SPEC-COFRGENET: REPARAMETRIZING CONTINUED FRACTION NETWORKS FOR STABLE AND ACCURATE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continued fractions have been shown to be a function class capable of constructing expressive yet parameter-efficient neural networks. These neural networks are not only effective for supervised learning, but also for training large language models (LLMs). However, the rational form of this function class introduces poles when the denominator approaches zero, which prior work has avoided by enforcing a small positive (non-smooth) lower bound. We show that performance and training stability can be highly sensitive to the choice of this lower bound. In this work, we introduce a novel transformation of the partial denominators, inspired by its connection to the Green's function in physics, that mitigates the above issue elegantly by constraining the spectra of the constructed tridiagonal Hamiltonian. Our approach further eliminates the need for incremental training used by previous works for stability, thus avoiding involved modifications to standard pretraining procedures. We demonstrate the effectiveness of our reparametrization by replacing FFNs with a previously proposed continued fraction architecture for both language and image generative modeling. Across diverse models, such as GPT-2 XL, (nano) Llama-MoE, and Granite, we show that our proposed method achieves performance comparable to the original (FFN-based) architectures while achieving $\sim 30\% - 50\%$ parameter savings, and in most cases even outperforms the more complicated incremental training baseline. We further demonstrate, for the first time for continued fraction networks, that their effectiveness also carries over to image generative modeling, that utilize diffusion objectives. On a state-of-the-art Diffusion Transformer, we show that our approach again achieves performance comparable to the original FFN-based models with $\sim 30\%$ parameter savings and significantly outperforms the previous incremental training approach.

1 INTRODUCTION

Generative AI has been rapidly and widely adopted in the recent years, being integrated into consumer, workspace and organizational settings (Bick et al., 2026; Maslej et al., 2025). This boom in adoption is driven by advances in several generative modeling paradigms, most notably for large language models (LLM) for language generation (Brown et al., 2020; Ouyang et al., 2022) and diffusion models for image generation (Peebles & Xie, 2023; Brooks et al., 2024; Labs et al., 2025). An important architectural component of these networks is the feedforward network (FFN) or multilayer perceptron (MLP), which have been shown to be universal function approximators (Hornik et al., 1989). Recent works have proposed alternatives such as continued fraction networks (CoFrNets) (Puri et al., 2021; Dhurandhar et al., 2026), which are also universal approximators, but additionally can represent functions compactly using much fewer parameters than FFNs.

We now provide some background on CoFrNets necessary to motivate our formulation with more background details in Section 3. A generalized continued fraction takes the form

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cdots}},$$

where the partial denominators a_k and partial numerators b_k may, in general, be complex-valued.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Through equivalence transformations, we can also represent this in a form where the numerators are normalized to one. This representation, referred to as the *canonical form*, is given by:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}}.$$

Note that we can truncate these continued fractions to certain depths, which are called *convergents* as they converge to the desired value. In CoFrNets (Puri et al., 2021), each of these partial denominators were parameterized using learnable affine functions of the input $a_k = w_k x$[1], to form a *ladder*. A network is instantiated from this representation by ensembling multiple such ladders, with the reciprocal operation serving as its nonlinearity. Considering $a := (a_1, \ldots, a_d)$, a ladder can then be written as

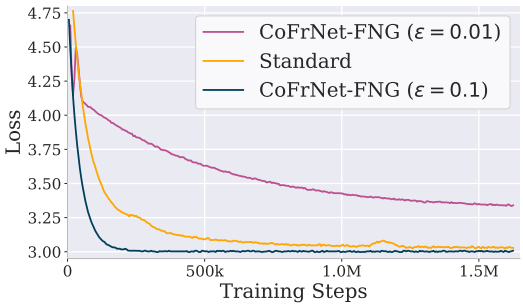


Figure 1: Training loss of **Llama-MoE** for the standard model and CoFrGeNet version with incremental training (Dhurandhar et al., 2026). FNG refers to FFNs replaced with gated continued fraction ladders without any expansion (See **Sec. 3** and **Sec. 5**). Stability of CoFrGeNet is sensitive to the choice ϵ, which is used to avoid avoid poles.

$$f(a_0, a) = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_{d-1} + \cfrac{1}{a_d}}}} = a_0 + \tilde{f}(a),$$

where a_0 is the leading linear term and $\tilde{f}(a)$ denotes the non-linear component of the ladder. This finite continued fraction in fact admits an equivalent representation as a ratio of two polynomials, known as *continuants* (Puri et al., 2021):

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

The continuants K_k satisfy the recurrence

$$K_0 = 1, \qquad K_1(a_d) = a_d,$$
$$K_k(a_{d-k+1}, \ldots, a_d) = a_{d-k+1} K_{k-1}(a_{d-k+2}, \ldots, a_d) + K_{k-2}(a_{d-k+3}, \ldots, a_d).$$

The fractional part $\tilde{f}(a)$ consequently can be expressed as $\frac{K_{d-1}(a_2, \ldots, a_d)}{K_d(a_1, \ldots, a_d)}$.

One practical challenge in training continued-fraction networks is the presence of *poles*. This is apparent from equation 1, where the fractional component is explicitly a rational function and therefore diverges whenever its denominator approaches zero i.e. $K_d(a_1, \ldots, a_d) = 0$. To mitigate this numerical instability, Puri et al. (2021) used the following stabilization:

$$K_d \longrightarrow \operatorname{sgn}(K_d) \max\left(|K_d|, \epsilon\right), \qquad \epsilon > 0. \tag{2}$$

This clipping ensures that the magnitude of the denominator never falls below ϵ, thus ensuring that the the continued fraction ladders never encounter a pole during training.

Dhurandhar et al. (2026) used the above principles of using continued fractions for training neural networks and extended that to a generative training framework to train LLMs. Despite having success in doing so, it remains unclear if such networks can be adopted to generative modeling of vision using diffusion style flow-matching losses or even other architectures such as Mixture-of-Experts (MoEs) for language. Additionally, in our work we find that the aforementioned ϵ based stabilization has some disadvantages. As shown in Fig. 1, the convergence of the training loss relies heavily on the choice of ϵ: smaller epsilon values (i.e. $\epsilon = 0.01$) lead to slower and less stable convergence. Conversely, a larger ϵ value can restrict the continuant value and decrease the range of expressible values. *More importantly, we find that when using diffusion style loss for vision generative modeling, the training often gets unstable even for larger ϵ such as* $0.1,\ 0.2$.

[1]The bias term was absorbed in x.

In this work, we introduce Spectral constrained Continued Fraction Generative Networks (CoFrGeNet), a novel reparameterization of continued fraction networks which does not rely on non-smooth ϵ-like clamps. We take inspiration from physics where the (0,0) element of a Krylov basis Hamiltonian forms a continued fraction, as shown in **Fig. 2** and demonstrate the effectiveness of our approach by replacing the feedforward networks of standard architectures since, the previous work (Dhurandhar et al., 2026) showed the best results with such a replacement. We summarize **our contributions** as follows:

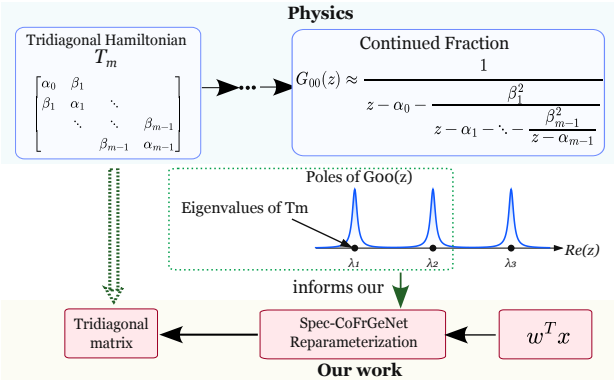


Figure 2: A Hamiltonian is transformed into a tridiagonal matrix in the Lanczos construction, whose Green's function has continued fraction representation. Poles of this fraction lie at the eigenvalues of the matrix. Motivated by this connection in physics, our conceptual contribution goes the other way, where our reparameterization constrains the spectrum of the associated tridiagonal matrix.

• We introduce a novel parameterization of continued fraction networks, which constrains the spectrum of an associated tridiagonal matrix, thus absolving the need for a non-smooth clamp that can make training and inference unstable even with incremental training.

• We perform experiments with three different LLM families, namely GPT-2 XL, Granite 4.1 and Llama-MoE. Across multiple evaluation datasets, Spec-CoFrGeNet outperforms the previous continued fraction approach. Spec-CoFrGeNet also outperforms standard training of these models in many cases, while using $\frac{2}{3}$ to $\frac{1}{2}$ the parameters.

• We show that our new approach makes continued fraction networks amenable to even diffusion models for vision, where the previous approach is considerably worse.

2 Related Work

The advent of the Transformer architecture (Vaswani et al., 2017) has played a significant contribution in the large scale growth of modern generative models. One defining part of this architecture is the self-attention which models interactions across tokens. Another important part is the FFN, which performs nonlinear transformation on each token. In fact, FFNs occupy a substantial amount of Transformer parameters. Thus existing research trying to improve the efficiency and expressivity of FFNs fall under 3 buckets. The first direction aims to create alternative activation functions and gating mechanisms, such as GELU (Hendrycks & Gimpel, 2016), Swish/SiLU (Ramachandran et al., 2017), GLU (Shazeer, 2020), as well as the widely adopted SwiGLU (Touvron et al., 2023a;b; Grattafiori et al., 2024). The second direction aims at changing computation through sparse mixture-of-experts (MoE), which routes inputs selectively to expert networks (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024; Zhu et al., 2024). The third direction aims to change the function class itself. These include Kolmogorov–Arnold Networks (KANs) (Liu et al., 2024), which consists of learnable univariaye functions, polynomial networks (Chrysos et al., 2020) which leverage higher-order polynomial interactions as well as Monarch (Dao et al., 2022). However, scalability, optimization behavior, efficiency, remain import open coniderations. Our closest work is CoFrGeNets (Dhurandhar et al., 2026), which propose continued fraction based architectures discussed in Section 1 and further discussed in Section 3.

Diffusion transformers constitute a major class of generative models for image generation. DiT (Peebles & Xie, 2023) established the Transformer architecture as a competitive backbone retaining the aforementioned MLP sublayers and attention blocks. Further developments have taken place based on the objective function such as adopting flow-based losses that learn a velocity field (Ma et al., 2024) as well as introducing an auxillary representation aligning objective (Yu et al., 2024). Finally DDT (Wang et al., 2026) introduced modern FFN designs such as SwiGLU, while decoupling the network into an encoder-decoder block.

In this work, we replace the FFNs in both language and vision models with a gated CoFrNet architecture shown in **Fig. A1**, but where each layer is reparametrized as described in Section 4.

3 BACKGROUND

Continued fraction generative networks. In Dhurandhar et al. (2026), the conventional transformer components are replaced by ensembles of the aforementioned continued fraction ladders (see **Sec. 1**). A pictorial representation of the architecture is shown in **Fig. A1** in the appendix. The inputs to these ladders, i.e., the partial denominators of the continued fractions, are provided by learned affine functions. Multiple such ladders are then combined through a linear projection. The **depth** of each ladder is the depth of the corresponding continued fraction, while the number of parallel ladders in the ensemble is called the **width**. Additionally, Dhurandhar et al. (2026) adopt an **incremental training** process, where the parameters of the continued fraction network are optimized in a progressive fashion, i.e., parameters at depth i are optimized only during the final $t/2^i$ iterations of a run containing t total iterations.

The FFNs in Transformer architectures have two features that are relevant to their replacement by continued fraction ladders. The first is a **gating** operation. Given an input $x \in \mathbb{R}^p$, two linear projections are computed, one of which passes through a non-linear activation (viz. Swish, SwiGLU, etc.). Gating refers to the element-wise multiplication of these two projections. Continued fraction replacements of FFNs unconditionally use such gating. The second feature is **expansion**, referring to how FFNs first project each token from the model dimension (d_{model}) to a higher-dimensional representation (d_{mlp}), and then project the representation back to the model dimension. We define the **expansion factor** α as the ratio $d_{\text{mlp}}/d_{\text{model}}$. While typical FFN architectures are expanded, i.e., $\alpha > 1$, usually CoFrNets adopt a non-expanded representation, i.e., $\alpha = 1$ to save parameters.

From Hamiltonians to continued fractions. The Hamiltonian operator plays a central role in the study of quantum physics. Not only does it govern the time evolution of a quantum system, but it also characterizes the energy structure of such a system (Sakurai & Tuan, 1994). Indeed, a central challenge in quantum physics is to compute the spectrum of the Hamiltonian, i.e., solve $H|\psi_n\rangle = E_n|\psi_n\rangle$, where E_n are the energy eigenvalues and $|\psi_n\rangle$ are the corresponding energy eigenstates. However, for large quantum systems, computing the full spectrum is computationally prohibitive and motivates the use of iterative methods such as the Lanczos algorithm, which can efficiently compute a number of extremal eigenvalues and eigenstates without the full decomposition (Lin, 1990; Dagotto, 1994). The Lanczos algorithm creates a Krylov subspace through the repeated application of the Hamiltonian to an initial state (Dagotto, 1994; Saad, 2011). An m-dimensional Krylov subspace is thus defined by

$$\mathcal{K}_m(H, |\phi_0\rangle) = \text{span}\left\{|\phi_0\rangle, H|\phi_0\rangle, H^2|\phi_0\rangle, \ldots, H^{m-1}|\phi_0\rangle\right\}.$$

Subsequently, the Krylov basis vectors are orthonormalized iteratively to construct an orthonormal basis $\{\phi_n\}_{n=0}^{m-1}$ as follows:

$$\begin{aligned}
&\beta_{n+1}|\phi_{n+1}\rangle = H|\phi_n\rangle - \alpha_n|\phi_n\rangle - \beta_n|\phi_{n-1}\rangle \\
&\alpha_n = \langle\phi_n|H|\phi_n\rangle, \qquad \beta_n = \langle\phi_{n-1}|H|\phi_n\rangle.
\end{aligned}$$

In this constructed basis, the Hamiltonian is projected into the Krylov subspace and its matrix representation becomes the following symmetric tridiagonal matrix (Dagotto, 1994):

$$T_m = \begin{bmatrix} \alpha_0 & \beta_1 & & \\ \beta_1 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_{m-1} \end{bmatrix}.$$

Meanwhile, the resolvent of the Hamiltonian, given by $(z - H)^{-1}$ (z is a complex parameter with the units of energy), is another important quantity frequently used in various domains of quantum mechanics. The matrix elements of the resolvent defines the Green's function. Using the m-dimensional Krylov subspace defined above, the (0,0)th element of the matrix representation of the Green's function can be approximated as:

$$G_{00}(z) \approx [(zI - T_m)^{-1}]_{00} = \cfrac{1}{z - \alpha_0 - \cfrac{\beta_1^2}{z - \alpha_1 - \ddots - \cfrac{\beta_{m-1}^2}{z - \alpha_{m-1}}}},$$

which we recognize as a continued fraction. This (0,0) element can be easily computed with recursive elimination with linear complexity.

Additionally, the above inverse also admits another form as follows:

$$(zI - T_m)^{-1} = \sum_{n=0}^{m-1} \frac{|\psi_n\rangle\langle\psi_n|}{z - E_n}$$

where ψ_n and E_n are the eigenvectors and eigenvalues of T_m.

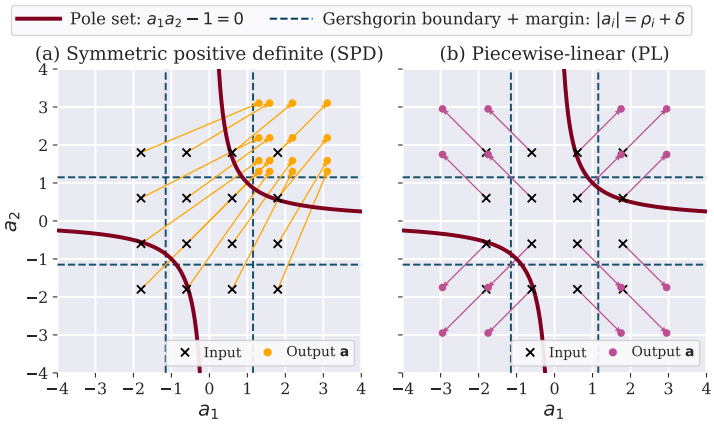


Figure 3: Our proposed parameterizations successfully avoid poles by moving the inputs away from the Gershgorin boundary, as visualized with two-dimensional datapoints. 'Input' refers to values of $w_i^T x$ in equation 4 and equation 5. $[a_1, a_2]$ refers to the resulting parameterized values.

4 METHODOLOGY

We use the end results of the progression of ideas presented in Section 3 as the basis of our proposed method. The crux is the following result: the poles of $G_{00}(z)$ are exactly the eigenvalues of T_m (Economou, 2006). Note that although the development in Section 3 comes from quantum mechanics, the consequences are actually linear algebraic, i.e., they can be independent of the "quantum nature." As a result, for our purposes we can freely set $\beta_i = 1$, $\alpha_i = a_i \in \mathbb{R}$, and $z = 0$. Thus our continued fraction reduces from $G_{00}(z)$ to $f(a) = \cfrac{1}{a_1 - \cfrac{1}{a_2 - \cdots}}$, where the corresponding tridiagonal matrix is $T(a) = \begin{bmatrix} a_1 & 1 & & \\ 1 & a_2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & a_d \end{bmatrix}$.

Recall our motivation from Section 1, where we wanted to avoid poles of the continued fractions for stable training, without the reliance on external parameters like ϵ (see equation 2). Since we have set $z = 0$ above, we wish to avoid poles at $z = 0$. This is equivalent to $T(a)$ not having any zero eigenvalues, i.e., it is a non-singular matrix.

Sufficient condition. To derive a sufficient condition for such non-singularity, we leverage the Gershgorin circle theorem (Horn & Johnson, 2012) which states the following : For a matrix $A \in \mathbb{C}^{d\times d}$, every eigenvalue lies in at least one disk

$$\mathcal{D}_i(A) = \{z \in \mathbb{C} : |z - A_{ii}| \le R_i(A)\}, \qquad R_i(A) = \sum_{j\neq i} |A_{ij}|.$$

For the tridiagonal matrix $T(a)$, Gershgorin disks have the radii ρ_i, given by:

$$\rho_i = \sum_{j\neq i} |T_{ij}| = \begin{cases} 1, & i \in \{1, d\}, \\ 2, & 1 < i < d, \end{cases}$$

and are centered at a_i. Thus a sufficient condition for avoiding poles is to simply push each disk away from the origin (i.e. the zero eigenvalue) by its radius:

$$|a_i| > \rho_i, \qquad i = 1, \ldots, d, \tag{3}$$

Reparameterizations. In Dhurandhar et al. (2026); Puri et al. (2021), the partial denominator scalars are parameterized as the inner product of a learnable weight vector and the input. Following this, we enforce the condition in equation 3 by reparameterizing the partial denominators as

$$\boxed{a_i(x) = \rho_i + \text{softplus}(w_i^T x) + \delta, \qquad \delta > 0} \tag{4}$$

where softplus is a smooth strictly positive mapping given by $\text{softplus}(z) = \log(1 + e^z)$, and δ is a small margin, set as $\delta = 10^{-3}$ in our experiments. This leads to a symmetric positive definite (SPD) $T(a)$ because all of its eigenvalues are positive.

Alternatively, we can preserve the sign of $w_i^T x$ by the following piecewise linear (PL) parameterization:

$$a_i(x) = \begin{cases} \rho_i + w_i^T x + \delta, & w_i^T x \geq 0, \\ -\rho_i + w_i^T x - \delta, & w_i^T x < 0. \end{cases} \tag{5}$$

*Note that the δ here is a small offset at the continued fraction **input** to enforce a condition in the spectral domain (i.e. avoid zero eigenvalues), and not a non-smooth clamp like ϵ that directly determines the scale of the continued fraction **output**.* The latter is something that has to be carefully tuned, while the former can be set to a fixed value much smaller than ϵ (i.e. 10^{-3}) across all our experiments, while still obtaining good results.

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
GPT-2 XL	1.5B	48	1600	4
Granite 4.1	2.14B	24	2560	3.2
Llama-MoE	403M	30	576	2.67

We explore the effects of the proposed parameterizations for a depth-2 CoFrNet in **Fig. 3**. Let us consider $\mathbf{a} = [a_1, a_2]$ to be the parameterized values. At depth 2, the matrix $T(a)$ becomes $\begin{bmatrix} a_1 & 1 \\ 1 & a_2 \end{bmatrix}$. The pole set corresponds to values of (a_1, a_2) for which $T(a)$ has a zero eigenvalue, i.e., $\det T(a) = 0 \implies a_1 a_2 = 1$. For this 2×2 $T(a)$, the Gershgorin radius is 1 and from equation 3, the sufficient condition becomes $|a_i| > 1$. Thus, the dashed lines in **Fig. 3** denote the margin-adjusted Gershgorin boundaries $|a_i| = 1 + \delta$ beyond which all points are pushed. Thus we can see from the illustration how our parameterizations avoid the complete pole set (the hyperbolas). However, the sign-dependent shift makes PL discontinuous whenever either a_i crosses zero. In contrast, SPD maps all inputs continuously through the softplus transformation, although it is restricted to be positive. This smoothness however favors optimization and is therefore our preferred parametrization in the experiments.

5 Experiments

In this section, we evaluate Spec-CoFrGeNet across both language and image generation tasks, allowing us to establish the generality of our approach. We **replaced the feedforward network** of all the models used, with the CoFrGeNet architecture (Dhurandhar et al., 2026) along with the modifications of Spec-CoFrGeNet (see **Fig. A1** in Appendix A).

For all experiments, we compare Spec-CoFrGeNet with incremental training of CoFrGeNets (henceforth referred to simply as 'CoFrGeNet') and also with the original feedforard network based model (henceforth referred to simply as 'Standard'). Typically, all CoFrNet variants have an expansion factor of 1, unless otherwise mentioned. Note that the continued fraction based architecture and the expansion factor significantly reduces the number of parameters for CoFrNet variants. Hence for fair comparison, we also consider the original models with an expansion factor of 1 (often marked with a †). For all experiments, $\epsilon = 0.1$ in CoFrGeNets, while width and depth are chosen as 5. Models are referred to as FNG (Feedforward Non-expanded Gated) when $\alpha = 1$ with gating, or FG (Feedforward Gated), when $\alpha > 1$.

Table 2: Perplexity of pretrained **GPT-2 XL** models on various evaluation datasets (lower is better). All CoFrGeNets have the **FNG** setting. The standard models do not have gating. † denotes the standard model trained with an expanion factor of 1. Light gray denotes the default standard setting, while light blue denotes our proposed method. The best results are highlighted in **bold**.

Dataset	Method	Params	PTB	Wikitext2	Lambada	AgNews	LM1B	Wikitext103
OWT	Standard	1.5B	30.12	18.30	8.66	37.13	41.20	17.50
	Standard †		35.98	27.74	16.35	50.75	51.95	29.29
	CoFrGeNet	985M	29.89	**17.12**	8.12	35.72	**40.14**	16.14
	Spec-CoFrGeNet		**29.87**	17.14	**8.11**	**35.69**	40.17	16.13
GW	Standard	1.5B	**29.07**	19.12	31.78	45.62	52.36	18.93
	Standard †		34.26	27.38	39.44	61.81	58.64	29.02
	CoFrGeNet	985M	29.72	18.13	**30.52**	41.63	**46.83**	18.11
	Spec-CoFrGeNet		29.68	**18.10**	30.55	**41.61**	46.86	**18.06**

5.1 LANGUAGE MODELING

For language modeling, we test Spec-CoFrGeNet on three diverse models GPT-2 XL , Granite 4.1 and Llama-MoE . Unless otherwise mentioned, all experiments were performed with a per-GPU batch size of 16 using the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.95$) and a weight decay of 0.1.

GPT-2 XL. We pre-train GPT-2 XL and the aforementioned baselines on OpenWebText (OWT) (Gokaslan et al., 2019) and GneissWeb-35B (GW) (Gohari et al., 2025). For each pretrained model, we report the perplexity on Penn Tree Bank (PTB) (Marcus et al., 1993), Wikitext2 (Merity et al., 2017), Wikitext103 (Merity et al., 2017), Lambada (Paperno et al., 2016), AgNews (Zhang et al., 2015) and One Billion Words (LM1B) (Chelba et al., 2014). We use a stride of 512 for WikiText-2 and WikiText-103, following prior evaluation protocols, and a stride of 256 for all other datasets. We use the recommended settings for pretraining GPT-2 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 (2K warmup). All models were trained on 16 H100 GPUs leveraging distributed data parallel (DDP) framework.

In most datasets in **Table 2**, Spec-CoFrGeNet outperforms previous CoFrGeNet versions and also performs better than the standard version of the model with a much higher number of parameters.

Table 3: Downstream accuracies of finetuned **GPT-2 XL** models on GLUE. All CoFrGeNets have the FNG setting. The standard models do not have gating. † denotes the standard model trained with no expansion.

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

We also evaluate finetuning performance on the pretrained GPT-2 XL models by finetuning on GLUE (Wang et al., 2019) (classification) tasks and compare accuracies as done in previous works (Dhurandhar et al., 2026). For finetuning, we choose the best performing learning rates for all the baselines, thus leading to 0.125×10^{-4} for Spec-CoFrGeNet and 0.25×10^{-4} for all other methods. Finetuning was achieved with a single A100 GPU for each model. As observed in **Table 3**, Spec-CoFrGeNet achieves better performance in most downstream tasks, when compared to Standard.

Granite 4.1. We pretrain Granite 4.1 on 125B tokens drawn from web and mathematical datasets in NVIDIA's Nemotron data repository, including Web-Nemotron-CC, MegaMath-Web, and FineMath-3Plus. We evaluate these pretrained models on OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), ARC-Easy (Clark et al., 2018), WinoGrande (Keisuke et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA (OpenAI) (Radford et al., 2018), BoolQ (Christopher et al., 2019), and SciQ (Welbl et al., 2017). These benchmarks cover open-domain and commonsense question answering, physical and commonsense reasoning, and language understanding.
We pre-train Granite 4.1 for 30K iterations, with 3K warmup steps. The training was performed using the Adam optimizer with an initial learning rate of 3×10^{-4} and an annealing schedule with no dropout. All Granite 4.1 models were trained using 64 H100 GPUs with FSDP.

Table 4: Zero-shot accuracies on open domain Q&A, reasoning and text understanding tasks for **Granite 4.1** and its baselines.

Method	Setting	Params	OpenBookQA	PIQA	ARC-Easy	WinoGrande	HellaSwag	LAMBADA	BoolQ	SciQ
Standard	FG	2.14B	35.2	**74.8**	69.3	56.9	**59.4**	48.7	57.7	**90.8**
CoFrGeNet			37.2	74.4	69.9	58.1	59.1	48.6	60.9	90.2
Spec-CoFrGeNet			**37.4**	74.6	**70.0**	**58.4**	**59.4**	**49.0**	**61.2**	90.0
Standard	FNG	1.1B	34.9	72.9	63.9	55.7	52.7	44.7	57.2	86.1
CoFrGeNet			35.2	73.1	63.9	55.9	52.7	45.0	59.1	86.5
Spec-CoFrGeNet			35.4	73.1	64.1	55.8	53.2	45.0	60.2	86.7

Table 5: Zero-shot accuracies on open domain Q&A, reasoning and text understanding tasks for **Llama-MoE** and its baselines. All CoFrGeNets have the **FNG** setting. † denotes the standard model trained with no expansion (i.e. $\alpha = 1$).

Model	Params	OpenBookQA	PIQA	ARC-Easy	WinoGrande	HellaSwag	LAMBADA	BoolQ	SciQ
Standard	403M	14.2	60.0	47.2	**50.1**	27.5	17.8	**60.8**	72.6
Standard †	204M	14.2	57.6	41.7	49.9	26.7	15.7	60.7	68.1
CoFrGeNet		15.2	61.0	51.1	49.8	**28.1**	**21.2**	53.2	77.6
Spec-CoFrGeNet		**15.4**	**61.2**	**51.2**	**50.1**	28.0	21.1	53.7	**78.0**

As observed in **Table 4**, the non-expanded versions contain almost half the parameters of the expanded versions yielding competitive but worse performance. Hence, we consider the expanded versions of CoFrGeNet and Spec-CoFrGeNet for Granite 4.1. Spec-CoFrGeNet outperforms the other baselines in six out of eight datasets, thus establishing the success of our approach.

Llama-MoE. We pretrain Llama-MoE (configured with 4 experts and one active expert per token) using the 2T token data mixture, which were used to train the Docling (Auer et al., 2024) models. This mixture contains web data (DCLM2 and DCLM3Plus (Li et al., 2024)), multilingual data (FineWeb-2-Edu (Lozhkov et al., 2024)), code (StarCoder and Stack-Edu (Allal et al., 2025)), mathematical data (FineMath (Allal et al., 2025), InfiWebMath (Han et al., 2024), OPC-FineWeb-Math-Corpus (Huang et al., 2024)), and synthetic data (Cosmopedia (Ben Allal et al., 2024)). The exact proportions of the different datasets are provided in the Appendix B. The evaluation setup remains the same as Granite 4.1. This was pretrained for about 2M iterations with 2K warmup steps. All other experimental details remain consistent with Granite 4.1.

Table 6: Efficiency metrics namely throughput (given in tokens/ day), training time (days) and number of floating point operations (GFLOPs) per token for all models. For GPT2 XL, training time is presented as (OWT / GW). † denotes the standard model trained with no expansion (i.e. $\alpha = 1$), while § denotes Spec-CoFrGeNet trained with expansion (FG).

Model	Method	Parallelism Strategy	Throughput	Training Time	FLOPs
GPT-2 XL	Standard	DDP	1.15B	7.9 / 17.2	9.1
	Standard†		1.21B	7.6 / 16.1	6.1
	Spec-CoFrGeNet		1.20B	7.6 / 16.2	6.1
Granite 4.1	Standard	FSDP	162B	0.77	12.7
	Standard†		203B	0.61	6.8
	Spec-CoFrGeNet		202B	0.62	6.8
	Spec-CoFrGeNet §		160B	0.78	12.8
Llama-MoE	Standard	FSDP	299B	6.7	1.1
	Standard†		310B	6.3	0.70
	Spec-CoFrGeNet		309B	6.4	0.71

As observed in **Table 5**, even though the CoFrGeNet models have almost half the number of parameters when compared to the standard model, Spec-CoFrGeNet outperforms all baselines in five out of eight tasks. In the other tasks, it maintains competitive performance with the CoFrGeNet model trained with incremental training.

Finally, we present various efficiency metrics in **Table 6**. As can be seen Spec-CoFrGeNet is more efficient than the corresponding standard models.

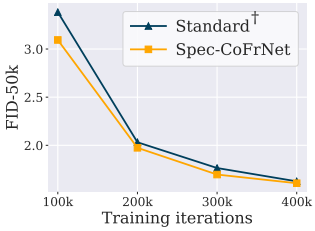


Figure 4: Convergence of FID measured for 50000 samples generated using DDT-XL/2 models.

Table 7: Performance of various metrics using DDT-XL/2 at 400000 steps. ↑ refers to higher the better and ↓ refers to lower the better.

Method	Params	FID ↓	sFID ↓	Inception Score ↑	Precision ↑	Recall ↑
Standard	678M	1.447	4.345	312.594	0.631	0.801
Standard †	493M	1.626	4.417	287.476	0.620	0.800
Spec-CoFrGeNet	493M	1.606	4.406	290.835	0.621	0.799

5.2 IMAGE GENERATION

To demonstrate the effectiveness of our approach in vision, we perform class-conditional image generation experiments with Decoupled Diffusion Transformers (DDT) (Wang et al., 2026) on ImageNet-1K at 256×256 resolution. Following Wang et al. (2026), images are center cropped using the ADM preprocessing protocol (Dhariwal & Nichol, 2021) and encoded into latent representations using the frozen Stability AI VAE (Rombach et al., 2022). We perform experiments using the DDT-XL/2 model as the primary backbone architecture, which contains 28 transformer blocks, with 22 blocks acting as the encoder and 6 blocks acting as the velocity decoder. Representation alignment is performed using REPA (Yu et al., 2024), which aligns the hidden representation of the output of the eighth encoder block with a frozen DINOv2-B/14 teacher (Oquab et al., 2024). All other architecture details follow the DDT implementation (Wang et al., 2026).

We train the model using a velocity-based linear flow matching objective (Ma et al., 2024) regularized with a REPA objective. The REPA loss is computed using a token-wise negative cosine similarity between projected DDT representations and the corresponding teacher DINOv2-B/14 representations. For classifier-free guidance, class labels are dropped with a probability of 0.1. An exponential moving average (EMA) of the model, with a decay of 0.9999 was maintained.

These models were trained for 400000 steps following previous works, using AdamW with a constant learning rate of 1×10^{-4}, $\beta_1 = 0.9, \beta_2 = 0.95$ and zero weight decay. We used a per-GPU batch size of 32 using BF16 mixed precision on eight NVIDIA A100s.

Table 8: Efficiency metrics namely throughput (given in images per second per GPU) and training time (in hours) for all variants of DDT-XL/2 models.

Method	Training Throughput	Training Time
Standard	64.78	54.88
Standard†	74.79	47.54
Spec-CoFrGeNet	76.57	43.41

We evaluate the EMA model by sampling 50,000 class-balanced images, using 250 deterministic Euler steps with the encoder representation recomputed at every step. During sampling, classifier-free guidance is applied for $0 < t < 0.7$ with a guidance scale of 2.0. Following the ADM evaluation protocol (Dhariwal & Nichol, 2021), we report the FID, sFID, Inception Score, Precision and Recall. FID measures the similarity between real and generated images in the Inception feature space, but it is sensitive to image spatial changes like resizing and preprocessing Leng et al. (2026). sFID provides a complementary metric by capturing spatial relationship. Inception score and Precision characterize sample fidelity while Recall is a proxy for sample diversity.

In our experiments, we observe that the **CoFrGeNet (with incremental training) cannot learn the input distribution**. Consequently, FID stays stuck at around 565.3 and eventually the loss becomes unstable. This is in fact a much more severe issue of instability, than the one we demonstrated in Section 1 and Fig. 1. As observed in **Fig. 4**, Spec-CoFrGeNet converges faster than non-expanded Standard model. Additionally, from **Table 7**, we observe that Spec-CoFrGeNet performs close to the Standard model while being while having about 30% less parameters. We also see that our method outperforms the non-expanded standard model in FID, sFID and IS.

6 CONCLUSION

In summary, inspired by the Green's function and its relation to continued fractions in physics we have provided a novel approach to stably and accurately train CoFrNets that does away with the non-smooth and rather artificial clamping. The trick is to reparametrize CoFrNets so as to spectrally constrain them that leads to a non-singular tridiagonal matrix inverse formed from the CoFrNet layers. Empirically, we obtain superior results to CoFrGeNets on language in many cases and significantly better results on vision foregoing the need for incremental training.

Our general idea of simply requiring a non-singular tridiagonal matrix formed from the CoFrNet layers opens up avenues for other reparametrizations. For instance, one could design other functions for each entry or even reparametrize a combination of entries as those would be weaker conditions for invertibility. Additionally, even though we consider generative modeling, our proposed reparameterization is general and can be applied in any discriminative learning task. Such explorations would be interesting to conduct in the future.

AI USE STATEMENT

In this work, we used generative AI tools for assistance in coding and assistance in preparing the manuscript such as polishing the writing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work provides a novel parameterization for training and we are not releasing a new model or dataset. Thus the risks associated with the method are similar to training any generative modeling architecture and depend on the dataset as well as downstream deployment. All existing safety measures and guardrails applied to other models, are applicable to our method as well.

REPRODUCIBILITY STATEMENT

We have provided the details of the training procedure as well as architectural details in the manuscript to facilitate reproducibility. Upon request, the code will be made available.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. Docling technical report. Technical report, 8 2024. URL https://arxiv.org/abs/2408.09869.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, 2024. URL https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.

Alexander Bick, Adam Blandin, and David J Deming. The rapid adoption of generative ai. *Management Science*, 2026.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (eds.), *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, pp. 2635–2639. ISCA, 2014. doi: 10.21437/INTERSPEECH.2014-564. URL https://doi.org/10.21437/Interspeech.2014-564.

Clark Christopher, Lee Kenton, Chang Ming-Wei, Kwiatkowski Tom, Collins Michael, and Toutanova Kristina. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Grigorios G. Chrysos, Stylianos Moschoglou, Giorgos Bouritsas, Yannis Panagakis, Jiankang Deng, and Stefanos Zafeiriou. π-nets: Deep polynomial neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7325–7335, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Elbio Dagotto. Correlated electrons in high-temperature superconductors. *Reviews of Modern Physics*, 66(3):763, 1994.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng 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.

Tri Dao, Beidi Chen, Nimit S. Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning (ICML)*, pp. 4690–4721. PMLR, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Amit Dhurandhar, Vijil Chenthamarakshan, Dennis Wei, Tejaswini Pedapati, Karthikeyan Natesan Ramamurthy, and Rahul Nair. Cofrgenet: Continued fraction architectures for language generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=wjc6oz63WL.

Eleftherios N Economou. *Green's functions in quantum physics*. Springer, 2006.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Hajar Emami Gohari, Swanand Ravindra Kadhe, Syed Yousaf Shah. Constantin Adam, Abdulhamid Adebayo, Praneet Adusumilli, Farhan Ahmed, Nathalie Baracaldo Angel, Santosh Borse, Yuan-Chi Chang, Xuan-Hong Dang, Nirmit Desai, Ravital Eres, Ran Iwamoto, Alexei Karve, Yan Koyfman, Wei-Han Lee, Changchang Liu, Boris Lublinsky, Takuyo Ohko, Pablo Pesce, Maroun Touma, Shiqiang Wang, Shalisha Witherspoon, Herbert Woisetschlager, David Wood, Kun-Lung Wu, Issei Yoshida, Syed Zawad, Petros Zerfos, Yi Zhou, and Bishwaranjan Bhattacharjee. Gneissweb: Preparing high quality data for llms at scale, 2025. URL https://arxiv.org/abs/2502.14907.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiaotian Han, Yiren Jian, Xuefeng Hu, Haogeng Liu, Yiqi Wang, Qihang Fan, Yuang Ai, Huaibo Huang, Ran He, Zhenheng Yang, and Quanzeng You. Infimm-webmath-40b: Advancing multimodal pre-training for enhanced mathematical reasoning, 2024. URL https://arxiv.org/abs/2409.12568.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Siming Huang, Tianhao Cheng, Jason Klein Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J. Yang, J. H. Liu, Chenchen Zhang, Linzheng Chai, Ruifeng Yuan, Zhaoxiang Zhang, Jie Fu, Qian Liu, Ge Zhang, Zili Wang, Yuan Qi, Yinghui Xu, and Wei Chu. Opencoder: The open cookbook for top-tier code large language models. 2024. URL https://arxiv.org/pdf/2411.04905.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Sakaguchi Keisuke, Le Bras Ronan, Bhagavatula Chandra, and Choi Yejin. Winogrande: An adversarial winograd schema challenge at scale. 2019.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Xingjian Leng, Jaskirat Singh, Zhanhao Liang, Ethan Smith, Martin Bell, Aninda Saha, Yuhui Yuan, and Liang Zheng. Diffusionbench: On holistic evaluation of diffusion transformers. *arXiv preprint arXiv:2606.24888*, 2026.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations (ICLR)*, 2021.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024.

HQ Lin. Exact diagonalization of quantum-spin models. *Physical Review B*, 42(10):6561, 1990.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-Arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu, May 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19(2):313–330, 1993.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, et al. Artificial intelligence index report 2025. *arXiv preprint arXiv:2504.07139*, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard 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. URL https://openreview.net/forum?id=Byj72udxe.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel 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. doi: 10.18653/V1/P16-1144. URL https://doi.org/10.18653/v1/p16-1144.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Isha Puri, Amit Dhurandhar, Tejaswini Pedapati, Karthikeyan Shanmugam, Dennis Wei, and Kush R Varshney. Cofrnets: interpretable neural architecture inspired by continued fractions. *Advances in neural information processing systems*, 34:21668–21680, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

Jun John Sakurai and San Fu Tuan. *Modern quantum mechanics*, volume 2. Addison-Wesley, 1994.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel 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.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 40633–40642, 2026.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. 2017.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 649–657, Cambridge, MA, USA, 2015. MIT Press.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu 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 Architecture details

B Additional dataset details

Dataset for pretraining Granite 4.1. We construct the data mixture from four datasets `web-nemotron-cc-hq-p2_0`, `web-nemotron-cc-hq-p2_1`, `megamath-web-pro_0`, `finemath-3plus-rewritten_0` with sampling ratios of 0.35, 0.30, 0.20 and 0.15 respectively.

Dataset for pretraining Llama-MoE. The Docling data mixture contains `dclm3plus`, `dclm2`, `cosmopedia_v2`, `fineweb_math_corpus`, `finemath-4plus`, `infiwebmath-4plus`, `finemath-3plus`, `infiwebmath-3plus`, `starcoder`, `stack-edu`, `fineweb-2-edu` with corresponding sampling weights of 0.5, 0.24, 0.06, 0.01, 0.015, 0.01, 0.015, 0.005, 0.01, 0.17 and 0.01 respectively.

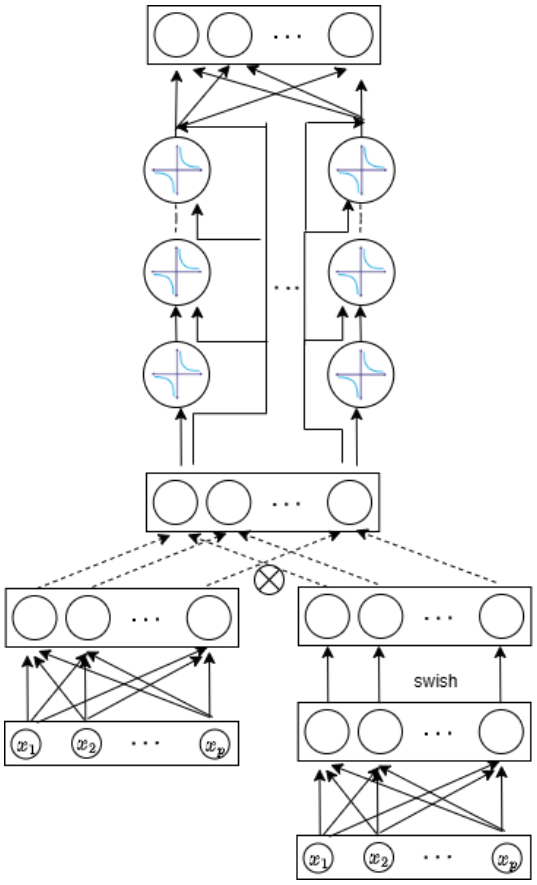


Figure A1: CoFrGeNet-FNG architecture replacing FFNs in a transformer block. We create a gated *non-expanded* (i.e. $\alpha = 1$) representation that we pass to the CoFrNet ladders. For each of the continued fraction ladders, we use our parameterization from equation 4, thus forming Spec-CoFrGeNet .