%
	Phi2	2.7B	4bit	22.14	1972 MB	27.36 mAH	34.92%
	large-base	1.2B	4bit	29.23	1163 MB	10.81 mAH	30.84%
	MobiLlama	0.5B	4bit	**36.32**	**799** MB	**4.86** mAH	**24.64**%
Snapdragon-685	Llama2	7B	4bit	1.193	4287 MB	10.07 mAH	77.41%
	Phi2	2.7B	4bit	2.882	1893 MB	14.61 mAH	56.82%
	large-base	1.2B	4bit	6.687	780 MB	6.00 mAH	17.15%
	MobiLlama	0.5B	4bit	**7.021**	**770** MB	**5.32** mAH	**13.02**%

Table 6: Comparison in terms of efficiency and resource consumption on different low-end hardware devices. We show the comparison on: a PC with RTX-2080Ti GPU, a laptop with i7 CPU and a smartphone with Snapdragon-685 processor. In addition to our *large-base* model, we also present the comparison with Llama2 7B and Phi2 2.7B. In case of CPU and smartphone, we use 4-bit GGUF format of the corresponding models, whereas the original models are deployed and tested on PC with RTX-2080Ti GPU. The different metrics measure the model's operational efficiency, model's footprint in the device's RAM and the energy efficiency of processing 1,000 tokens. Our *MobiLlama* performs favorably in terms of efficiency on these low-end hardware devices. We note that both Phi2 and Llama2 are not fully transparent in that the complete data pipeline for pre-training is not publicly available.

Model	#Slice	#Params	HellaS	Arc_C	piqa	wino	Average
OPT-1.3B	30%	0.91B	39.81	25.77	60.77	54.7	45.26
OPT-6.7B	30%	4.69B	54.56	29.01	68.61	60.69	53.21
Llama-2-7B	30%	4.9B	49.62	31.23	63.55	61.33	51.43
Phi2-2.7B	30%	1.89B	47.56	30.29	65.94	63.14	51.73
MobiLlama	Dense	0.5B	52.52	29.52	72.03	57.53	52.90
	Dense	0.8B	54.09	30.20	73.17	57.45	53.72

Table 7: Comparison on *4 open LLM benchmarks* when parameters are sliced down to 30% using Wiki2Text dataset, following (Ashkboos et al., 2024).

Model	GQA	SQA	TextQA	MME
MobiLlama-V	58.5	53.1	41.4	1191.9

Table 8: Quantitative performance of our multimodal design, *MobiLlama-V* 0.8B, on different benchmarks.

We further perform an efficiency comparison to a recent post-training sparsification scheme (Ashkboos et al., 2024), where each weight matrix is substituted with a smaller (dense) matrix, thereby reducing dimensions of the embeddings in the model. In such a scheme, the parameters of the original LLM are reduced significantly up to 70% followed by post-slicing fine-tuning using a dataset such as WikiText-2 (Merity et al., 2016). Tab. 7 shows the comparison of our *MobiLlama* with existing LLMs (e.g., Llama-2-7B, OPT-6.7B) on four benchmarks following (Ashkboos et al., 2024). Our *MobiLlama* 0.5B and 0.8B models perform favorably against representative LLMs, with an average score of 53.72 computed over four benchmarks. These results highlight the potential of designing new fully transparent SLMs that can achieve comparable capabilities of their larger sliced model counterparts.

Multimodal MobiLlama: We further build a multimodal model on top of our *MobiLlama* by combining it with a vision encoder to develop a general-purpose visual assistant having visual reasoning capabilities. Our multimodal model, *MobiLlama-V*, is trained by bridging the visual encoder of CLIP (Radford et al., 2021) with the language decoder of our *MobiLlama*, and fine-tuning it in an end-to-end fashion on a 665k vision-language instruction set (Liu et al., 2023a). We conduct evaluation on GQA (Hudson and Manning, 2019), SQA (Lu et al., 2022), TextQA (Singh et al., 2019), and MME (Fu et al., 2023). Tab. 8 shows the performance of *MobiLlama-V* 0.8B model.

Evaluating Large-base Model: As discussed earlier, we strive to develop fully transparent models for democratization of SLMs and fostering future research. To this end, we compare our *large-base* 1.2B with existing fully transparent SLMs falling within the less than 2B category. These existing SLMs are: Boomer, pythia (Biderman et al., 2023), Falcon-RW (Penedo et al., 2023), TinyLlama (Zhang et al., 2024b), OLMo (Groeneveld et al., 2024), cerebras-gpt (Dey et al., 2023), Lamini-LM (Wu et al., 2023), opt (Zhang et al., 2022) and gpt-neo (Black et al., 2021). Tab. 9 shows that compared to recent OLMo and TinyLlama that are pre-trained on a larger dataset of 3T

Model	#Params	HellaSwag	Truthfulqa	MMLU	Arc_C	CrowsPairs	piqa	race	siqa	winogrande	Average
Boomer	1B	31.62	39.42	25.42	22.26	61.26	57.99	28.99	40.32	50.98	39.80
Pythia-Dedup	1B	49.63	38.92	24.29	29.09	67.11	70.23	32.44	42.63	53.98	45.36
Falcon-RW	1B	63.12	35.96	25.36	35.06	69.04	74.10	36.07	40.23	61.88	48.98
TinyLlama	1.1B	60.22	37.59	26.11	33.61	70.60	73.28	36.45	41.65	59.18	48.74
OLMo	1.2B	62.50	32.94	25.86	34.45	69.59	73.70	36.74	41.14	58.90	48.42
Cerebras-GPT	1.3B	38.51	42.70	26.66	26.10	63.67	66.75	30.33	42.42	53.59	43.41
Lamini	1.3B	38.05	36.43	28.47	26.62	64.62	67.89	33.39	43.19	50.59	43.25
OPT	1.3B	54.50	38.67	24.63	29.6	70.70	72.47	34.16	42.47	59.74	47.43
GPT-NEO	1.3B	48.49	39.61	24.82	31.31	65.67	71.05	34.06	41.81	57.06	45.98
Pythia-Deduped	1.4B	55.00	38.63	25.45	32.59	67.33	72.68	34.64	42.68	56.90	47.32
large-base	1.2B	62.99	35.90	24.79	34.55	68.49	75.57	35.31	41.96	62.03	**49.06**

Table 9: Comprehensive comparisons with existing < *2B params fully open-source LLM models* on *9* benchmarks. Our 1.2B *large-base* model pre-trained on 1.2T tokens achieves superior performance compared to both the recent OLMo 1.17B model (Groeneveld et al., 2024) and TinyLlama 1.1B model (Zhang et al., 2024a), which are pre-trained on a substantially larger data of 3T tokens.

Model	#Params	Training Time	GPU Hours	GPU Memory
MobiLlama-0.5B	**0.52B**	**0.58 days**	**2.20 K**	**3.0 GB**
large-base-1.2B	1.2B	1.7 days	4.92 K	6.2 GB

Table 10: Training comparison of our MobiLlama-0.5B vs. standard non-shared SLM large-base pre-trained on 120B tokens from Amber dataset.

Model	Load (ms)	Init (ms)	Forward-Pass (ms)
large-base-1.2B	52	1896	15.7
MobiLlama-0.5B	**27**	**642**	**9.3**

Table 11: Latency analysis of our MobiLlama-0.5B vs. large-base using a profiler at inference time on RTX2080Ti.

tokens, our *large-base* 1.2B model pre-trained on 1.2T tokens achieves favourable results with an average score of 49.06 over nine benchmarks. We hope that our *large-base* model will serve as a solid baseline and help ease future research in SLM.

Why proposed FFN Sharing Works?: Our MobiLlama model utilizes a shared FeedForward Network (FFN) across all transformer layers to significantly enhance the efficiency of Small Language Models (SLMs). This approach reduces the total number of unique parameters, thereby lowering memory usage and accelerating the training process. By employing the same FFN unit across different layers, MobiLlama optimizes neural processing uniformity, enhancing the model's ability to generalize across various contexts. This not only saves computational resources during extensive pre-training but also sustains high performance, essential for deployment on resource-limited devices. During pre-training, gradient computation and backpropagation occur once per FFN instead of repeatedly across layers, streamlining the learning process as shown in Tab. 10. Moreover, during inference, MobiLlama's architecture avoids the frequent weight switching of FFN blocks between consecutive layers, leading to faster processing speeds and improved latency as demonstrated in Tab. 11. This highlights the operational benefits and advantages of our shared FFN design.

5 Conclusion

We present a fully transparent SLM, *MobiLlama*, that alleviates redundancy in the transformer block. Within *MobiLlama*, we propose to utilize a shared FFN design for all the blocks within the SLM. Our *MobiLlama* is accurate yet efficient in terms of training cost, on-device memory and storage efficiency. We evaluate *MobiLlama* on nine benchmarks, achieving favourable results compared to existing methods falling under less than 1B category. We also build a multimodal model on top of *MobiLlama* SLM to demonstrate visual reasoning capabilities. We hope that our *MobiLlama* will help accelerate research efforts towards building fully-transparent, accurate yet efficient SLMs that bridge the gap with their resource hungry LLM counterparts.

Limitation and Future Direction: A potential direction is to further improve *MobiLlama* for enhanced context comprehension and understanding subtlety of linguistic nuances. Domain-specific expertise of the model can also be explored (e.g., healthcare). While *MobiLlama* offers a fully transparent SLM framework, a follow-up study to understand any misrepresentations and biases is desired to improve model's robustness. While MobiLlama marks a significant stride in the development of lightweight, efficient language models, it is not without limitations.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Srinadh Bhojanapalli, Ayan Chakrabarti, Andreas Veit, Michal Lukasik, Himanshu Jain, Frederick Liu, Yin-Wen Chang, and Sanjiv Kumar. 2021. Leveraging redundancy in attention with reuse transformers. *arXiv preprint arXiv:2110.06821*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Nolan Dey, Gurpreet Gosal, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, Joel Hestness, et al. 2023. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

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.

Shangqian Gao, Burak Uzkent, Yilin Shen, Heng Huang, and Hongxia Jin. 2022. Learning to jointly share and prune weights for grounding based vision and language models. In *The Eleventh International Conference on Learning Representations*.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, A. Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Daniel Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hanna Hajishirzi. 2024. Olmo: Accelerating the science of language models. *arXiv preprint*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research*, 22(1):10882–11005.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023a. Starcoder: may the source be with you!

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021a. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021b. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. 2023b. Llm360: Towards fully transparent open-source llms.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. *arXiv preprint arXiv:2010.00133*.

Bowen Pan, Rameswar Panda, Rogerio Schmidt Feris, and Aude Jeanne Oliva. 2023. Interpretability-aware redundancy reduction for vision transformers. US Patent App. 17/559,053.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Telmo Pessoa Pires, António V Lopes, Yannick Assogba, and Hendra Setiawan. 2023. One wide feedforward is all you need. *arXiv preprint arXiv:2309.01826*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pages 287–297.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

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

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

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.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

A Appendix

Qualitative Analysis: Fig. 3 shows example responses obtained when interacting with *MobiLlama* 0.5B with conversation capabilities. We show examples covering different tasks such as, text completion, code generation and conversation capabilities. Our model generates faithful responses to these diverse interactions such as, asking to generate specific code snippet, cooking recipe and generating a poem about a specific topic (e.g., climate change). Fig. 4 shows examples demonstrating visual reasoning capabilities of our multimodal *MobiLlama-V*. For instance, *MobiLlama-V* accurately describes the atypical aspects of the image when asked to describe the given image.

More on Experimental Comparisons: Our work strives towards achieving two objectives: (i) improved accuracy while maintaining similar pre-training cost (pre-training time, GPU hours and GPU memory), (ii) better trade-off at inference/deployment in terms of accuracy and speed. To achieve the first objective, we empirically show in Tab. 5 that the proposed MobiLLama 0.5B model achieves superior accuracy compared to the two baseline 0.5B models of similar parameters under identical pre-training settings in terms of pre-training data (120B tokens), number of iterations, and hyper-parameters. Further, Tab. 1 in our paper shows that the proposed MobiLLama 0.5B model requires comparable pre-training cost compared to the two baseline 0.5B models. The comparable pre-training time between our MobiLLama 0.5B and the two baseline 0.5B models is likely due to identical unique trainable parameters. In the table below, we summarize the comparison between our MobiLLama 0.5B and the two baseline 0.5B models in terms of pre-training cost and accuracy under identical pre-training settings.

More on Pre-training Dataset: *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,

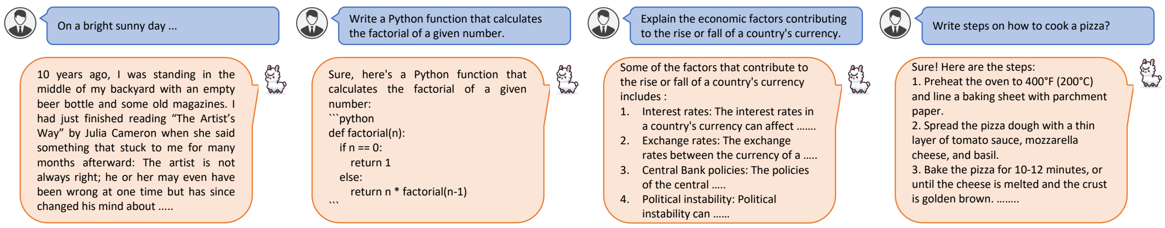

Figure 3: Example responses from our *MobiLlama* across a variety of tasks, including creative storytelling, coding exercises, economic analysis, and cooking instructions. The responses highlight the models' ability to engage with both abstract concepts and practical, step-by-step processes, demonstrating its broad applicability and sophisticated language processing capabilities.

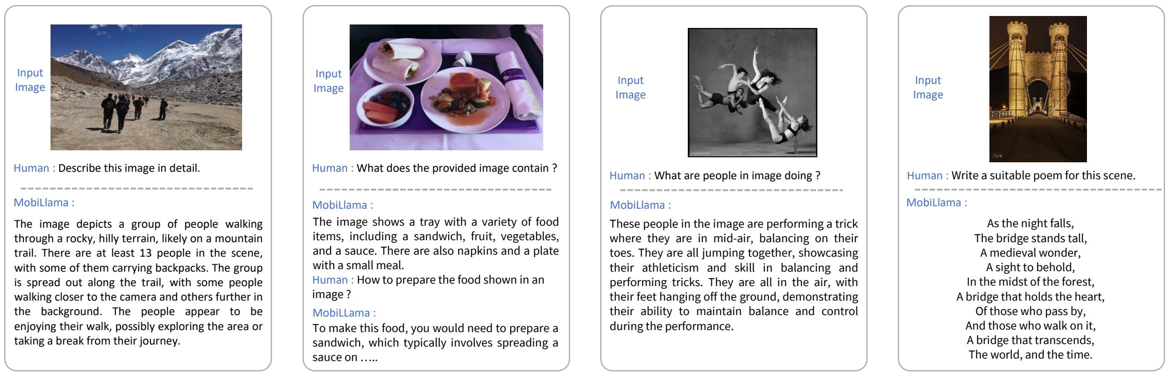

Figure 4: Example responses of *MobiLlama-V* in responding to visual stimuli across a range of scenarios. From describing a group's mountainous trek to poetic reflections on a scene, MobiLlama demonstrates a nuanced understanding of both the physical and emotive layers present in images. These qualitative responses highlight MobiLlama's capacity for detailed observation, creative interpretation, and generating contextually relevant textual content, affirming its potential in bridging the gap between visual perception and linguistic expression..

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.