Normalized Architectures are Natively 4-Bit

Maxim Fishman [^*] **Brian Chmiel** [^*] **Ron Banner** [^] **Daniel Soudry** [^°] **Boris Ginsburg** [^]

[^] NVIDIA

[°] Department of Electrical and Computer Engineering - Technion, Haifa, Israel

{mfishman, bchmiel, rbanner, bginsburg}@nvidia.com
{daniel.soudry}@gmail.com

Abstract

Training large language models at 4-bit precision is critical for efficiency. We show that nGPT, an architecture that constrains weights and hidden representations to the unit hypersphere, is inherently more robust to low-precision arithmetic. This removes the need for interventions—such as applying random Hadamard transforms and performing per-tensor scaling calculations—to preserve model quality, and it enables stable end-to-end NVFP4 training. We validate this approach on both a 1.2B dense model and hybrid (Mamba-Transformer) MoE models of up to 3B/30B parameters. We trace this robustness to the dot product: while quantization noise remains largely uncorrelated in both standard and normalized architectures, the signal behaves differently. In nGPT, the hypersphere constraint enhances weak positive correlations among the element-wise products, leading to a constructive accumulation of the signal across the hidden dimension while the noise continues to average out. This yields a higher effective signal-to-noise ratio and a flatter loss landscape, with the effect strengthening as the hidden dimension grows, suggesting increasing advantages at scale. A reference implementation is available at `https://github.com/anonymous452026/ngpt-nvfp4`.

1 Introduction

Deploying large transformers at 4-bit precision is becoming essential for efficiency, yet low-bit training remains fragile. In practice, standard transformer architectures often require fixes such as randomized Hadamard transforms (RHT), dynamic per-tensor scaling, or mixed-precision exceptions to maintain model quality [3]—which all require overhead. These interventions can make 4-bit training possible, but they also raise a more fundamental question: Can we find an architecture that has an intrinsic 4-bit quantization robustness without requiring these quantization tricks?

In this work, we show that quantization robustness can be architectural. We study **nGPT** [10], a transformer that constrains hidden states and model parameters to the unit hypersphere, and find that this normalization makes the model natively robust to NVFP4 arithmetic. We focus on this low-precision format since it is supported natively by NVIDIA's Blackwell GPUs [2]. Across diverse architectures, including a 1.2B dense model and hybrid Mamba-Transformer Mixture-of-Experts (MoE) configurations ranging from 400M/600M to 3B/30B, nGPT supports stable end-to-end full NVFP4 training, without RHT, without dynamic per-tensor scaling, and without the divergence typically observed in standard transformer baselines when these operations are omitted.

The central question is why. At first glance, one might expect quantization robustness to come from reducing quantization noise. That is the logic behind most existing approaches: control outliers, im-

[*]Equal contribution

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

prove rounding, rescale tensors, or selectively restore higher precision where noise is too large. Our analysis points to a different mechanism. In nGPT, the advantage does not come from making the quantization noise unusually small. It comes from making the *signal* accumulate more effectively.

To trace this effect, we perform a layer-wise structural analysis on a 3.6B parameters transformer, using NVFP4 quantization. We find that quantization noise remains largely uncorrelated in both standard GPT and nGPT. The key difference appears in the signal. In nGPT, the hypersphere constraint induces weak but consistent positive correlations among the element-wise products inside the dot product. These correlations are tiny at the level of any single coordinate, but they act coherently across thousands of dimensions. As a result, the true dot product grows more constructively, while the quantization noise continues to average out like an incoherent random walk.

This behavior is a direct consequence of the training dynamics under the hypersphere constraint. In a standard GPT, the network can rely on a small number of large, unbounded coordinates to dominate the dot product. In nGPT, this path is blocked: because both activations and weights are normalized, no single element can arbitrarily scale the output. Instead, to produce a large dot product, we observe empirically that the model creates alignment across many coordinates. Specifically, although the marginal distributions of individual weights and activations are similar in both architectures (Fig. 1a), the element-wise products in nGPT exhibit a systematic positive correlation across coordinates that is absent in GPT. That structural bias is weak per coordinate, but overall, across the dimensions it produces a robust signal drift that low-precision noise cannot easily disrupt.

This leads to a different view of low-bit robustness. The main determinant is not the suppression of local quantization error, but the coherence of signal accumulation. Under the hypersphere constraint, the signal scales more like a coordinated sum, while the noise remains essentially incoherent. This creates a higher effective dot-product SNR, and the effect compounds with depth into a substantially flatter loss landscape. In other words, nGPT is structurally robust to low-bit quantization not as a side effect, but as a direct consequence of how normalized dot products accumulate signal.

Our results therefore suggest a shift in emphasis for low-precision model design. Much of the literature treats quantization as a compression problem applied after the architecture is fixed. Our findings suggest that the architecture itself can be made quantization-ready. No post-hoc quantization trick is needed to create the effect we observe. The hyperspherical geometry already biases the model toward a representation in which signal survives 4-bit arithmetic unusually well. As training and inference continue to move toward 4-bit precision and below, this structural property may prove more valuable than increasingly elaborate correction mechanisms layered on top of standard architectures.

Our main contributions are:

- **Architecture-driven robustness:** We show that nGPT is natively "quantization-ready", enabling stable end-to-end NVFP4 training. Unlike standard Transformers, it requires no fixes that introduce overhead like randomized Hadamard transforms (RHT) or dynamic per-tensor scaling to keep model quality.

- **The Signal-Accumulation Mechanism:** We identify that nGPT's robustness stems from signal coherence rather than noise suppression. By forcing the signal to accumulate constructively across thousands of dimensions, nGPT increases the SNR per layer, allowing the true dot-product signal to outpace incoherent quantization noise.

- **Driven by Training Dynamics:** We demonstrate that this robustness is a direct consequence of training dynamics under the unit hypersphere constraint. By blocking the model from relying on a few large, unbounded coordinates to drive outputs, nGPT forces the optimizer to learn distributed alignments across thousands of dimensions. This structural bias ensures the signal remains coherent even under heavy 4-bit quantization.

- **Empirical Validation across Architectures:** We confirm these advantages across diverse configurations, including a 1.2B dense model and a hybrid Mamba-Transformer Mixture-of-Experts (MoE) architectures of 400m/600m and 3B/30B. In all cases, nGPT maintains stability and achieves lower relative error than standard transformers.

2 Related works

The scaling of large language models into the trillion-parameter regime has driven a massive shift toward low-precision arithmetic, specifically utilizing NVIDIA's NVFP4 format. The first work to successfully demonstrate the viability of fully quantized NVFP4 pretraining was [6]. To further reduce the inherent quantization error of NVFP4 formats, subsequent methods such as [8] introduced adaptive block scaling, modifying the NVFP4 algorithm to make the distribution of representable values more uniform and reduce error for near-maximal values. The work most directly related to ours–and the current benchmark for stable 4-bit pretraining—is [3]. This work establishes the state-of-the-art recipe for NVFP4 in large scale, demonstrating that to prevent divergence and handle severe block-level outliers, standard Transformers require a complex and overhead-heavy intervention pipeline. Specifically, this "best recipe" relies on Randomized Hadamard Transforms (RHT) to computationally disperse outlier features, dynamic per-tensor scaling, and Stochastic Rounding (SR) for unbiased gradient estimation. Other concurrent efforts have attempted to mitigate 4-bit quantization errors through differentiable quantization estimators [14]. Furthermore, frameworks such as [5, 12] have advanced native FP4 pre-training by constructing unbiased gradient estimators designed specifically to minimize the variance inherent in low-precision backpropagation.

All of these methods treat quantization robustness as an algorithmic challenge to be patched via post-hoc scaling, rather than as an intrinsic architectural property. In contrast, our investigation focuses on the structural properties of the normalized Transformer (nGPT) [10], an architecture that constrains all representations and weights to the unit hypersphere. While nGPT was originally introduced to accelerate training, recent literature has begun to explore the broader geometric implications of this design. For instance, recent work on transferable hypersphere optimization [13] has re-evaluated language model scaling under these conditions, demonstrating that hyperspherical constraints facilitate stable optimization, predictable scaling laws, and hyperparameter transferability across different model sizes. However, despite these advancements in scaling, no prior work has analyzed nGPT's behavior in the context of low-precision arithmetic. To the best of our knowledge, we are the first to demonstrate that nGPT's inherent hyperspherical constraints make it natively robust to 4-bit quantization, entirely bypassing the need for overhead-heavy mechanisms.

Our empirical finding that nGPT's normalization leads to a flatter loss landscape with robust signal-to-noise ratios connects to a rich body of theoretical literature on optimization geometry. [15] showed that weight normalization introduces implicit regularization that guides overparameterized models towards flatter, minimum-norm solutions. Furthermore, additional works reveal that structural constraints inherently bias gradient descent toward flatter minima, as rigorously studied in both linear diagonal [7] and homogeneous nonlinear neural networks [11]. Our work provides a critical link between these theoretical observations of hypersphere geometry and the practical demands of ultra-low-precision LLM training.

3 Analysis of nGPT Quantization Robustness

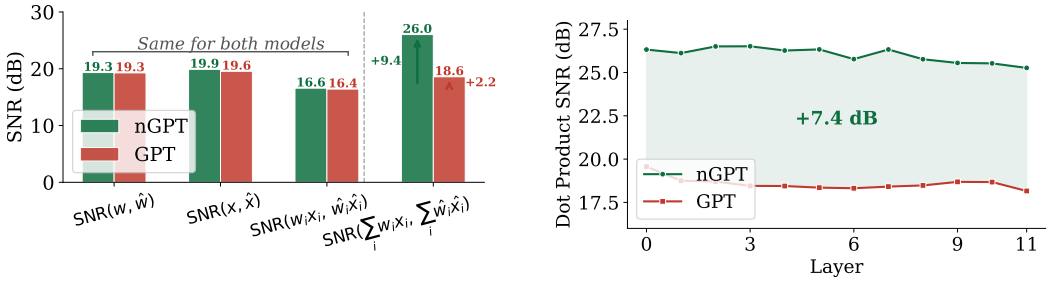
In the following section, we trace the origins of nGPT robustness using a 12-layer transformer with a hidden dimension of $D = 4096$, totaling 3.6B parameters, built on the nanoGPT baseline [9]. We first demonstrate that this robustness stems from the signal-to-noise ratio (SNR) in the summation. We then attribute the improved SNR to a stronger underlying signal, and finally we show it is driven by an enhanced correlation between weights and activations in nGPT.

3.1 Where Does the Robustness Come From?

Every linear layer computes the matrix multiplication $Y = WX$. Each scalar output element y is a dot product $y = \sum_{k=1}^{D} w_k x_k$ of D=4096 terms. To locate where nGPT's advantage arises, we measure the Signal-to-Noise Ratio (SNR) at each stage of this computation under NVFP4 quantization (Fig. 1a). We define the SNR(dB) for a given tensor T and its quantized counterpart $\hat{T}$ as:

$$\text{SNR} = \frac{\|T\|_2^2}{\|T - \hat{T}\|_2^2}\,. \tag{1}$$

When dB units are used, we replace SNR with $10\log_{10}(\text{SNR})$.



(a) (b)

Figure 1: **(a):** SNR at each stage of a matrix multiplication under NVFP4, averaged over all 12 layers. The first three stages (individual weights, activations, and their products) are identical for both models. The difference appears only at the final summation step. **(b):** Dot product SNR under NVFP4 at each layer. nGPT maintains a consistent $\sim$7 dB advantage in all layers. Both graphs are based on a 3.6B model, built on the nanoGPT baseline [9].

Before the sum: no difference. The first three groups in Figure 1a show the SNR of individual quantities: weights (w vs $\hat{w}$), activations (x vs $\hat{x}$), and their element-wise products ($w_k x_k$ vs $\hat{w}_k \hat{x}_k$). In all three cases, nGPT and GPT are nearly identical—both around 19 dB for elements and 16.5 dB for products. Normalization does not make individual values easier to quantize.

After the sum: 7 dB gap. The rightmost group shows the SNR of the full dot product $\sum_k w_k x_k$ vs $\sum_k \hat{w}_k \hat{x}_k$. Here the models diverge sharply: nGPT achieves 26 dB; GPT only 18.6 dB. The improvement from products to dot product—the *averaging gain*—is +9.4 dB for nGPT but only +2.2 dB for GPT.

The advantage is entirely in the summation. This is the central finding: normalization does not reduce quantization noise on individual elements or products—those are identical across architectures. What changes is how all terms behave when added together. Rather than the noise canceling out, the *signal* in nGPT accumulates constructively during summation, whereas in standard GPT, it does not. Because this coherent signal grows much faster than the accumulated noise, nGPT achieves a fundamentally higher signal-to-noise ratio. The next section explains why.

The gap is consistent across all layers. Figure 1b shows the dot product SNR at each of the 12 layers individually averaged across all forward matmuls. nGPT maintains 25–26 dB uniformly; GPT achieves 18–19 dB. The advantage is not concentrated in particular layers—it is a structural property present throughout the network.

3.2 What drives the SNR in the summation?

To isolate the structural drivers of dot product SNR, we decompose each matrix multiplication row into a normalized signal and a normalized noise component. Let $s_k = w_k x_k$ denote the k-th element-wise product, and let $n_k = \hat{w}_k \hat{x}_k - w_k x_k$ represent the corresponding quantization error, where $\hat{w}$ and $\hat{x}$ are the low-precision quantized values. We define the normalized signal (z_s) and normalized noise ($\tilde{z}_n$) by utilizing the signal's random-walk standard deviation as a shared scaling factor:

$$z_s = \frac{\left|\sum_{k=1}^{D} s_k\right|}{\sqrt{D}\,\sigma_s}, \qquad \tilde{z}_n = \frac{\left|\sum_{k=1}^{D} n_k\right|}{\sqrt{D}\,\sigma_s}, \tag{2}$$

where

$$\sigma_s^2 = \frac{1}{D-1}\sum_{k=1}^{D}(s_k - \bar{s})^2, \qquad \bar{s} = \frac{1}{D}\sum_{k=1}^{D} s_k$$

Under this formulation, the dot product SNR factorizes as $\mathrm{SNR}_{\mathrm{dot}} = (z_s/\tilde{z}_n)^2$.

Figure 2 illustrates the relationship between these decomposed quantities and the empirical dot product SNR evaluated across validation activations. The analysis yields a clear structural distinction:

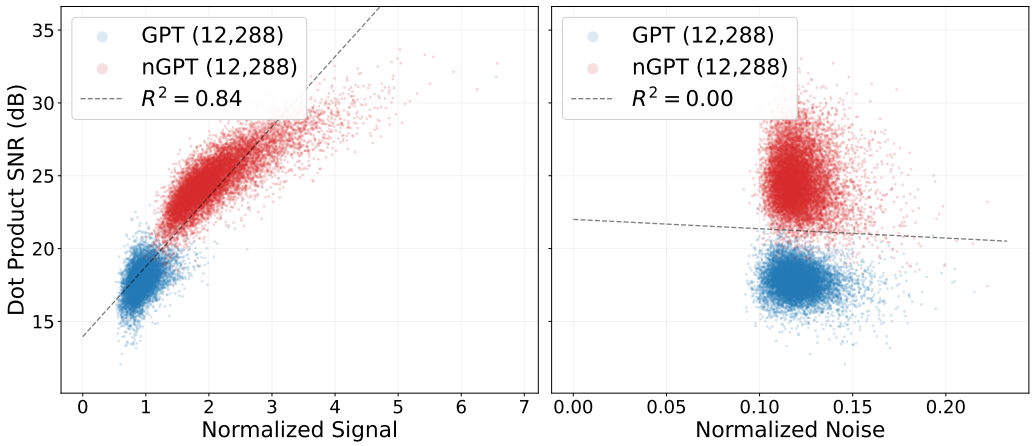


Figure 2: **Normalized signal vs. normalized noise as predictors of dot product SNR.** Evaluated in MLP layers under NVFP4 quantization after training. **Left:** The normalized signal z_s demonstrates a strong positive correlation with SNR, cleanly separating the highly coherent representations of nGPT (red) from the baseline GPT (blue). **Right:** The normalized noise $\tilde{z}_n$ is nearly identical across both architectures and exhibits no predictive power over the final SNR. This demonstrates that nGPT's SNR advantage originates entirely from coherent signal accumulation rather than structural noise reduction.

the normalized signal z_s acts as a robust predictor of the overall SNR, establishing a distinct separation between nGPT and standard GPT representations. In stark contrast, the normalized noise $\tilde{z}_n$ remains statistically indistinguishable between the two architectures and demonstrates no predictive capacity for the final SNR. **nGPT's superior SNR is driven exclusively by a more robust signal representation rather than a reduction in noise**.

3.3 Why is the Signal Stronger in nGPT?

We now trace the SNR advantage to the structure of the dot-product summation itself. First, we measure statistical dependence between the individual signal and noise terms. Next, we show that nGPT introduces weak but systematic positive correlations in the signal, whereas the noise remains near-uncorrelated in both architectures. Finally, we explain why these correlations strengthen the signal: when positively correlated terms are summed over many dimensions, they accumulate more coherently and produce a larger dot product.

3.3.1 Measuring Dependence Between Elements in the Dot-Product Sum

We begin by directly measuring whether the elements in the dot-product sum behave independently or exhibit statistical dependence. If the terms s_k are independent, the dot product behaves like a sum of unrelated contributions. However, if they are positively correlated, they tend to increase and decrease together, leading to more coherent accumulation.

To quantify this, we define an effective pairwise correlation directly from the variance decomposition $\mathrm{Var}(\sum_k u_k) = \sum_k \mathrm{Var}(u_k) + 2\sum_{j<k} \mathrm{Cov}(u_j, u_k)$:

$$\bar{\rho}_u \;=\; \frac{1}{D-1}\left(\frac{\mathrm{Var}(\sum_{k=1}^{D} u_k)}{\sum_{k=1}^{D}\mathrm{Var}(u_k)} \;-\; 1\right), \qquad u \in \{s, n\}, \tag{3}$$

where $\mathrm{Var}(\,\cdot\,)$ in Eq. (3) denotes variance taken across input samples x. This quantity is the variance-weighted average of pairwise Pearson correlations between elements of the sum, and is equivalent to the standard average correlation when all $\mathrm{Var}(u_k)$ are comparable.

As shown in Fig. 3, GPT exhibits small signal correlation ($\bar{\rho}_s = 9.31 \times 10^{-5}$), while nGPT shows a consistently larger correlation ($\bar{\rho}_s = 1.32 \times 10^{-3}$, about $14\times$ larger). In contrast, noise correlations remain small in both models ($\bar{\rho}_n \sim 10^{-6}$).

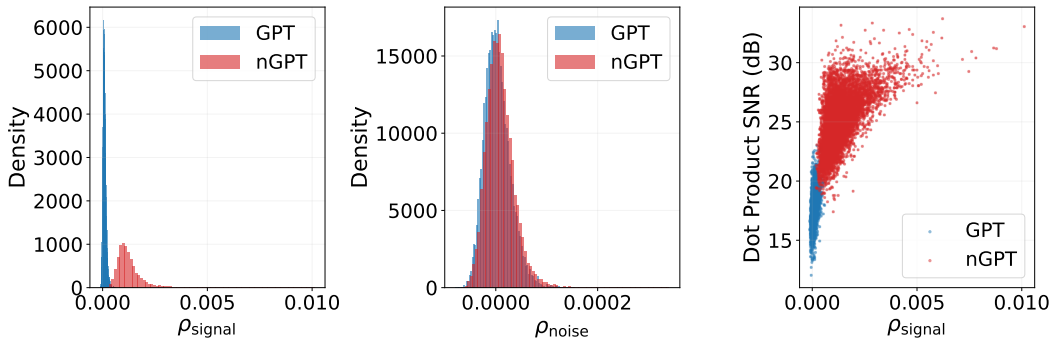


Figure 3: Mean-centered pairwise Pearson correlation of quantized dot-product elements (NVFP4) after training. **Left:** Distribution of signal correlation $\bar{\rho}_s$; nGPT is shifted toward positive values. **Middle:** Distribution of noise correlation $\bar{\rho}_n$; both architectures remain near zero. **Right:** $\bar{\rho}_s$ versus dot-product SNR; higher signal coherence is associated with higher SNR.

3.3.2 Why correlation makes the signal stronger?

In Section B.1 we show that the SNR of the dot product is:

$$\text{SNR} = \frac{\mathbb{E}[S^2]}{\mathbb{E}[N^2]} = \frac{\text{Var}(S) + \mu_S^2}{\text{Var}(N) + \mu_N^2} \approx \frac{D\sigma_s^2\big(1 + (D-1)\rho_s\big) + \mu_S^2}{D\sigma_n^2 + \mu_N^2}. \tag{4}$$

where $\mu_S = \mathbb{E}[S]$ and $\mu_N = \mathbb{E}[N]$. This clearly shows how the positive correlation ρ_s acts as a multiplier on the baseline SNR.

Summary. The SNR advantage in nGPT arises from the collective summation of elements rather than individual contributions; this stems not from noise reduction, but from enhanced signal correlation that is inherently stronger in the nGPT architecture. To confirm that increased correlation is a phenomenon driven by nGPT's training dynamics, we trained a one-layer model in Section B.3. The results demonstrate that while both GPT and nGPT achieve comparable training loss, nGPT exhibits higher correlation.

3.4 Scaling of the SNR Advantage with Width

We found empirically that $\mu_S^2/\sigma_s^2 \ll D$ and $\mu_N^2/\sigma_n^2 \ll D$ and that the ratio σ_s^2/σ_n^2 is matched across architectures. Therefore, we can use Eq. (4) to calculate the SNR ratio as follows:

$$\frac{\text{SNR}_{\text{nGPT}}(D)}{\text{SNR}_{\text{GPT}}(D)} \approx \frac{1 + D\,\bar{\rho}_{\text{nGPT}}}{1 + D\,\bar{\rho}_{\text{GPT}}}. \tag{5}$$

This ratio predicts how much more nGPT benefits from increasing width.

Because $\bar{\rho}_{\text{GPT}} \ll \bar{\rho}_{\text{nGPT}}$, Eq. (5) exhibits three distinct regimes: small, intermediate, and large width. In Section B.2 we analyze these regimes.

Empirical validation. Using the measured values from Section 3.3.1 the two transition points are

$$\frac{1}{\bar{\rho}_{\text{nGPT}}} \approx 755, \qquad \frac{1}{\bar{\rho}_{\text{GPT}}} \approx 10{,}738,$$

and the saturation value is $\bar{\rho}_{\text{nGPT}}/\bar{\rho}_{\text{GPT}} \approx 14.2$. A key empirical finding is that $\bar{\rho}_s$ is essentially *constant* across all partial summation lengths $k \in [16, 4096]$ for both models. This means a single measured value of $\bar{\rho}_s$ is sufficient to predict the full width-scaling behavior. Investigating the underlying mechanisms that maintain this constant correlation and why it is inherently higher within the nGPT architecture remains a compelling direction for future research.

Figure 4(a) shows the absolute SNR for both models together with the theory curves from Eq. (4) using the measured $\bar{\rho}_s$. Figure 4(b) shows the ratio in Eq. (5), with the three regimes marked.

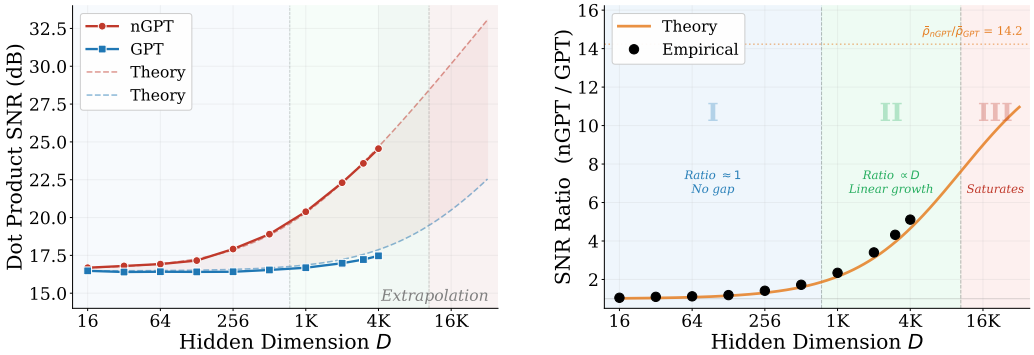


Figure 4: **Three-regime scaling of nGPT's SNR advantage. (Left):** Dot-product SNR vs. summation length D for both architectures, with theory curves from the measured $\bar{\rho}_s$ (dashed). The theory, which has no free parameters beyond the single measured $\bar{\rho}_s$, closely tracks the data across two orders of magnitude in D. **(Right):** SNR ratio (nGPT/GPT) vs. D, showing the three predicted regimes: **I** (blue, $D \ll 755$): no gap; **II** (green, $755 \ll D \ll 10738$): linear growth; **III** (red, $D \gg 10738$): saturation at $\bar{\rho}_{\mathrm{nGPT}}/\bar{\rho}_{\mathrm{GPT}} \approx 14.2$. Current models operate in Regime II; the advantage is expected to continue growing at larger scales.

Current models (D=4096) sit firmly in Regime II, where the gap is still growing. At D=16384 (405B-scale), the theory predicts the ratio will approach $10\times$.

The practical implication is direct: *larger models benefit more from nGPT*, because positive signal correlation makes SNR grow faster with width.

4 From Per-Layer SNR to a Flat Loss Landscape

The averaging gain is measured per layer, but quantization affects all layers simultaneously. We now show that the per-layer advantage directly predicts a flatter loss landscape for nGPT.

Per-layer: $\sim$ 7 dB improvement in SNR. Figure 8(a) shows the gain (dB):

$$\text{gain(dB)} = \text{SNR}(\sum_i w_i x_i,\ \sum_i \hat{w}_i \hat{x}_i) - \text{SNR}(w_i x_i,\ \hat{w}_i \hat{x}_i) \tag{6}$$

at each of the layers. At every matrix multiplication, nGPT suppresses quantization noise more effectively than GPT—not because the noise is smaller, but because the signal accumulates faster.

Full network: 3.5$\times$ flatter. When all layers are perturbed simultaneously—as happens during quantized training—the per-layer effects compound. To measure this directly, we add Gaussian noise to all weights at scale $\alpha \cdot \|W\|_F/\sqrt{n}$, where n is the number of entries in W, and record the validation loss increase (Figure 8b). GPT's loss degrades $3.5\times$ faster than nGPT's. Since quantization is a structured perturbation applied to every weight and activation, a flatter landscape means quantization has less impact on the loss.

This connection is direct: a model with higher averaging gain at each layer tolerates more perturbation per layer, and therefore tolerates more total perturbation across the full network.

5 Experiments

We now present our empirical evaluation demonstrating nGPT's robustness to NVFP4 quantization. The quantization is applied across three GEMMs: Fprop, Dgrad and Wgrad. We first analyze the training dynamics and downstream task performance of a 1.2B parameter dense model. Following this, we extend our evaluation to a Hybrid Mamba-Transformer MoE, to confirm that nGPT structural advantages hold across different model topologies. All training was conducted using the Nemotron-Pretraining-Datasets [1] , AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$. All experiments use Blackwell GPUs, which natively support NVFP4 datatype. In Table 3 we compare the different

training overhead operations, While in nGPT-NVFP4 runs we eliminate the need for Randomized Hadamard Transforms (RHT) and per-tensor scaling, the standard NVFP4 still keeps these operations to avoid divergence [3]. Relative error is defined as $(\text{Loss}_{\text{quantized}} - \text{Loss}_{\text{BF16}})/\text{Loss}_{\text{BF16}}$. In Section A we show the changes required to transform a standard architecture to nGPT.

5.1 1.2B dense

Fig. 5(a) shows the relative error of a 1.2B dense model trained on 1T tokens using NVFP4 and nNVFP4. A key advantage of the nGPT architecture is its ability to mitigate the overhead of specific quantization operations, such as RHT and per-tensor-scaling while improving the relative error. As shown in Table 1 , the normalized architecture consistently outperforms the standard baseline across various downstream tasks in both BF16 and NVFP4 precisions. In Table 4 we show the hyperparameters used.

Table 1: Downstream tasks evaluation of 1.2B dense model. Note that the normalized architecture outperforms the standard architecture over all tasks, both in BF16 and NVFP4 datatype.

Config	Loss ↓	Hell.↑	PIQA ↑	Win. ↑	MMLU ↑ (5-shot)	MMLU Pr ↑ (5-shot)	GSM8K ↑ (8-shot)
bf16	1.47	60.54	73.72	58.17	45.23	13.98	33.66
n bf16	**1.46**	**61.57**	**74.27**	**59.51**	**45.50**	**14.34**	**36.24**
nvfp4	1.52	58.86	73.78	58.64	42.56	13.06	29.49
n nvfp4	**1.50**	**59.93**	**74.37**	58.64	**43.57**	**14.59**	**35.56**

5.2 Hybrid Mamba-Transformer MoE

400m/600m In Fig. 9, we compare the training loss and relative error for a Hybrid Mamba-Transformer MoE model (400M/600M) trained over 36B tokens. These results highlight that the benefits of nGPT architecture extend to Hybrid Mamba-Transformer MoE configurations; specifically, it achieves a lower relative error while improving the pipeline through the removal of RHT and per-tensor scaling overhead. In Table 5 we show the hyperparameters used.

3B/30B In Fig. 5(b), we compare the relative error for a Hybrid Mamba-Transformer MoE model (3B/30B) trained over $\sim$ 500B tokens of a 1T token horizon, achieving $\sim$ 0% relative error for nGPT. These results highlight that the benefits of the normalized architecture also extends to large models. In Table 2 we showed the normalized architecture outperforms the standard baseline across most of downstream tasks also with hybrid MoE architecture. Following the methodology in [3], the NVFP4 experiments maintain high precision for the final 15% of layers in both GPT and nGPT architectures. In Fig. 10 we show that both BF16 training runs achieved similar training loss. In Table 6 we show the hyperparameters used.

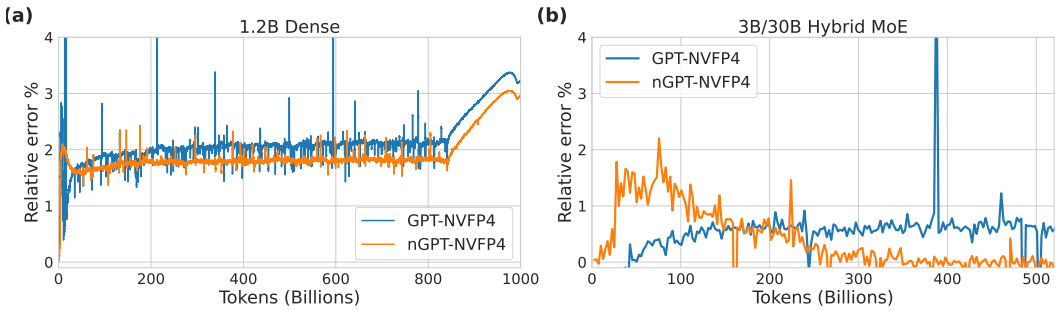
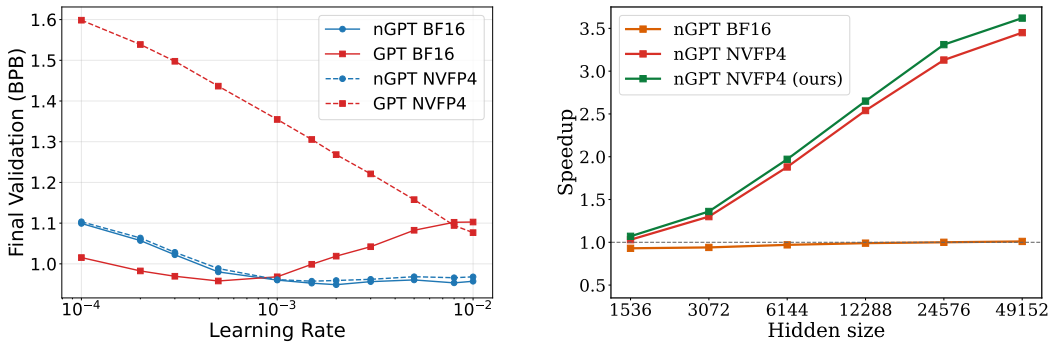


Figure 5: **(a):** 1.2B dense model relative error (1T tokens); nGPT-NVFP4 reduces the loss gap and eliminates RHT/scaling overhead. **(b):** 3B/30B hybrid MoE relative error; nGPT achieves $\sim$0% relative error without standard NVFP4 interventions (RHT and per-tensor scaling).

Table 2: Downstream tasks evaluation of 3B/30B hybrid MoE model. Note that the normalized architecture outperforms the standard architecture over most tasks, both in BF16 and NVFP4 datatype.

Config	Hell.↑	PIQA ↑	Win. ↑	MMLU ↑ (5-shot)	MMLU Pr ↑ (5-shot)	GSM8K ↑ (8-shot)
bf16	76.9	80.3	68.43	60.13	36.44	62.17
n bf16	**77.51**	**81.18**	**70.72**	**63.04**	**37.83**	**66.57**
nvfp4	76.54	**81.28**	69.22	59.14	32.78	58.68
n nvfp4	**77.63**	80.3	**69.30**	**63.05**	**38.17**	**64.82**



(a) (b)

Figure 6: **(a):** Validation BPB vs. learning rate for nGPT and GPT. nGPT maintains a flat optimum across precisions, while GPT is LR-sensitive under quantization. nGPT's optimal LR transfers directly from BF16 to NVFP4. **(b):** One-layer training speedup on a GB200 GPU as a function of hidden size. Speedup is measured relative to the BF16 GPT layer baseline. The nGPT NVFP4 configuration labeled "ours" removes both dynamic per-tensor amax scaling computation and RHT.

5.3 nGPT Learning Rate Robustness Under Quantization

A practical consequence of the higher dot-product SNR (Section 3.1) is that nGPT should tolerate a wider range of hyperparameters under quantization. To test this, we sweep the learning rate over two orders of magnitude for all four configurations: nGPT and GPT, each in BF16 and NVFP4. We run the same model as in Section 3 and use the final validation bits-per-byte (BPB) as a normalized metric to compare between the runs. Figure 6a reveals two findings:

nGPT is remarkably LR-insensitive. In BF16, nGPT achieves nearly identical validation BPB across the entire range. GPT, by contrast, peaks sharply and degrades rapidly at higher rates, reaching a wider spread. This flat minimum is consistent with the flatter loss landscape predicted by the per-layer SNR analysis (Fig. 8b): a model that is less sensitive to weight perturbations is also less sensitive to the effective perturbation introduced by a suboptimal learning rate.

The flat minimum transfers to NVFP4. Under NVFP4 quantization, nGPT maintains the same flat profile. The best NVFP4 learning rate for nGPT coincides with the BF16 optimum, with minimal BPB difference at the optimal LR. This means the BF16-optimal hyperparameters can be directly transferred to NVFP4 training without retuning-a significant practical advantage, since hyperparameter sweeps at low precision are expensive. GPT shows the opposite behavior: its NVFP4-optimal LR is $16\times$ larger than the BF16 optimum, requiring a separate sweep.

5.4 Acceleration

The same structural property that improves quantization robustness also simplifies the runtime path. In the standard NVFP4 recipe, activations must be dynamically rescaled and RHT are used to smooth

outliers before quantization. nGPT hidden states are already constrained to a bounded hypersphere, so the activation scale can be fixed and the RHT path can be removed. We isolate this effect with the single-layer benchmark on a Blackwell GB200 GPU. Details appear in Section C.4.

BF16 nGPT has negligible layer overhead. The BF16 nGPT curve remains close to the BF16 GPT baseline across the sweep, indicating that the normalization and interpolation operations themselves do not dominate the layer runtime once the fused kernels are used. The acceleration comes from moving the expensive GEMMs to NVFP4, not from changing the BF16 architecture.

The NVFP4 benefit grows with width. Figure 6b shows that the nGPT NVFP4 path becomes increasingly favorable as the hidden dimension grows. At small widths, fixed costs from Python, autograd, and CUDA launches are still visible; and dominate the runtime. At larger widths, GEMM time dominates and the benefit of Blackwell NVFP4 tensor cores becomes clear: the optimized nGPT NVFP4 path reaches roughly 3.3–$3.6\times$ speedup over the BF16 GPT baseline at the largest hidden sizes. Removing both the per-tensor amax reduction and RHT consistently tracks the fastest nGPT NVFP4 configuration in the high-throughput regime, showing that the overhead eliminated by nGPT is not only useful for model quality, but also visible in the measured layer time.

6 Conclusions

In this work, we have demonstrated that the robustness of normalized transformers to low-precision arithmetic is not merely a function of quantization algorithms, but a fundamental property of model geometry. Our structural analysis reveals that nGPT's robustness does not stem from a reduction in local quantization noise, but from a superior signal accumulation mechanism. By inducing weak but consistent positive correlations among element-wise products, the architecture ensures that the intended signal accumulates constructively across high-dimensional dot products. The source of this positive correlation in normalized architectures is left for future work.

We identify a unique scaling advantage where nGPT's signal-to-noise ratio (SNR) grows faster with model width compared to standard architecture. This suggests that as models scale toward larger hidden dimensions, the architectural benefits of normalization become increasingly pronounced.

A critical practical outcome of this structural robustness is nGPT's remarkable insensitivity to LR selection. While standard GPT models exhibit sharp sensitivity to learning rate shifts under quantization—often requiring a large shift in optimal LR when moving from BF16 to NVFP4—nGPT maintains a broad, flat optimum.

Our empirical evaluations demonstrate that these advantages are consistent across diverse and large-scale architectures. In a 1.2B dense model trained on 1T tokens, nGPT in NVFP4 achieved lower relative error and superior performance across various downstream benchmarks, despite the removal of RHT and per-tensor scaling overhead. These benefits extend to Hybrid Mamba-Transformer MoE configurations, where evaluations of up to 3B/30B variants showed that the normalized architecture maintains significantly lower relative error under NVFP4 quantization than standard baselines. Our results demonstrate that nGPT's reduced overhead provides a dual benefit: maintaining high model quality while significantly accelerating layer-wise throughput.

Acknowledgments and Disclosure of Funding

The research of DS was funded by the European Union (ERC, A-B-C-Deep, 101039436). Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (ERCEA). Neither the European Union nor the granting authority can be held responsible for them. The research of DS was also supported by the Council for Higher Education in Israel under the Moonshot Project.

References

[1] Nemotron pretraining data - huggingface. URL https://huggingface.co/collections/nvidia/nemotron-pre-training-datasets.

[2] Nvidia blackwell architecture. URL https://resources.nvidia.com/en-us-blackwell-architecture.

[3] Felix Abecassis, Anjulie S. Agrusa, Dong Ahn, Jonah Alben, et al. Pretraining large language models with nvfp4. *ArXiv*, abs/2509.25149, 2025. URL https://api.semanticscholar.org/CorpusID:281674055.

[4] Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, and et al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *ArXiv*, abs/2512.20848, 2025. URL https://api.semanticscholar.org/CorpusID:283936671.

[5] Roberto L. Castro, Andrei Panferov, Soroush Tabesh, Oliver Sieberling, Jiale Chen, Mahdi Nikdan, Saleh Ashkboos, and Dan Alistarh. Quartet: Native fp4 training can be optimal for large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/pdf?id=XMzxZ6h68o.

[6] Brian Chmiel, Maxim Fishman, Ron Banner, and Daniel Soudry. Fp4 all the way: Fully quantized training of llms. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/pdf?id=kuzye4EPLR.

[7] Hung-Hsu Chou, Holger Rauhut, and Rachel Ward. Robust implicit regularization via weight normalization. *Information and Inference: A Journal of the IMA*, 13(3):iaae022, 2024. doi: 10.1093/imaiai/iaae022. URL https://doi.org/10.1093/imaiai/iaae022.

[8] Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, and Song Han. Four over six: More accurate nvfp4 quantization with adaptive block scaling, 2025. URL https://arxiv.org/abs/2512.02010.

[9] Andrej Karpathy. nanogpt. https://github.com/karpathy/nanoGPT, 2022.

[10] Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.01131.

[11] Depen Morwani and Harish G. Ramaswamy. Inductive bias of gradient descent for weight normalized smooth homogeneous neural nets. In Sanjoy Dasgupta and Nika Haghtalab, editors, *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 827–880. PMLR, 29 Mar–01 Apr 2022. URL https://proceedings.mlr.press/v167/morwani22a.html.

[12] Andrei Panferov, Erik Schultheis, Soroush Tabesh, and Dan Alistarh. Quartet ii: Accurate llm pre-training in nvfp4 by improved unbiased gradient estimation, 2026. URL https://arxiv.org/abs/2601.22813.

[13] Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking language model scaling under transferable hypersphere optimization. 2026. URL https://api.semanticscholar.org/CorpusID:286974216.

[14] Ruizhe Wang, Yeyun Gong, Xiao Liu, Guoshuai Zhao, Ziyue Yang, Baining Guo, Zhengjun Zha, and Peng Cheng. Optimizing large language model training using fp4 quantization. *ArXiv*, abs/2501.17116, 2025. URL https://api.semanticscholar.org/CorpusID:275932373.

[15] Xiaoxia Wu, Edgar Dobriban, Tongzheng Ren, Shanshan Wu, Zhiyuan Li, Suriya Gunasekar, Rachel Ward, and Qiang Liu. Implicit regularization and convergence for weight normalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2835–2847. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1de7d2b90d554be9f0db1c338e80197d-Paper.pdf.

A nGPT architecture

The architectural modifications introduced by nGPT [10] are outlined below. The highlighted box emphasizes the weight and activation normalization steps, which are fundamental to our low-precision nGPT regime. Notably, in our approach, we normalize all inputs. This contrasts with the official nGPT implementation, which omits normalization for the inputs to the out projection GEMM in the attention block and the FFN2 GEMM in the MLP block.

1. **Remove** RMSNorm / LayerNorm layers

2. **Normalize** all weights along embedding dim (after each step)
3. **Replace residual updates:**

$$h \leftarrow \text{Norm}(h+\alpha_A(\text{Norm}(\text{ATTN}(h))-h)), \quad h \leftarrow \text{Norm}(h+\alpha_M(\text{Norm}(\text{MLP}(h))-h))$$

$$\alpha_A, \alpha_M \in R^{d_{\text{model}}}_{\geq 0} \text{ learnable}; \quad \alpha_{\text{init}} = 0.05, \quad \alpha_{\text{scale}} = 1/\sqrt{d_{\text{model}}}$$

4. **Attention:** softmax scale $1/\sqrt{d_k} \rightarrow \sqrt{d_k}$; normalize & rescale q, k:

$$q \leftarrow \text{Norm}(q) \cdot s_{qk}, \quad k \leftarrow \text{Norm}(k) \cdot s_{qk} \qquad (s_{qk-\text{init}} = 1,\ s_{qk-\text{scale}} = 1/\sqrt{d_{\text{model}}})$$

5. **MLP rescaling:**

$$u \leftarrow u \cdot s_u, \quad \nu \leftarrow \nu \cdot s_\nu \sqrt{d_{\text{model}}} \qquad (s_{u,v-\text{init}} = 1,\ s_{u,v-\text{scale}} = 1)$$

6. **Logit rescaling:** $z \leftarrow z \cdot s_z \qquad (s_{z\text{init}} = 1,\ s_{z\text{scale}} = 1/\sqrt{d_{\text{model}}})$
7. **Remove** weight decay and LR warmup

B Analysis of nGPT quantization robustness - proofs

B.1 SNR of the dot product

To understand the effect of this difference, consider the sum of the signal terms

$$S = \sum_{k=1}^{D} s_k$$

If the terms are uncorrelated, then the variance of the sum is simply the sum of the variances:

$$\text{Var}(S) = \sum_{k=1}^{D} \text{Var}(s_k)$$

However, if different terms are positively correlated, the variance increases:

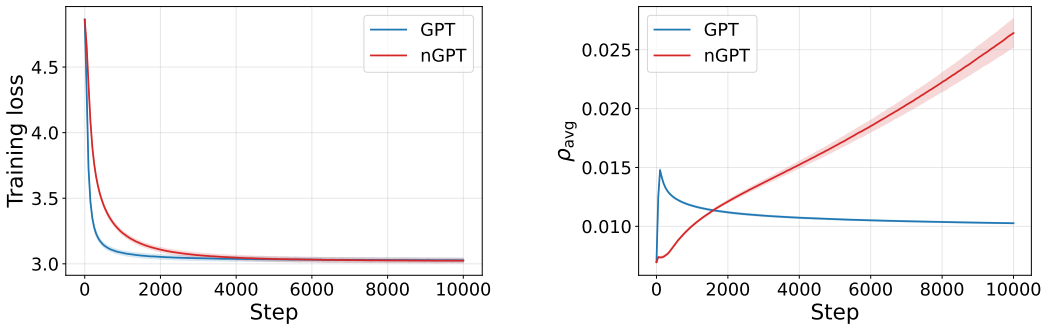
$$\text{Var}(S) = \sum_{k=1}^{D} \text{Var}(s_k) + 2 \sum_{j<k} \text{Cov}(s_j, s_k)$$

Assume the signal terms have similar variance σ_s^2 and an average pairwise correlation ρ_s. Then

$$\text{Var}(S) \approx D\sigma_s^2\big(1 + (D-1)\rho_s\big)$$

Because the quantization noise remains largely uncorrelated in both models ($\rho_n \approx 0$), the variance of the noise sum $N = \sum_{k=1}^{D} n_k$ scales strictly linearly:

$$\text{Var}(N) \approx D\sigma_n^2$$



(a) (b)

Figure 7: Training loss **(a)** and average correlation **(b)** of a one-layer mlp model for GPT and nGPT architectures. While both models achieve similar training loss, the correlation ρ in nGPT is higher.

If the sums have non-zero means, let $\mu_S = \mathbb{E}[S]$ and $\mu_N = \mathbb{E}[N]$. Since $\mathbb{E}[S^2] = \mathrm{Var}(S) + \mu_S^2$ and $\mathbb{E}[N^2] = \mathrm{Var}(N) + \mu_N^2$, the SNR of the dot product is:

$$\mathrm{SNR} = \frac{\mathbb{E}[S^2]}{\mathbb{E}[N^2]} = \frac{\mathrm{Var}(S) + \mu_S^2}{\mathrm{Var}(N) + \mu_N^2} \approx \frac{D\sigma_s^2\big(1 + (D-1)\rho_s\big) + \mu_S^2}{D\sigma_n^2 + \mu_N^2}. \tag{7}$$

B.2 Scaling SNR regimes

To derive Eq. (5) from Eq. (7), we assume, based on our empirical observations, that $\mu_S^2/\sigma_s^2 \ll D$ and $\mu_N^2/\sigma_n^2 \ll D$, and that the ratio σ_s^2/σ_n^2 are matched across architecture. Based on these assumptions, we can identify three distinct regimes when $\bar{\rho}_{\mathrm{GPT}} \ll \bar{\rho}_{\mathrm{nGPT}}$:

Regime I: small width, $D \ll 1/\bar{\rho}_{\mathrm{nGPT}}$**.** Both $D\bar{\rho}$ terms are much smaller than 1, so

$$\frac{\mathrm{SNR}_{\mathrm{nGPT}}}{\mathrm{SNR}_{\mathrm{GPT}}} \approx 1. \tag{8}$$

At small width, neither model accumulates enough correlated signal to create a gap.

Regime II: intermediate width, $1/\bar{\rho}_{\mathrm{nGPT}} \ll D \ll 1/\bar{\rho}_{\mathrm{GPT}}$**.** The numerator is dominated by $D\bar{\rho}_{\mathrm{nGPT}}$, while the denominator remains close to 1:

$$\frac{\mathrm{SNR}_{\mathrm{nGPT}}}{\mathrm{SNR}_{\mathrm{GPT}}} \approx D\,\bar{\rho}_{\mathrm{nGPT}}. \tag{9}$$

The advantage of nGPT grows *linearly* with width. Every additional dimension contributes to the gap.

Regime III: large width, $D \gg 1/\bar{\rho}_{\mathrm{GPT}}$**.** Both terms are dominated by their correlation contributions, and D cancels:

$$\frac{\mathrm{SNR}_{\mathrm{nGPT}}}{\mathrm{SNR}_{\mathrm{GPT}}} \approx \frac{\bar{\rho}_{\mathrm{nGPT}}}{\bar{\rho}_{\mathrm{GPT}}}. \tag{10}$$

The gain saturates at a constant set by the ratio of the two correlations.

B.3 One layer alignment

To confirm that nGPT's higher correlation is a reproducible phenomenon rather than model-specific, we trained a single-layer MLP. As illustrated in Figs. 7a and 7b, both architectures achieve comparable training loss, yet nGPT exhibits significantly higher average correlation, further validating the analysis in Section 3.3.

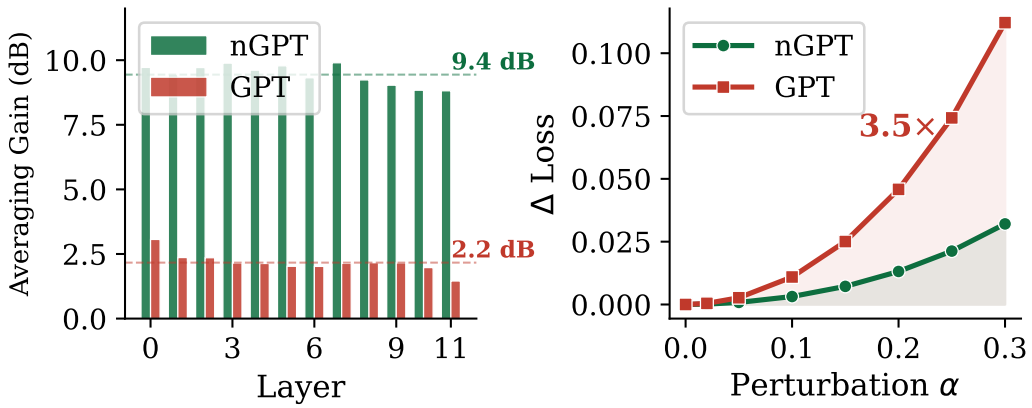


Figure 8: **(Left):** Per-layer gain(dB): nGPT cancels noise better at every matmul. Gain (dB) refers to Eq. (6). **(Right):** Loss landscape flatness: perturbing all weights simultaneously, GPT degrades 3.5× faster. The per-layer advantage compounds into a measurably flatter loss surface.

Table 3: Comparison of training overhead between nGPT-NVFP4 and GPT-NVFP4

Method	Per-tensor scaling	RHT	SR
nGPT-NVFP4	✗	✗	✓
GPT-NVFP4	✓	✓	✓

C Experiments details

C.1 1.2B dense

The hyperparameters for the 1.2B dense model experiments, discussed in Section 5.1, are detailed in Table 4. Notably, the nGPT architecture omits both warmup and weight decay while utilizing a learning rate half that of the standard model. In both 1.2B dense NVFP4 experiments we quantize the GEMMs of all layers. We keep the embeddings, non linear operations and batched-GEMMs in BF16.

Table 4: Hyperparameters for 1.2B Dense GPT and nGPT models

	GPT	**nGPT**
Number of layers	20	
Model dimension	2048	
FFN Dimension	6144	
Head dimension	128	
Q heads	16	
KV heads	4	
Sequence length	8192	
Scheduler	WSD	
Batch size	768	
Warmup (Samples)	1M	0
lr	1.2×10^{-3}	0.6×10^{-3}
Weight decay	0.1	0
Init std	0.0198	0.0221

C.2 Hybrid Mamba-Transformer MoE 400m/600m

The hyperparameters for the 400m/600m hybrid Mamba-Transformer MOE model experiments, discussed in Section 5.2, are detailed in Table 5. Notably, the nGPT architecture omits both warmup

and weight decay while utilizing a learning rate half that of the standard model. For both NVFP4 hybrid 400m/600m experiments we quantize the GEMMs of all layers. We keep the embeddings, non linear operations and batched-GEMMs in BF16.

Table 5: Hyperparameters for 400m/600m hybrid Mamba-Transformer MOE model. The model uses the following hybrid pattern: "MEMEM*EMEMEM" where "M" refer to Mamba block, "E" to MOE block and "*" to attention block.

	GPT	**nGPT**
Number of layers	12	
Model dimension	1024	
FFN Dimension	1536	
Head dimension	128	
Q heads	8	
KV heads	4	
Sequence length	8192	
Scheduler	WSD	
Batch size	64	
Num experts	16	
Activated experts	4	
Mamba num heads	16	
Mamba head dim	64	
Warmup (Samples)	640K	0
lr	1.2×10^{-3}	0.6×10^{-3}
Weight decay	0.1	0
Init std	0.0198	0.031

C.3 Hybrid Mamba-Transformer MoE 3B/30B

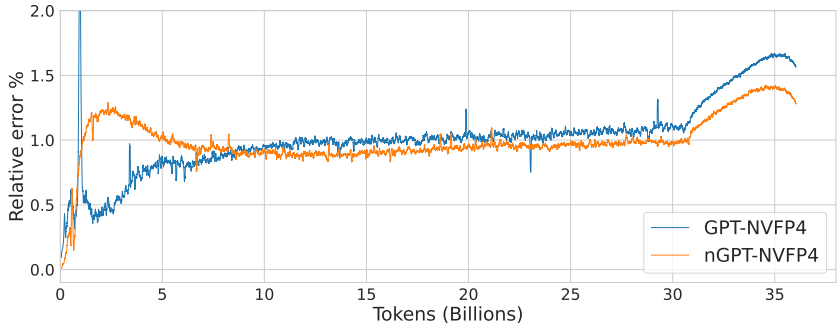
The hyperparameters for the 3B/30B hybrid Mamba-Transformer MoE model experiments, discussed in Section 5.2, are detailed in Table 6. Notably, the nGPT architecture omits both warmup and weight decay while utilizing a learning rate half that of the standard model. In Fig. 10 we compare the BF16 training of GPT and nGPT, showing they achieve similar training loss. For both hybrid 3B/30B NVFP4 experiments we maintain high precision for the final 15% of layers in both GPT and nGPT architectures, similar to [3]. In Fig. 10

C.4 Acceleration

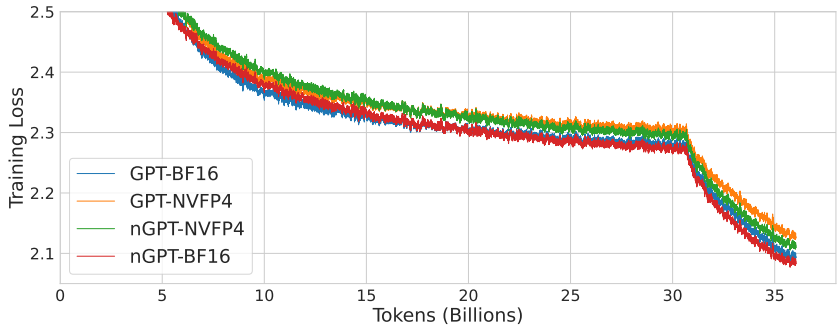
The experiments in Fig. 6b include a single transformer layer benchmark with the same transformer-layer structure as the training runs with RMSNorm fused into GEMMs, SwiGLU MLPs, no linear bias, and input gradients enabled so that backward includes the activation-gradient GEMMs. Forward and backward are timed in the same iteration, with an explicit output gradient and with gradient cleanup excluded from the measured time. We sweep over multiple hidden sizes (D), use FFN size $3D$, set the number of attention heads to $D/128$, disable GQA, and run with MBS one. Each point is averaged over 10 independent repetitions of 20 timed iterations.

D Broader Impact

Accelerating LLM runtime is pivotal as AI becomes increasingly embedded in daily life. By making 4-bit training an intrinsic architectural feature rather than a complex patch, this work improves efficiency, reduces memory demands, and lowers the carbon footprint of large-scale deployment. These advancements democratize AI by enabling high-performance models to run on consumer hardware, fostering broader innovation. However, while increased accessibility empowers diverse users, it also underscores the necessity for responsible oversight and ethical development to mitigate potential misuse as these powerful tools become more widely available.



(a)



(b)

Figure 9: Relative error (a) and training loss (b) of NVFP4 and nNVFP4 with their corresponding Bf16 and nBF16 for hybrid Mamba-Transformer MoE 400m/600m. The normalized architecture achieves lower training loss and lower relative error, while avoiding the use of RHT and per-tensor-scaling.

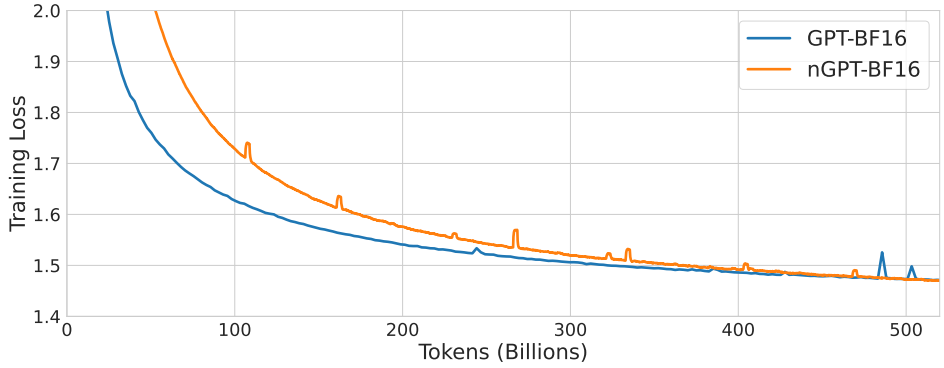


Figure 10: Training loss of nGPT and standard GPT for BF16 datatype for the hybrid Mamba-Transformer 3B/30B model. The architecture is similar to [4]. Both training achieved similar training loss.

E Limitations

While nGPT shows clear architectural advantages, our analysis primarily uses smaller models ($D = 4096$) to isolate the SNR mechanism. Additionally, while the 3B/30B MoE results demonstrate stability, the 500B token training duration is a relatively short horizon compared to the 20T+ tokens used for SOTA production models; it is possible that performance gaps could evolve over longer training periods or at even greater scales.

Table 6: Hyperparameters for 3B/30B hybrid Mamba-Transformer MoE model. The model uses the following hybrid pattern: "MEMEM*EMEMEM*EMEMEM*EMEMEM*EMEMEM*EMEMEMEM*EMEMEMEME" where "M" refer to Mamba block, "E" to MOE block and "*" to attention block.

	GPT	**nGPT**
Number of layers	52	
Model dimension	2688	
FFN Dimension	1856	
Head dimension	128	
Q heads	32	
KV heads	2	
Sequence length	8192	
Scheduler	WSD	
Batch size	3072	
Num experts	128	
Activated experts	6	
Mamba num heads	64	
Mamba head dim	64	
Warmup (Samples)	1M	0
lr	1×10^{-3}	0.5×10^{-3}
Weight decay	0.1	0
Init std	0.0173	0.0192

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: As stated in the abstract, the primary contribution of this paper is demonstrating that the nGPT architecture is inherently better suited for low-precision training.

   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 the appendix we have a limitation section that we discuss it.

   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: [Yes]

   Justification: The main theoretical part of the paper, appears in Section 3 and in the appendix. It includes full and complete proof as required.

   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 the appendix we clarify all hyperparameters and architectures choices we use. It include all information requires to reproduce the main experiments. Moreover we added an anonymous github with all the code to reproduce the experiments.

   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: [Yes]

   Justification: The paper use open source dataset, as mentioned in the experiments section. We added also an anonymous github with all the code to reproduce the experiments.

   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 the appendix we clarify all hyperparameters for the experiments section.

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: In the required section, such as in the acceleration paragraph, we include it. For the main experiments we use one run with random seed.

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: We clarify in the experiments section that we use B200 GPUs. We also show the acceleration achieved using NVFP4 datatype.

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: The paper conform the Code of Ethics.

   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: Yes, the paper include broader impact section in the appendix.

    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: [N/A]

    Justification: The paper poses no such risks.

    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: All data used in the paper is open source. The paper include full citation of all related works.

    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: [Yes]

    Justification: As explained in experiments section, the data used is open source.

    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: The paper does not involve crowdsourcing nor research with human subjects

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: The paper does not involve crowdsourcing nor research with human subjects.

    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: LLM was used only for editing or formatting purposes.

    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.