Non-Expanding Gated Continued Fraction Architecture for Feedforward Layers in Language Models

Amit Dhurandhar
IBM Research
adhuran@us.ibm.com

Vijil Chenthamarakshan
IBM Research
ecvijil@us.ibm.com

Karthikeyan Natesan Ramamurthy
IBM Research
knatesa@us.ibm.com

Dennis Wei
IBM Research
dwei@us.ibm.com

Rahul Nair
IBM Research
rahul.nair@ie.ibm.com

Abstract

Recent work has shown that continued fraction inspired neural architectures (CoFrGeNets) can be highly parameter efficient yet performant for language modeling. In this work, we revisit continued fraction-based channel mixing with the goal of improving its performance and simplifying integration into standard pre-training protocols for language models. We achieve this by two simple yet critical contributions: i) we propose a novel non-expanding gated continued fraction inspired architectural component to replace feedforward layers in established language model architectures that may use vanilla multilayer perceptrons or gated linear units or (vanilla/gated) experts in mixture-of-experts (MoE) type architectures. ii) We propose a way to re-implement divisions – the non-linearity in these architectures – in terms of logarithms and exponentials. By experimenting on different language modeling architectures such as GPT-2, Llama-3, Mamba-2 and (nano) Llama-MoE we find that our proposed architecture performs similarly or significantly better than the continued fraction architecture proposed in prior work along with having higher throughput. In addition, our re-implementation of divisions makes training of deeper CoFrGeNets stable without the need for incremental training which was proposed in prior work as a means to stabilize training, but added notable complexity to existing large language model training implementations. Code is available at http://github.com/IBM/CoFrNets.

1 Introduction

Large Language Models (LLMs) have taken the world by storm [38, 19] revolutionizing how tasks such as web search, writing, coding, planning and even research are being done. A few neural architectures seem to be lying at the core of these advancements. Transformers [46] are arguably the most leveraged architecture appearing in many popular models such as in Generative Pre-trained Transformers (GPTs) [37], Llama [18] and even in the mixture-of-experts type of architectures [9] which are in vogue currently. State space architectures such as Mamba [19] have gained popularity given that they are more efficient than Transformers especially over long sequences. Diffusion models [40] which leverage Transformers are also gaining traction given their ability to generate sequences in parallel as opposed to auto-regressively. A major and the most parameter heavy component of these architectures is the feedforward network (FFN) which consists of a 2-layer perceptron with the hidden state being expanded by a factor of two or more over the input or embedding dimension. Large number of parameters implies slow and expensive training and inference with a large carbon

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

footprint. Hence, having alternatives that are parameter efficient yet performant is a highly sought after goal [43].

Recently, a new architecture inspired by the mathematical concept of continued fractions CoFrGeNets [14] was proposed for language generation. The main contribution was proposing readily pluggable components for attention and FFNs in transformers that are parameter efficient. Replacing FFNs in dense Transformer models such as GPT2 and Llama led to models that were roughly $\frac{2}{3}$ the size, while replacing both attention and FFNs led to roughly $\frac{1}{2}$ the size with some degradation in performance. A continued fraction can be written as $a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \cdots}}$ (ladder like structure) where, a_ks are complex numbers. In the recent CoFrGeNets and the original CoFrNets [36], which was introduced for supervised settings, the a_ks were replaced by linear functions of the input $x \in \mathbb{R}^p$ by taking the inner product of x with weight vector $w_k \in \mathbb{R}^p$ in each layer k.[1] The reciprocal of the function thus far is applied as a nonlinearity in each layer leading to the following kind of form for a single ladder:

$$w_0^T x + \frac{1}{w_1^T x + \frac{1}{w_2^T x + \cdots}} \tag{1}$$

The w_ks are parameters that can be learned. Here, the input x gets multiplied by the parameter vector in each layer with the non-linearity being a reciprocal. This simplistic architecture has nice properties including universal approximability when enough ladders are ensembled together [36].

In this work, inspired by CoFrGeNets and motivated by the need to have parameter efficient yet performant alternatives to FFNs which contributes the most to model size, we propose a novel non-expanding, gated continued fraction inspired architectural component to replace FFNs. The resulting module can be used as a drop-in replacement for FFNs in standard MLP, GLU, and MoE settings, enabling direct integration into architectures such as GPT-2, Llama-style models, and state-space models. This new variant can have slightly more number of parameters than the component proposed in [14], due to the inclusion of the non-expanding gating, but having similar or significantly better performance across models. Additionally, our variant also has higher throughput, a consequence of improved arithmetic intensity.

One challenge with training continued fraction architectures is that deep ladders can result in slow training and inference as well as training instabilities due to reciprocal calculations. To address the former, in [14] a ladder was cast in terms of *continuants* –a mathematically equivalent formulation (see Section 3)– which reduced the number of divisions from d to 1 for a d depth ladder. However, training instabilities persisted for deep ladders (i.e. $d > 1$) for which the authors introduced an incremental training protocol whereby training starts at a single depth and incrementally larger depths are trained following a dyadic schedule. This however, added non-trivial complexity to standard pre-training recipes for LLMs. To allow for a drop-in replacement that would require no change to standard pre-training recipes, in this work, we propose a simple reformulation that entirely avoids these stability issues. The reformulation involves transforming the division-based nonlinearities using logarithmic and exponential transformations. By avoiding explicit division, we improve gradient flow leading to improved stability.

As such, our contributions in this work are threefold:

- We propose a novel non-expanding gated continued fraction architecture to replace (gated and vanilla) feedforward layers in LLMs (see Figure 1), which offers performance and throughput improvements over previously proposed continued fraction architectures [14].
- We propose a simple reformulation of the division operation, which accounts for the non-linearity in CoFrNets, in terms for logarithms and exponentials to make training stable for deeper ladders. This facilitates drop-in replacement with a broad class of language models without the need to modify pre-training strategies.
- We replace experts in MoE architectures as well as gated FFNs in Mamba-2 with our novel continued fraction architectural component further distinguishing it from prior work. We showcase the broad efficacy of our architectural component by reporting promising results on diverse LLM architectures such as GPT-2, Llama-3, Mamba-2 and Llama-MoE. Notably, the parameter savings for dense models are close to 33% which is already considerable, but for MoEs it reaches even higher to about 50%.

[1] Constant term is subsumed in x.

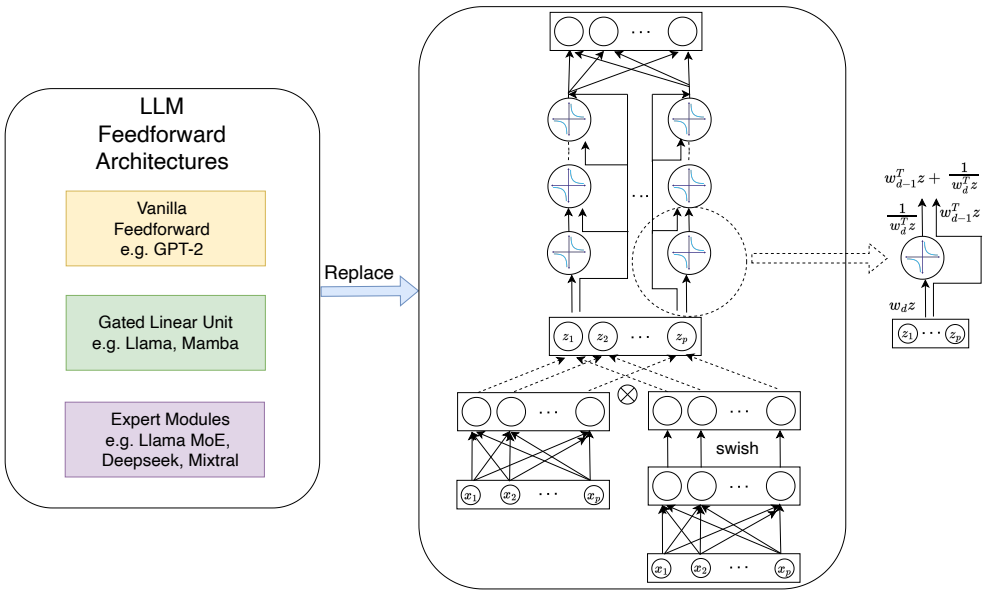

Figure 1: Above we see our continued fraction inspired architecture that can replace feedforward layers in dense as well as MoE architectures. Our architecture performs a *non-expansive* gating of the input that then passes through an (ensemble of) continued fraction ladders. The fact that we do not expand the hidden representation and each layer of our ladders requires only a linear number of parameters we obtain significant parameter savings over typical (gated and vanilla) feedforward layers leading to dense models that are $\approx \frac{2}{3}$ the size and MoEs that are $\approx \frac{1}{2}$ the size. In our implementations we create a different class for CoFrGeNet that has the same api as the FFN class, that is, same parameters are passed and the same size output tensor (equal to embedding size) is returned. The depth and width parameters for CoFrGeNet could be additionally passed or just set in the CoFrGeNet class itself. Hence, no changes are made to the residual stream or other aspects of the original architecture. The input $x_1, ..., x_p$ in the figure is thus simply the output of the attention or SSM layer where the residual stream is added and the representation normalized in the particular block. In other words, nothing else in the model architectures changes and the input and output tensors are both size embedding dimension.

Overall, our contributions retain the lower parameter count and compactness of continued-fraction architectures, while simplifying (stable) training and improving performance in many cases with also improved throughput.

2 Related Work

Modern large language models are predominantly built on the transformer backbone [46], in which each block alternates a self-attention sublayer with a position-wise feedforward network (FFN). The FFN is typically implemented as a multilayer perceptron with a pointwise nonlinearity, typically ReLU [34], GELU [21] or Swish/SiLU [39]. Recently, the gated linear unit family [13] was shown to outperform vanilla MLPs at matched parameter budgets [41] in several settings, motivating the adoption of SwiGLU in the Llama family of models [44, 45, 18]. A parallel line of work is mixture-of-expert (MoE) models, where each token is routed to a small set of experts that themselves are MLPs or gated MLPs [42, 26, 15, 23, 9, 51]. Beyond attention-based models, state-space architectures such as Mamba [19] and Mamba-2 [11] replace quadratic attention with selective recurrences but still rely on MLP- or GLU-style channel mixers. Across all of these designs, the FFN sublayer dominates the parameter count, making it a natural target for architectural redesign.

Because of this reason, few works have explored alternatives to dense MLPs. Kolmogorov–Arnold Networks (KANs) [28] replace fixed activations with learnable univariate functions and have shown promise on small-scale scientific tasks in terms of performance and interpretability, although their

scalability and training cost remain open questions. Many times KANs are 10x slower to train than MLPs of the same size. Structured-matrix approaches such as Monarch [10] factorize dense weights into block-diagonal components to reduce FLOPs and memory, and are not really a novel architecture but a strategy than can be applied across architectures. Polynomial networks [7] introduce multiplicative interactions between linear projections and have been shown effective in vision tasks. These alternatives however, have not been shown to match the empirical performance, scalability and stability of MLPs across varied large language model architectures.

The closest work to ours is the recent work CoFrGeNets [14], which proposed continued fraction architectures for attention and FFNs in dense Transformers. The FFN replacements produced models that were $\frac{2}{3}$ the size and with higher throughput. These replacements were however, not incorporated into other models such as Mamba and MoEs. In this work, we incorporate our component into these architectures as well and show that it should be the preferred alternative as we obtain similar level of parameter savings but much better performance in many cases and even higher throughput.

3 Background and Methods

We first introduce some notation. We denote the input or embedding dimension by p, the number of ladders in an ensemble by L, the sequence length by l and the depth of a ladder by d.

Next, we describe what a continued fraction is. The generalized form for a continued fraction is $a_0 + \frac{b_1}{a_1 + \frac{b_2}{a_2 + \cdots}}$, where a_ks and b_ks can be complex numbers. If none of the a_k or b_k are zero $\forall k \in \mathbb{N}$, then using equivalence transformations [24], one can create simpler equivalent forms where either the $b_k = 1$ or the $a_k = 1\ \forall k \in \mathbb{N}$, with $a_0 = 0$ in the latter form. A more concise way to write these two forms is as follows: i) $a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \cdots}} \equiv a_0 + \frac{1}{a_1 +}\frac{1}{a_2 + \cdots}$ and ii) $\frac{b_1}{1 + \frac{b_2}{1 + \cdots}} \equiv \frac{b_1}{1 +}\frac{b_2}{1 + \cdots}$. Form i) is known as the *canonical form*. One of the nice properties of continued fractions is that they are the best possible rational approximation to real numbers [24, 33].

In this work, we consider continued fractions in canonical form, with partial numerators $b_k = 1$ for $k = 1, \ldots, d$ and depth d. We thus view continued fractions as functions f of the partial denominators, where we separate a_0 from the others and use $a \coloneqq (a_1, \ldots, a_d)$ as a shorthand. Hence,

$$f(a_0, a) = a_0 + \frac{1}{a_1 +}\frac{1}{a_2 +} \cdots \frac{1}{a_{d-1} +}\frac{1}{a_d} = a_0 + \tilde{f}(a), \tag{2}$$

where we also define $\tilde{f}(a)$ as the "fractional part" of $f(a_0, a)$. Another way of representing a continued fraction is in terms of *continuants*, which we describe next. The continued fraction in (2) can be expressed as the following ratio of polynomials K_{d+1} and K_d,

$$f(a_0, a) = \frac{K_{d+1}(a_0, \ldots, a_d)}{K_d(a_1, \ldots, a_d)}. \tag{3}$$

Polynomials K_d, K_{d+1} are part of a sequence of polynomials K_k, $k = 0, 1, \ldots$, known as *continuants*. They satisfy the recursion

$$K_0 = 1, \qquad K_1(a_d) = a_d, \tag{4}$$

$$\begin{aligned} K_k(a_{d-k+1}, \ldots, a_d) &= a_{d-k+1} K_{k-1}(a_{d-k+2}, \ldots, a_d) \\ &\quad + K_{k-2}(a_{d-k+3}, \ldots, a_d). \end{aligned} \tag{5}$$

Using (5), (3) can also be written as $f(a_0, a) = a_0 + \frac{K_{d-1}(a_2, \ldots, a_d)}{K_d(a_1, \ldots, a_d)}$ and hence,

$$\tilde{f}(a) = \frac{K_{d-1}(a_2, \ldots, a_d)}{K_d(a_1, \ldots, a_d)}. \tag{6}$$

This formalism as shown in previous work [14] significantly enhances training and inference speed over standard layerwise forward and backward passes done in [36], since the number of divisions for a depth d ladder reduces from d to 1 and divisions are significantly more expensive than multiplications or additions on modern hardware. Nonetheless, it still causes some training instability especially for deep ladders (even $d \geq 2$). This resulted in the proposal for incremental training requiring non-trivial

modifications to language model pre-training procedures. Below we propose a simple alternative that stabilizes training and absolves the need for these training modifications.

We design neural networks inspired by continued fractions by letting the partial denominators a_k be affine functions of an input x, $a_k = w_k x$, where w_k is a row vector and a 1 is prepended to the elements of x so that the corresponding coefficient w_{k0} is the intercept or *bias* term. We will often refer to a continued fraction with $a_k = w_k x$ as a (CoFrNet) *ladder*, and we will also construct ensembles of such ladders.

3.1 Continued Fraction Architecture for FFNs

In Figure 1 we observe our continued fraction architecture for FFNs. We pass the p-variate input to two linear layers where the output of one of them (bottom right portion in the figure) is gated by a Swish non-linearity. Then the outputs of the two layers are element-wise multiplied producing a (potentially) sparse representation $(z_1, ..., z_p)$. This representation is then input to L continued fraction ladders that produce an L-variate output. Note that one could have an arbitrary number of ladders in the ensemble and one projects to p dimensions using the linear layer. The input to the ladders is a gated non-expanded (i.e. the expansion factor $\alpha = 1$) representation. The purpose of expansion and sparsification is typically understood to unpack information in a representation [12]. CoFrNets being a compact yet expressive function class are able to extract information with just partial unpacking, where they benefit from sparsification but do not necessarily need the expansion. Not performing expansion produces significant parameter savings as seen in the experiments. In particular, the number of parameters is $Lp(d+1) + 2p^2$ as opposed to the typical $2\alpha p^2$ for vanilla FFNs and $3\alpha p^2$ for gated FFNs, where $\alpha \geq 2$.

The continued fraction architecture suggested in [14] and depicted in Figure 3 in the appendix consists of two continued fraction ladder ensembles, which directly ingest the input. The output of the ladders is element-wise multiplied for better representation power. Thus no gating of the input was performed and two rather than one ensemble of continued fraction ladders was leveraged. The number of parameters for this variant was $2Lp(d+1)$.

Although our new architecture has a quadratic dependence on p, the non-expansion of the intermediate (hidden) FFN layer leads to significant parameter savings in practice as seen for the different models in the experiments. Additionally, as seen in these experiments the number of parameters is only slightly more or similar to the previous continued fraction architecture. More interestingly, even though our architecture may sometimes have more parameters than the previous one, since we have only one ensemble rather than two the *arithmetic intensity is higher* for our model as the input does not have to be read twice and processed separately by two ensembles. The gating is implemented so that the input is projected by a linear map to a representation twice its size and then Swish is applied to the latter half of this representation followed by elementwise multiplication with the prior half needing the input to be read just once. This leads to higher throughput as seen in the experiments, thus mitigating the adverse effects of (the slightly) increased parameter count.

3.2 Replacing Divisions

The continuant formulation described in the previous section does reduce the number of divisions significantly as a d-depth ladder which would normally require d divisions now requires only one. Nonetheless, the number of divisions scale with the size of the ensemble requiring L-divisions for L ladders. More importantly though even a single division can cause poles or abrupt discontinuities that can hurt gradient based optimization. This was observed in [14] and hence for higher depths incremental training was introduced where one would freeze deeper layers and start training them only after the shallower layers were trained reasonably well for the training to be stable.

We propose here a pytorch favorable replacement for the division operation which is performed in CoFrGeNets. In particular, the non-linearity in CoFrGeNets is division which is the ratio of two continuants $\frac{K_{d-1}(w_2 x, \ldots, w_d x)}{K_d(w_1 x, \ldots, w_d x)}$ for a d depth ladder. To avoid poles the denominator was capped from below to a small $\epsilon > 0$ producing the following non-linearity:

$$\operatorname{sgn}\left(K_d(w_1 x, \ldots, w_d x)\right) \frac{K_{d-1}(w_2 x, \ldots, w_d x)}{\max(|K_d(w_1 x, \ldots, w_d x)|, \epsilon)} \tag{7}$$

where $|.|$ denotes absolute value and $\mathrm{sgn}(.)$ the sign function which returns a value of $+1$ or -1. We rewrite (7) as follows:

$$\mathrm{sgn}(K_{d-1}(w_2x, \ldots, w_dx))\,\mathrm{sgn}\left(K_d(w_1x, \ldots, w_dx)\right) \cdot \\ \exp(\log(|K_{d-1}(w_2x, \ldots, w_dx)|) - \log(\max(|K_d(w_1x, \ldots, w_dx)|, \epsilon)) \tag{8}$$

This simple re-implementation absolves the need for incremental training. The logarithms and exponential smoothen out the division operation as logarithms compress the dynamic range of values, which reduces the skew in gradients and helps the optimizer move more smoothly towards a minima. This is confirmed in our experiments where CoFrGeNet-FNG is effectively optimized by optimizers such as AdamW for larger (> 1) depth d and width L of the continued fraction ensemble.

4 Experiments

4.1 Setup

Pre-Training Datasets and Models: We pre-train GPT2-xl and its variants on OpenWebText (OWT) [17] and the GneissWeb 35B (GW) [16] datasets. The remaining architectures, that is, Llama-3, Mamba-2 and Llama-MoE (4 experts, 1 activated) are pre-trained on the 2T data mix used to train Docling [2] models. This data mix contains web (DCLM2, DCLM3Plus [27]), multilingual (FineWeb-2-edu [29]), code (Starcoder, stack-edu [1]), math (Finemath [1], Infiwebmath [20], opc-fineweb-math-corpus [22]) and synthetic data (Cosmopedia [3]), which are heavily used to train models for diverse document understanding. The mixture proportions for the different datasets are as follows: `dclm3plus` $\rightarrow$ 0.5, `dclm2` $\rightarrow$ 0.24, `cosmopedia_v2` $\rightarrow$ 0.06, `fineweb_math_corpus` $\rightarrow$ 0.01, `finemath-4plus` $\rightarrow$ 0.015, `infiwebmath-4plus` $\rightarrow$ 0.01, `finemath-3plus` $\rightarrow$ 0.015, `infiwebmath-3plus` $\rightarrow$ 0.005, `starcoder` $\rightarrow$ 0.01, `stack-edu` $\rightarrow$ 0.17 and `fineweb-2-edu` $\rightarrow$ 0.01.

Baselines: We compare our non-expansive gated architecture CoFrGeNet-FNG with i) the original models with the existing FFNs, ii) CoFrGeNet-F which corresponds to replacing FFNs with the architecture proposed in [14] and depicted in Figure 3 and iii) the original models where non-expansive gating (NG) is applied for the up projection in the FFNs, but where the down projection is a typical (although non-expanded) linear layer (no CoFrNet) seen in the original models. This baseline checks the value of having our continued fraction architecture in the down projection.

Evaluations: For GPT2-xl and its variants we report perplexity on Penn Tree Bank (PTB) [30], Wikitext2 [31], Wikitext103 [31], Lambada [35], AgNews [50] and One Billion Words (LM1B) [5] datasets. We use a stride of 512 for wikitext2, wikitext103 as recommended in these works. For all the other datasets, we use a stride of 256. We then fine tune our models on GLUE [47] (classification) tasks and compare accuracies as done in previous works [40, 14]. We average results over five runs. The other models and their variants we evaluate on openbookqa [32], piqa [4], arc-easy [8], winogrande [25], hellaswag [49], lambada open AI [37], boolq [6] and sciq [48] which cover open domain Q&A, reasoning and text understanding tasks. For all the models we also report throughput, train time and floating point operations (FLOPS). For the Llama model we also report NF4 post training quantization (PTQ) results using bitsandbytes (`https://huggingface.co/docs/bitsandbytes/en/index`).

We also compare the effect of our division replacement with the previous implementation that performs divisions with and without incremental training. This experiment we report on GPT2-xl as the results are representative for other models which for all the reported experiments leveraged our division re-implementation from section 3.2. We employ a dyadic training schedule for incremental training as recommended in [14].

Parameter Settings: We use the recommended settings for pre-training GPT2-xl in `https://github.com/karpathy/nanoGPT` where, the learning rate is 6×10^{-4}, weight decay is 0.1, there is no dropout and the maximum iterations is $600K$.

For fine tuning the GPT2-xl model the learning rate is 0.25×10^{-4}, batch size is 64 and there is no dropout. This is the same for all the baselines. For our models the learning rate was 0.125×10^{-4} with other parameters being the same. These learning rates produced the best results for the respective models.

We pre-train the other models for about 2M iterations. The initial learning rate is 3×10^{-4} and follows an annealing schedule with no dropout.

For the continued fraction architectures we set $\epsilon = 0.1$. We experiment with d equal to $1, 2, 5, 8$ with the widths (i.e. number of ladders in each ensemble) taking the same values as well when replacing FFNs.

All experiments were performed with a batch size of 16 using the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.95$) and a weight decay of 0.1. More details of each architecture are given in Table 1.

Table 1: Below we see architectural details such as number of parameters, number of layers, embedding dimension (p) and expansion factor (α) of the different (original) models.

Model	Size	Layers	p	α
GPT2-xl	1.5B	48	1600	4
Llama	3.2B	24	3072	2.67
Mamba-2	3.2B	9	4096	3.5
Llama-MoE	403M	30	576	2.67

Table 2: Downstream task accuracies (best results bolded) for GPT2-xl and its variants on GLUE benchmark after finetuning. The first column is the pre-training dataset. Standard deviations are reported in Table 10 in the appendix. The CoFrGeNet variants perform similarly (best) in this case.

Data	Model	MNLI	QQP	QNLI	SST2	COLA	MRPC	RTE	WNLI
OWT	GPT2-xl (1.5B)	86.89	88.93	91.35	93.56	81.78	79.83	60.27	58.28
	CoFrGeNet-F (972M)	**87.24**	**89.95**	**91.87**	**94.12**	**82.57**	**80.17**	**61.36**	**58.30**
	CoFrGeNet-FNG (985M)	**87.26**	**89.95**	**91.89**	**94.16**	**82.59**	**80.21**	**61.35**	**58.30**
	GPT2-xl-NG (985M)	81.17	81.35	83.94	87.22	75.99	74.37	56.42	58.28
GW	GPT2-xl (1.5B)	78.28	86.83	**82.93**	91.82	74.18	77.72	60.19	**58.33**
	CoFrGeNet-F (972M)	**79.23**	**87.24**	82.69	**92.38**	**74.79**	**78.04**	**61.37**	**58.33**
	CoFrGeNet-FNG (985M)	**79.62**	**87.26**	82.73	**92.36**	**74.83**	**78.01**	**61.35**	**58.33**
	GPT2-xl-NG (985M)	74.83	84.74	77.82	83.94	68.93	71.83	57.85	**58.33**

Table 3: Perplexities of GPT2-xl and its variants with best results bolded.

Data	Model	PTB	Wikitxt2	Lbda	AgNews	Lm1b	Wikitxt103
OWT	GPT2-xl (1.5B)	30.12	18.30	8.66	37.13	41.20	17.50
	CoFrGeNet-F (972M)	29.94	**17.09**	8.15	35.72	**40.11**	16.17
	CoFrGeNet-FNG (985M)	**29.87**	17.14	**8.11**	**35.69**	40.17	**16.13**
	GPT2-xl-NG (985M)	35.98	27.74	16.35	50.75	51.95	29.29
GW	GPT2-xl (1.5B)	**29.07**	19.12	31.78	45.62	52.36	18.93
	CoFrGeNet-F (972M)	29.83	**18.08**	**30.55**	41.77	**46.59**	18.11
	CoFrGeNet-FNG (985M)	29.68	18.10	**30.55**	**41.65**	46.86	**18.06**
	GPT2-xl-NG (985M)	34.26	27.38	39.44	61.81	58.64	29.02

Hardware: The GPT2 models were pre-trained on 16 H100 GPUs leveraging just distributed data parallel (ddp) training as done previously. Fine tuning was done using a single A100 GPU for each model. The Llama-3 and Mamba-2 models were pre-trained using 128 H100 GPUs with fully sharded distributed data parallel (fsdp) training. The Llama-MoE models were pre-trained using 64 H100 GPUs with fsdp.

Table 4: Zero-shot accuracies on open domain Q&A, reasoning and text understanding tasks for Llama and its variants. As can be seen CoFrGeNet-FNG is much better than CoFrGeNet-F and Llama-NG in most cases and much closer to the original model Llama being roughly $\frac{2}{3}$ the latters size. The docling data mix of 2 trillion tokens was used for pre-training.

Model	openqa	piqa	arc	wino	hswag	lambada	boolq	sciq
Llama (3.2B)	.282	.76	.77	**.654**	**.503**	.581	**.691**	.941
CoFrGeNet-F (1.9B)	.292	**.764**	.765	.643	.482	.581	.659	.941
CoFrGeNet-FNG (2.1B)	**.294**	**.764**	**.778**	.649	.491	**.583**	.668	**.944**
Llama-NG (2.1B)	.277	.755	.762	.639	.477	.576	.660	.938

4.2 Observations

Performance: From Tables 2 and 3 we see that our CoFrGeNet-FNG architecture performs best or close to it on different benchmarks for GPT2-xl. It has slightly more parameters than the previously suggested CoFrNet architecture CoFrGeNet-F. Since, GPT2-xl employs a vanilla FFN it seems the gating in our architecture is not as needed, although it does not harm the performance either. Not using a CoFrNet and doing non-expansive gating however, hurts performance.

Looking at Tables 4 and 5 we see the advantage of our architecture where it is closer to the original Llama and Mamba-2 models than the CoFrGeNet-F. It is in fact significantly better than CoFrGeNet-F on many benchmarks and where it is not worse than it

Table 5: Zero-shot accuracies on open domain Q&A, reasoning and text understanding tasks for Mamba-2 and its variants. Here too CoFrGeNet-FNG is much better than CoFrGeNet-F and Mamba2-NG in most cases and closer to the original model Mamba-2 being roughly $\frac{2}{3}$ the latters size. The docling data mix of 2 trillion tokens was used for pre-training.

Model	openqa	piqa	arc	wino	hswag	lambada	boolq	sciq
Mamba-2 (3.2B)	**.324**	.761	.768	.615	**.486**	**.548**	**.655**	.919
CoFrGeNet-F (1.9B)	.302	.751	.765	.613	.482	.541	.651	.913
CoFrGeNet-FNG (2.1B)	.322	**.763**	**.773**	**.619**	.484	.543	.651	**.924**
Mamba-2-NG (2.1B)	.297	.757	.762	.609	.478	.539	.646	.918

Table 6: Zero-shot accuracies on open domain Q&A, reasoning and text understanding tasks for Llama-MoE and its variants. In this case, CoFrGeNet-FNG is much better than CoFrGeNet-F, Llama-MoE-NG and even the original model Llama-MoE in most cases being roughly half the latters size. The docling data mix of 2 trillion tokens was used for pre-training.

Model	openqa	piqa	arc	wino	hswag	lambada	boolq	sciq
Llama-MoE (403M)	.142	.6	.472	**.501**	.275	.178	**.608**	.726
CoFrGeNet-F (204M)	.148	.609	**.511**	.499	.280	.196	.414	.761
CoFrGeNet-FNG (204M)	**.152**	**.610**	**.511**	.498	**.281**	**.212**	.532	**.776**
Llama-MoE-NG (204M)	.142	.576	.417	.499	.267	.157	.607	.681

even in a single instance. The non-expansive gated version without CoFrNet is again worse. This shows our architecture is particularly useful for models that use gated FFNs.

For the MoE case, we observe in Table 6 that the non-expansive versions are roughly half the size of the original MoE, and our model outperforms even the original model in most cases. Here the size becomes roughly half because more (4 in this case) FFNs (a.k.a. experts) are replaced by our non-expansive gated architecture. As the number of experts increase, our parameter savings should approach the expansion factor in these models as a larger fraction of the model would be our continued fraction architecture. For Llama-MoE the expansion factor was 2.67 and so we start coming close to it in terms of scale of parameter savings as we move from dense models to MoEs. Hence, with more experts the percentage savings in parameters will be even higher. Interestingly here, the new architecture has the same size as CoFrGeNet-F. Also observing the validation loss during pre-training in Figure 2, we see faster convergence and better loss when compared to the original Llama-MoE.

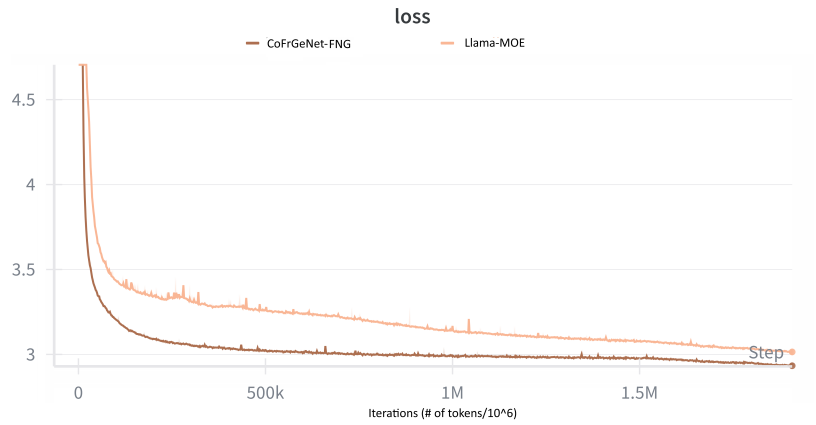

Figure 2: Validation loss for Llama-MoE and CoFrGeNet-FNG as a function of the number of training tokens. As can be seen the loss for CoFrGeNet-FNG converges faster and to a lower value (2.92 vs 3.01) than the original model qualitatively consistent with the benchmark results in Table 6.

Efficiency: In Table 7 we observe the throughput, train times and FLOPS for the different models. Our variant, CoFrGeNet-FNG, is much faster than the original models and requires much fewer FLOPS. While CoFrGeNet-FNG although is slightly larger than the previously proposed CoFrGeNet-F variant, in most cases it has higher throughput and lower training time. This is because the (gated) input has to pass only through a single ensemble of ladders as opposed to two distinct ones and hence has to be read only once leading to higher arithmetic intensity for the new architecture. This improves training efficiency albeit with slightly more parameters and FLOPS/token.

Post Training Quantization: We see in Table 8 that the accuracy drop after performing 4-bit PTQ on the CoFrGeNet-FNG variant of Llama is minimal for all benchmarks. This implies that our variants are robust to quantization. The peak memory usage dropped from 8.3548 GB to 2.3564 GB for the quantized variant.

Replacing Divisions: Note that all models were trained using equation 8 as a replacement for divisions. We see from the results we discussed before that the performance for CoFrGeNet-F matches that of those reported in [14], which did not use this replacement, but used incremental training, thus showing the efficacy of our no division replacement. Moreover, the results for our variant CoFrGeNet-FNG is competitive or better than the original models in several cases further strengthening our claim.

Table 7: Throughput, training time and the number of (training) floating point operations (FLOPS) for the different models and our variants of them. For GPT2-xl the train time is for OWT, GW respectively, while for others its on the docling data mix.

Model	Tokens/day	Train Time (days)	FLOPS/token (GF)
GPT2-xl (1.5B)	1.15B	7.9, 17.2	9.1
CoFrGeNet-F (972M)	1.17B	7.8, 16.9	5.9
CoFrGeNet-FNG (985M)	1.20B	7.6, 16.2	6.1
GPT2-xl-NG (985M)	1.21B	7.6, 16.1	6.1
Llama (3.2B)	235B	8.5	19.3
CoFrGeNet-F (1.9B)	288B	7	11.7
CoFrGeNet-FNG (2.1B)	306B	6.5	12.5
Llama-NG (2.1B)	306B	6.5	12.5
Mamba-2 (3.2B)	475B	4.2	19.1
CoFrGeNet-F (1.9B)	478B	4.2	11.5
CoFrGeNet-FNG (2.1B)	486B	4.1	12.2
Mamba2-NG (2.1B)	491B	4	12.1
Llama-MoE (403M)	299B	6.7	1.1
CoFrGeNet-F (204M)	301B	6.6	0.70
CoFrGeNet-FNG (204M)	309B	6.4	0.71
Llama-MoE-NG (204M)	310B	6.3	0.70

Table 8: Zero-shot accuracy difference between (bf16) CoFrGeNet-FNG for Llama and its post training quantized NF4 variant. As can be seen the difference in accuracies is minimal on open domain Q&A, reasoning and text understanding tasks. The docling data mix of 2 trillion tokens was used for pre-training.

Model	openqa	piqa	arc	wino	hswag	lambada	boolq	sciq
CoFrGeNet-FNG (2.1B)	.0120	.0141	.0126	.0032	.0120	.0136	.0141	$-.0070$

In Table 9, we see a comparison of the three approaches –no incremental training (NIT) using divisions, incremental training (IT) using divisions, and no incremental training and no divisions (NIT-ND)– for depth 5 CoFrGeNet-FNG networks replacing FFNs in GPT2-xl. As can be seen the perplexities are quite similar for IT and NIT-ND with NIT being much worse. This apples-to-apples comparison clearly shows that our reformulation is effective in stabilizing the training of these networks foregoing the need for incremental training.

Table 9: Perplexities of CoFrGeNet-FNG (GPT2-xl) with no incremental training (NIT), with incremental training (IT) and with no incremental training and using no divisions log/exp trick (NIT-ND). NIT-ND performs similar to IT absolving the need for complicating the standard training procedure of language models. Best results bolded.

Data	Model	PTB	Wikitxt2	Lbda	AgNews	Lm1b	Wikitxt103
OWT	NIT	33.72	26.71	12.56	42.18	47.28	22.65
	IT	29.89	**17.12**	8.12	35.72	**40.14**	16.14
	NIT-ND	**29.87**	17.14	**8.11**	**35.69**	40.17	**16.13**
GW	NIT	35.88	25.55	37.33	45.46	49.53	20.44
	IT	29.72	18.13	**30.52**	**41.63**	**46.83**	18.11
	NIT-ND	**29.68**	**18.10**	30.55	41.65	46.86	**18.06**

5 Discussion

In this work we have proposed a novel continued fraction architecture to replace FFNs that performs similarly or significantly better than prior work, while still being parameter efficient and having higher throughput. We have evaluated this across different models such as GPT2, Llama-3, Mamba-2 and Llama-MoE, where the latter two replacements were not performed in prior work. We also proposed a simple way to re-implement divisions in these architectures in terms of logarithms and exponentials making training stable and accurate precluding the need for incorporating complex incremental training procedures.

Even though our work significantly enhances the efficacy and applicability of continued fraction architectures for language modeling it has certain *limitations*. Our networks are still susceptible to prompt injection attacks, hallucinations and other kinds of output manipulation that other architectures are. A memory efficient implementation possibly leveraging Triton kernels is something missing in our implementations, which would possibly further speed up models. We would also like to test our contributions on larger MoE models, although that is constrained by the amount of compute we have available.

We believe, our work makes an important contribution to the field of efficient architectures which can have *significant societal implications*. Having to train more compact models can lead to savings in time, cost and a reduced carbon footprint. They could be easier to audit, maintain and track. The favorable post training quantization results further support that compact CoFrGeNet models can be performant. Moreover, given that our proposed component has favorable interpretability properties [36] may open new avenues to appropriate trust in these models rather than relying solely on post-hoc (mechanistic) interpretability methods, which could have varying degrees of faithfulness.

In the future, it would be exciting to understand if there are ways to guide this unique architecture, which has multiple intervention points as it accepts an input at every layer, to inherently become more robust or safe without simply doing the standard data augmented training. Foraying into such and other trust and efficiency related directions would be, in our opinion, a worthwhile endeavor.

References

[1] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav, J. Lochner, C. Fahlgren, X.-S. Nguyen, C. Fourrier, B. Burtenshaw, H. Larcher, H. Zhao, C. Zakka, M. Morlon, C. Raffel, L. von Werra, and T. Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025.

[2] C. Auer, M. Lysak, A. Nassar, M. Dolfi, N. Livathinos, P. Vagenas, C. B. Ramis, M. Omenetti, F. Lindlbauer, K. Dinkla, L. Mishra, Y. Kim, S. Gupta, R. T. de Lima, V. Weber, L. Morin, I. Meijer, V. Kuropiatnyk, and P. W. J. Staar. Docling technical report. Technical report, 8 2024.

[3] L. Ben Allal, A. Lozhkov, G. Penedo, T. Wolf, and L. von Werra. Cosmopedia, 2024.

[4] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[5] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. In H. Li, H. M. Meng, B. Ma, E. Chng, and L. Xie, editors, *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA, 2014.

[6] C. Christopher, L. Kenton, C. Ming-Wei, K. Tom, C. Michael, and T. Kristina. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

[7] G. G. Chrysos, S. Moschoglou, G. Bouritsas, Y. Panagakis, J. Deng, and S. Zafeiriou. π-nets: Deep polynomial neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7325–7335, 2020.

[8] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[9] D. Dai, C. Deng, C. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. K. Li, P. Huang, F. Luo, C. Ruan, Z. Sui, and W. Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[10] T. Dao, B. Chen, N. S. Sohoni, A. Desai, M. Poli, J. Grogan, A. Liu, A. Rao, A. Rudra, and C. Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning (ICML)*, pages 4690–4721. PMLR, 2022.

[11] T. Dao and A. Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

[12] S. Dasgupta and C. Tosh. Expressivity of expand-and-sparsify representations. Technical report, 6 2020.

[13] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *International Conference on Machine Learning (ICML)*, 2017.

[14] A. Dhurandhar, V. Chenthamarakshan, D. Wei, T. Pedapati, K. N. Ramamurthy, and R. Nair. Cofrgenet: Continued fraction architectures for language generation. In *Proceedings of the 43rd International Conference on Machine Learning*, ICML'26. JMLR.org, 2026.

[15] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[16] H. E. Gohari, S. R. Kadhe, S. Y. S. C. Adam, A. Adebayo, P. Adusumilli, F. Ahmed, N. B. Angel, S. Borse, Y.-C. Chang, X.-H. Dang, N. Desai, R. Eres, R. Iwamoto, A. Karve, Y. Koyfman, W.-H. Lee, C. Liu, B. Lublinsky, T. Ohko, P. Pesce, M. Touma, S. Wang, S. Witherspoon, H. Woisetschlager, D. Wood, K.-L. Wu, I. Yoshida, S. Zawad, P. Zerfos, Y. Zhou, and B. Bhattacharjee. Gneissweb: Preparing high quality data for llms at scale, 2025.

[17] A. Gokaslan, V. Cohen, E. Pavlick, and S. Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[18] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[19] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[20] X. Han, Y. Jian, X. Hu, H. Liu, Y. Wang, Q. Fan, Y. Ai, H. Huang, R. He, Z. Yang, and Q. You. Infimm-webmath-40b: Advancing multimodal pre-training for enhanced mathematical reasoning, 2024.

[21] D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[22] S. Huang, T. Cheng, J. K. Liu, J. Hao, L. Song, Y. Xu, J. Yang, J. H. Liu, C. Zhang, L. Chai, R. Yuan, Z. Zhang, J. Fu, Q. Liu, G. Zhang, Z. Wang, Y. Qi, Y. Xu, and W. Chu. Opencoder: The open cookbook for top-tier code large language models. 2024.

[23] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[24] W. B. Jones and W. Thron. *Continued fractions. Analytic theory and applications*. Encyclopedia of Mathematics and its Applications. Addison-Wesley, 1980.

[25] S. Keisuke, L. B. Ronan, B. Chandra, and C. Yejin. Winogrande: An adversarial winograd schema challenge at scale. 2019.

[26] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations (ICLR)*, 2021.

[27] J. Li, A. Fang, G. Smyrnis, M. Ivgi, M. Jordan, S. Gadre, H. Bansal, E. Guha, S. Keh, K. Arora, S. Garg, R. Xin, N. Muennighoff, R. Heckel, J. Mercat, M. Chen, S. Gururangan, M. Wortsman, A. Albalak, Y. Bitton, M. Nezhurina, A. Abbas, C.-Y. Hsieh, D. Ghosh, J. Gardner, M. Kilian, H. Zhang, R. Shao, S. Pratt, S. Sanyal, G. Ilharco, G. Daras, K. Marathe, A. Gokaslan, J. Zhang, K. Chandu, T. Nguyen, I. Vasiljevic, S. Kakade, S. Song, S. Sanghavi, F. Faghri, S. Oh, L. Zettlemoyer, K. Lo, A. El-Nouby, H. Pouransari, A. Toshev, S. Wang, D. Groeneveld, L. Soldaini, P. W. Koh, J. Jitsev, T. Kollar, A. G. Dimakis, Y. Carmon, A. Dave, L. Schmidt, and V. Shankar. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024.

[28] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark. KAN: Kolmogorov-Arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[29] A. Lozhkov, L. Ben Allal, L. von Werra, and T. Wolf. Fineweb-edu, May 2024.

[30] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19(2):313–330, 1993.

[31] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[32] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[33] K. Milton. Summation techniques, Padé approximants, and continued fractions. 2011. `http://www.nhn.ou.edu/~milton/p5013/chap8.pdf`.

[34] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010.

[35] D. Paperno, G. Kruszewski, A. Lazaridou, Q. N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[36] I. Puri, A. Dhurandhar, T. Pedapati, K. Shanmugam, D. Wei, and K. R. Varshney. Cofrnets: Interpretable neural architecture inspired by continued fractions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21668–21680. Curran Associates, Inc., 2021.

[37] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[39] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[40] S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models, 2024.

[41] N. Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[42] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

[43] W. Sun, J. Hu, Y. Zhou, J. Du, D. Lan, K. Wang, T. Zhu, X. Qu, Y. Zhang, X. Mo, D. Liu, Y. Liang, W. Chen, G. Li, and Y. Cheng. Speed always wins: A survey on efficient architectures for large language models. Technical report, 8 2025.

[44] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[45] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[47] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 24th International Conference on Learning Representations*, 2019.

[48] J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions. 2017.

[49] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[50] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA, 2015. MIT Press.

[51] T. Zhu, X. Qu, D. Dong, J. Ruan, J. Tong, C. He, and Y. Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

A Appendix

Table 10: Downstream task accuracies *with standard deviations* (best results bolded) for GPT2-xl and its variants on GLUE benchmark after finetuning. The first column is the pre-training dataset.

Data	Model	MNLI	QQP	QNLI	SST2	COLA	MRPC	RTE	WNLI
OWT	GPT2-xl (1.5B)	86.89±.15	88.93±.67	91.35±.34	93.56±.24	81.78±.38	79.83±.26	60.27±.22	58.28±.28
	CoFrGeNet-F (972M)	**87.24**±.14	**89.95**±.17	**91.87**±.22	**94.12**±.11	**82.57**±.28	**80.17**±.33	**61.36**	**58.30**±.09
	CoFrGeNet-FNG (985M)	**87.26**±.18	**89.95**±.12	**91.89**±.34	**94.16**±.29	**82.59**±.23	**80.21**±.19	**61.35**±.32	**58.30**±.16
	GPT2-xl-NG (985M)	81.17±.23	81.35±.35	83.94±.21	87.22±.31	75.99±.42	74.37±.27	56.42±.08	58.30±.19
GW	GPT2-xl (1.5B)	78.28±.82	86.83±.17	**82.93**±.37	91.82±.22	74.18±.82	77.72±.93	60.19±.01	**58.33**±.07
	CoFrGeNet-F (972M)	**79.23**±.54	**87.24**±.35	82.69±.61	**92.38**±.32	**74.79**±.25	**78.04**±.21	**61.37**±.51	**58.33**±.44
	CoFrGeNet-FNG (985M)	**79.62**±.63	**87.26**±.25	82.73±.53	**92.36**±.45	**74.83**±.56	**78.01**±.34	**61.35**±.08	**58.33**±.04
	GPT2-xl-NG (985M)	74.83±.66	84.74±.37	77.82±.49	83.94±.42	68.93±.29	71.83±.38	57.85±.41	**58.33**±.18

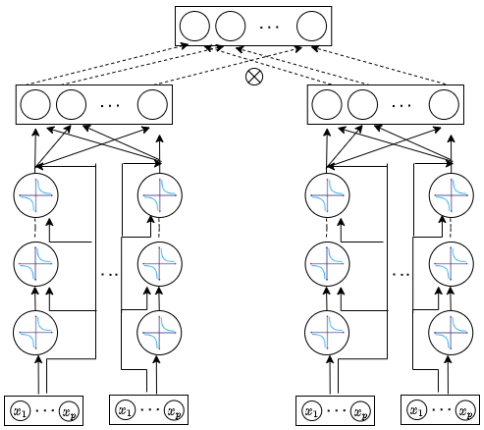

Figure 3: CoFrNet architecture from [14] to simulate FFNs. Two ensembles are used each consisting of specified number of p-variate ladders. Element wise multiplication is done between the representations coming out of the linear layer of each ensemble for higher expressivity.

NeurIPS Paper Checklist

1. **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Section 3 describes the methods and Section 4 describes experiments justifying the claims made in the introduction and abstract.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: In Section 5 second paragraph we discuss limitations of our work.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: NA

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 4 we provide architectural details, information about datasets, parameter settings, evaluation metrics and hardware details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

Justification: The datasets are public, however, our institution has not allowed us to share code yet.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4 we provide experimental details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide standard deviations in Table 10 for fine tuning results. However, pre-training is expensive precluding multiple runs and hence we do not provide statistical significance for zero-shot results from these experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4 we have a paragraph on hardware which provides these details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not have experiments with human subjects and nor do we release any data or models. We mention societal significance and safety in the third paragraph of section 5.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss this in paragraph three in section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We do not release data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite appropriate sources for data, models and other assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: NA

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.