E-Tamba: Efficient Transformer-Mamba Layer Transplantation

Anonymous Author(s)
Affiliation
Address
email

Abstract

With the growing popularity of Transformer (Vaswani et al., 2017) and State Space Models (SSMs), hybrid designs like Jamba (Lieber et al., 2024) and Recurrent-Gemma (Botev et al., 2024) have gained significant attention for their abilities to integrate the long-context processing strengths of Transformers with the low-memory demands of SSMs. However, most hybrid models require extensive pre-training, making them inaccessible to researchers with limited resources who want to experiment with different model architectures. To address this challenge, we introduce E-Tamba, a novel method for constructing hybrid models through only fine-tuning pre-trained Transformer and SSM models. Using layer-wise importance analysis, E-Tamba-1.1B replaces the non-critical upper Transformer layers of Pythia-1.4B (Biderman et al., 2023) with key layers from Mamba-1.4B (Gu and Dao, 2023). Following only 0.9B tokens of fine-tuning, E-Tamba-1.1B delivers excellent results in perplexity scores and various NLP downstream tasks. Additionally, it achieves a 3X reduction in inference memory compared to the baseline Pythia-1.4B, while offering superior long-context retrieval capabilities over Mamba-1.4B. Our code is available at `https://anonymous.4open.science/r/Transformer_Mamba_Transplantation-FFBE`.

1 Introduction

In recent years, Transformer-based Large Language Models (LLMs) have achieved significant breakthroughs, particularly as their scale has expanded significantly (Bender et al., 2021). However, this growth has come with substantial resource costs, especially regarding memory. Due to the nature of the attention mechanism, Transformers are known for their linear inference and quadratic training costs, which place heavy demands on hardware.

Conversely, State Space Models (SSMs), such as Mamba, have emerged as promising alternatives due to their linear training and constant inference costs. While more efficient, SSMs come with trade-offs. Recent research has highlighted Mamba's limitations in long-context copying and retrieval tasks (Jelassi et al., 2024), likely due to its fixed-size hidden state. As a result, there is increasing interest in hybrid models that integrate the strengths of both approaches: the superior long-context capabilities of Transformers and the memory efficiency of Mamba. However, the development of powerful hybrid models typically requires extensive pre-training on massive datasets (De et al., 2024), making them inaccessible for researchers with limited resources who wish to experiment with different hybrid model architectures.

To make such hybrid models accessible to the broader research community, we introduce E-Tamba, a novel approach that achieves Transformer-Mamba hybrid architecture based only on fine-tuning. E-Tamba is built through a series of key steps. First, we perform a layer-wise importance analysis, identifying non-critical Transformer layers and critical Mamba layers by measuring the tokens' average hidden state distance between different layers. Layers with larger distances from other layers are deemed more important. We then replace the non-critical Transformer layers with the more efficient Mamba layers. Finally, we conduct full-parameter fine-tuning on the merged model using the regular cross-entropy loss.

Through this fine-tuning process, E-Tamba-1.1B, based on Pythia-1.4B and Mamba-1.4B, demonstrates outstanding language modeling capabilities. E-Tamba-1.1B outperforms Pythia-1.4B by 38% and Mamba-1.4B by 33% in terms of perplexity. Moreover, E-Tamba-1.1B matches Pythia-1.4B's performance on various NLP downstream tasks while exhibiting nearly 2X the long-context retrieval ability of Mamba-1.4B. Finally, system performance analysis shows that E-Tamba reduces inference memory usage by 3X compared to the Transformer-based Pythia-1.4B.

In summary, our contributions are as follows:

- We introduce a novel layer importance analysis and transplantation method, enabling the creation of Transformer-Mamba hybrid models through fine-tuning alone.
- We present E-Tamba-1.1B, a hybrid model based on Pythia-1.4B and Mamba-1.4B, which delivers exceptional downstream NLP and system performance, offering a middle-ground model solution between Transformer and Mamba.

2 Related Work

Transformer-SSM hybrid models have gained significant attention in recent research. Jamba (Lieber et al., 2024) introduces a pre-trained hybrid model that vertically stacks Jamba layers, interleaving attention and Mamba layers in a `1:7` ratio. Zamba (Glorioso et al., 2024) offers a novel architecture that employs a global shared self-attention layer to optimize memory efficiency. Similarly, Griffin (De et al., 2024) proposes an innovative attention and gated linear recurrent block, achieving comparable performance of Llama-2 (Touvron et al., 2023) but requires fewer training tokens.

In contrast, the strategy of replacing specific model layers with layers from other models remains under-explored. BERT-of-Theseus (Xu et al., 2020) introduces a distilled version of BERT (Devlin, 2018), in which every two BERT layers are replaced with a reinitialized BERT layer. During training, a Bernoulli random variable determines the forward path between the original layers and the new one. Sajjad et al. (2023) finds that up to 40% of BERT layers can be removed while retaining 98% of the original performance. For decoder-only models, Gromov et al. (2024) demonstrates that the upper Transformer layers contribute minimally to overall performance and can be pruned, with the model compensating for their removal by fine-tuning.

3 Methodologies

In this section, we first present our analysis of which layers in Transformer and SSM models are critical. We then explain how these insights inform the architecture design of E-Tamba-1.1B. Finally, we describe the efficient layer transplantation and fine-tuning process to train this hybrid model. Throughout the paper, we refer to layers as the stacked components of modern deep learning models. For instance, Pythia-1.4B contains 24 layers (Biderman et al., 2023), while Mamba-1.4B has 48 layers (Gu and Dao, 2023).

3.1 Layers Importance Analysis

We evaluate the significance of a model's different layers using the layers' pairwise distance method inspired by Gromov et al. (2024). However, we introduce a crucial improvement to the original algorithm: instead of calculating the distance between two layers based solely on the hidden state difference of the final token in a sequence, we compute the average distance across the hidden states of all tokens. This enhancement is significant because the first few tokens in a sequence typically establish the context and often exhibit higher perplexities than the later ones. Thus, we hypothesize that incorporating all tokens offers a more comprehensive measure of a layer's importance in a language model.

$$\bar{d}(x^{(l)}, x^{(l+n)}) \equiv \frac{1}{m} \sum_{T=1}^{m} \frac{1}{\pi} \arccos \left(\frac{x_T^{(l)} \cdot x_T^{(l+n)}}{\|x_T^{(l)}\| \|x_T^{(l+n)}\|} \right) \tag{1}$$

To measure the distance between layer l and layer $l+n$ (the n^{th} layer after layer l), we use the distance formula presented in Equation 1. In this formula, x represents the hidden state of a sequence across different model layers, m denotes the number of the tokens in the sequence, and T refers to the T^{th} token currently being iterated over in the sentence. For example, $x_T^{(l)} \cdot x_T^{(l+n)}$ represents the dot product distance between the hidden states of the T^{th} token in layer l and layer $l+n$. At a high level, layers with greater distances from later layers are considered more important as they induce more substantial changes to the hidden states of the tokens.

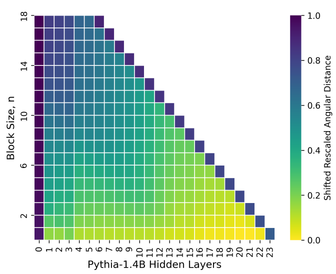

Figure 1: Pythia-1.4B

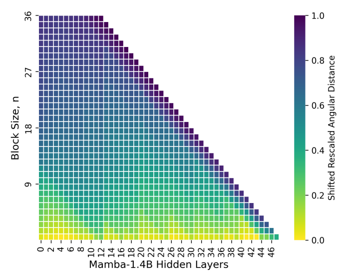

Figure 2: Mamba-1.4B

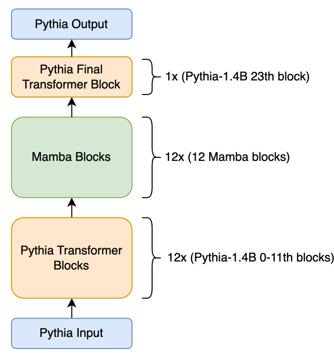

Figure 3: E-Tamba Architecture

As illustrated in Figures 1 and 2, each block on the graph, with an x-axis coordinate l and y-axis coordinate n, represents the average token hidden state distance between layer l and the layer $l+n$ (the n^{th} layer following l) using the formula outlined in Equation 1. Generally, darker blocks indicate larger distances between the respective layer pairs, signifying greater importance of the earlier layer in the pair. The experiment uses a subset of the C4 validation dataset (`en/c4-validation.00000-of-00008`) with an input sequence length of 1024.

Specifically, by examining the x-coordinates and analyzing the layers above them, we observe that the lower $0-11^{\text{th}}$ Pythia-1.4B layers exhibit darker colors, indicating greater importance. Additionally, the final layer of Pythia-1.4B stands out, as evidenced by the leftward diagonal pattern in the figure. In contrast, Mamba's layers display more complex dynamics, with specific groups of layers showing heightened importance. This is reflected by the interleaving darker color pattern along the x-axis. In particular, layers $3-14^{\text{th}}$, $14-25^{\text{th}}$, $25-36^{\text{th}}$, and $36-47^{\text{th}}$ exhibit this interleaving pattern. These patterns indicate four potential groups of Mamba layers that could be prioritized when replacing non-critical Transformer layers with important Mamba layers.

3.2 Layers Transplantation & Fine-tuning

In summary, we identify the $0-11^{\text{th}}$ and the final Transformer layer of Pythia-1.4B as critical, while the intermediate layers are deemed as non-crucial and can be replaced by lightweight Mamba layers. For Mamba, we select four key groups of layers: $3-14^{\text{th}}$, $14-25^{\text{th}}$, $25-36^{\text{th}}$, and $36-47^{\text{th}}$, which can substitute for the non-critical Transformer layers. The resulting merged E-Tamba-1.1B architecture is shown in Figure 3.

To determine the best-performing group of Mamba layers, we conduct extensive experiments to evaluate their respective effectiveness in replacing the non-essential Transformer layers. Specifically, we test the performance of each of the four candidate Mamba layer groups in scenarios where they replace the previously identified non-critical $12-22^{\text{th}}$ Pythia-1.4B layers. We also explore two additional conditions: one where an untrained (reinitialized) group of 12 Mamba layers is used, and another where no Mamba layers at all (only pruning intermediate Pythia layers). For each scenario, we conduct exploratory full-parameter language modeling fine-tuning, using validation perplexity score as the metrics for comparison. The exploratory training uses a subset of C4 training subset (`en/c4-train.00000-of-01024`) with a sequence length of 1024.

Table 1: Layer Transplantation Ablation Studies

Mamba Layers	C4-val (ppl)
None (deleted)	31.42
Non-pre-trained	29.39
3-14th	29.21
14-25th	29.28
25-36th	**28.68**
36-48th	31.42

Table 2: E-Tamba's end-to-end performance

Model	C4-val (ppl)	Lambada (acc)	Winogrande (acc)	Memory (MiB)
Pythia-1.4B	19.87	61.7	57.2	9114
Mamba-1.4B	18.83	**64.9**	**61.5**	3100
E-Tamba-1.1B	**12.48**	60.6	56.5	**3082**

As shown in Table 1, the configuration using Mamba $25-36^{\text{th}}$ layers outperforms other candidate groups. Therefore, we finalize E-Tamba's final architecture with $0-11^{\text{th}}$ Pythia-1.4B layers, followed by $25-36^{\text{th}}$ Mamba layers, and conclude with the final Pythia-1.4B layer. Using this architecture, we fine-tune E-Tamba-1.1B on three subsets of C4's train split (`en/c4-train.00000-of-01024, en/c4-train.00001-of-01024, en/c4-train.00002-of-01024]`), with a total of 0.9B tokens. We perform a search over `[1e-3, 1e-4, 1e-5]` constant learning rate, ultimately selecting `1e-4`. The experiments are conducted with a global batch size of `72`, one epoch, and a warm-up ratio of `0.03`, on a single NVIDIA A100 GPU.

4 Experiments

The experiments section is organized as follows: First, we present the end-to-end performance of the fine-tuned E-Tamba-1.1B, assessing both perplexity and various NLP evaluation benchmarks. Next, we highlight E-Tamba-1.1B's advantages in GPU inference memory usage. Finally, we explore how E-Tamba-1.1B addresses Mamba's limitations in long-context retrieval tasks. Throughout our experiments, we have excluded comparisons with other hybrid architectures, such as RecurrentGemma (Botev et al., 2024), due to the unavailability of comparably sized models at the time of writing.

4.1 Language Modeling Capabilities

We begin by evaluating the language modeling capabilities of E-Tamba-1.1B on a subset of the C4 validation split (`en/c4-validation.00000-of-00008`). As shown in Table 2, with a test sequence length of 1024, E-Tamba-1.1B achieves significantly lower perplexity scores than both baseline models, despite having the fewest parameters. These findings indicate that fine-tuning pre-trained Transformer and SSM models offers a promising approach for building robust hybrid architecture.

4.2 Downstream Tasks

Given that perplexities are often heavily influenced by the accuracy of generating the first token in a sequence, a factor determined by the unigram distribution in the training data, we further assess E-Tamba-1.1B on two widely used downstream NLP tasks to evaluate its broader performance. Specifically, we use the Lambada (Paperno et al., 2016) and WinoGrande (Sakaguchi et al., 2021) benchmarks to measure E-Tamba-1.1B's commonsense reasoning abilities. For Lambada, we reference the performance of Pythia-1.4B and Mamba-1.4B reported by Gu and Dao (2023). For WinoGrande, we reproduce these two models' results on the `winogrande_xl` dataset for consistency.

As shown in Table 2, although E-Tamba-1.1B does not achieve the top performance due to its smallest parameter count, it delivers competitive results on both challenging NLP benchmarks. Notably, E-Tamba-1.1B's performance closely matches that of Pythia-1.4B across both tasks. With this downstream task performance validation and considering its significant memory efficiency, E-Tamba-1.1B emerges as a strong alternative to traditional Transformer architectures.

4.3 Inference Memory

To recap, a key objective of hybrid models is to integrate the memory efficiency of Mamba with Transformer models, which have historically been constrained by the attention mechanism's memory demands. To evaluate this, we measure GPU memory usage during long-context inference with a batch size of 1 and a sequence length of 4096. As shown in Table 2, E-Tamba-1.1B achieves nearly 3X memory savings compared to Transformer models. Even with fewer parameters, these substantial memory savings highlight E-Tamba's potential as a balanced solution between Transformer and SSM architectures in memory-limited situations.

4.4 Long-Context Performance

In addition to memory efficiency, another key objective of hybrid models like E-Tamba is to leverage the Transformer's strength in handling long-context retrieval tasks, an area where Mamba has been shown to under-perform (Jelassi et al., 2024). To evaluate this, we assess the models' performance on two tasks: long-context copying and phone book retrieval.

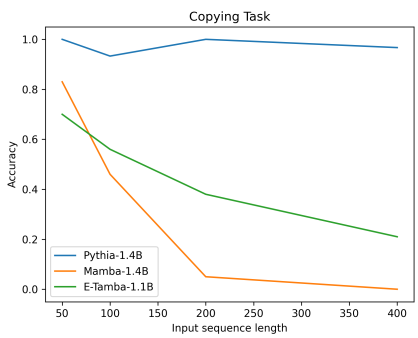

Figure 4: Copying Task

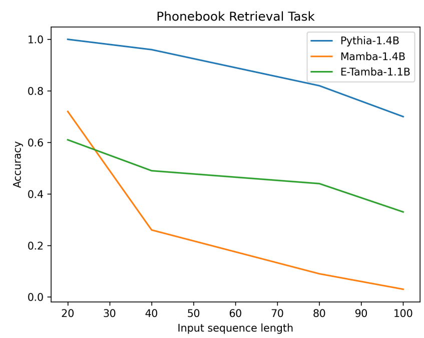

Figure 5: Phonebook Retrieval Task

For the copying task, the input consists of a sequence of tokens repeated twice, followed by the first token from the sequence. We evaluate the models' ability to copy across various input sequence lengths, considering the task successful only when the copied sequence exactly matches the input. For the phone book retrieval task, we manually generate a test dataset of `<name, number>` pairs, formatted as `<Jack, 123-456-7890>`. The input consists of the entire phone book concatenated with a randomly selected name, and performance is evaluated across different phone book sizes. A test is deemed successful when the retrieved phone number exactly matches the ground truth. For both copying and phonebook experiments, the final results are averaged over 30 test cases.

As shown in Figures 4.4 and 4.4, E-Tamba-1.1B exhibits significant performance improvements over the Mamba model on both long-context tasks. While Mamba can maintain strong performance with short sequences, its accuracy declines sharply with longer inputs, likely due to its fixed-size hidden state. In contrast, E-Tamba-1.1B continues to perform well on long inputs, despite performing worse than Pythia-1.4B because of fewer parameters. This highlights the effectiveness of E-Tamba in overcoming Mamba's limitations in long-context tasks.

5 Conclusion

In this paper, we introduce E-Tamba, a novel approach to creating a Transformer-Mamba hybrid through layer transplantation and fine-tuning. With only 0.9B tokens of fine-tuning, E-Tamba-1.1B achieves competitive language modeling perplexities and various downstream NLP task performances. Moreover, E-Tamba-1.1B reduces inference memory usage by 3X compared to the Transformer-based Pythia-1.4B, while significantly outperforming Mamba-1.4B in long-context retrieval tasks. Therefore, the E-Tamba fine-tuning method proves as an effective approach for resource-limited researchers to experiment with different hybrid model architectures. Meanwhile, the E-Tamba-1.1B model itself becomes a strong middle-ground solution to combine Transformer's long context and Mamba's memory-saving abilities. We hope this work will inspire further exploration of resource-efficient hybrid models.

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

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.

Aleksandar Botev, Soham De, Samuel L Smith, Anushan Fernando, George-Cristian Muraru, Ruba Haroun, Leonard Berrada, Razvan Pascanu, Pier Giuseppe Sessa, Robert Dadashi, et al. 2024. Recurrentgemma: Moving past transformers for efficient open language models. *arXiv preprint arXiv:2404.07839*.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. 2024. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. 2024. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*.