



POLITECNICO
MILANO 1863

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**

EXECUTIVE SUMMARY OF THE THESIS

Beyond Attention: Efficient Small Language Models for Domain-Specific Tasks

LAUREA MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING

Author: VALERIO CECCARELLI

Advisor: PROF. MATTEO MATTEUCCI

Co-advisor: GRETA CORTI, PAOLO CUDRANO, SARA PIDÒ

Academic year: 2025-2026

1. Introduction

For the past few years, language models have improved mainly through scaling [1, 2]. Each new generation uses more parameters, trains on more data, and consumes more computation, producing systems of remarkable generality that are served from large data centers, where clusters of specialized accelerators consume hundreds of watts and depend on gigabytes of fast memory [3].

This work follows the opposite direction. Instead of measuring how a model's capabilities improve as it scales, it asks how much useful capability a model can still retain once it is small enough to run not in a data center but on an edge device such as a wearable, a sensor, or a battery-powered board [4, 5]. In this setting a self-contained model keeps the data it processes local, responds without a round trip to a remote server, and continues to work where connectivity is slow or absent, all at a fraction of the energy and cost of a cloud deployment.

Shrinking a model below a certain size is not only a matter of using fewer parameters. Hardware components below a certain size shift the model constraints from raw capability toward memory, energy, and the structure of the hard-

ware itself: the self-attention mechanism that made the Transformer [6] successful becomes ill-suited to the small, power-efficient accelerators found at the edge, which are designed for the static, weight-stationary linear algebra of convolutional networks rather than for the dynamic products of activations that attention computes [7]. For this reason, alternative architectures with sub-quadratic complexity are considered, among which this work adopts Mamba [8] for its competitive performances and its mature, widely supported implementations. A second idea guides this work. A model with few parameters cannot hold broad world knowledge, so its strength must lie in focus on a specific domain. Small models, however, are usually judged on the same benchmarks used for large models [9, 10]; yet, these benchmarks focus on general-purpose wide and multi-domain knowledge, which small models cannot have. A fair assessment must do the opposite and test knowledge depth in a narrow field. For this reason, four dedicated question-answering benchmarks were designed, covering two narrow domains, medicine and finance, each cast into two task formats: a balanced yes/no format and a multiple-choice one.

Existing baseline models BabyLlama (58M parameters) [11], MobileLLM (125M parameters) [12] and Mamba (130M parameters) [8] are adapted and evaluated for these benchmarks. The core contribution is BabyMamba, a 50 million parameters model based on the Mamba architecture and trained from scratch for this work, first through pretraining and offline distillation, and then fine-tuned on the new datasets. Its size is deliberately placed below the smallest model of the original Mamba study [8] while matching that of BabyLlama, which enables two comparisons at once: two architectures at the same size, and two sizes of the same architecture.

The results are encouraging. BabyMamba reaches performance in line with state-of-the-art models of comparable size, losing only a few points on the evaluation metrics with respect to models twice its size. The Mamba architecture can therefore be pushed well below the smallest size of the original work and still be specialized on a narrow domain, which is precisely the regime that matters for the constrained hardware where these models are meant to live.

2. Data Collection

Evaluating a specialized small model requires benchmarks that measure how deep a model goes in a single domain, not how widely it ranges, which is something the common general-purpose benchmarks are not built to do [9, 10]. Such benchmarks therefore had to be created from scratch, starting from a suitable source of domain text.

Two domains were chosen, medicine and finance, each covered by a public collection of question-answer pairs: WikiDoc [13] for medicine, taken from a collaborative platform where clinicians share up-to-date knowledge, and Investopedia [14] for finance, taken from a reference site written for readers with no financial background. Both keep the vocabulary and the concepts narrow, which is exactly what a small model needs in order to specialize, and both are restricted to English to avoid the curse of multilinguality, by which a fixed parameter budget shared across many languages lowers performance in each one [15]. A third, general-purpose corpus, a 900k-article English subset of Wikipedia [16], is reserved for pretraining new models from

scratch.

The original pairs are written as free text (opened questions with detailed answers), where the same fact can be phrased in many ways, which makes automatic and reproducible scoring unreliable. To obtain answers in a fixed, checkable form, the pairs were reshaped with the large language model GPT-5 [17] into two structured formats through a few-shot prompting strategy: a binary yes/no benchmark inspired by BoolQ [9] and a four-option multiple-choice benchmark inspired by MMLU [10]. Unlike those two benchmarks, no context passage is supplied, so a model must answer from what it learned during training. The result is four benchmarks, one binary and one multiple-choice for each of the two domains.

Example 1 Example of multiple-choice prompt generated from the Investopedia dataset.

Question: What type of rate do home equity loans have?

A: Variable

B: Fixed

C: Unsecured

D: HELOC

Answer:B

The benchmarks were also designed so that they would not reward superficial cues instead of genuine understanding. Because the prompts are produced by a language model, a model under test might guess the answer from the form of a question instead of its content. To prevent this in the yes/no benchmark, GPT-5 generated two questions per source item, one answered “yes” and one answered “no”, identical in structure and differing only in content, and the dataset was then balanced between the two answers. In the multiple-choice benchmark a deterministic shuffling algorithm spreads the correct option evenly across the four positions, so that position alone reveals nothing.

3. Baseline training

The small models BabyLlama, MobileLLM, and the 130M Mamba are already pretrained and they can reproduce the results of their original papers on standard benchmarks, yet they perform poorly on the new ones, because they

have never been trained to answer in the required format. These baselines therefore had to be adapted to the four benchmarks before the new model could be built, so that the comparison would be fair.

Two strategies were compared. The first is direct fine-tuning on each task, with the loss computed only on the answer token and not on the question, so that the model learns the required format and acquires the task without degrading the language understanding and the knowledge gained during pretraining. The second is a two-stage procedure in which a phase of continual pretraining precedes the fine-tuning. Continual pretraining continues the next-token training with the same cross-entropy loss, but on the plain domain text and with a deliberately small learning rate, so that the model absorbs the specific facts of the domain without its pre-trained weights being ruined by catastrophic forgetting [18]. The fine-tuning step then teaches it to answer in the required format.

The continual pretraining variant was consistently, if only modestly, better than direct fine-tuning, which indicates that the model does absorb part of the domain knowledge from the unlabeled text. This improvement varies with the task format. On the multiple-choice benchmarks both strategies reach accuracies clearly above the random classifier (Figure 1), whereas the binary benchmarks remain hard and unstable: none of the models performs significantly better than a random classifier, and their results swing widely with the particular split of the data. To check whether this difficulty was an artifact of model size, the same task was given to Llama 3 3B Instruct [19], a far larger and already instruction-tuned model, which reached the same level. The binary task therefore seems genuinely hard when no supporting passage is provided, and not just beyond the reach of very small models.

4. BabyMamba

BabyMamba is a very small instance of the Mamba architecture, built to test how far the design can be pushed below the smallest size of the original study.

It has 50 million parameters, roughly the size of BabyLlama, obtained from the configuration of the 130M Mamba by reducing the hidden size

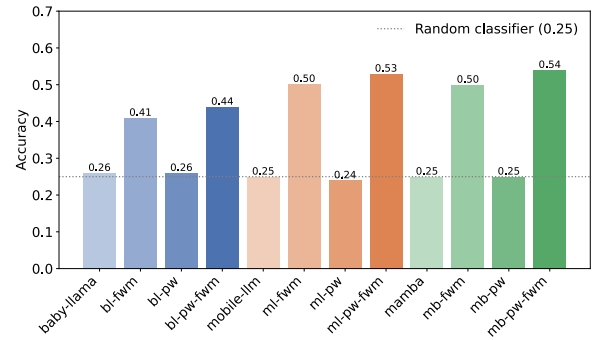


Figure 1: Accuracy of the three baseline model families on the wikidoc-multi benchmark. Each family appears in four configurations: the base model (e.g. baby-llama), fine-tuned (-fwm), continually pretrained (-pw), and fine-tuned after continual pretraining (-pw-fwm). The base models perform at the level of a random classifier; fine-tuning teaches the output format and clearly improves accuracy; and fine-tuning after continual pretraining achieves the best performance for every family.

from 768 to 384 while leaving the number of layers unchanged. Reducing the width rather than the depth keeps the model deep and thin, in line with the MobileLLM finding that thin, deep architectures outperform wide, shallow ones at this scale [12]. At such small sizes the embedding layers take up a large share of the parameter budget, so a smaller vocabulary would help; the model nonetheless keeps the GPT-NeoX tokenizer [20] shared by the whole Mamba family, because distillation requires the student and the teacher to share the same output space.

Three base models were produced from scratch on the 900k-article Wikipedia corpus. The first is pretrained in the ordinary way with the cross-entropy loss. The other two are produced by offline distillation from a Mamba 2.8B teacher of the same family. Because running teacher and student together was infeasible with the available resources, the teacher soft labels are generated once and cached, which makes storage the binding constraint: only the 40 most relevant tokens of each distribution are kept, already about 200GB on disk. Two ways of choosing those 40 tokens are compared, top-k, which keeps the most probable ones, and random sampling, which draws them from the teacher distribution to obtain an unbiased estimate, fol-

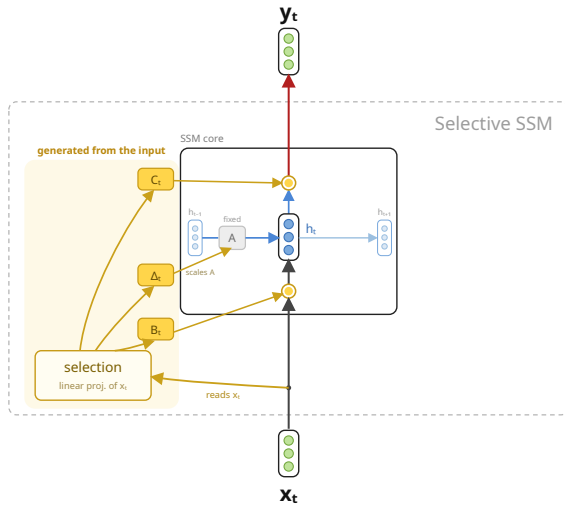


Figure 2: Selective state-space model block diagram, on which Mamba is based; the matrices that manipulate the recurrent state and the input are a function of the input X_t itself, allowing it to selectively accept or discard information.

lowing the sparse-logit method of Anshumann et al. [21]. Each of the three base models is then specialized with the same two-stage procedure used for the baselines, continual pretraining on the domain text followed by fine-tuning on the task.

Two of the findings went against expectation. Distillation did not improve on plain pretraining, most likely because of how severe the truncation is: keeping only 40 tokens leaves the distillation signal only marginally informative, a regime in which, as the same reference reports, sparse distillation is barely better than plain cross-entropy. For the same reason top-k and random sampling did not reproduce the ordering reported in the literature. On the downstream tasks, however, the outcome is more encouraging. On the multiple-choice benchmarks BabyMamba reaches the same class as the best BabyLlama models, of equal size but different architecture, while the larger Mamba and MobileLLM models keep a clear advantage (Figure 3). A 50 million parameters Mamba trained from scratch can therefore be specialized on a narrow domain and hold its own against a model of the same size built on a different architecture.

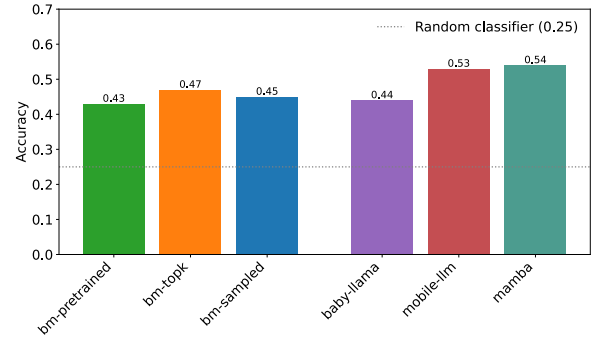


Figure 3: Accuracy of the best BabyMamba models on the wikidoc-multi benchmark, compared with the best baselines; the best BabyMamba model is the one trained with the top-k distillation technique: it exceeds the performance of BabyLlama (a Transformer of the same size) and it loses only a few points against models twice its size.

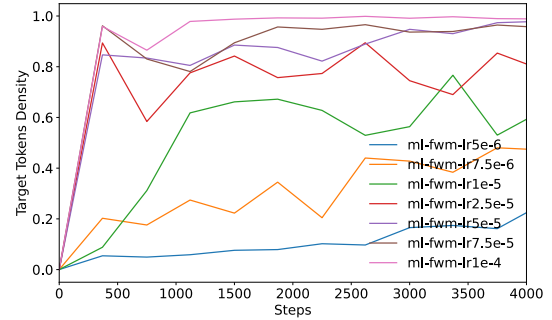


Figure 4: Sum of the probability assigned to each of the possible answers during the fine-tuning phase for different learning rates of MobileLLM on the wikidoc-multi dataset.

5. Technical notes

One methodological point deserves emphasis. All accuracies are computed with the standard tool `lm-evaluation-harness` [22], which for closed-form tasks does not read the text the model generates but instead compares the probabilities it assigns to the valid answers and selects the most probable one. This scores a model as correct whenever it ranks the right option highest, even when the text it would actually generate is not in the required format. Because the two can diverge, the generated output was also inspected throughout this work, to confirm that the reported accuracy reflects answers the model really produces rather than an artifact of

the metric.

Figure 4 shows that at the start the base model assigns very low probabilities to the possible answers of the task (A, B, C and D), and only after a few steps of fine-tuning with high learning rates these probabilities reach a critical mass that makes the model actually output one of the chosen tokens.

6. Conclusions

This work explored a frontier opposite to the prevailing one, asking how little capability is lost as a language model is made small enough to run on an edge device. The argument that attention is ill-suited to small, power-efficient accelerators motivated the choice of Mamba, a state-space architecture with sub-quadratic complexity. To measure the focused competence that small models are actually meant to provide, four question-answering benchmarks were built over two narrow domains, in a balanced yes/no format and a multiple-choice one. The existing small models were adapted to them through direct fine-tuning and through a two-stage continual-pretraining procedure, and BabyMamba, a 50M parameters Mamba trained from scratch with pretraining and offline distillation, was built and specialized in the same way. On the multiple-choice benchmarks BabyMamba performed in line with BabyLlama, a model of equal size, confirming that the architecture can be pushed well below the smallest size of the original work and still be specialized on a narrow domain; the binary benchmarks, by contrast, stayed difficult for every model tested, including the much larger Llama 3 3B Instruct. Several directions remain open. The clearest one concerns the data. The amount and quality of domain-specific text available, and sources such as WikiDoc and Investopedia are limited, so collecting more and cleaner corpora would help the most. The training of BabyMamba can also be improved. The distillation kept only 40 tokens of the teacher distribution, and storing more of it is the natural next step. The final goal, however, is deployment: running a model such as BabyMamba on the embedded hardware described at the start, and measuring its accuracy, latency, and energy use on real devices.

References

- [1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.
- [2] Hoffmann et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.
- [3] Samsi et al. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, 2023.
- [4] STMicroelectronics. STM32N6 series: The first STM32 MCU with ST Neural-ART accelerator NPU. Product page, 2025. URL <https://www.st.com/en/microcontrollers-microprocessors/stm32n6-series.html>.
- [5] GreenWaves Technologies. GAP9: Ultra-low-power parallel RISC-V processor for the edge. Product brief, 2023. URL https://greenwaves-technologies.com/gap9_processor/.
- [6] Vaswani et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [7] Sze et al. Efficient processing of deep neural networks: A tutorial and survey. In *Proceedings of the IEEE*, 2017.
- [8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL <https://arxiv.org/abs/2312.00752>.
- [9] Clark et al. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*:

- Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019.
- [10] Hendrycks et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.
- [11] Inar Timiryasov and Jean-Loup Tastet. 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*. Association for Computational Linguistics, 2023.
- [12] Zechun Liu et al. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2024. URL <https://proceedings.mlr.press/v235/liu24ce.html>.
- [13] medalpaca. Medical meadow wikidoc. https://huggingface.co/datasets/medalpaca/medical_meadow_wikidoc, 2023. Accessed: 2026-05-18.
- [14] FinLang. Investopedia instruction tuning dataset. <https://huggingface.co/datasets/FinLang/investopedia-instruction-tuning-dataset>, 2024. Accessed: 2026-05-18.
- [15] Conneau et al. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020. Association for Computational Linguistics.
- [16] Wikimedia Foundation. Wikipedia dataset: 20231101.en. <https://huggingface.co/datasets/wikimedia/wikipedia/viewer/20231101.en>, 2023. English Wikipedia dump processed for Hugging Face datasets. Accessed: 2026-05-18.
- [17] OpenAI. Gpt-5 system card. System card, OpenAI, 8 2026. URL <https://cdn.openai.com/gpt-5-system-card.pdf>.
- [18] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.
- [19] Meta AI. The llama 3 herd of models, 2024. URL <https://ai.meta.com/research/publications/the-llama-3-herd-of-models/>.
- [20] Black et al. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, virtual+Dublin, 2022. Association for Computational Linguistics.
- [21] Anshumann et al. Sparse logit sampling: Accelerating knowledge distillation in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025.
- [22] Gao et al. The language model evaluation harness, 2024. URL <https://zenodo.org/records/12608602>.