Transformers Get Stable: An End-to-End Signal Propagation Theory for Language Models

Akhil Kedia [*1] **Mohd Abbas Zaidi** [*1] **Sushil Khyalia** [*2] **JungHo Jung** [1] **Harshith Goka** [1] **Haejun Lee** [1]

Abstract

In spite of their huge success, transformer models remain difficult to scale in depth. In this work, we develop a unified signal propagation theory and provide formulae that govern the moments of the forward and backward signal through the transformer model. Our framework can be used to understand and mitigate vanishing/exploding gradients, rank collapse, and instability associated with high attention scores. We also propose DeepScaleLM, an initialization and scaling scheme that conserves unit output/gradient moments throughout the model, enabling the training of very deep models with 1000 layers. We find that transformer models could be much deeper – our deep models with fewer parameters outperform shallow models in Language Modeling, Speech Translation, and Image Classification, across encoder-only, decoder-only and encoder-decoder variants, for both Pre-LN and Post-LN transformers, for multiple datasets and model sizes. These improvements also translate into improved performance on downstream Question Answering tasks and improved robustness for Image Classification.

1 Introduction

Transformer models are extremely popular across different domains of machine learning, however, deep transformers are plagued with issues of gradient explosion/vanishing (Rae et al., 2021; Shleifer et al., 2021; Smith et al., 2022; Takase et al., 2022; Smith et al., 2022; Zhang et al., 2022c; Dehghani et al., 2023; Chowdhery et al., 2023; Molybog et al., 2023; Wortsman et al., 2024) and rank collapse (Zhou et al., 2021; Noci et al., 2022) that adversely affect training stability. Proposed remedies include residual scaling, changing initialization or extra/modified layernorms (Zhang et al., 2019a; Xiong et al., 2020; Bachlechner et al., 2021; Wang et al., 2024; Dehghani et al., 2023).

Theoretical analysis via signal propagation and kernel methods has led to an improved understanding of these issues. Several works in the signal propagation domain (Glorot & Bengio, 2010; Arpit et al., 2016; Xu et al., 2019; Dong et al., 2021; Davis et al., 2021; Wang et al., 2022) have analysed the propagation of moments for some components of deep transformers, but often make simplifying assumptions of IID inputs, uncorrelated outputs, ignoring effect of query/key initialization, simplifying non-linearity, etc. We observed break down of each of these assumptions with real world data, adversely affecting model stability.

These issues highlight the need for a holistic theoretical framework that can fully explain signal propagation through transformer models with real data. In this work, we provide a framework to fully explain signal propagation through transformer models, by deriving closed-form expressions for the first and second-order moments (mean and variance) of the outputs and gradients of each of the components of the transformer model (Embeddings, FFN, ReLU/GeLU, LayerNorm, Dropout, Softmax, Single-Head Attention), Attention and FFN blocks, and through the entire model. Our derived equations are empirically verified within strict error bounds with real world data[1].

We apply this framework to understand and mitigate instability issues with deep transformers – vanishing/exploding gradients, rank collapse, and instability caused by high QK values. To harness the improved complexity of deeper models (Montúfar et al., 2014; Poole et al., 2016; Raghu et al., 2017), we propose DeepScaleLM, a novel initialization scheme that augments residual/output scaling, and ensures the moments of outputs and gradients remain fully conserved throughout the model. DSLM enables us to break the depth barrier and train models with 100s of layers which outperform shallow models for BERT, GPT, Encoder-Decoder models across text, vision and speech modalities.

[*]Equal contribution [1]Language Intelligence Lab, Samsung Research, Seoul, South Korea. [2]Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA (work done while at Samsung). Correspondence to: Akhil Kedia <akhil.kedia@samsung.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]Code: https://github.com/akhilkedia/TranformersGetStable

Table 1. Signal propagation for forward and backward passes through components of a transformer (GeLU in Appendix A.5). The expressions here are illustrative simplification of full closed form formulae in Appendices A and C.

Component	$\mu_{\mathbf{x}_{\text{out}}}$	$\sigma^{\mathbf{2}}_{\mathbf{x}_{\text{out}}}$	$\sigma^{\mathbf{2}}_{\mathbf{g}_{\text{in}}}$	$\mathbf{r}^{\mathbf{l}}_{\mathbf{x}_{\text{out}}}$	$\mathbf{r}^{\mathbf{l}}_{\mathbf{g}_{\text{in}}}$
Embeddings	0	$\sum \sigma^2_{w_{\text{embd}}}$	-	$\frac{\pi^2}{18 * \log(\lvert V\rvert)^2} + \frac{2}{9}$	-
Linear ($d_{\text{in}} \to d_{\text{out}}$)	0	$d_{\text{in}}\sigma^2_w(\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})$	$d_{\text{out}}\sigma^2_w\sigma^2_{g_{\text{out}}}$	$\frac{r^l_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}/\sigma^2_{x_{\text{in}}}}{1 + \mu^2_{x_{\text{in}}}/\sigma^2_{x_{\text{in}}}}$	$r^l_{g_{\text{out}}}$
ReLU	$\frac{\sigma_{x_{\text{in}}}}{\sqrt{(2\pi)}}$	$\frac{(\pi-1)}{(2\pi)}\sigma^2_{x_{\text{in}}}$	$\frac{1}{2}\sigma^2_{g_{\text{out}}}$	$0.7r^l_{x_{\text{in}}} + 0.3{r^l_{x_{\text{in}}}}^2$	$(\frac{1}{2} + \frac{\sin^{-1}(r^l_{x_{\text{in}}})}{\pi})\mathrm{r}^{\mathrm{l}}_{g_{\text{out}}}$
LayerNorm (d)	0	1	$\frac{\sigma^2_{g_{\text{out}}}}{\sigma^2_{x_{\text{in}}}}$	$r^l_{x_{\text{in}}}$	$r^l_{g_{\text{out}}}$
Dropout (p)	$\mu_{x_{in}}$	$\frac{\sigma^2_{x_{\text{in}}} + p\mu^2_{x_{\text{in}}}}{1-p}$	$\frac{1}{1-p}\sigma^2_{g_{\text{out}}}$	$\frac{r^l_{x_{\text{in}}}(1-p)}{1 + p\mu^2_{x_{\text{in}}}/\sigma^2_{x_{\text{in}}}}$	$(1-p)r^l_{g_{\text{out}}}$
SHA-without V	0	$r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}$	$r^l_{g_{\text{out}}}\sigma^2_{g_{\text{out}}}$	1	1
Softmax	$\frac{1}{L}$	$\frac{e^{(1-r^d_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}} - 1}{L^2}$	$\frac{e^{(1-r^d_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}}}{L^2}\sigma^2_{g_{\text{out}}}$	-	-

Table 2. Moment Propagation through the blocks of a transformer layer. Exact closed forms / proofs are provided in Appendices B and C.

Component	$\sigma^{\mathbf{2}}_{\mathbf{x_{out}}}$	$\mathbf{r}^{\mathbf{l}}_{\mathbf{x_{out}}}$	$\sigma^{\mathbf{2}}_{\mathbf{g_{in}}}$	$\mathbf{r}^{\mathbf{l}}_{\mathbf{g_{in}}}$
Attention Block	$\frac{d^2\sigma^2_o\sigma^2_v\sigma^2_{x_{\text{in}}} * r^l_{x_{\text{in}}}}{(1-p)}$	$1-p$	$\frac{d^2\sigma^2_o\sigma^2_v * \sigma^2_{g_{\text{out}}}}{(1-p)}r^l_{g_{\text{out}}}$	$1-p$
FFN Block	$\frac{2d^2\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{x_{\text{in}}}}{(1-p)}$	$(1-p)(\frac{1}{\pi} + \frac{r^l_{x_{\text{in}}}}{2} + (\frac{1}{2} - \frac{1}{\pi}){r^l_{x_{\text{in}}}}^2)$	$\sigma^2_{x_{\text{out}}} * \sigma^2_{g_{\text{out}}}$	$(1-p)(\frac{1}{2} + \frac{\sin^{-1}(r^l_{x_{\text{in}}})}{\pi})r^l_{g_{\text{out}}}$

2 Moments of Transformer Models

2.1 Moments of Transformer Components

Following an analysis similar to that of Xavier initialization (Glorot & Bengio, 2010), we derive closed-form expressions for the mean and variance of the output and of the backpropagated gradient for all the components of the transformer model in Table 1.

Here $\mu_{x_{\text{in}}}$, $\sigma^2_{x_{\text{in}}}$, $\mu_{x_{\text{out}}}$, $\sigma^2_{x_{\text{out}}}$ are the mean and variance of the input/outputs, $\sigma^2_{g_{\text{out}}}$, $\sigma^2_{g_{\text{in}}}$ are the variance of the gradient back-propagated to/from the component, and r^l, r^d are the correlations across sequence length and hidden dimension. p is the dropout probability, L sequence length, $d_{\text{in}}, d_{\text{out}}$ input/output dimensions of Linear layer, σ^2_w, $\sigma^2_{w_{\text{embd}}}$ variances of the weights of the Linear layer and the Embeddings table. At the input side, $r^l_{x_{\text{in}}}$ originates from repeated tokens. For text, we estimate input correlation theoretically by assuming that input tokens follow Zipf (Kingsley, 1935) distribution. Detailed proofs are provided in Appendix A, and all assumptions are summarized in Appendix L.2.

2.2 Moments of Transformer Blocks

Combining the expressions reported in Table 1, we derive closed-form expressions for the moment transformation during the forward and backward pass of the transformer Attention and FFN blocks. The Attention block refers to the Q, K, V projection, followed by Multi-Head Attention and Output-Projection Layer. The FFN block refers to the Linear layer followed by non-linearity (ReLU) and output Linear layer. Table 2 provides our derived equations for these, where σ^2_v, σ^2_o, $\sigma^2_{w_1}$, $\sigma^2_{w_2}$ are the variances for V weights, Output-Projection weights, and weights of FFN block Linear layers, and d is model the hidden size. These results show that considering correlation r^l, dropout p and effects

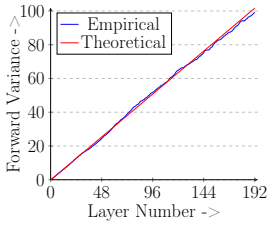

Figure 1. Pre-LN: Variance of forward signal increases linearly across layers N.

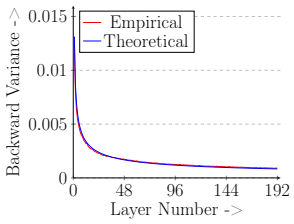

Figure 2. Pre-LN: Backward gradient variance increases hyperbolically across layers N.

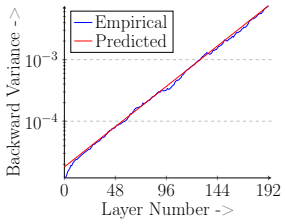

Figure 3. Post-LN: Backward gradient variance vanishes exponentially (y-axis log-scale).

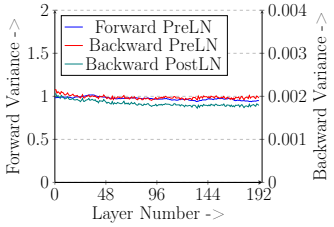

Figure 4. DeepScaleLM: The variances remain conserved for both forward and backward pass.

of non-linearity are crucial for correctly modelling signal propagation through Transformer blocks.

2.3 Moments of Entire Transformer Model

By repeatedly applying the expressions in Table 2 for each layer, we calculate the propagation of moments of outputs and gradients through the entire transformer model. We do this for both Pre-LN style transformers, in which the skip connection bypasses the LayerNorm, and for Post-LN style transformers, in which the Layernorm is applied before the skip-connection. The method is fully detailed in Appendices E.1 and E.2. Figures 1, 2 and 3 provide the forward (left to right) and backward (right to left) signal propagation at initialization through the layers of a very deep 192-layer model with Xavier initialization.

2.4 Numerical Validation of Theoretical Results

We verify the theoretical formulae of transformer components and blocks by running simulations with real/synthetic data, (detailed in Appendix D, code released). Even at 99 percentile, no error (other than SHA gradient σ^2) is larger than 10%, verifying our assumptions.

All our derivations are modality-agnostic. We verify our formulae for the entire transformer model using real textual MLM data, as shown in Figures 1, 2 and 3 (Reproducible using our released code), and using ImageNet data (as shown in Appendix H). Our formulae predict the observed gradient and forward/backward norms with remarkable accuracy, with mean and median relative errors of 6.8% and 5.2% respectively, and an R^2 of 0.998. We further verify that for model depths in range $[1 - 768]$, and model dimensions $[128 - 6096]$, the reported formulae are within 10% error, even across 768 layers of the transformer model.

2.5 Validity of Theoretical Predictions even after Training

Interestingly, our theoretical estimates hold approximately even after the models have been trained for a large number of steps. The model stays in the regime it is initialized with (as has also been shown in Li & Liang (2018); Arora et al. (2019a); Lee et al. (2019); Jesus et al. (2021); Arora et al. (2019b); Dettmers et al. (2023)), highlighting the importance of correct initialization. We analyze gradient explosion in a $30B$ parameter 64-layer PreLN model (after 150k training steps) and use our theory to predict the moments. Our hyperbolic estimation for the gradient explosion match closely with the observed moments as shown in Figure 5. Similarly, forward growth in a 48-layer 1024-d PreLN model (after 100k training steps) matches our linear estimations (Figure 6).

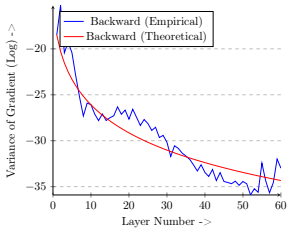

Figure 5. Backward gradient variance increases hyperbolically after 150k train steps.

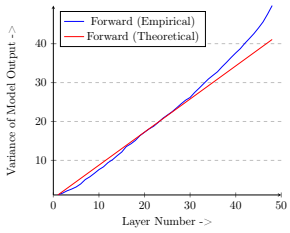

Figure 6. Linear growth in the forward pass for a 48-layer after 100k train steps.

3 Applications

3.1 Explaining Variance Explosion in Transformer

Our approach theoretically proves the gradient vanishing/explosion (Table 3) for both Pre-LN and Post-LN transformers.

Exploding Output and Gradient in Pre-LN The forward output for Pre-LN transformer increases linearly with increasing depth N (Appendix E.1) since each layer's output is directly added to the skip connection, as seen in Figure 1. For the backward pass, the gradient increases hyperbolically with increasing N, as seen in Figure 2. Intuitively, this is because the gradient increases in every layer when a block's gradient is added to the skip connection, and the fractional increase in gradient is inversely proportional to the forward variance (which increases by N) because of LayerNorm.

Vanishing/Exploding Gradient in Post-LN While layernorm solves the explosion in the forward pass of networks with residual connections (De & Smith, 2020), it has the opposite impact on the gradient. As proved in Appendix E.2, the gradient in a Post-LN transformer grows/decays exponentially with the number of layers (Figure 3).

Intuitively, the gradient is first transformed within the layer and then at the LayerNorm placed before the layer. The multiplicative factor is applied repeatedly, and causes gradient to vanish or explode exponentially, as was also observed in Schoenholz et al. (2017). This explains why Post-LN models are more challenging to train than Pre-LN for deeper networks (Wang et al., 2024; Shleifer et al., 2021; Takase et al., 2022).

Table 3. Comparison of maximum theoretical forward pass and backward pass growth in variance for the entire transformer model across methods (See Appendix E for proofs). Here β is the initial value of residual scaling for LayerScale.

Method	Post-LN		Pre-LN		
	Backward	Sensitivity	Forward	Backward	Sensitivity
Vanilla	$\mathcal{O}(c^{\pm N})$	$\boldsymbol{\mathcal{O}(N)}$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\boldsymbol{\mathcal{O}(logN)}$
DSInit	$\mathcal{O}(1)$	$\mathcal{O}(N^{-1})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(N^{-1})$
LayerScale	$\mathcal{O}(1)$	$\mathcal{O}(\beta N)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\beta N)$
DeepNet	$\mathcal{O}(1)$	$\mathcal{O}(N^{-0.5})$	-	-	-
DSLM (Ours)	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(1)$	**1**	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(1)$

3.2 Explaining Higher Pruning of Deeper Layers

Gromov et al. (2024) found that LLMs such as Llama-2-70B (Touvron et al., 2023) have minimal degradation in performance on Question Answering tasks until almost half the deeper layers are removed – suggesting that parameters in deeper layers are less effective in current LLMs. As we prove in Appendix E.1, the output of a Pre-LN transformer grows proportionally with depth (Figure 1). For an 80-layer model like Llama-2, this implies the deeper layers will have a significantly reduced impact on changing the output.

3.3 Explaining Impact of Large QK Values

In Dehghani et al. (2023), the authors observed large QK values destabilized the training, and solved this empirically by adding a layernorm after attention scores. Unlike prior works (Wang et al., 2024; Noci et al., 2022), note from our derivations of softmax(Appendix A.7) that the backwards gradients from Q/K are exponentially related to their variance, highlighting the critical significance of correct initialization of Q/K. For example, by initializing them to only 2x the xavier values (all other initializations the same), backwards gradients exploded 10000x through a 192 layer model. Our theory explains these empirical observations, and suggests a simple initialization strategy to fix this problem, achieving the same variance on QK without the overhead of LayerNorm (Section 3.5).

3.4 Explaining and Mitigating Rank Collapse

Similar to our work, Noci et al. (2022) also analyze moment propagation through the transformer, and observed the rank collapse of the token's representations at initialization after just a few layers, i.e., all the token representations became the same ($r_x^l \approx 1$ after just 12 layers) at initialization. This has also been reported in Shi et al. (2022); Zhou et al. (2021); Wang et al. (2022); He et al. (2023); Bachlechner et al. (2021); Zhai et al. (2023), and suggested modifications such as adding a skip connection on attention scores, initializing Q/other weights to 0, or normalizing all FFN weights.

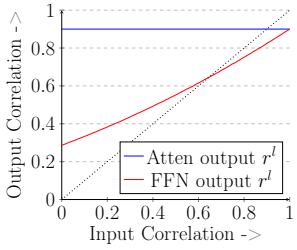

Figure 7. Forward $r^l_{x_{out}}$ for FFN and Attention blocks with $p = 0.1$. FFN reduces $r^l_{x_{out}}$ for $r^l_{x_{in}} > 0.65$, and attention always has $r^l_{x_{out}} < 1$.

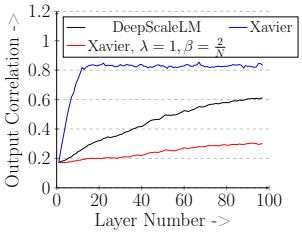

Figure 8. No rank collapse is observed with Xavier init and dropout. r^l increases slower with $\beta^2 = \frac{2}{N}$ or for DeepScaleLM.

Our theory suggests a very simple solution – Dropout. As our closed form expressions show, both FFN block (because of ReLU) and dropout reduce the correlation (Figure 7). With dropout, our method shows that such a rank collapse will not occur, and r_x^l will quickly reach a stable value < 1 (Appendix F), as verified empirically in Figure 8.

Alternatively, scaling the block output by $\beta = \frac{1}{\sqrt{N}}$, or equivalently initializing the weights very small in Post-LN will also prevent rank collapse, even without Dropout. For Pre-LN, $\lambda = 1$ slows down increase in r^l compared to $\lambda^2 = 1 - \frac{1}{N}$ (but the same slowdown can be achieved by decreasing β). This highlights the criticality of correct initialization, dropout and scaling for deep transformer models, as well as the explainability power of our theoretical framework.

3.5 DeepScaleLM: Enabling Deep Transformers

We propose DeepScaleLM (DSLM), a new initialization / scaling scheme that alleviates the issues discussed above.

Residual/Skip-Connection Scaling Let σ^2_{skip}, σ^2_{block}, σ^2_{model} be the variances of the skip connection, the block, and the output of the final layer of the model, respectively. Let $\sigma^2_{\text{skip}} = \sigma^2_{\text{block}}$, and we scale them by scalars λ and β respectively. Then, as has been proven in numerous works (Appendix K.3), if $\lambda^2 + \beta^2 = 1$, this scaling will maintain the variance after addition of the residual.

Initialization However while ensuring $\sigma^2_{\text{skip}} = \sigma^2_{\text{block}}$ (and equal to the variance of model input) has been done for ResNets (Appendix K.1), it is difficult to achieve theoretically for transformers. By leveraging the equations in Table 2, our theory provides us the tools to achieve this. We modify the initialization of the components of the transformer FFN and Attention blocks such that the variance of their output is 1, as further detailed in Appendix M –

1. We set the variance of embedding weights as $\sigma^2_e = \frac{1-p}{num_{\text{embd}}}$, where num_{embd} is the number of embeddings types. As embeddings are followed by a dropout, this ensures the input variance to the model is 1.
2. We set $\sigma^2_{w_2} = \sigma^2_{w_1} = \frac{1}{d} * \sqrt{\frac{1-p}{2}}$, to make the output of the FFN block 1.
3. We iteratively calculate layer-by-layer $r^l_{x_{\text{in}}}$, $r^l_{x_{\text{out}}}$ using expressions from Table 2, and calculate the initial variance of the attention block weights to make the output variance 1.

This initialization of transformer blocks, combined with the scaling of the skip connection and residual, and correct initialization of the embeddings, results $\sigma^2_{\text{model}} = 1$, irrespective of the number of layer N. This initialization also preserves the backward gradient, as proved for Pre-LN and Post-LN, in Appendices E.3 and E.4. Empirically, we show the backward gradient being preserved for both Pre-LN and Post-LN even across 192 layers at initialization (Figure 4).

Choice of Scaling Parameters While any choice of β will work at initialization, higher values of β, for example $\beta^2 = 0.5$ causes gradients to vanish (Figure 9, Table 4). This is because covariance between residual and skip connection increases the forward variance, which causes normalization to decrease backward gradient (De & Smith, 2020).

Similar to other prior works (Appendix K.3), we use $\beta^2 = \frac{k}{N}$ in all our experiments, where k is some small constant. This enables us to bound the fall in gradient (Appendix E.3) for Pre-LN. For Post-LN, $\beta^2 \leq \frac{k}{N^2}$ is theoretically required to bound the gradient (Appendix E.6). In practice, with $\beta^2 = \frac{2}{N}$, even with 768 layers, we empirically observed the final output variance from the model does not exceed 30, and all our models converge. We hence use $\beta^2 = \frac{k}{N}$ (Figure 10), but a practitioner may choose $\beta^2 = \frac{k}{N^\alpha}$, with $\alpha > 1$ if more stability is required at the expense of performance/"sensitivity" (Refer to discussion of relative strength in Section 4.6 and comparison to prior works in Section 4.5). While the above analysis assumes positive covariance (which we always observed experimentally), negative covariance follows a similar reasoning, and will cause gradient explosion instead.

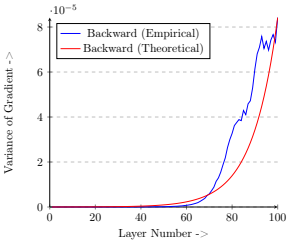

Figure 9. Gradient vanishes using $\lambda^2 = 0.9$ and $\beta^2 = 0.1$, after 50k training steps.

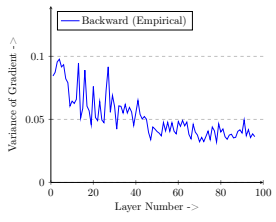

Figure 10. Gradient remains conserved using $\lambda^2 = 1 - \frac{1}{N}$ and $\beta^2 = \frac{1}{N}$, after 50k steps.

Preventing Rank Collapse For DSLM, applying block equations iteratively shows that $r^l_x < 1 - \frac{1}{e^2}$ after N layers.

Simpler Initialization Another avenue to handle the covariance between residual and skip connection could be to set $\lambda^2 + \beta^2 < 1$. We therefore also consider a simpler initialization method(Appendix M), in which we modify the initialization of attention value and output matrices to be the same as those of FFN block. This decreases the "effective" β of the attention block, but as the attention block has 2x fewer params than FFN, this change in weightage seems reasonable. As we show in Appendices E.5 and E.6 while variances are no longer unit at initialization, they are still bounded. This change does not impact performance significantly, as we show in Table 14. All further experiments in Section 4 used this simpler initialization.

Folding Scaling into Weights for Inference The scaling parameters introduced here can be fully absorbed into the model checkpoint weights by recursively scaling layernorm gain and output linear weights, hence and do not require any changes to vanilla transformers inference code.

DeepScaleLM enables training deeper-narrower models with 100s of layers, outperforming standard models across transformer variants, tasks and modalities.

4 DeepScaleLM Results

4.1 Improvements on Encoder-only Models (BERT)

Implementation Details We test our method on the Masked Language Modelling task with the BERT (Devlin et al., 2019) model. Pile-CC dataset (Gao et al., 2021) was used to pre-train our model. We use $k = 2$ for β while keep-

ing all the original hyper-parameters of BERT the same, except for learning rate (LR). We find that higher LR is needed for our deeper-narrower models (similar to Yang et al. (2021)). Hence, we search for LR for all the models. The training steps were decided based on Chinchilla (Hoffmann et al., 2022), at 6.6B tokens. Table 25 provides all hyper-parameter details. For DSLM, model output was down-scaled by $\sqrt{d}$ before being passed to the LM-head.

We train different language models with the same number of parameters and compute – while increasing the depth (N), we reduce the hidden dimension d keeping number of transformer parameters (Nd^2) constant. When changing from 12-layer 1024-d model to 192-layer 256-d model, compute negligibly increases by only 6.6% when keeping Nd^2 constant (Table 23), while the number of parameters decreases by $5-15\%$ due to decreased embedding parameters.

Evaluation Metrics Pre-training Perplexity (exponential of pre-training test-set loss) is often used to measure MLM pre-training performance (RoBERTa (Liu et al., 2019b), Megatron-LM (Shoeybi et al., 2019), Tay et al. (2023), or similar variants in Salazar et al. (2020); Lu et al. (2023)), and is well-correlated with downstream performance (Geiping & Goldstein, 2023). We use the perplexity as reported by Megatron-LM here. Calling this measure "perplexity" is a slight abuse of notation (as previous words which are masked are not available, and future words are). For downstream fine-tuning, we use accuracy while for Speech-to-Text translation, we use BLEU score.

Pre-Training Improvements In Table 4, we provide the results obtained on scaling model depth after applying DSLM to Post-LN. Post-LN models often diverge while scaling model depth. DSLM stabilizes the training of Post-LN models, and even a 768 layer Post-LN model (with 2300 Linear and 768 attention layers) converges.

Table 4. Performance (perplexity) of BERT models with different shapes. Deep-Thin models provide large improvements with fewer parameters.

Model N/D (# Params)	**12/1024** (185M)	**48/512** (168M)	**192/256** (160M)	**768/128** (156M)
Baseline	14.2	14.8	17.2	diverge
DSLM	15.5	13.1	**12.9**	18.4
Model N/D (# Params)	**24/1024** (336M)	**96/512** (319M)	**384/256** (311M)	- -
Baseline	13.2	diverge	diverge	-
DSLM	14.0	**11.7**	12.3	-

Our method is comparable to the baseline for shallow models but starts to outperform as the model gets deeper. Our 192-layer model outperforms the vanilla 12-layer, and our 96 layer outperforms the vanilla 24-layer model. The 160M 192-layer model outperforms the vanilla 24-layer 336M model with more than $2\times$ the params.

Reading Table 4 vertically, we can compare the performance of our approach with the baseline as we vary the model depth (N) while keeping the hidden dimension (d) constant. The baseline models often diverge at larger depths. By stabilizing the training, DSLM allows training larger models with better performance, with consistent improvements at larger depths.

Pre-training Improvements for Pre-LN We also applied DSLM to the deep Pre-LN models, trained for 3.3B tokens. Table 5 show that DSLM significantly improves the performance of the Pre-LN model across a range of model depths.

Table 5. DSLM with Pre-LN Models.

Model N/D	**12/512**	**96/512**	**192/256**	**768/128**
Baseline	29.4	20.6	19.8	26.9
DSLM	**26.0**	**15.4**	**17.0**	**25.9**

Sustained Improvements after Longer Pre-training Due to compute limitations, our models were trained for Chinchilla optimal steps. To ensure reproducibility of our work (scripts provided in released code), and demonstrate sustained improvements for standard models, we trained the BERT-base model using public Wikipedia data for 64B tokens ($30x$ chinchilla tokens). We train a $4x$ deeper, 10% smaller model using DSLM (N/d = 48 / 384). We finetune these models on the public RACE-M, RACE-H (Lai et al., 2017), MNLI (Williams et al., 2018) and QQP [2] datasets. As shown in Table 6, our model provides better pretraining performance which is translated into downstream Question-Answering tasks' performance across all datasets.

Table 6. BERT-base (trained for 64B tokens) pre-training and finetuning results (mean accuracy across 5 runs with stderr).

Dataset	**Baseline**	**DSLM**
Pretraining Performance		
Validation PPL	8.3	**7.8**
Finetuning Accuracy		
MNLI	82.4 ± 0.1	**83.7** ± 0.1
QQP	90.8 ± 0.03	**91.1** ± 0.05
RACE-Middle	71.1 ± 0.2	**74.0** ± 0.3
RACE-High	63.7 ± 0.1	**65.7** ± 0.2

[2] Quora Question Pairs dataset

Downstream Low Rank Finetuning DSLM continues to outperform the baseline on finetuning for downstream tasks with Low Rank Adapters (Hu et al., 2022), as shown in Table 7. Following QLoRA (Dettmers et al., 2023), we apply LoRA on all linear modules, with $r = 32$, $\alpha = 16$, and searched for LR.

Table 7. Accuracy on MNLI after low rank finetuning using LoRA

Model	Model Size		Score (Accuracy)
	Layers (N)	Hidden Dim (d)	
Baseline	12	768	82.2 ±0.1
DSLM	48	384	**82.9** ±0.1

4.2 Improvements on Decoder-only Models (GPT)

We applied DSLM to the decoder-only GPT model, trained for 8B tokens (slightly more than Chinchilla-optimal). Similar to BERT, increasing model depth by $4x$ with DSLM while keeping the parameters constant results in improved performance (Table 8).

Table 8. Application of DSLM to Decoder-only model (GPT), while increasing model depth to 4x (token-level PPL).

Model	Model Size			LM Perplexity	
	Layers (N)	Dim (d)	Params	Pre-LN	Post-LN
Baseline	12	1024	204M	11.6	12.7
DSLM	12	1024	204M	11.5	**11.5**
DSLM	48	512	178M	**11.2**	11.7
Baseline	24	1024	355M	10.4	11.6
DSLM	24	1024	355M	10.2	**10.5**
DSLM	96	512	329M	**10.1**	10.6

4.3 Improvements on Speech (Encoder-Decoder)

We apply DSLM on encoder/decoder style transformer for Speech-to-Text translation task. Applying our method to speech additionally requires handling the input embeddings. Instead of theoretical estimates as in the case of text inputs (Appendix A.1), the moments for speech embedding were replaced by the empirically observed values. This input variance and correlation was observed as 2.2 and 0.29.

The baseline was trained on the MuST-C (Di Gangi et al., 2019) dataset using fairseq (Ott et al., 2019). Using DSLM, we successfully train 4x deeper models which outperforms the 18-layer (12-encoder, 6-decoder layers) baseline with 9% less parameters as seen in Table 9.

4.4 Improvements on Vision Modality

Similar to speech domain, applying our method to vision modality simply requires handling the input embedding (Appendix H). Using ImageNet-1k (Russakovsky et al., 2015) data with ViT (Dosovitskiy et al., 2021) model, our method can also constrain the growth of moments in Vision Transformers, as we show in Figure 11.

Table 9. Application of DSLM to Speech-to-Text translation. N_{enc} and N_{dec} refer to number of layers in the encoder and the decoder respectively. For models marked with *, maximum source sequence length was limited to 1024 due to compute limitations, and longer examples were discarded for both train and test.

Model	Lang	Model Size			BLEU
		N_{enc}, N_{dec}	Dim (d)	Params	
Baseline Pre-LN	en→de	12,6	256	31.1M	24.9
DSLM Pre-LN	en→de	48,24	128	28.4M	**25.6**
Baseline Post-LN	en→de	12,6	256	31.1M	21.9
DSLM Post-LN	en→de	48,24	128	28.4M	**23.8**
Baseline Pre-LN*	en → es	12,6	256	31.1M	21.61
DSLM Pre-LN*	en → es	48,24	128	28.4M	**23.03**
Baseline Pre-LN*	en → fr	12,6	256	31.1M	23.74
DSLM Pre-LN*	en → fr	48,24	128	28.4M	**26.30**

We train our models on the Image Classification task using ViT baselines provided by Beyer et al. (2022), and trained a 4x deeper model with same params. The deeper DSLM model outperforms the baseline ViT both in both 90 and 300 epoch settings. The improvements also translate to improved robustness on ImageNet-v2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021) and ImageNet-Sketch (Wang et al., 2019).

Table 10. Applying DSLM to Image classification using ViT.

Eval Set	90-epoch		300-epoch	
	Baseline	DSLM	Baseline	DSLM
ImageNet	76.5	**77.2**	79.8	**80.3**
ImageNet-Real	83.2	**83.8**	85.4	**85.5**
ImageNet-v2	63.7	**65.2**	67.9	**68.3**
ImageNet-R	23.9	**24.4**	27.8	**28.3**
ImageNet-Sketch	24.4	**25.5**	28.7	**29.9**

4.5 Comparison with Prior Methods

In Table 11, we compare DSLM with several prior methods for deep transformers. DSInit and DeepNet stabilize the model training at the expense of reduced "sensitivity" (Section 4.6) by using smaller effective values of β^2, at $\mathcal{O}(N^{-2})$ and $\mathcal{O}(N^{-1.5})$ respectively. Interestingly, 96-layer model diverges with DSInit, despite DSInit using a smaller β asymptotically – this is because the constants hidden in $\mathcal{O}(N^{-2})$ are much larger for DSInit. Our method, by analysing signal propagation, sets constants exactly at 1.

Bamboo method is a vanilla Pre-LN transformer, which our method out-performs. SkipInit, ReZero, LayerScale and Value-Skipinit all initialize β to zero/very small values – this choice may slow down learning initially by reducing back-propagated gradients, and a learnable β under-performs compared to fixed (Table 13). Vanilla μP targets hyper-parameter transfer from thinner to wider models, and also diverges. Zero-initializing the output layers solves this divergence, but under-performs similar to SkipInit. Noci et al. (2022) initializes Query and Key matrices to a large value, causing divergence (Section 3.3). ADMIN requires an extra profiling pass through the model, and more importantly, cannot stop vanishing gradients (Appendix K.1), causing the 192-Layer model to diverge.

Table 11. Comparison with prior methods for deep Transformers.

Method	192/256	96/512
DSInit (Zhang et al., 2019a)	15.9	diverge
ADMIN (Liu et al., 2020a)	diverge	25.2
SkipInit (De & Smith, 2020)	15.1	13.1
ReZero (Bachlechner et al., 2021)	diverge	diverge
LayerScale (Touvron et al., 2021b)	13.2	14.4
μP-Tensor Programs V (Yang et al., 2021)	diverge	diverge
DeepNorm (Wang et al., 2024)	14.4	13.4
Noci et al. (2022)	diverge	diverge
Bamboo (Xue et al., 2023)	17.1	diverge
Value-SkipInit (He et al., 2023)	18.8	17.1
DeepScaleLM (ours)	**12.9**	**11.7**

4.6 Analysis of DSLM

Model Quantization Similar to Unit Scaling (Blake et al., 2023), conserving unit activations and gradients from our method results in models which lose much less performance when quantized (via direct casting) to FP8 precision compared to original models. We apply 8-bit quantization to the 48-Layer 512-dim BERT baseline model and the model trained with DSLM. Table 12 provides the performance corresponding to the full precision inference and FP8 inferences (corresponding to two different FP8 standards, E5M2 and E4M3). DSLM model can be compressed to 25% of the original size with significantly lower performance loss.

Table 12. Model performance on direct casting to FP8

Model	FP32	E5M2	E4M3
Baseline	14.8	42.5 (Δ 27.7)	16.5 (Δ 1.7)
DSLM	**13.1**	**21.4** (Δ 8.3)	**13.9** (Δ 0.8)

Ablation of Residual Scaling Table 13 provides the results corresponding to the different components of our proposed DSLM scheme for training 96-layer 512-d model Post-LN model. The model fails to converge without the proposed residual scaling. β may also be set as learnable (similar to BatchNorm (Ioffe & Szegedy, 2015)), after initializing it with $\beta^2 = \frac{2}{N}$. We find that this does not significantly impact performance, and β remains within $[0.2-5]\times$ of its initialized values.

Table 13. Ablation of various DeepScaleLM components.

Model	Perf
Vanilla Xavier (with or w/o $\beta^2 = 0.5$)	diverge
DSLM-Init (with or w/o $\beta^2 = 0.5$)	diverge
DSLM-Init + $\beta^2 = \frac{2}{N}$ (learnable β)	12.2
DSLM-Init + $\beta^2 = \frac{2}{N}$ (fixed β)	11.7

Ablation of Initialization Table 14 provides ablation results for our proposed initialization. All experiments in Table 14 were conducted for the Pre-LN model with our proposed scaling (λ, β), since the Post-LN model diverged with Xavier initialization. Xavier initialization performs significantly worse for very deep models, due to higher QK initialization. BERT default initialization with $\sigma = 0.02$ also performs worse. Finally, DSLM simpler initialization performs comparably to DSLM.

Table 14. Ablation of the initializations.

Model	Model Size (N/d)	Perf
Xavier	192/256 (160M)	38.2
DSLM	192/256 (160M)	**17.0**
DSLM (simple)	192/256 (160M)	17.9
Fixed $\sigma = 0.02$	96/512 (319M)	20.5
DSLM	96/512 (319M)	**17.9**

Compute Appendix I provides detailed theoretical and wall-clock compute overheads for making models deeper. We observe that up to 200 layers, the theoretical compute is within $6-7\%$ and wall-clock times is within 15% of the original shallow model. While our 192-layer 256-d model requires 6% extra compute than the 12-layer 1024-d model, it manages to outperform the 24-layer 1024-d model, that has 62.5% more parameters, at equal wall-clock time and at equal number of tokens.

Discussion of Relative Strength In general, for a β of the form $\beta^2 = \frac{k}{N^\alpha}$, we can choose from a wide range of values for the constant k and exponent α. There is an expressivity-trainability trade-off in training deep networks (Yang & Schoenholz, 2017) – having lower β (smaller k or higher α) will result in networks where observed issues (forward growth or gradient explosion/vanishing) are mitigated, but they may converge slowly/sub-optimally.

Davis et al. (2021) defines "sensitivity" as the variance of relative change in output for small perturbations in parameters, averaged across all parameters. If $\sigma^2_{\text{skip}} = 1$, sensitivity can be shown to be mean across layers of $N * (1/\sigma^2_{\text{block}}) = N * \beta^2$. Mean is not robust to outliers, and hence we suggest median may provide a more robust measure. For e.g., for vanilla pre-LN, Davis et al. (2021)'s definition gives sensitivity as $\mathcal{O}(log(N))$, whereas using median provides a more robust measure as $\mathcal{O}(1)$. But only the first $N/10$ layers have $\mathcal{O}(log(N))$ sensitivity, and the last $9N/10$ layers have $\mathcal{O}(1)$ sensitivity. We will use median in the discussion below.

In Appendix G, we show that the fall in gradient for both pre-LN and post-LN for $\beta^2 = k/N^\alpha$ is $\mathcal{O}(e^{kN^{1-\alpha}})$. The sensitivity is hence $kN^{1-\alpha}$. For DSLM, we chose $\alpha = 1$, that is the sweet spot on the stability-expressivity curve where both the gradient fall bound and sensitivity expressions become independent of model depth. For higher values of α such as $\alpha = 2$ (DS-Init) and, $\alpha = 1.5$ (DeepNet), the gradient becomes stable using but the model expressivity reduces with depth, as shown in Table 3. Such models might not be able to extract better results when going deeper, as we indeed verify empirically in the comparison with prior works paragraph in Section 4.5.

5 Related Works

For detailed discussion of prior works, refer to Appendix K.

Initialization Several works (Glorot & Bengio, 2010; He et al., 2015; Brock et al., 2021a; Poole et al., 2016; Schoenholz et al., 2017) improved the initialization of ResNets/ReLU networks. These works do not consider transformers, and are unable to handle Softmax/Attention. Others, such as ADMIN (Liu et al., 2020a), Mishkin & Matas (2016); Liu et al. (2020b) achieve unit variance for faster convergence by scaling the weights and/or outputs based on empirical profiling of a forward pass. Blake et al. (2023) also tries to achieve this, but does not completely handle correlation and non-zero mean of ReLU. We demonstrate that this profiling is unnecessary, and can instead be done theoretically in our work.

Signal Propagation Signal propagation in Neural Networks (Neal, 1995; LeCun et al., 1996) has a long history, such as for ResNets (He et al., 2015; De & Smith, 2020; Brock et al., 2021a; Schoenholz et al., 2017; Hoedt et al., 2022; Labatie et al., 2021; Marion et al., 2022; Klambauer et al., 2017; Balduzzi et al., 2017), and for transformers in (Xu et al., 2019; Dong et al., 2021; Davis et al., 2021; Noci et al., 2022; Martens et al., 2021; He et al., 2023; Shi et al., 2022; Wang et al., 2022). Our work considers previously often neglected effects of dropout, input correlation, activation non-linearity, and QK initialization, providing closed forms with verifiable correctness of signal propagation. This allows us to constrain the output and gradient to almost exactly unit variance.

Moment Control & Residual Scaling Bounded gradients have been shown to results in better/faster convergence (Shen et al., 2020; Yu et al., 2017; You et al., 2017; 2020; Takase et al., 2022; Shleifer et al., 2021; Hayou et al., 2019). Different scaling schemes for residual networks (λ for skip connections and β for residual output) have been explored by prior works, such as $\lambda^2+\beta^2 = 1$ for ResNets (Balduzzi et al., 2017; Szegedy et al., 2017; Hanin & Rolnick, 2018; Arpit et al., 2019; Zhang et al., 2019b; Hoedt et al., 2022). Learnable $\beta \approx 0$ was used in SkipInit (De & Smith, 2020), ReZero (Bachlechner et al., 2021), LayerScale (Touvron et al., 2021b), Value-SkipInit (He et al., 2023). Others proposed $\beta^2 = \mathcal{O}(\frac{1}{N})$, where N is max/current layer was used in Arpit et al. (2019); Brock et al. (2021a); Marion et al. (2022); Zhang et al. (2022b); He et al. (2023); Noci et al. (2022); De & Smith (2020); Liu et al. (2020a;b); Davis et al. (2021); Blake et al. (2023), while DSInit (Zhang et al., 2019a), T-Fixup (Huang et al., 2020a), DeepNorm (Wang et al., 2024) used $\beta^2 < \mathcal{O}(\frac{1}{N})$. However, the optimal initialization/scaling can vary based on data/model characteristics (Zhang et al., 2022b; Marion et al., 2022). Our contribution goes beyond providing an optimal scaling scheme – our theory enables informed choices about these initialization/scaling schemes based on their expressivity-trainability trade-off. Some works such as DeepNet, ADMIN show performance improvements by making the model deeper, but much larger. In this work, we explore a stricter setting of keeping transformer parameters and compute constant while making the model deeper.

Other Network modifications for Deep Networks Architectural modifications such as Zhai et al. (2023); Zhou et al. (2021); Shleifer et al. (2021) can only stabilize the model later during training and not at initialization. They are orthogonal to our approach, and our equations can be easily extended to cover these.

6 Conclusion

We theoretically derive closed forms for the growth of variances for forward and backward pass through individual transformer components as well as the entire transformer model. These formulae enable us to identify and solve the key reasons for vanishing/exploding gradients and rank collapse in very deep transformers. Via scaling and correct initialization, we also enable training very deep transformers with 1000 layers. Our experiments suggest that deeper transformers should be explored – using our method, models with 100s of layers outperform larger standard models across multiple modalities, tasks, and transformer variants.

Acknowledgements

We would like to thank Dr. Kangwook Lee and Dr. Joohyung Lee of Samsung Research, Seoul, Korea, for their guidance and leadership. We would also like to thank all the reviewers for their valuable feedback and suggestions, which helped greatly improve the paper.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, some which we feel must be specifically highlighted here. Using crawled web data for pre-training language models is questionable, something which society has yet to finalize its views on. Language modelling in particular suffers from hallucinations, and may be used for misinformation.

References

Anil, C., Lucas, J., and Grosse, R. B. Sorting out lipschitz function approximation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 291–301. PMLR, 2019. URL http://proceedings.mlr.press/v97/anil19a.html.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8139–8148, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/dbc4d84bfcfe2284ba11beffb853a8c4-Abstract.html.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 2019b. URL http://proceedings.mlr.press/v97/arora19a.html.

Arpit, D., Zhou, Y., Kota, B. U., and Govindaraju, V. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1168–1176. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/arpitb16.html.

Arpit, D., Campos, V., and Bengio, Y. How to initialize your network? robust initialization for weightnorm i& resnets. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 10900–10909, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e520f70ac3930490458892665cda6620-Abstract.html.

Bachlechner, T., Majumder, B. P., Mao, H. H., Cottrell, G., and McAuley, J. J. Rezero is all you need: fast convergence at large depth. In de Campos, C. P., Maathuis, M. H., and Quaeghebeur, E. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Balduzzi, D., Frean, M., Leary, L., Lewis, J. P., Ma, K. W., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 342–350. PMLR, 2017. URL http://proceedings.mlr.press/v70/balduzzi17b.html.

Beyer, L., Zhai, X., and Kolesnikov, A. Better plain vit baselines for imagenet-1k, 2022.

Bingham, G. and Miikkulainen, R. AutoInit: Analytic signal-preserving weight initialization for neural networks. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pp. 6823–6833. AAAI Press,

2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i6.25836.

Blake, C., Orr, D., and Luschi, C. Unit scaling: Out-of-the-box low-precision training. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2548–2576. PMLR, 2023. URL https://proceedings.mlr.press/v202/blake23a.html.

Bordelon, B., Noci, L., Li, M. B., Hanin, B., and Pehlevan, C. Depthwise Hyperparameter Transfer in Residual Networks: Dynamics and Scaling Limit. In *The Twelfth International Conference on Learning Representations*, 2023.

Brock, A., De, S., and Smith, S. L. Characterizing signal propagation to close the performance gap in unnormalized resnets. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=IX3Nnir2omJ.

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1059–1071. PMLR, 2021b. URL http://proceedings.mlr.press/v139/brock21a.html.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20v.html.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL http://jmlr.org/papers/v24/22-1144.html.

Clark, K., Luong, M., Le, Q. V., and Manning, C. D. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Dasoulas, G., Scaman, K., and Virmaux, A. Lipschitz normalization for self-attention layers with application to graph neural networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2456–2466. PMLR, 2021. URL http://proceedings.mlr.press/v139/dasoulas21a.html.

Daunizeau, J. Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables, 2017. URL https://arxiv.org/abs/1703.00091.

Davis, J. Q., Gu, A., Choromanski, K., Dao, T., Ré, C., Finn, C., and Liang, P. Catformer: Designing stable transformers via sensitivity analysis. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2489–2499. PMLR, 2021. URL http://proceedings.mlr.press/v139/davis21a.html.

De, S. and Smith, S. L. Batch normalization biases residual blocks towards the identity function in deep networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6b738eca0e6792ba8a9cbcba6c1881d-Abstract.html.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Ruiz, C. R., Minderer,

M., Puigcerver, J., Evci, U., Kumar, M., van Steenkiste, S., Elsayed, G. F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M., Gritsenko, A. A., Birodkar, V., Vasconcelos, C. N., Tay, Y., Mensink, T., Kolesnikov, A., Pavetic, F., Tran, D., Kipf, T., Lucic, M., Zhai, X., Keysers, D., Harmsen, J. J., and Houlsby, N. Scaling vision transformers to 22 billion parameters. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Deshpande, A. and Narasimhan, K. Guiding attention for self-supervised learning with transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4676–4686, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.419. URL https://aclanthology.org/2020.findings-emnlp.419.

Dettmers, T. and Zettlemoyer, L. The case for 4-bit precision: k-bit inference scaling laws. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7750–7774. PMLR, 2023. URL https://proceedings.mlr.press/v202/dettmers23a.html.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OUIFPHEgJU.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Di Gangi, M. A., Cattoni, R., Bentivogli, L., Negri, M., and Turchi, M. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2012–2017, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1202. URL https://aclanthology.org/N19-1202.

Dinan, E., Yaida, S., and Zhang, S. Effective Theory of Transformers at Initialization, 2023. URL https://arxiv.org/abs/2304.02034.

Dong, Y., Cordonnier, J., and Loukas, A. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2793–2803. PMLR, 2021. URL http://proceedings.mlr.press/v139/dong21a.html.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Galanti, T. A note on the implicit bias towards minimal depth of deep neural networks. *ArXiv preprint*, abs/2202.09028, 2022. URL https://arxiv.org/abs/2202.09028.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv preprint*, abs/2101.00027, 2021. URL https://arxiv.org/abs/2101.00027.

Geiping, J. and Goldstein, T. Cramming: Training a language model on a single GPU in one day. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11117–11143. PMLR, 2023. URL https://proceedings.mlr.press/v202/geiping23a.html.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, D. M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 249–256. JMLR.org,

2010. URL http://proceedings.mlr.press/v9/glorot10a.html.

Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P., and Roberts, D. A. The unreasonable ineffectiveness of the deeper layers. *ArXiv preprint*, abs/2403.17887, 2024. URL https://arxiv.org/abs/2403.17887.

Hanin, B. and Rolnick, D. How to start training: The effect of initialization and architecture. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 569–579, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d81f9c1be2e08964bf9f24b15f0e4900-Abstract.html.

Hayou, S., Doucet, A., and Rousseau, J. On the impact of the activation function on deep neural networks training. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2672–2680. PMLR, 2019. URL http://proceedings.mlr.press/v97/hayou19a.html.

He, B. and Hofmann, T. Simplifying transformer blocks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RtDok9eS3s.

He, B., Martens, J., Zhang, G., Botev, A., Brock, A., Smith, S. L., and Teh, Y. W. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=NPrsUQgMjKK.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.123. URL https://doi.org/10.1109/ICCV.2015.123.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=XPZIaotutsD.

He, R., Ravula, A., Kanagal, B., and Ainslie, J. RealFormer: Transformer likes residual attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 929–943, Online, 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.81. URL https://aclanthology.org/2021.findings-acl.81.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 8320–8329. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00823. URL https://doi.org/10.1109/ICCV48922.2021.00823.

Hoedt, P.-J., Hochreiter, S., and Klambauer, G. Normalisation is dead, long live normalisation! In *ICLR Blog Track*, 2022. URL https://iclr-blog-track.github.io/2022/03/25/unnormalized-resnets/. https://iclr-blog-track.github.io/2022/03/25/unnormalized-resnets/.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. An empirical analysis of compute-optimal large language model training. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, X. S., Pérez, F., Ba, J., and Volkovs, M. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4475–4483. PMLR, 2020a.

URL http://proceedings.mlr.press/v119/huang20f.html.

Huang, X. S., Pérez, F., Ba, J., and Volkovs, M. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4475–4483. PMLR, 2020b. URL http://proceedings.mlr.press/v119/huang20f.html.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ioffe15.html.

Jesus, R. J., Antunes, M. L., da Costa, R. A., Dorogovtsev, S. N., Mendes, J. F. F., and Aguiar, R. L. Effect of Initial Configuration of Weights on Training and Function of Artificial Neural Networks. *Mathematics*, 9(18):2246, 2021. ISSN 2227-7390. doi: 10/gsshxg.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and Improving the Training Dynamics of Diffusion Models, 2023. URL https://arxiv.org/abs/2312.02696.

Kim, H., Papamakarios, G., and Mnih, A. The lipschitz constant of self-attention. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5562–5571. PMLR, 2021. URL http://proceedings.mlr.press/v139/kim21i.html.

Kingsley, Z. G. *The psycho-biology of language: an introduction to dynamic philology*. Houghton Mifflin, 1935.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 971–980, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5d44ee6f2c3f71b73125876103c8f6c4-Abstract.html.

Korotkov, N. E. and Korotkov, A. N. *Integrals related to the error function*. Chapman & Hall/CRC, Philadelphia, PA, 2020. ISBN 9780367408206. URL https://www.taylorfrancis.com/books/mono/10.1201/9780367809232/integrals-related-error-function-nikolai-korotkov-alexander-korotkov.

Labatie, A., Masters, D., Eaton-Rosen, Z., and Luschi, C. Proxy-normalizing activations to match batch normalization while removing batch dependence. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 16990–17006, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8d2a5f7d4afa5d0530789d3066945330-Abstract.html.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K. Effiicient backprop. In Orr, G. B. and Müller, K. (eds.), *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pp. 9–50. Springer, 1996. doi: 10.1007/3-540-49430-8_2. URL https://doi.org/10.1007/3-540-49430-8_2.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8570–8581, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Levine, Y., Wies, N., Sharir, O., Bata, H., and Shashua, A. Limits to depth efficiencies of self-attention. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*

2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ff4dfdf5904e920ce52b48c1cef97829-Abstract.html.

Li, A. C., Efros, A. A., and Pathak, D. Understanding Collapse in Non-Contrastive Siamese Representation Learning, 2022. URL https://arxiv.org/abs/2209.15007.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8168–8177, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/54fe976ba170c19ebae453679b362263-Abstract.html.

Lin, Z., Sekar, V., and Fanti, G. Why spectral normalization stabilizes gans: Analysis and improvements. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9625–9638, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4ffb0d2ba92f664c2281970110a2e071-Abstract.html.

Liu, F., Gao, M., Liu, Y., and Lei, K. Self-adaptive scaling for learnable residual structure. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 862–870, Hong Kong, China, 2019a. Association for Computational Linguistics. doi: 10.18653/v1/K19-1080. URL https://aclanthology.org/K19-1080.

Liu, L., Liu, X., Gao, J., Chen, W., and Han, J. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5747–5763, Online, 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.463. URL https://aclanthology.org/2020.emnlp-main.463.

Liu, X., Duh, K., Liu, L., and Gao, J. Very deep transformers for neural machine translation. *ArXiv preprint*, abs/2008.07772, 2020b. URL https://arxiv.org/abs/2008.07772.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692, 2019b. URL https://arxiv.org/abs/1907.11692.

Lo, C. F. WKB approximation for the sum of two correlated lognormal random variables. *Applied Mathematical Sciences*, 7:6355–6367, 2013. ISSN 13147552. doi: 10.12988/ams.2013.39511. URL http://www.m-hikari.com/ams/ams-2013/ams-125-128-2013/39511.html.

Lu, J., Zhu, D., Han, W., Zhao, R., Namee, B. M., and Tan, F. What makes pre-trained language models better zero-shot learners? In Rogers, A., Boyd-Graber, J. L., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 2288–2303. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.128. URL https://doi.org/10.18653/v1/2023.acl-long.128.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6231–6239, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/32cbf687880eb1674a07bf717761dd3a-Abstract.html.

Marion, P., Fermanian, A., Biau, G., and Vert, J. Scaling resnets in the large-depth regime. *ArXiv preprint*, abs/2206.06929, 2022. URL https://arxiv.org/abs/2206.06929.

Martens, J., Ballard, A., Desjardins, G., Swirszcz, G., Dalibard, V., Sohl-Dickstein, J., and Schoenholz, S. S. Rapid training of deep neural networks without skip connections or normalization layers using deep kernel shaping. *ArXiv preprint*, abs/2110.01765, 2021. URL https://arxiv.org/abs/2110.01765.

Micikevicius, P., Stosic, D., Burgess, N., Cornea, M., Dubey, P., Grisenthwaite, R., Ha, S., Heinecke, A., Judd, P., Kamalu, J., Mellempudi, N., Oberman, S. F., Shoeybi, M., Siu, M. Y., and Wu, H. FP8 formats for deep learning. *ArXiv preprint*, abs/2209.05433, 2022. URL https://arxiv.org/abs/2209.05433.

Mishkin, D. and Matas, J. All you need is a good init. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06422.

Molybog, I., Albert, P., Chen, M., DeVito, Z., Esiobu, D., Goyal, N., Koura, P. S., Narang, S., Poulton, A., Silva, R., Tang, B., Liskovich, D., Xu, P., Zhang, Y., Kambadur, M., Roller, S., and Zhang, S. A theory on adam instability in large-scale machine learning. *ArXiv preprint*, abs/2304.09871, 2023. URL https://arxiv.org/abs/2304.09871.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2924–2932, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/109d2dd3608f669ca17920c511c2a41e-Abstract.html.

Neal, R. M. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, Canada, 1995. URL https://librarysearch.library.utoronto.ca/permalink/01UTORONTO_INST/14bjeso/alma991106438365706196.

Ng, E. W. and Geller, M. A table of integrals of the Error functions. *Journal of Research of the National Bureau of Standards, Section B: Mathematical Sciences*, 73B(1):1, 1969. ISSN 0098-8979. doi: 10/gdtk9p. URL https://nvlpubs.nist.gov/nistpubs/jres/73B/jresv73Bn1p1_A1b.pdf.

Nguyen, T. Q. and Salazar, J. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, 2019. Association for Computational Linguistics. URL https://aclanthology.org/2019.iwslt-1.17.

Noci, L., Anagnostidis, S., Biggio, L., Orvieto, A., Singh, S. P., and Lucchi, A. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ae0cba715b60c4052359b3d52a2cff7f-Abstract-Conference.html.

Noci, L., Li, C., Li, M. B., He, B., Hofmann, T., Maddison, C. J., and Roy, D. The shaped transformer: Attention models in the infinite depth-and-width limit. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/aa31dc84098add7dd2ffdd20646f2043-Abstract-Conference.html.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://aclanthology.org/N19-4009.

Peer, D., Keulen, B., Stabinger, S., Piater, J. H., and Rodríguez-Sánchez, A. J. Improving the trainability of deep neural networks through layerwise batch-entropy regularization. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=LJohl5DnZf.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3360–3368, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/148510031349642de5ca0c544f31b2ef-Abstract.html.

Qi, X., Wang, J., Chen, Y., Shi, Y., and Zhang, L. Lipsformer: Introducing lipschitz continuity to vision transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=cHf1DcCwcH3.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, H. F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S. M., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D.,

Lazaridou, A., Mensch, A., Lespiau, J., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., de Masson d'Autume, C., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., de Las Casas, D., Guy, A., Jones, C., Bradbury, J., Johnson, M. J., Hechtman, B. A., Weidinger, L., Gabriel, I., Isaac, W., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv preprint*, abs/2112.11446, 2021. URL https://arxiv.org/abs/2112.11446.

Raghu, M., Poole, B., Kleinberg, J. M., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2847–2854. PMLR, 2017. URL http://proceedings.mlr.press/v70/raghu17a.html.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 2019. URL http://proceedings.mlr.press/v97/recht19a.html.

Roberts, D. A., Yaida, S., and Hanin, B. The principles of deep learning theory. *ArXiv preprint*, abs/2106.10165, 2021. URL https://arxiv.org/abs/2106.10165.

Rong, Y., Huang, W., Xu, T., and Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. doi: 10.1007/S11263-015-0816-Y. URL https://doi.org/10.1007/s11263-015-0816-y.

Salazar, J., Liang, D., Nguyen, T. Q., and Kirchhoff, K. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2699–2712, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.240. URL https://aclanthology.org/2020.acl-main.240.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. 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=H1W1UN9gg.

Shao, J., Hu, K., Wang, C., Xue, X., and Raj, B. Is normalization indispensable for training deep neural networks? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pp. 13434–13444, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 978-1-71382-954-6.

Shen, S., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Powernorm: Rethinking batch normalization in transformers. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8741–8751. PMLR, 2020. URL http://proceedings.mlr.press/v119/shen20e.html.

Shi, H., Gao, J., Xu, H., Liang, X., Li, Z., Kong, L., Lee, S. M. S., and Kwok, J. T. Revisiting over-smoothing in BERT from the perspective of graph. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=dUV91uaXm3.

Shleifer, S., Weston, J., and Ott, M. Normformer: Improved transformer pretraining with extra normalization. *ArXiv preprint*, abs/2110.09456, 2021. URL https://arxiv.org/abs/2110.09456.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv preprint*, abs/1909.08053, 2019. URL https://arxiv.org/abs/1909.08053.

Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., Zheng, E., Child, R., Aminabadi, R. Y., Bernauer, J., Song, X., Shoeybi, M., He, Y., Houston, M., Tiwary, S., and Catanzaro, B. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *ArXiv preprint*, abs/2201.11990, 2022. URL https://arxiv.org/abs/2201.11990.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Singh, S. P. and Markovitch, S. (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 4278–4284. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

Takase, S., Kiyono, S., Kobayashi, S., and Suzuki, J. On layer normalizations and residual connections in transformers. *ArXiv preprint*, abs/2206.00330, 2022. URL https://arxiv.org/abs/2206.00330.

Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H. W., Narang, S., Yogatama, D., Vaswani, A., and Metzler, D. Scale efficiently: Insights from pretraining and finetuning transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=f2OYVDyfIB.

Tay, Y., Dehghani, M., Abnar, S., Chung, H. W., Fedus, W., Rao, J., Narang, S., Tran, V. Q., Yogatama, D., and Metzler, D. Scaling laws vs model architectures: How does inductive bias influence scaling? In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12342–12364. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.825.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021a. URL http://proceedings.mlr.press/v139/touvron21a.html.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 32–42. IEEE, 2021b. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.org/10.1109/ICCV48922.2021.00010.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton-Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288, 2023. URL https://arxiv.org/abs/2307.09288.

Trockman, A. and Kolter, J. Z. Mimetic initialization of self-attention layers. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34456–34468. PMLR, 2023. URL https://proceedings.mlr.press/v202/trockman23a.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wang, H., Ge, S., Lipton, Z. C., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 10506–10518, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3eefceb8087e964f89c2d59e8a249915-Abstract.html.

Wang, H., Ma, S., Huang, S., Dong, L., Wang, W., Peng, Z., Wu, Y., Bajaj, P., Singhal, S., Benhaim, A., Patra, B., Liu, Z., Chaudhary, V., Song, X., and Wei, F. Magneto: A foundation transformer. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of

Proceedings of Machine Learning Research, pp. 36077–36092. PMLR, 2023. URL https://proceedings.mlr.press/v202/wang23u.html.

Wang, H., Ma, S., Dong, L., Huang, S., Zhang, D., and Wei, F. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Wang, P., Zheng, W., Chen, T., and Wang, Z. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=O476oWmiNNp.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Wortsman, M., Liu, P. J., Xiao, L., Everett, K. E., Alemi, A. A., Adlam, B., Co-Reyes, J. D., Gur, I., Kumar, A., Novak, R., Pennington, J., Sohl-Dickstein, J., Xu, K., Lee, J., Gilmer, J., and Kornblith, S. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 2020. URL http://proceedings.mlr.press/v119/xiong20b.html.

Xu, J., Sun, X., Zhang, Z., Zhao, G., and Lin, J. Understanding and improving layer normalization. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4383–4393, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/2f4fe03d77724a7217006e5d16728874-Abstract.html.

Xue, F., Chen, J., Sun, A., Ren, X., Zheng, Z., He, X., Chen, Y., Jiang, X., and You, Y. A study on transformer configuration and training objective. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38913–38925. PMLR, 2023. URL https://proceedings.mlr.press/v202/xue23b.html.

Yang, G. and Schoenholz, S. S. Mean field residual networks: On the edge of chaos. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 7103–7114, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/81c650caac28cdefce4de5ddc18befa0-Abstract.html.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tuning large neural networks via zero-shot hyperparameter transfer. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17084–17097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8df7c2e3c3c3be098ef7b382bd2c37ba-Abstract.html.

Yang, G., Yu, D., Zhu, C., and Hayou, S. Tensor programs VI: Feature learning in infinite depth neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=17pVDnpwwl.

You, Y., Gitman, I., and Ginsburg, B. Scaling SGD batch size to 32k for imagenet training. *ArXiv preprint*, abs/1708.03888, 2017. URL https://arxiv.org/abs/1708.03888.

You, Y., Li, J., Reddi, S. J., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C. Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Syx4wnEtvH.

Yu, A. W., Lin, Q., Salakhutdinov, R., and Carbonell, J. G. Normalized gradient with adaptive stepsize method for deep neural network training. *ArXiv preprint*, abs/1707.04822, 2017. URL https://arxiv.org/abs/1707.04822.

Zhai, S., Likhomanenko, T., Littwin, E., Busbridge, D., Ramapuram, J., Zhang, Y., Gu, J., and Susskind, J. M. Stabilizing transformer training by preventing attention entropy collapse. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40770–40803. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhai23a.html.

Zhang, B., Titov, I., and Sennrich, R. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 898–909, Hong Kong, China, 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1083. URL https://aclanthology.org/D19-1083.

Zhang, G., Botev, A., and Martens, J. Deep learning without shortcuts: Shaping the kernel with tailored rectifiers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=U0k7XNTiFEq.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=H1gsz30cKX.

Zhang, H., Yu, D., Yi, M., Chen, W., and Liu, T. Stabilize deep resnet with a sharp scaling factor τ. *Mach. Learn.*, 111(9):3359–3392, 2022b. doi: 10.1007/S10994-022-06192-X. URL https://doi.org/10.1007/s10994-022-06192-x.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M. T., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. OPT: open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068, 2022c. URL https://arxiv.org/abs/2205.01068.

Zhao, H., Ma, S., Zhang, D., Deng, Z., and Wei, F. Are more layers beneficial to graph transformers? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=uagC-X9XMi8.

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Hou, Q., and Feng, J. Deepvit: Towards deeper vision transformer. *ArXiv preprint*, abs/2103.11886, 2021. URL https://arxiv.org/abs/2103.11886.

Zhu, C., Ni, R., Xu, Z., Kong, K., Huang, W. R., and Goldstein, T. Gradinit: Learning to initialize neural networks for stable and efficient training. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 16410–16422, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/88ae6372cfdc5df69a976e893f4d554b-Abstract.html.

Contents

A Moment Propagation through Transformer Components

We provide detailed proofs of the closed-form expression for each of the transformer component – Linear layer, Dropout, ReLU, GeLU, LayerNorm, and Softmax.

For any component, input is represented as $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ is the output. The gradient flowing in into the component from the output side is represented as $\mathbf{g}_{\text{out}}$ and the backpropagated gradient towards the input is $\mathbf{g}_{\text{in}}$. We switch from vector to matrix notation ($\mathbf{X}_{\text{in}}$, $\mathbf{X}_{\text{out}}$) whenever needed. We assume that the input is distributed normally $\mathcal{N}(0, \sigma_{x_{in}})$. No assumptions are made regarding the covariance of the input – it is not assumed to be IID, and it may/may-not have covariance both along the sequence length and hidden dimension. Additional assumptions needed to derive the proofs for softmax and attention can be found in the respective proofs. A detailed list of terms/notations used in the proofs is provided at the end of this work in Appendix O.

A.1 Embeddings

The BERT model's embedding component consists of 3 look up tables - token embeddings, position embeddings, and segment embeddings. For a given input token, each of these 3 embeddings are added before being passed to the transformer model. Other transformer models, such as decoder-only GPT lack some (eg. segment) of these, but the derivations remain similar. In the general case, these theoretical derivations can be replaced by the empirically observed moments of the inputs fed to the transformer model (as we did for Speech-to-Text translation). We derive formulae for each of these embedding types below.

Token Embeddings We do not assume the input embeddings to be IID. Repetition of same token introduces correlation across the sequence length. We assume that the input tokens have been sampled from a multinomial distribution. The words / token ids are distributed almost according to Zipf's law (Kingsley, 1935). Assuming we initialize all the embeddings with variance $\sigma^2_{w_{embd}}$, the relevant statistics for word embeddings output $x_{\text{out}_{we}}$ are as follows

$$\begin{aligned}
\mu_{x_{\text{out}_{we}}} &= 0 \\
\sigma^2_{x_{\text{out}_{we}}} &= \sigma^2_{w_{\text{embd}}} \\
\text{Cov}^l(x_{\text{out}_{we}}) &= \sum \frac{N_i * (N_i - 1)}{L * (L - 1)} * \sigma^2_{w_{\text{embd}}} \\
\text{r}^l(x_{\text{out}_{we}}) &= \sum \frac{N_i * (N_i - 1)}{L * (L - 1)} \\
\text{Cov}^d(x_{\text{out}_{we}}) &= 0
\end{aligned}$$

Assume ith word occurs N_i times, it contributes $\frac{N_i*(N_i-1)}{L*(L-1))}$ to the covariance along sequence length. Similarly, we can calculate the correlation for segment-type embeddings output $x_{\text{out}_{se}}$. Zipf's law states that the probability for each token is inversely proportional to its rank. For the word with rank i, $p_i = \frac{c}{i}$, where $c = \frac{1}{\sum_i \frac{1}{i}} = \frac{1}{\gamma + \log(|V|)}$, where $\gamma \approx 0.58$ is the Euler's constant.

For a sentence of length L, the token with probability p_i is expected to occur $p_i.L$ times. Hence, for a given vocabulary size $|V|$, we can calculate the correlation as follows

$$\begin{aligned}
r^l(x_{\text{out}_{we}}) &= \sum \frac{N_i * (N_i - 1)}{L * (L - 1)} \\
&= \sum_i^{|V|} \frac{p_i L * (p_i L - 1)}{L * (L - 1)} \\
&= \frac{\sum_i p_i^2 * L - 1}{L - 1} \\
&= \frac{\sum_i \frac{c^2}{i^2} * L - 1}{L - 1} \\
&\approx \frac{\frac{L\pi^2}{6.(\gamma + \log(|V|))^2} - 1}{L - 1} \\
&\approx \frac{\pi^2}{6.\log(|V|)^2} \quad \text{, assuming } \gamma \approx 0.58 << \log(|V|) \approx 10.4, L >> 1
\end{aligned}$$

Segment Type Embeddings Similarly, the segment type embeddings have two possible values denoting the sentence order. If first sentence has length x, we can consider this as a special case of the analysis performed above with two possible tokens, where $N_1 = x$ and $N_2 = L - x$. Assuming x is distributed uniformly between 0 to L, $L - x$ also has the same

distribution. Hence,

$$r^l(x_{\text{out}_{se}}, N_1, N_2) = \frac{N_1^2 + N_2^2 - L}{L * (L-1)}$$

Taking expectation, we get

$$\begin{aligned} r^l(x_{\text{out}_{se}}) &= \frac{\frac{2}{3} * L^2 - L}{L * (L-1)} \\ &\approx \frac{2}{3} \end{aligned}$$

Position Embeddings Since learnt position embeddings are lookup tables with unique inputs, the correlation from position embeddings is 0.

Final Model Input Embeddings Each of the above embeddings are added before being passed to the transformer model. Since the variance is same for all embedding types, the final correlation is the average of the three. Hence:

$$\begin{aligned} r^l(x_{\text{out}}) &= \frac{1}{3}(r^l(x_{\text{out}_{we}}) + r^l(x_{\text{out}_{se}})) \\ &= \frac{\pi^2}{18 * \log(|V|)^2} + \frac{2}{9} \end{aligned}$$

For our case, $|V| = 32000$ and sequence length $L = 256$, the theoretically predicted correlation $r^l_{x_{in}} = 0.227$ which is within 3% of the empirically observed correlation (0.221).

Hence, the final moments for the embedding output are

$$\boxed{\begin{aligned} \mu_{x_{\text{out}}} &= 0 \\ \sigma^2_{x_{\text{out}}} &= 3 * \sigma^2_{w_{\text{embd}}} \\ \text{Cov}^l_{x_{\text{out}}} &= (\frac{\pi^2}{18 * \log(|V|)^2} + \frac{2}{9})\sigma^2_{x_{\text{out}}} \\ \text{Cov}^d_{x_{\text{out}}} &= 0 \end{aligned}}$$

A.2 Linear

For linear layer with d_{in} dimensional input $\mathbf{x}_{\text{in}}$, and d_{out} dimensional output $\mathbf{x}_{\text{out}}$, we can define the forward pass mathematically as,

$$\begin{aligned} \mathbf{x}_{\text{out}} &= \mathbf{x}_{\text{in}}\mathbf{W} \\ \implies x_{\text{out}_j} &= \sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} W_{i,j} \end{aligned}$$

Similarly, we define the backward pass as,

$$\begin{aligned} \mathbf{g}_{\text{in}} &= \mathbf{g}_{\text{out}}\mathbf{W}^{\mathbf{T}} \\ \implies g_{\text{in}_j} &= \sum_{i=1}^{d_{\text{out}}} g_{\text{out}_i} W_{j,i} \end{aligned}$$

For expectation of output we have,

$$\mathbb{E}[x_{\text{out}_j}] = \mathbb{E}[\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} W_{i,j}] = \sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{in}_i} W_{i,j}]$$
$$= \sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{in}_i}]\mathbb{E}[W_{i,j}] = \mu_{x_{\text{in}}}\mu_w$$

(As weights and input are independent of each other)

$$\boxed{\mu_{x_{\text{out}}} = 0} \qquad (\forall j)$$

To get variance of the output of forward pass we have,

$$\text{Var}(x_{\text{out}_j}) = \text{Var}(\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} W_{i,j})$$

As the weights are initialized independently each term in summation is independent of each other

$$= \sum_{i=1}^{d_{\text{in}}} (\text{Var}(x_{\text{in}_i} W_{i,j}))$$
$$= \sum_{i=1}^{d_{\text{in}}} ((\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})(\sigma^2_w + \mu^2_w) - \mu^2_{x_{\text{in}}}\mu^2_w)$$

(As weights and input are independent of each other)

$$= \sum_{i=1}^{d_{\text{in}}} (\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w$$
$$\text{Var}(x_{\text{out}_j}) = d_{\text{in}}(\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w \qquad (\forall j)$$
$$\boxed{\sigma^2_{x_{\text{out}}} = d_{\text{in}}(\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w}$$

If we have two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ such that for all i we have $\text{Corr}(x_{\text{in}_i}, y_{\text{in}_i}) = r^l_{x_{\text{in}}}$, and $\mathbf{x}_{\text{out}} = \mathbf{x}_{\text{in}}\mathbf{W}$ and $\mathbf{y}_{\text{out}} = \mathbf{y}_{\text{in}}\mathbf{W}$. Then for any j we have

$$\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}]}{\sqrt{\text{Var}(x_{\text{out}_j})\text{Var}(y_{\text{out}_j})}}$$
$$= \frac{\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}]}{\sqrt{\sigma^2_{x_{\text{out}}}\sigma^2_{x_{\text{out}}}}}$$
$$= \frac{\mathbb{E}[\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} W_{i,j} \sum_{k=1}^{d_{in}} y_{\text{in}_k} W_{k,j}]}{\sigma^2_{x_{\text{out}}}}$$
$$= \frac{\mathbb{E}[\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i} W^2_{i,j} + \sum_{k=1,k\neq i}^{d_{in}} \sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} y_{\text{in}_k} W_{i,j} W_{k,j}]}{\sigma^2_{x_{\text{out}}}}$$

In second summation all terms are independent of each other and as the expectation of weights is 0 we have

$$\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{\mathbb{E}[\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i} W^2_{i,j}]}{\sigma^2_{x_{\text{out}}}}$$
$$= \frac{\sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{in}_i} y_{\text{in}_i} W^2_{i,j}]}{\sigma^2_{x_{\text{out}}}} \qquad \text{(Independence of weight initialization)}$$
$$= \frac{\sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{in}_i} y_{\text{in}_i}]\mathbb{E}[W^2_{i,j}]}{\sigma^2_{x_{\text{out}}}}$$

$$= \frac{\sum_{i=1}^{d_{\text{in}}}(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w}{\sigma^2_{x_{\text{out}}}} \qquad \text{(Definition of correlation)}$$

$$= \frac{d_{\text{in}}(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w}{d_{\text{in}}(\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})\sigma^2_w}$$

$$\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}}$$

$$\boxed{r^l_{x_{\text{out}}} = \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}}}$$

As the backward pass has similar structure, assuming $\mu_{g_{\text{out}}} = 0$ we can use the same analysis to get,

$$\boxed{\begin{aligned}\mu_{g_{\text{in}}} &= 0\\ \sigma^2_{g_{\text{in}}} &= d_{\text{out}}\sigma^2_{g_{\text{out}}}\sigma^2_w\end{aligned}}$$

A.3 Dropout

We can define Dropout mathematically as,

$$\mathbf{x}_{\text{out}} = \text{Dropout}(\mathbf{x}_{\text{in}})$$

$$\implies x_{\text{out}_i} = \begin{cases}\frac{x_{\text{in}_i}}{(1-p)} & \text{with probability } 1-p\\ 0 & \text{else}\end{cases}$$

To calculate expectation of dropout,

$$\mathbb{E}[x_{\text{out}_i}] = 0 * p + (1-p) * \mathbb{E}[\frac{x_{\text{in}_i}}{(1-p)}]$$

$$\boxed{\mu_{x_{\text{out}}} = \mu_{x_{\text{in}}}}$$

For variance,

$$\text{Var}(x_{\text{out}_i}) = \mathbb{E}[x^2_{\text{out}_i}] - \mathbb{E}[x_{\text{out}_i}]^2$$

$$= 0 * p + (1-p) * \mathbb{E}[\frac{x^2_{\text{in}_i}}{(1-p)^2}] - \mu^2_{x_{\text{in}}}$$

$$= \frac{\mathbb{E}[x^2_{\text{in}_i}]}{(1-p)} - \mu^2_x$$

$$= \frac{\sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}}}{(1-p)} - \mu^2_{x_{\text{in}}}$$

$$\boxed{\sigma^2_{x_{\text{out}}} = \frac{\sigma^2_{x_{\text{in}}} + p\mu^2_{x_{\text{in}}}}{(1-p)}}$$

If we have two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ such that for all i we have $\text{Corr}(x_{\text{in}_i}, y_{\text{in}_i}) = r^l_{x_{\text{in}}}$, and $\mathbf{x}_{\text{out}} = \text{Dropout}(\mathbf{x}_{\text{in}})$ and $\mathbf{y}_{\text{out}} = \text{Dropout}(\mathbf{y}_{\text{in}})$. Then for any j we have

$$\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{\mathbb{E}[x_{\text{out}_j}y_{\text{out}_j}] - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}]}{\sqrt{\text{Var}(x_{\text{out}_j})\text{Var}(y_{\text{out}_j})}}$$

$$= \frac{\mathbb{E}[x_{\text{out}_j}y_{\text{out}_j}] - \mu_{x_{\text{out}}}\mu_{x_{\text{out}}}}{\sqrt{\sigma^2_{x_{\text{out}}}\sigma^2_{x_{\text{out}}}}}$$

$$= \frac{p^2 * 0 + 2 * p * (1-p) * 0 + (1-p)^2 * \mathbb{E}[\frac{x_{\text{in}_j}y_{\text{in}_j}}{(1-p)*(1-p)}] - \mu^2_{x_{\text{out}}}}{\sigma^2_{x_{\text{out}}}}$$

$$= \frac{\mathbb{E}[x_{\text{in}_j} y_{\text{in}_j}] - \mu^2_{x_{\text{out}}}}{\sigma^2_{x_{\text{out}}}}$$

$$\boxed{\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{(r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}})(1-p)}{\sigma^2_{x_{\text{in}}} + p\mu^2_{x_{\text{in}}}} = r^l_{x_{\text{out}}}}$$

We can define the backward pass of Dropout as,

$$g_{\text{in}_i} = \begin{cases} \frac{g_{\text{out}_i}}{(1-p)} & \text{if } x_i \text{ isn't dropped out (which has probability } (1-p)) \\ 0 & \text{else} \end{cases}$$

Again we can see that backward has similar definition to that of forward pass. Assuming $\mu_{g_{x_{\text{out}}}} = 0$ and using similar analysis we get,

$$\boxed{\begin{aligned} \mu_{g_{\text{in}}} &= 0 \\ \sigma^2_{g_{\text{in}}} &= \frac{\sigma^2_{g_{\text{out}}}}{(1-p)} \end{aligned}}$$

A.4 ReLU

Formuale functionally equivalent to ours for μ_x, σ^2_x, and σ^2_g have also been derived in Arpit et al. (2016).

We can define ReLU mathematically as,

$$\mathbf{x}_{\text{out}} = \text{ReLU}(\mathbf{x}_{\text{in}})$$
$$\implies x_{\text{out}_i} = \begin{cases} x_{\text{in}_i} & \text{if } x_{\text{in}_i} > 0 \\ 0 & \text{else} \end{cases}$$

For getting expectation of output of ReLU for normally distributed input we have,

$$\begin{aligned} \mathbb{E}[x_{\text{out}_i}] &= \int_{-\infty}^{\infty} \frac{\text{ReLU}(x_{\text{in}_i}) \exp\left(\frac{-x^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right)}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} \\ &= \int_{-\infty}^{0} \frac{0 * \exp\left(\frac{-x^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right)}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} + \int_{0}^{\infty} \frac{x_{\text{in}_i} \exp\left(\frac{-x^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right)}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} \\ &= \int_{0}^{\infty} \frac{x_{\text{in}_i} \exp\left(\frac{-x^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right)}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} \end{aligned}$$

Substituting $t = \frac{x^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}$ we have $dt = \frac{x_{\text{in}_i} dx_{\text{in}_i}}{\sigma^2_{x_{\text{in}}}}$ we get,

$$\begin{aligned} \mathbb{E}[x_{\text{out}_i}] &= \int_{0}^{\infty} \frac{\sigma_{x_{\text{in}}} \exp(-t) dt}{\sqrt{2\pi}} \\ &= \frac{\sigma_{x_{\text{in}}}}{\sqrt{2\pi}} [-\exp(-t)]_0^{\infty} = \frac{\sigma_{x_{\text{in}}}}{\sqrt{2\pi}} \end{aligned}$$

Hence, the mean of output

$$\boxed{\mu_{x_{\text{out}}} = \frac{\sigma_{x_{\text{in}}}}{\sqrt{2\pi}}} \tag{1}$$

Variance of output can be calculated by,

$$\text{Var}(x_{\text{out}_i}) = \mathbb{E}[{x_{\text{out}_i}}^2] - \mathbb{E}[x_{\text{out}_i}]^2$$

$$= \int_{-\infty}^{\infty} \frac{(\mathrm{ReLU}(x_{\mathrm{in}_i}))^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i} - \frac{\sigma_{x_{\mathrm{in}}}^2}{2\pi}$$

$$= \int_{-\infty}^{0} \frac{0 * \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i} + \int_{0}^{\infty} \frac{x_{\mathrm{in}_i}^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i} - \frac{\sigma_{x_{\mathrm{in}}}^2}{2\pi}$$

$$= \int_{0}^{\infty} \frac{x_{\mathrm{in}_i}^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i} - \frac{\sigma_{x_{\mathrm{in}}}^2}{2\pi}$$

Let $I = \int_{0}^{\infty} \frac{x_{\mathrm{in}_i}^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i}$, then substituting $t = -x_{\mathrm{in}_i}$ we have,

$$I = \int_{0}^{-\infty} \frac{-t^2 \exp\left(\frac{-t^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dt$$

$$= \int_{-\infty}^{0} \frac{t^2 \exp\left(\frac{-t^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dt$$

$$\implies I + I = \int_{-\infty}^{0} \frac{t^2 \exp\left(\frac{-t^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dt + \int_{0}^{\infty} \frac{x_{\mathrm{in}_i}^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i}$$

$$2I = \int_{-\infty}^{\infty} \frac{x_{\mathrm{in}_i}^2 \exp\left(\frac{-x_{\mathrm{in}_i}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} dx_{\mathrm{in}_i} = \sigma_{x_{\mathrm{in}}}^2$$

$$\implies \mathrm{Var}(x_{\mathrm{out}_i}) = \frac{\sigma_{x_{\mathrm{in}}}^2}{2} - \frac{\sigma_{x_{\mathrm{in}}}^2}{2\pi} = \frac{\sigma_{x_{\mathrm{in}}}^2}{2}(1 - \frac{1}{\pi})$$

$$\boxed{\sigma_{x_{\mathrm{out}}}^2 = \frac{\sigma_{x_{\mathrm{in}}}^2}{2}(1 - \frac{1}{\pi})}$$

Now for two inputs $\mathbf{x}_{\mathrm{in}}$ and $\mathbf{y}_{\mathrm{in}}$ such that for all i we have $\mathrm{Corr}(\mathrm{x}_{\mathrm{in}_i}, \mathrm{y}_{\mathrm{in}_i}) = r_{x_{\mathrm{in}}}^l$, and $\mathbf{x}_{\mathrm{out}} = \mathrm{ReLU}(\mathbf{x}_{\mathrm{in}})$ and $\mathbf{y}_{\mathrm{out}} = \mathrm{ReLU}(\mathbf{y}_{\mathrm{in}})$. Then for any j we have,

$$\mathrm{Corr}(x_{\mathrm{out}_j}, y_{\mathrm{out}_j}) = \frac{\mathbb{E}[x_{\mathrm{out}_j} y_{\mathrm{out}_j}] - \mathbb{E}[x_{\mathrm{out}_j}]\mathbb{E}[y_{\mathrm{out}_j}]}{\sqrt{\mathrm{Var}(x_{\mathrm{out}_j})\mathrm{Var}(y_{\mathrm{out}_j})}}$$

$$\mathbb{E}[x_{\mathrm{out}_j} y_{\mathrm{out}_j}] = \int_0^{\infty}\int_0^{\infty} \frac{x_{\mathrm{in}_j} y_{\mathrm{in}_j}}{2\pi\sigma_{x_{\mathrm{in}}}^2\sqrt{1-(r_{x_{\mathrm{in}}}^l)^2}} \exp\left(\frac{-(x_{\mathrm{in}_j}^2 + y_{\mathrm{in}_j}^2 - 2r_{x_{\mathrm{in}}}^l x_{\mathrm{in}_j} y_{\mathrm{in}_j})}{2\sigma_{x_{\mathrm{in}}}^2(1-(r_{x_{\mathrm{in}}}^l)^2)}\right) dx_{\mathrm{in}_j} dy_{\mathrm{in}_j}$$

$$= \int_0^{\infty}\int_0^{\infty} \frac{x_{\mathrm{in}_j} y_{\mathrm{in}_j}}{2\pi\sigma_{x_{\mathrm{in}}}^2\sqrt{1-(r_{x_{\mathrm{in}}}^l)^2}} \exp\left(\frac{-(x_{\mathrm{in}_j} - r_{x_{\mathrm{in}}}^l y_{\mathrm{in}_j})^2}{2\sigma_{x_{\mathrm{in}}}^2(1-(r_{x_{\mathrm{in}}}^l)^2)}\right) \exp\left(\frac{-y_{\mathrm{in}_j}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right) dx_{\mathrm{in}_j} dy_{\mathrm{in}_j}$$

Substituting $t = x_{\mathrm{in}_j} - r_{x_{\mathrm{in}}}^l y_{\mathrm{in}_j}$, and assuming y_{in_j} is constant for the inner integral,$dx_{\mathrm{in}_j} = dt$

$$\mathbb{E}[x_{\mathrm{out}_j} y_{\mathrm{out}_j}] =$$

$$= \int_0^{\infty} \frac{y_{\mathrm{in}_j} \exp\left(\frac{-y_{\mathrm{in}_j}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right)}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} \int_{-r_{x_{\mathrm{in}}}^l y_{\mathrm{in}_j}}^{\infty} \frac{t + r_{x_{\mathrm{in}}}^l y_{\mathrm{in}_j}}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}\sqrt{1-(r_{x_{\mathrm{in}}}^l)^2}} \exp\left(\frac{-t^2}{2\sigma_{x_{\mathrm{in}}}^2(1-(r_{x_{\mathrm{in}}}^l)^2)}\right) dt dy_{\mathrm{in}_j}$$

$$= \int_0^{\infty} \frac{y_{\mathrm{in}_j}}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}} \exp\left(\frac{-y_{\mathrm{in}_j}^2}{2\sigma_{x_{\mathrm{in}}}^2}\right) \int_{-r_{x_{\mathrm{in}}}^l y_{\mathrm{in}_j}}^{\infty} \frac{t}{\sqrt{2\pi}\sigma_{x_{\mathrm{in}}}\sqrt{1-(r_{x_{\mathrm{in}}}^l)^2}} \exp\left(\frac{-t^2}{2\sigma_{x_{\mathrm{in}}}^2(1-(r_{x_{\mathrm{in}}}^l)^2)}\right) dt dy_{\mathrm{in}_j}$$

$$+\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_x}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}$$

Let us first define I_1 and I_2 as:

$$I_1=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{t}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}$$

$$I_2=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_x}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}$$

$$I_1=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{t}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}$$

Substituting $p=\frac{t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}$ we have $dp=\frac{tdt}{\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}$

$$\begin{aligned}
I_1&=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{\frac{(r^l_{x_{\text{in}}}y_{\text{in}_j})^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}}^\infty \frac{\sigma_{x_{\text{in}}}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{\sqrt{2\pi}}\exp(-p)dpdy_{\text{in}_j}\\
&=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\frac{\sigma_{x_{\text{in}}}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{\sqrt{2\pi}}\exp\left(\frac{-(r^l_{x_{\text{in}}}y_{\text{in}_j})^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dy_{\text{in}_j}\\
&=\int_0^\infty \frac{y_{\text{in}_j}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{2\pi}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dy_{\text{in}_j}
\end{aligned}$$

Substituting $m=\frac{y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}$, $dm=\frac{y_{\text{in}_j}dy_{\text{in}_j}}{\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}$,

$$\begin{aligned}
I_1&=\int_0^\infty \frac{\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{2\pi}(1-(r^l_{x_{\text{in}}})^2)\sigma_{x_{\text{in}}}^2\exp(-m)dm\\
&=\frac{(1-(r^l_{x_{\text{in}}})^2)^{\frac{3}{2}}\sigma_{x_{\text{in}}}^2}{2\pi}\\
I_2&=\int_0^\infty \frac{y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}\\
&=\int_0^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}^2}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-r^l_{x_{\text{in}}}y_{\text{in}_j}}^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-t^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dtdy_{\text{in}_j}
\end{aligned}$$

Substituting $p=-t$, where Φ is CDF of Standard Normal Distribution

$$\begin{aligned}
I_2&=\int_0^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}^2}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{r^l_{x_{\text{in}}}y_{\text{in}_j}}^{-\infty} \frac{-1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-p^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dpdy_{\text{in}_j}\\
&=\int_0^\infty \frac{r^l_{x_{\text{in}}}y_{\text{in}_j}^2}{\sqrt{2\pi}\sigma_{x_{\text{in}}}}\exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right)\int_{-\infty}^{r^l_{x_{\text{in}}}y_{\text{in}_j}} \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\exp\left(\frac{-p^2}{2\sigma_{x_{\text{in}}}^2(1-(r^l_{x_{\text{in}}})^2)}\right)dpdy_{\text{in}_j}
\end{aligned}$$

$$= \int_0^\infty \frac{r^l_{x_{\text{in}}} y^2_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)\Phi\left(\frac{r^l_{x_{\text{in}}} y_{\text{in}_j}}{\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}\right)dy_{\text{in}_j}$$

$$= \int_0^\infty \frac{r^l_{x_{\text{in}}} y^2_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)\left[\frac{1}{2}\left(1+\text{erf}\left(\frac{r^l_{x_{\text{in}}} y_{\text{in}_j}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r^l_{x_{\text{in}}})^2)}}\right)\right)\right]dy_{\text{in}_j}$$

$$= \frac{r^l_{x_{\text{in}}}}{2}\int_0^\infty \frac{y^2_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)dy_{\text{in}_j} +$$

$$\frac{r^l_{x_{\text{in}}}}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_0^\infty y^2_{\text{in}_j} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)\text{erf}\left(\frac{r^l_{x_{\text{in}}} y_{\text{in}_j}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r^l_{x_{\text{in}}})^2)}}\right)dy_{\text{in}_j}$$

Let us define $I_{2,1}$ and $I_{2,2}$ as

$$I_{2,1} = \frac{r^l_{x_{\text{in}}}}{2}\int_0^\infty \frac{y^2_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)dy_{\text{in}_j}$$

$$I_{2,2} = \frac{r^l_{x_{\text{in}}}}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_0^\infty y^2_{\text{in}_j} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)\text{erf}\left(\frac{r^l_{x_{\text{in}}} y_{\text{in}_j}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r^l_{x_{\text{in}}})^2)}}\right)dy_{\text{in}_j}$$

$$I_{2,1} = \frac{r^l_{x_{\text{in}}}}{2}\int_0^\infty \frac{y^2_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}}\right)dy_{\text{in}_j}$$

$$I_{2,1} = \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{4} \qquad \text{(Same integral as in variance calculation)}$$

From Ng & Geller (1969) we have $\int_0^\infty x^2 \exp(-b^2x^2)\text{erf}(ax)dx = \frac{\sqrt{\pi}}{4b^3} - \frac{\tan^{-1}(\frac{b}{a})}{2\sqrt{\pi}b^3} + \frac{a}{2\sqrt{\pi}b^2(a^2+b^2)}$.

Hence, putting $a = \frac{r^l_{x_{\text{in}}}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r^l_{x_{\text{in}}})^2)}}$ and $b = \frac{1}{\sigma_{x_{\text{in}}}\sqrt{2}}$ we get,

$$I_{2,2} = \frac{r^l_{x_{\text{in}}}}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\left[\frac{2\sqrt{2}\sigma^3_{x_{\text{in}}}}{4} - \frac{\tan^{-1}\left(\frac{\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{r^l_{x_{\text{in}}}}\right)2\sqrt{2}\sigma^3_{x_{\text{in}}}}{2\sqrt{\pi}} + \frac{\sqrt{2}r^l_{x_{\text{in}}}\sigma^3_{x_{\text{in}}}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{\sqrt{\pi}}\right]$$

$$= \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{4} - \frac{r^l_{x_{\text{in}}}\cos^{-1}(r^l_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}}{2\pi} + \frac{(r^l_{x_{\text{in}}})^2\sqrt{(1-(r^l_{x_{\text{in}}})^2)}\sigma^2_{x_{\text{in}}}}{2\pi}$$

$$\mathbb{E}[x_{\text{out}_j}y_{\text{out}_j}] = I_1 + I_{2,1} + I_{2,2}$$

$$= \frac{(1-(r^l_{x_{\text{in}}})^2)^{\frac{3}{2}}\sigma^2_{x_{\text{in}}}}{2\pi} + 2*\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{4} - \frac{r^l_{x_{\text{in}}}\cos^{-1}(r^l_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}}{2\pi} + \frac{(r^l_{x_{\text{in}}})^2\sqrt{(1-(r^l_{x_{\text{in}}})^2)}\sigma^2_{x_{\text{in}}}}{2\pi}$$

$$= \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2} - \frac{r^l_{x_{\text{in}}}\cos^{-1}(r^l_{x_{\text{in}}})\sigma^2_{x}}{2\pi} + \frac{\sqrt{(1-(r^l_{x_{\text{in}}})^2)}\sigma^2_{x_{\text{in}}}}{2\pi}$$

$$\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{\mathbb{E}[x_{\text{out}_j}y_{\text{out}_j}] - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}]}{\sqrt{\text{Var}(x_{\text{out}_j})\text{Var}(y_{\text{out}_j})}}$$

$$= \frac{\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2} - \frac{r^l_{x_{\text{in}}}\cos^{-1}(r^l_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}}{2\pi} + \frac{\sqrt{(1-(r^l_{x_{\text{in}}})^2)}\sigma^2_{x}}{2\pi} - \frac{\sigma^2_{x_{\text{in}}}}{2\pi}}{\frac{\sigma^2_{x_{\text{in}}}}{2}(1-\frac{1}{\pi})}$$

$$\boxed{r^l_{x_{\text{out}}} = \frac{\frac{\pi r^l_{x_{\text{in}}}}{2} + r^l_{x_{\text{in}}} \sin^{-1}(r^l_{x_{\text{in}}}) + \sqrt{(1-(r^l_{x_{\text{in}}})^2)} - 1}{\pi - 1}}$$

Backward pass on ReLU can be defined as,

$$g_{\text{in}_i} = \begin{cases} g_{\text{out}_i} & \text{if } x_{\text{in}_i} > 0 \text{ (which has probability } \frac{1}{2}) \\ 0 & \text{else} \end{cases}$$

Assuming $\mu_{g_{\text{out}}} = 0$,

$$\begin{aligned} \mathbb{E}[g_{\text{in}_i}] &= \frac{1}{2} * 0 + \frac{1}{2} * \mathbb{E}[g_{\text{out}_i}] \\ \boxed{\mu_{g_{\text{in}}} = 0} & \\ \text{Var}(g_{\text{in}_i}) = \mathbb{E}[g^2_{\text{in}_i}] - \mathbb{E}[g_{\text{in}_i}]^2 &= \mathbb{E}[g^2_{\text{in}_i}] \\ &= \frac{1}{2} * 0 + \frac{1}{2} * \mathbb{E}[g^2_{\text{out}}] \\ \boxed{\sigma^2_{g_{\text{in}}} = \frac{\sigma^2_{g_{\text{out}}}}{2}} & \end{aligned}$$

If for two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ for all i we have $\text{Corr}(g_{\text{out}_{x_i}}, g_{\text{out}_{y_i}}) = r^l_{g_{\text{out}}}$, and $g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}}$ be the gradient after passing through ReLU layer. Then we have,

$$\begin{aligned} \mathbb{E}[g_{\text{in}_{x_i}} g_{\text{in}_{y_i}}] &= \mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0)\mathbb{E}[g_{\text{out}_{x_i}} g_{\text{out}_{y_i}}] \\ &= \mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0) r^l_{g_{\text{out}}} \sigma^2_{g_{\text{out}}} \end{aligned}$$

$$\begin{aligned} &\mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0) = \\ &= \int_0^\infty \int_0^\infty \frac{x_{\text{in}_i} y_{\text{in}_i}}{2\pi\sigma^2_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}} \exp\left(\frac{-(x^2_{\text{in}_i} + y^2_{\text{in}_i} - 2r^l_{x_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i})}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right) dx_{\text{in}_i} dy_{\text{in}_i} \\ &= \int_0^\infty \int_0^\infty \frac{x_{\text{in}_i} y_{\text{in}_i}}{2\pi\sigma^2_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}} \exp\left(\frac{-(x_{\text{in}_i} - r^l_{x_{\text{in}}} y_{\text{in}_i})^2}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right) \exp\left(\frac{-y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right) dx_{\text{in}_i} dy_{\text{in}_i} \end{aligned}$$

Substituting $t = x_{\text{in}_i} - r^l_{x_{\text{in}}} y_{\text{in}_i}$, and assuming y_{in_i} is constant for the inner integral,$dx_{\text{in}_i} = dt$

$$\begin{aligned} &\mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0) = \\ &\int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right) \int_{-r^l_{x_{\text{in}}} y_{\text{in}_i}}^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}} \exp\left(\frac{-t^2}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right) dt dy_{\text{in}_i} \end{aligned}$$

Substituting $p = -t$, where Φ is CDF of Standard Normal Distribution

$$\begin{aligned} &\mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0) = \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right) \int_{r^l_{x_{\text{in}}} y_{\text{in}_i}}^{-\infty} \frac{-1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}} \exp\left(\frac{-p^2}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right) dp dy_{\text{in}_i} \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}}\right) \int_{-\infty}^{r^l_{x_{\text{in}}} y_{\text{in}_i}} \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}} \exp\left(\frac{-p^2}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right) dp dy_{\text{in}_i} \end{aligned}$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)\Phi\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sigma_{x_{\text{in}}}\sqrt{1-(r_{x_{\text{in}}}^l)^2}}\right)dy_{\text{in}_i}$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)\left[\frac{1}{2}\left(1+\text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r_{x_{\text{in}}}^l)^2)}}\right)\right)\right]dy_{\text{in}_i}$$

$$= \frac{1}{2}\int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)dy_{\text{in}_i} + \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_0^\infty \exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)\text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r_{x_{\text{in}}}^l)^2)}}\right)dy_{\text{in}_i}$$

$$= \frac{1}{4} + \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_0^\infty \exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)\text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r_{x_{\text{in}}}^l)^2)}}\right)dy_{\text{in}_i}$$

From Ng & Geller (1969) we have $\int_0^\infty \exp(-b^2x^2)\text{erf}(ax)dx = \frac{\sqrt{\pi}}{2b} - \frac{1}{b\sqrt{\pi}}\tan^{-1}\left(\frac{b}{a}\right)$

Putting $a = \frac{r_{x_{\text{in}}}^l}{\sigma_{x_{\text{in}}}\sqrt{2(1-(r_{x_{\text{in}}}^l)^2)}}$ and $b = \frac{1}{\sigma_{x_{\text{in}}}\sqrt{2}}$ we get,

$$\begin{aligned}
\mathbb{P}(x_{\text{in}_i} > 0, y_{\text{in}_i} > 0) &= \frac{1}{4} + \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}}\left[\frac{\sqrt{\pi}\sigma_{x_{\text{in}}}\sqrt{2}}{2} - \frac{\sigma_{x_{\text{in}}}\sqrt{2}}{\sqrt{\pi}}\tan^{-1}\left(\frac{\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}{r_{x_{\text{in}}}^l}\right)\right] \\
&= \frac{1}{4} + \frac{1}{2\pi}\left[\frac{\pi}{2} - \cos^{-1}(r_{x_{\text{in}}}^l)\right] \\
&= \frac{1}{4} + \frac{\sin^{-1}(r_{x_{\text{in}}}^l)}{2\pi} \\
\implies \mathbb{E}[g_{\text{in}_{x_i}} g_{\text{in}_{y_i}}] &= \left(\frac{1}{4} + \frac{\sin^{-1}(r_{x_{\text{in}}}^l)}{2\pi}\right) r_{g_{\text{out}}}^l \sigma_{g_{\text{out}}}^2 \\
\text{Corr}(g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}}) &= \frac{\left(\frac{1}{4} + \frac{\sin^{-1}(r_{x_{\text{in}}}^l)}{2\pi}\right) r_{g_{\text{out}}}^l \sigma_{g_{\text{out}}}^2}{\frac{\sigma_{g_{\text{out}}}^2}{2}}
\end{aligned}$$

$$\boxed{r_{g_{\text{out}}}^l = \left(\frac{1}{2} + \frac{\sin^{-1}(r_{x_{\text{in}}}^l)}{\pi}\right) r_{g_{\text{out}}}^l}$$

A.5 GeLU

Forward pass through GeLU is defined as,

$$\begin{aligned}
\mathbf{x}_{\text{out}} &= \text{GeLU}(\mathbf{x}_{\text{in}}) \\
\implies x_{\text{out}_i} &= x_{\text{in}_i}\Phi(x_{\text{in}_i})
\end{aligned}$$

where $\Phi(x)$ is CDF of Standard Normal Distribution at x

$$= \frac{x_{\text{in}_i}}{2}\left(1 + \text{erf}\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right)\right)$$

To get the mean of output of GeLU, we have

$$\begin{aligned}
\mathbb{E}[x_{\text{out}_i}] &= \int_{-\infty}^\infty \frac{x_{\text{out}_i}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)dx_{\text{in}_i} \\
&= \int_{-\infty}^\infty \frac{x_{\text{in}_i}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right)dx_{\text{in}_i}
\end{aligned}$$

$$= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i} + \int_{-\infty}^{\infty} \frac{x_{\text{in}_i} \text{erf}\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right)}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i} \text{erf}\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right)}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i} \qquad \text{(Integral of odd function)}$$

$$= \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} x_{\text{in}_i} \text{erf}\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right) \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

From 2.6.1.4 of Korotkov & Korotkov (2020), $\int_{-\infty}^{\infty} z \text{erf}(az) \exp\left(-a_1 z^2\right) dz = \frac{a}{a_1\sqrt{a^2 + a_1}}$

Substituting, $a = \frac{1}{\sqrt{2}}, a_1 = \frac{1}{2\sigma_{x_{\text{in}}}^2}$, we have

$$\mathbb{E}[x_{\text{out}_i}] = \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \frac{\frac{1}{\sqrt{2}}}{\frac{1}{2\sigma_{x_{\text{in}}}^2}\sqrt{\frac{1}{2} + \frac{1}{2\sigma_{x_{\text{in}}}^2}}}$$

$$= \frac{1}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \frac{2\sigma_{x_{\text{in}}}^3}{\sqrt{\sigma_{x_{\text{in}}}^2 + 1}}$$

$$\boxed{\mu_{x_{\text{out}}} = \frac{\sigma_{x_{\text{in}}}^2}{\sqrt{2\pi(\sigma_{x_{\text{in}}}^2 + 1)}}}$$

For calculating variance of output,

$$\mathbb{E}[x_{\text{out}_i}^2] = \int_{-\infty}^{\infty} \frac{x_{\text{out}_i}^2}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}^2 (1 + \text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))^2}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}^2}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$+ \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}^2 \text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{2\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i} + \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}^2 \text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$= \frac{\sigma_{x_{\text{in}}}^2}{4} + \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}^2 \text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i} \qquad \text{(Definition of variance, and integral of odd function)}$$

$$= \frac{\sigma_{x_{\text{in}}}^2}{4} + \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} x_{\text{in}_i}^2 \text{erf}^2\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right) \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

From 2.7.3.3 of Korotkov & Korotkov (2020)

$$\int_{-\infty}^{\infty} z^2 \exp\left(-az^2\right) \text{erf}(a_1 z) \text{erf}(a_2 z) =$$

$$\frac{1}{\sqrt{\pi}}\left(\frac{1}{a\sqrt{a}} \tan^{-1}\left(\frac{a_1 a_2}{\sqrt{a^2 + a a_1^2 + a a_2^2}}\right) + \frac{a_1 a_2 (2a + a_1^2 + a_2^2)}{a\sqrt{a + a_1^2 + a_2^2}(a^2 + a a_1^2 + a a_2^2 + a_1^2 a_2^2)}\right)$$

Substituting $a = \frac{1}{2\sigma_{x_{\text{in}}}^2}, a_1 = a_2 = \frac{1}{\sqrt{2}}$

$$\int_{-\infty}^{\infty} x_{\text{in}_i}^2 \text{erf}^2\left(\frac{x_{\text{in}_i}}{\sqrt{2}}\right) \exp\left(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}\right) dx_{\text{in}_i}$$

$$= \frac{1}{\sqrt{\pi}}(2\sqrt{2}\sigma_{x_{\text{in}}}^3 \tan^{-1}(\frac{\frac{1}{2}}{\sqrt{\frac{1}{4\sigma_{x_{\text{in}}}^4}+\frac{1}{2\sigma_{x_{\text{in}}}^2}}}) + \frac{\frac{1}{2}(\frac{1}{\sigma_{x_{\text{in}}}^2}+1)}{\frac{1}{2\sigma_{x_{\text{in}}}^2}\sqrt{\frac{1}{2\sigma_{x_{\text{in}}}^2}+1}(\frac{1}{4\sigma_{x_{\text{in}}}^4}+\frac{1}{2\sigma_{x_{\text{in}}}^2}+\frac{1}{4})})$$

$$= \frac{1}{\sqrt{\pi}}(2\sqrt{2}\sigma_{x_{\text{in}}}^3 \tan^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-\sigma_{x_{\text{in}}}^4}}) + \frac{4\sqrt{2}\sigma_{x_{\text{in}}}^5(\sigma_{x_{\text{in}}}^2+1)}{\sqrt{2\sigma_{x_{\text{in}}}^2+1}(\sigma_{x_{\text{in}}}^4+2\sigma_{x_{\text{in}}}^2+1)})$$

$$= \frac{1}{\sqrt{\pi}}(2\sqrt{2}\sigma_{x_{\text{in}}}^3 \sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{4\sqrt{2}\sigma_{x_{\text{in}}}^5}{\sqrt{2\sigma_{x_{\text{in}}}^2+1}(\sigma_{x_{\text{in}}}^2+1)})$$

$$= \frac{2\sqrt{2}\sigma_{x_{\text{in}}}^3}{\sqrt{\pi}}(\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{2\sigma_{x_{\text{in}}}^2}{\sqrt{2\sigma_{x_{\text{in}}}^2+1}(\sigma_{x_{\text{in}}}^2+1)}))$$

$$\mathbb{E}[x_{\text{out}_i}^2] = \frac{\sigma_{x_{\text{in}}}^2}{4} + \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty} x_{\text{in}_i}^2 \text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})dx_{\text{in}_i}$$

$$= \frac{\sigma_{x_{\text{in}}}^2}{4} + \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\frac{2\sqrt{2}\sigma_{x_{\text{in}}}^3}{\sqrt{\pi}}(\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{2\sigma_{x_{\text{in}}}^2}{\sqrt{2\sigma_{x_{\text{in}}}^2+1}(\sigma_{x_{\text{in}}}^2+1)}))$$

$$\mathbb{E}[x_{\text{out}_i}^2] = \frac{\sigma_{x_{\text{in}}}^2}{4} + \frac{\sigma_{x_{\text{in}}}^2}{2\pi}(\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{2\sigma_{x_{\text{in}}}^2}{\sqrt{2\sigma_{x_{\text{in}}}^2+1}(\sigma_{x_{\text{in}}}^2+1)}))$$

$$\text{Var}(x_{\text{out}_i}) = \mathbb{E}[x_{\text{out}_i}^2] - (\mathbb{E}[x_{\text{out}_i}])^2$$

$$\boxed{\sigma_{x_{out}}^2 = \frac{\sigma_{x_{in}}^2}{2\pi}(\frac{\pi}{2} - \frac{\sigma_{x_{in}}^2}{1+\sigma_{x_{in}}^2} + \sin^{-1}(\frac{\sigma_{x_{in}}^2}{1+\sigma_{x_{in}}^2}) + \frac{2\sigma_{x_{in}}^2}{(1+\sigma_{x_{in}}^2)\sqrt{1+2\sigma_{x_{in}}^2}})}$$

Now if we have two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ such that for all values of i, we have $\text{Corr}(x_{\text{in}_i}, y_{\text{in}_i}) = r_{x_{\text{in}}}^l$, then we can calculate the covariance $\text{Cov}(x_{\text{out}_j}, y_{\text{out}_j})$ for any j as,

$$\text{Cov}(x_{\text{out}_j}, y_{\text{out}_j}) = \mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}]$$

$$\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}]$$

$$= \iint_{-\infty}^{\infty} \frac{x_{\text{out}_j} y_{\text{out}_j}}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j} - y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j}dy_{\text{in}_j} = I$$

$$= \iint_{-\infty}^{\infty} \frac{x_{\text{in}_j}(1+\text{erf}(\frac{x_{\text{in}_j}}{\sqrt{2}}))y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j} - y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j}dy_{\text{in}_j}$$

$$= \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})I_X dy_{\text{in}_j}$$

Where $I_X = \int_{-\infty}^{\infty} x_{\text{in}_j}(1+\text{erf}(\frac{x_{\text{in}_j}}{\sqrt{2}}))\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j}$

$$I_X = \int_{-\infty}^{\infty} x_{\text{in}_j}(1+\text{erf}(\frac{x_{\text{in}_j}}{\sqrt{2}}))\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j}$$

$$= \int_{-\infty}^{\infty} x_{\text{in}_j}\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j} +$$

$$\int_{-\infty}^{\infty} x_{\text{in}_j}\text{erf}(\frac{x_{\text{in}_j}}{\sqrt{2}})\exp(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dx_{\text{in}_j}$$

Let, $I_{X,1} = \int_{-\infty}^{\infty} x_{\text{in}_j} \exp\left(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j}$

$I_{X,2} = \int_{-\infty}^{\infty} x_{\text{in}_j} \text{erf}\left(\frac{x_{\text{in}_j}}{\sqrt{2}}\right) \exp\left(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j}$

$$
\begin{aligned}
I_{X,1} &= \int_{-\infty}^{\infty} x_{\text{in}_j} \exp\left(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j} \\
&= \int_{-\infty}^{\infty} x_{\text{in}_j} \exp\left(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \exp\left(\frac{-(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j} \\
&= \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \int_{-\infty}^{\infty} x_{\text{in}_j} \exp\left(\frac{-(x_{\text{in}_j} - r_{x_{\text{in}}}^l y_{\text{in}_j})^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j} \\
&= \int_{-\infty}^{\infty} \frac{x_{\text{in}_j}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-(x_{\text{in}_j} - r_{x_{\text{in}}}^l y_{\text{in}_j})^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j} \\
&= r_{x_{\text{in}}}^l y_{\text{in}_j} \sqrt{2\pi}\sigma_{x_{\text{in}}} \sqrt{(1-(r_{x_{\text{in}}}^l)^2)} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \\
I_{X,2} &= \int_{-\infty}^{\infty} x_{\text{in}_j} \text{erf}\left(\frac{x_{\text{in}_j}}{\sqrt{2}}\right) \exp\left(\frac{-x_{\text{in}_j}^2 + 2r_{x_{\text{in}}}^l x_{\text{in}_j} y_{\text{in}_j}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dx_{\text{in}_j}
\end{aligned}
$$

From 2.7.2.4 of Korotkov & Korotkov (2020),

$$
\begin{aligned}
&\int_{-\infty}^{\infty} z \text{erf}(a_1 z) \exp(-az^2 + bz) dz = \\
&= \frac{\sqrt{\pi} b}{2a\sqrt{a}} \exp\left(\frac{b^2}{4a}\right) \text{erf}\left(\frac{a_1 b}{2\sqrt{a^2 + a a_1^2}}\right) + \frac{a_1}{a\sqrt{a + a_1^2}} \exp\left(\frac{b^2}{4a + 4a_1^2}\right)
\end{aligned}
$$

Substituting $a_1 = \frac{1}{\sqrt{2}}, a = \frac{1}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}, b = \frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}$, we get

$$
\begin{aligned}
I_{X,2} &= \frac{\sqrt{\pi} \frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}}{2\frac{1}{2\sqrt{2}\sigma_{x_{\text{in}}}^3(1-(r_{x_{\text{in}}}^l)^2)^{\frac{3}{2}}}} \exp\left(\frac{\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{\sigma_{x_{\text{in}}}^4(1-(r_{x_{\text{in}}}^l)^2)^2}}{4\frac{1}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}}\right) \text{erf}\left(\frac{\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2}\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}}{2\sqrt{\frac{1}{4\sigma_{x_{\text{in}}}^4(1-(r_{x_{\text{in}}}^l)^2)^2} + \frac{1}{4\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}}}\right) \\
&+ \frac{\frac{1}{\sqrt{2}}}{\frac{1}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\sqrt{\frac{1}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)} + \frac{1}{2}}} \exp\left(\frac{\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{\sigma_{x_{\text{in}}}^4(1-(r_{x_{\text{in}}}^l)^2)^2}}{4\frac{1}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)} + \frac{4}{2}}\right) \\
&= r_{x_{\text{in}}}^l y_{\text{in}_j} \sqrt{2\pi}\sigma_{x_{\text{in}}} \sqrt{(1-(r_{x_{\text{in}}}^l)^2)} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) \\
&+ \frac{2\sigma_{x_{\text{in}}}^3(1-(r_{x_{\text{in}}}^l)^2)^{\frac{3}{2}}}{\sqrt{\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1}} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)
\end{aligned}
$$

Let us define $I_{X,2,1}$ and $I_{X,2,2}$ as:

$$
\begin{aligned}
I_{X,2,1} &= r_{x_{\text{in}}}^l y_{\text{in}_j} \sqrt{2\pi}\sigma_{x_{\text{in}}} \sqrt{(1-(r_{x_{\text{in}}}^l)^2)} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) \\
I_{X,2,2} &= \frac{2\sigma_{x_{\text{in}}}^3(1-(r_{x_{\text{in}}}^l)^2)^{\frac{3}{2}}}{\sqrt{\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1}} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)
\end{aligned}
$$

$$I = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) I_X dy_{\text{in}_j}$$

$$= \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) (I_{X,1} + I_{X,2,1} + I_{X,2,2}) dy_{\text{in}_j}$$

$$I_1 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) I_{X,1} dy_{\text{in}_j}$$

$$I_2 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) I_{X,2,1} dy_{\text{in}_j}$$

$$I_3 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) I_{X,2,2} dy_{\text{in}_j}$$

We have $I = I_1 + I_2 + I_3$

$$I_1 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) r_{x_{\text{in}}}^l y_{\text{in}_j}$$

$$\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l}{4} \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}^2(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{\sqrt{2\pi}\sigma_{x_{\text{in}}}^2} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l}{4} \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}^2}{\sqrt{2\pi}\sigma_{x_{\text{in}}}^2} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) dy_{\text{in}_j} + \frac{r_{x_{\text{in}}}^l}{4} \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}^2 \text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}^2} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l \sigma_{x_{\text{in}}}^2}{4}$$ (Definition of variance, and integral of odd function)

$$I_2 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) r_{x_{\text{in}}}^l y_{\text{in}_j}$$

$$\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)} \exp\left(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} y_{\text{in}_j}^2(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}})) \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} y_{\text{in}_j}^2 \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) dy_{\text{in}_j}$$

$$+ \frac{r_{x_{\text{in}}}^l}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} y_{\text{in}_j}^2 \text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}) \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) dy_{\text{in}_j}$$

$$= \frac{r_{x_{\text{in}}}^l}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} y_{\text{in}_j}^2 \text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}) \exp\left(\frac{-y_{\text{in}_j}^2}{2\sigma_{x_{\text{in}}}^2}\right) \text{erf}\left(\frac{r_{x_{\text{in}}}^l y_{\text{in}_j}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}\right) dy_{\text{in}_j}$$ (Integral of Odd function)

From 2.7.3.3 of Korotkov & Korotkov (2020),

$$\int_{-\infty}^{\infty} z^2 \exp\left(-az^2\right)\mathrm{erf}(a_1 z)\mathrm{erf}(a_2 z) =$$
$$\frac{1}{\sqrt{\pi}}\left(\frac{1}{a\sqrt{a}}\tan^{-1}\left(\frac{a_1 a_2}{\sqrt{a^2+aa_1^2+aa_2^2}}\right)+\frac{a_1 a_2(2a+a_1^2+a_2^2)}{a\sqrt{a+a_1^2+a_2^2(a^2+aa_1^2+aa_2^2+a_1^2a_2^2)}}\right)$$

Substituting $a=\frac{1}{2\sigma_{x_{\text{in}}}^2}, a_1=\frac{1}{\sqrt{2}}, a_2=\frac{r^l_{x_{\text{in}}}}{\sqrt{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}$

$$a_1 a_2 = \frac{r^l_{x_{\text{in}}}}{2\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}$$

$$\begin{aligned}
a^2+aa_1^2+aa_2^2 &= \frac{1}{4\sigma^4_{x_{\text{in}}}}+\frac{1}{4\sigma^2_{x_{\text{in}}}}+\frac{(r^l_{x_{\text{in}}})^2}{4\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1+\sigma^4_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+\sigma^2_{x_{\text{in}}}+(r^l_{x_{\text{in}}})^2\sigma^2_{x_{\text{in}}}}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{\sigma^4_{x_{\text{in}}}+2\sigma^2_{x_{\text{in}}}+1-(r^l_{x_{\text{in}}})^2\sigma^4_{x_{\text{in}}}}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} = \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
a+a_1^2+a_2^2 &= \frac{a^2+aa_1^2+aa_2^2}{a} = \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} * 2\sigma^2_{x_{\text{in}}} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
a^2+aa_1^2+aa_2^2+a_1^2a_2^2 &= \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}+\frac{(r^l_{x_{\text{in}}})^2}{4(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2+(r^l_{x_{\text{in}}})^2\sigma^4_{x_{\text{in}}}}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} = \frac{(\sigma^2_{x_{\text{in}}}+1)^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
2a+a_1^2+a_2^2 &= \frac{1}{2\sigma^2_{x_{\text{in}}}}+\frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2+\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)^2+\sigma^2_{x_{\text{in}}}+1-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2-\sigma^2_{x_{\text{in}}}(r^l_{x_{\text{in}}})^2}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)(\sigma^2_{x_{\text{in}}}+2)-(r^l_{x_{\text{in}}})^2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}+1)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)} \\
&= \frac{(\sigma^2_{x_{\text{in}}}+1)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+2)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}
\end{aligned}$$

$$I_2 = \frac{r^l_{x_{\text{in}}}}{4\sqrt{2}\pi\sigma_{x_{\text{in}}}}\left(2\sqrt{2}\sigma^3_{x_{\text{in}}}\tan^{-1}\left(\frac{\frac{r^l_{x_{\text{in}}}}{2\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}}{\sqrt{\frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}}\right)\right)$$
$$+\frac{r^l_{x_{\text{in}}}}{4\sqrt{2}\pi\sigma_{x_{\text{in}}}}\left(\frac{\frac{r^l_{x_{\text{in}}}}{2\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\frac{(\sigma^2_{x_{\text{in}}}+1)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+2)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}{\frac{1}{2\sigma^2_{x_{\text{in}}}}\sqrt{\frac{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\frac{(\sigma^2_{x_{\text{in}}}+1)^2}{4\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)$$

$$= \frac{r^l_{x_{\text{in}}}}{4\sqrt{2}\pi\sigma_{x_{\text{in}}}}(2\sqrt{2}\sigma^3_{x_{\text{in}}}\tan^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}})) + \frac{r^l_{x_{\text{in}}}}{4\sqrt{2}\pi\sigma_{x_{\text{in}}}}(\frac{2\sqrt{2}r^l_{x_{\text{in}}}\sigma^5_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+2)}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}})$$

$$I_2 = \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2\pi}(\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+2)}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}})$$

$$I_3 = \int_{-\infty}^{\infty} \frac{y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{8\pi\sigma^2_{x_{\text{in}}}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}\exp(\frac{-y^2_{\text{in}_j}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}) \frac{2\sigma^3_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)^{\frac{3}{2}}}{\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\exp(\frac{(r^l_{x_{\text{in}}})^2y^2_{\text{in}_j}}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)})dy_{\text{in}_j}$$

$$= \int_{-\infty}^{\infty} \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1-(r^l_{x_{\text{in}}})^2)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)})dy_{\text{in}_j}$$

$$= \int_{-\infty}^{\infty} \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}+1)(1-(r^l_{x_{\text{in}}})^2)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)})dy_{\text{in}_j}$$

$$= \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\int_{-\infty}^{\infty} y_{\text{in}_j}(1+\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}}))\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}+1)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}})dy_{\text{in}_j}$$

$$= \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\int_{-\infty}^{\infty} y_{\text{in}_j}\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}+1)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}})dy_{\text{in}_j}$$

$$+ \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\int_{-\infty}^{\infty} y_{\text{in}_j}\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}})\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}+1)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}})dy_{\text{in}_j}$$

$$= \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\int_{-\infty}^{\infty} y_{\text{in}_j}\text{erf}(\frac{y_{\text{in}_j}}{\sqrt{2}})\exp(\frac{-y^2_{\text{in}_j}(\sigma^2_{x_{\text{in}}}+1)}{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)\sigma^2_{x_{\text{in}}}})dy_{\text{in}_j} \quad \text{(Integral of Odd function)}$$

From 2.6.1.4 of Korotkov & Korotkov (2020), $\int_{-\infty}^{\infty} z\text{erf}(az)\exp(-a_1z^2)dz = \frac{a}{a_1\sqrt{a^2+a_1}}$

Substituting, $a = \frac{1}{\sqrt{2}}, a_1 = \frac{(\sigma^2_{x_{\text{in}}}+1)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}$, we have

$$I_3 = \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}(\frac{\frac{1}{\sqrt{2}}}{\frac{(\sigma^2_{x_{\text{in}}}+1)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}\sqrt{\frac{1}{2}+\frac{(\sigma^2_{x_{\text{in}}}+1)}{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}})$$

$$= \frac{\sigma_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{4\pi\sqrt{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1}}\frac{2\sigma^3_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)^{\frac{3}{2}}}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{\sigma^4_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+\sigma^2_{x_{\text{in}}}+\sigma^2_{x_{\text{in}}}+1}}$$

$$I_3 = \frac{\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)(1-(r^l_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}}$$

Finally we have,

$$\begin{aligned}I &= I_1 + I_2 + I_3 \\ &= \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{4} + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2\pi}(\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+2)}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}}) \\ &\quad + \frac{\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)(1-(r^l_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}}\end{aligned}$$

$$I = \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{4} + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2\pi}\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) + \frac{\sigma^4_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1+(r^l_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}}$$

$$I = \frac{\sigma^2_{x_{\text{in}}}}{4}\left[r^l_{x_{\text{in}}} + \frac{2r^l_{x_{\text{in}}}}{\pi}\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) + \frac{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1+(r^l_{x_{\text{in}}})^2)}{\pi(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}}\right]$$

We have,

$$\begin{aligned}\text{Cov}(x_{\text{out}_j}, y_{\text{out}_j}) &= I - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}] \\ \text{Cov}(x_{\text{out}_j}, y_{\text{out}_j}) &= I - \frac{\sigma^4_{x_{\text{in}}}}{2\pi(\sigma^2_{x_{\text{in}}}+1)}\end{aligned}$$

$$\boxed{\begin{aligned}\text{Cov}(x_{\text{out}_j}, y_{\text{out}_j}) = \frac{\sigma^2_{x_{\text{in}}}}{4\pi}(\pi r^l_{x_{\text{in}}} + 2r^l_{x_{\text{in}}}\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) \\ + \frac{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1+(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}} - \frac{2\sigma^2_{x_{\text{in}}}}{(\sigma^2_{x_{\text{in}}}+1)})\end{aligned}}$$

The backward pass through GeLU is defined as,

$$\begin{aligned}g_{\text{in}_i} &= (\Phi(x_{\text{in}_i}) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))g_{\text{out}_i} \\ &= (\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))g_{\text{out}_i}\end{aligned}$$

So the mean of gradient is obtained as following,

$$\begin{aligned}\mathbb{E}[g_{\text{in}_i}] &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))g_{\text{out}_i}] \\ &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))]\mathbb{E}[g_{\text{out}_i}] = 0 \\ &\boxed{\mu_{g_{\text{in}}} = 0}\end{aligned}$$

Similarly for variance,

$$\begin{aligned}\mathbb{E}[g^2_{\text{in}_i}] &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))^2 g^2_{\text{out}_i}] \\ &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))^2]\mathbb{E}[g^2_{\text{out}_i}] \\ &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x^2_{\text{in}_i}}{2}))^2]\sigma^2_{g_{\text{out}}}\end{aligned}$$

$$I = \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x_{\text{in}_i}^2}{2}))^2]$$

$$= \int_{-\infty}^{\infty}(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-x_{\text{in}_i}^2}{2}))^2 \frac{\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i}$$

$$I = \int_{-\infty}^{\infty}(\frac{1}{4} + \frac{\text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})}{4} + \frac{x_{\text{in}_i}^2\exp(-x_{\text{in}_i}^2)}{2\pi} + \frac{\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{2} + \frac{x_{\text{in}_i}\exp(\frac{-x_{\text{in}_i}^2}{2})}{\sqrt{2\pi}} + \frac{x_{\text{in}_i}\exp(\frac{-x_{\text{in}_i}^2}{2})\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{\sqrt{2\pi}})\frac{\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i}$$

$$I_1 = \int_{-\infty}^{\infty}\frac{1}{4}\frac{\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i}$$

$$I_1 = \frac{1}{4}$$

$$I_2 = \int_{-\infty}^{\infty}\frac{\text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})}{4}\frac{\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i}$$

$$= \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty}\text{erf}^2(\frac{x_{\text{in}_i}}{\sqrt{2}})\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}) dx_{\text{in}_i}$$

From 2.7.1.3 of Korotkov & Korotkov (2020),

$$\int_{-\infty}^{\infty}\text{erf}(a_1 z)\text{erf}(a_2 z)\exp(-az^2)dz = \frac{2}{\sqrt{\pi a}}\tan^{-1}(\frac{a_1 a_2}{\sqrt{a^2 + aa_1^2 + aa_2^2}})$$

Substituting $a = \frac{1}{2\sigma_{x_{\text{in}}}^2}, a_1 = a_2 = \frac{1}{\sqrt{2}}$

$$I_2 = \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\frac{2}{\sqrt{\pi\frac{1}{2\sigma_{x_{\text{in}}}^2}}}\tan^{-1}(\frac{\frac{1}{2}}{\sqrt{\frac{1}{4\sigma_{x_{\text{in}}}^4} + \frac{1}{4\sigma_{x_{\text{in}}}^2} + \frac{1}{4\sigma_{x_{\text{in}}}^2}}})$$

$$= \frac{1}{2\pi}\tan^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sqrt{2\sigma_{x_{\text{in}}}^2 + 1}}) = \frac{1}{2\pi}\tan^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2 - \sigma_{x_{\text{in}}}^4}})$$

$$I_2 = \frac{1}{2\pi}\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2 + 1})$$

$$I_3 = \int_{-\infty}^{\infty}\frac{x_{\text{in}_i}^2\exp(-x_{\text{in}_i}^2)}{2\pi}\frac{\exp(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i}$$

$$= \frac{1}{2\pi\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty}\frac{x_{\text{in}_i}^2}{\sqrt{2\pi}}\exp(\frac{-x_{\text{in}_i}^2(2\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}) dx_{\text{in}_i}$$

$$= \frac{1}{2\pi\sigma_{x_{\text{in}}}}\frac{\sigma_{x_{\text{in}}}}{\sqrt{(2\sigma_{x_{\text{in}}}^2+1)}}\int_{-\infty}^{\infty}\frac{x_{\text{in}_i}^2}{\sqrt{2\pi}\frac{\sigma_{x_{\text{in}}}}{\sqrt{(2\sigma_{x_{\text{in}}}^2+1)}}}\exp(\frac{-x_{\text{in}_i}^2(2\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}) dx_{\text{in}_i}$$

$$= \frac{1}{2\pi\sigma_{x_{\text{in}}}}\frac{\sigma_{x_{\text{in}}}}{\sqrt{(2\sigma_{x_{\text{in}}}^2+1)}}\frac{\sigma_{x_{\text{in}}}^2}{(2\sigma_{x_{\text{in}}}^2+1)}$$ (Definition of variance)

$$I_3 = \frac{\sigma_{x_{\text{in}}}^2}{2\pi(2\sigma_{x_{\text{in}}}^2+1)^{\frac{3}{2}}}$$

$$I_4 = \int_{-\infty}^{\infty} \frac{\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{2} \frac{\exp{(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} = 0 \qquad \text{(Integral of odd function)}$$

$$I_5 = \int_{-\infty}^{\infty} \frac{x_{\text{in}_i} \exp{(\frac{-x_{\text{in}_i}^2}{2})}}{\sqrt{2\pi}} \frac{\exp{(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} = 0 \qquad \text{(Integral of odd function)}$$

$$\begin{aligned} I_6 &= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i} \exp{(\frac{-x_{\text{in}_i}^2}{2})}\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{\sqrt{2\pi}} \frac{\exp{(\frac{-x_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})}}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} dx_{\text{in}_i} \\ &= \frac{1}{2\pi\sigma_{x_{\text{in}}}} \int_{-\infty}^{\infty} x_{\text{in}_i} \text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}) \exp{(\frac{-x_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}}) dx_{\text{in}_i} \end{aligned}$$

From 2.6.1.4 of Korotkov & Korotkov (2020), $\int_{-\infty}^{\infty} z\text{erf}(az)\exp{(-a_1 z^2)}dz = \frac{a}{a_1\sqrt{a^2+a_1}}$

Substituting, $a = \frac{1}{\sqrt{2}}, a_1 = \frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}$, we have

$$\begin{aligned} I_6 &= \frac{1}{2\pi\sigma_{x_{\text{in}}}} \frac{\frac{1}{\sqrt{2}}}{\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}\sqrt{\frac{1}{2}+\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}}} \\ &= \frac{1}{2\pi\sigma_{x_{\text{in}}}} \frac{2\sigma_{x_{\text{in}}}^3}{(\sigma_{x_{\text{in}}}^2+1)\sqrt{2\sigma_{x_{\text{in}}}^2+1}} \\ I_6 &= \frac{\sigma_{x_{\text{in}}}^2}{\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{2\sigma_{x_{\text{in}}}^2+1}} \\ I &= I_1+I_2+I_3+I_4+I_5+I_6 \\ &= \frac{1}{4}+\frac{1}{2\pi}\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{\sigma_{x_{\text{in}}}^2}{2\pi(2\sigma_{x_{\text{in}}}^2+1)^{\frac{3}{2}}} + \frac{\sigma_{x_{\text{in}}}^2}{\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{2\sigma_{x_{\text{in}}}^2+1}} \\ &= \frac{1}{4}+\frac{1}{2\pi}\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{\sigma_{x_{\text{in}}}^2(4\sigma_{x_{\text{in}}}^2+2+\sigma_{x_{\text{in}}}^2+1)}{2\pi(\sigma_{x_{\text{in}}}^2+1)(2\sigma_{x_{\text{in}}}^2+1)^{\frac{3}{2}}} \\ I &= \frac{1}{4}+\frac{1}{2\pi}\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{\sigma_{x_{\text{in}}}^2(5\sigma_{x_{\text{in}}}^2+3)}{2\pi(\sigma_{x_{\text{in}}}^2+1)(2\sigma_{x_{\text{in}}}^2+1)^{\frac{3}{2}}} \end{aligned}$$

So the variance of gradient of input of GeLU comes out to be

$$\mathbb{E}[g_{\text{in}_i}^2] = I\sigma_{g_{\text{out}}}^2$$

$$\boxed{\sigma_{g_{\text{in}}}^2 = \left[\frac{1}{4}+\frac{1}{2\pi}\sin^{-1}(\frac{\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1}) + \frac{\sigma_{x_{\text{in}}}^2(5\sigma_{x_{\text{in}}}^2+3)}{2\pi(\sigma_{x_{\text{in}}}^2+1)(2\sigma_{x_{\text{in}}}^2+1)^{\frac{3}{2}}}\right]\sigma_{g_{\text{out}}}^2}$$

If for two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ for all i we have $\text{Corr}(g_{\text{out}_{x_i}}, g_{\text{out}_{y_i}}) = r_{g_{\text{out}}}^l$, and $g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}}$ be the gradient after passing through GeLU layer. Then we have,

$$\begin{aligned} &\mathbb{E}[g_{\text{in}_{x_i}} g_{\text{in}_{y_i}}] = \\ &= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})) + \frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp{(\frac{-x_{\text{in}_i}^2}{2})})g_{\text{out}_{x_i}}(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp{(\frac{-y_{\text{in}_i}^2}{2})})g_{\text{out}_{y_i}}] \end{aligned}$$

$$\mathbb{E}[g_{\text{in}_{x_i}} g_{\text{in}_{y_i}}] = \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))+$$

$$\frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-x_{\text{in}_i}^2}{2}\right))\left(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-y_{\text{in}_i}^2}{2}\right)\right)]\mathbb{E}[g_{\text{out}_{x_i}}g_{\text{out}_{y_i}}]$$

$$= \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))+$$

$$\frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-x_{\text{in}_i}^2}{2}\right))\left(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-y_{\text{in}_i}^2}{2}\right)\right)]r_{g_{\text{out}}}^l\sigma_{g_{\text{out}}}^2$$

$$I = \mathbb{E}[(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))+$$

$$\frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-x_{\text{in}_i}^2}{2}\right))\left(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-y_{\text{in}_i}^2}{2}\right)\right)]$$

$$= \int_{-\infty}^{\infty}(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))+$$

$$\frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-x_{\text{in}_i}^2}{2}\right))\left(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-y_{\text{in}_i}^2}{2}\right)\right)p_{x_{\text{in}_i},y_{\text{in}_i}}dx_{\text{in}_i}dy_{\text{in}_i}$$

Where $p_{x_{\text{in}_i},y_{\text{in}_i}} = \frac{1}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)$

$$I = \int_{-\infty}^{\infty}\frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-y_{\text{in}_i}^2}{2}\right))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp\left(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)I_X dy_{\text{in}_i}$$

Where,

$$I_X = \int_{-\infty}^{\infty}(\frac{1}{2}(1+\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}}))+\frac{x_{\text{in}_i}}{\sqrt{2\pi}}\exp\left(\frac{-x_{\text{in}_i}^2}{2}\right))\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

$$I_{X,1} = \int_{-\infty}^{\infty}\frac{1}{2}\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty}\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)\exp\left(\frac{-(r_{x_{\text{in}}}^l)^2y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)\exp\left(\frac{(r_{x_{\text{in}}}^l)^2y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

$$= \frac{1}{2}\exp\left(\frac{(r_{x_{\text{in}}}^l)^2y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)\int_{-\infty}^{\infty}\exp\left(\frac{-(x_{\text{in}_i}-r_{x_{\text{in}}}^l y_{\text{in}_i})^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

$$= \frac{1}{2}\exp\left(\frac{(r_{x_{\text{in}}}^l)^2y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}\int_{-\infty}^{\infty}\frac{\exp\left(\frac{-(x_{\text{in}_i}-r_{x_{\text{in}}}^l y_{\text{in}_i})^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)}{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}dx_{\text{in}_i}$$

$$I_{X,1} = \frac{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}{2}\exp\left(\frac{(r_{x_{\text{in}}}^l)^2y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)$$

$$I_{X,2} = \int_{-\infty}^{\infty}\frac{\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})}{2}\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty}\text{erf}(\frac{x_{\text{in}_i}}{\sqrt{2}})\exp\left(\frac{-x_{\text{in}_i}^2+2r_{x_{\text{in}}}^l x_{\text{in}_i}y_{\text{in}_i}}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\right)dx_{\text{in}_i}$$

From 2.7.1.6 of Korotkov & Korotkov (2020),

$$\int_{-\infty}^{\infty}\text{erf}(a_1z)\exp\left(-az^2+bz\right)dz = \sqrt{\frac{\pi}{a}}\exp\left(\frac{b^2}{4a}\right)\text{erf}\left(\frac{a_1 b}{2\sqrt{a^2+aa_1^2}}\right)$$

Substituting $a_1 = \frac{1}{\sqrt{2}}, a = \frac{1}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}, b = \frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}$

$$I_{X,2} = \frac{1}{2}\sqrt{\frac{\pi}{\frac{1}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}}} \exp\left(\frac{\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{\sigma^4_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)^2}}{4\frac{1}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}}\right)\text{erf}\left(\frac{\frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{\sqrt{2}\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}}{2\sqrt{\frac{1}{4\sigma^4_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)^2} + \frac{1}{4\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}}}\right)$$

$$I_{X,2} = \frac{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r^l_{x_{\text{in}}})^2)}}{2} \exp\left(\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right)\text{erf}\left(\frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{\sqrt{2(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)$$

$$\begin{aligned}
I_{X,3} &= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}} \exp\left(\frac{-x^2_{\text{in}_i}}{2}\right)\exp\left(\frac{-x^2_{\text{in}_i} + 2r^l_{x_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right)dx_{\text{in}_i} \\
&= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}} \exp\left(\frac{-x^2_{\text{in}_i}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1) + 2r^l_{x_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}\right)dx_{\text{in}_i} \\
&= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}} \exp\left(\frac{-x^2_{\text{in}_i} + \frac{2r^l_{x_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)dx_{\text{in}_i}
\end{aligned}$$

$$= \int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}} \exp\left(\frac{-x^2_{\text{in}_i} + \frac{2r^l_{x_{\text{in}}} x_{\text{in}_i} y_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)\exp\left(\frac{\frac{-(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)^2}}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)*$$
$$\exp\left(\frac{\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)^2}}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)dx_{\text{in}_i}$$

$$= \exp\left(\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}\right)*$$
$$\int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}} \exp\left(\frac{-(x_{\text{in}_i} - \frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)})^2}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)dx_{\text{in}_i}$$

$$= \exp\left(\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}\right)\frac{\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}{\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}$$
$$\int_{-\infty}^{\infty} \frac{x_{\text{in}_i}}{\sqrt{2\pi}\frac{\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}{\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}} \exp\left(\frac{-(x_{\text{in}_i} - \frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)})^2}{\frac{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\right)dx_{\text{in}_i}$$

$$= \exp\left(\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}\right).$$
$$\frac{\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}{\sqrt{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}}\frac{r^l_{x_{\text{in}}} y_{\text{in}_i}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}$$

$$I_{X,3} = \frac{r^l_{x_{\text{in}}} y_{\text{in}_i}\sigma_{x_{\text{in}}}\sqrt{1-(r^l_{x_{\text{in}}})^2}}{(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)^{\frac{3}{2}}} \exp\left(\frac{(r^l_{x_{\text{in}}})^2 y^2_{\text{in}_i}}{2\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1)}\right)$$

$$I = \int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})(I_{X,1}+I_{X,2}+I_{X,3})dy_{\text{in}_i}$$

$$I_1 = \int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})I_{X,1}dy_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})$$
$$\frac{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}{2} \exp(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})dy_{\text{in}_i}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})dy_{\text{in}_i}$$

$$I_{1,1} = \frac{1}{4}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})dy_{\text{in}_i} = \frac{1}{4}$$

$$I_{1,2} = \frac{1}{4}\int_{-\infty}^{\infty} \frac{\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})dy_{\text{in}_i} = 0 \qquad \text{(Integral of odd function)}$$

$$I_{1,3} = \frac{1}{2}\int_{-\infty}^{\infty} \frac{y_{\text{in}_i}\exp(\frac{-y_{\text{in}_i}^2}{2})}{2\pi\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})dy_{\text{in}_i} = 0 \qquad \text{(Integral of odd function)}$$

$$I_2 = \int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})I_{X,2}dy_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})$$
$$\frac{\sqrt{2\pi}\sigma_{x_{\text{in}}}\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}{2} \exp(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} \frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})) + \frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2}).$$
$$\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i}$$

$$I_{2,1} = \frac{1}{4}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{x_{\text{in}}}} \exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i} = 0 \qquad \text{(Integral of odd function)}$$

$$I_{2,2} = \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty} \text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})\exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i}$$

From 2.7.1.3 of Korotkov & Korotkov (2020),

$\int_{-\infty}^{\infty} \text{erf}(a_1 z)\text{erf}(a_2 z)\exp(-az^2)dz = \frac{2}{\sqrt{\pi a}}\tan^{-1}(\frac{a_1 a_2}{\sqrt{a^2+aa_1^2+aa_2^2}})$

Substituting $a = \frac{1}{2\sigma_{x_{\text{in}}}^2}, a_1 = \frac{1}{\sqrt{2}}, a_2 = \frac{r_{x_{\text{in}}}^l}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}$

$$I_{2,2} = \frac{1}{4\sqrt{2\pi}\sigma_{x_{\text{in}}}}\frac{2}{\sqrt{\pi\frac{1}{2\sigma_{x_{\text{in}}}^2}}}\tan^{-1}(\frac{\frac{r_{x_{\text{in}}}^l}{2\sqrt{(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}}{\sqrt{\frac{1}{4\sigma_{x_{\text{in}}}^4}+\frac{1}{4\sigma_{x_{\text{in}}}^2}+\frac{(r_{x_{\text{in}}}^l)^2}{4\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}})$$

$$I_{2,2} = \frac{1}{2\pi}\tan^{-1}(\frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{\sqrt{\sigma_{x_{\text{in}}}^4+2\sigma_{x_{\text{in}}}^2+1-(r_{x_{\text{in}}}^l)^2\sigma_{x_{\text{in}}}^4}}) = \frac{1}{2\pi}\tan^{-1}(\frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2}})$$

$$I_{2,2} = \frac{1}{2\pi}\sin^{-1}(\frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{\sigma_{x_{\text{in}}}^2+1})$$

$$I_{2,3} = \frac{1}{4\pi\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty} y_{\text{in}_i}\exp(\frac{-y_{\text{in}_i}^2}{2})\exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2})\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i}$$

$$= \frac{1}{4\pi\sigma_{x_{\text{in}}}}\int_{-\infty}^{\infty} y_{\text{in}_i}\exp(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2})\text{erf}(\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}})dy_{\text{in}_i}$$

From 2.6.1.4 of Korotkov & Korotkov (2020), $\int_{-\infty}^{\infty} z\text{erf}(az)\exp(-a_1 z^2)dz = \frac{a}{a_1\sqrt{a^2+a_1}}$

Substituting, $a = \frac{r_{x_{\text{in}}}^l}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}, a_1 = \frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}$, we have

$$I_{2,3} = \frac{1}{4\pi\sigma_{x_{\text{in}}}}\frac{\frac{r_{x_{\text{in}}}^l}{\sqrt{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}}{\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}\sqrt{\frac{(r_{x_{\text{in}}}^l)^2}{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}+\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2}}}$$

$$= \frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{2\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{\sigma_{x_{\text{in}}}^4+2\sigma_{x_{\text{in}}}^2+1-(r_{x_{\text{in}}}^l)^2\sigma_{x_{\text{in}}}^4}}$$

$$I_{2,3} = \frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{2\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2}}$$

$$I_3 = \int_{-\infty}^{\infty}\frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})I_{X,3}dy_{\text{in}_i}$$

$$= \int_{-\infty}^{\infty}\frac{(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))}{2\pi\sigma_{x_{\text{in}}}^2\sqrt{(1-(r_{x_{\text{in}}}^l)^2)}}\exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})$$

$$\frac{r_{x_{\text{in}}}^l y_{\text{in}_i}\sigma_{x_{\text{in}}}\sqrt{1-(r_{x_{\text{in}}}^l)^2}}{(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\exp(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)})dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty} y_{\text{in}_i}(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))$$

$$\exp(\frac{-y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)})\exp(\frac{(r_{x_{\text{in}}}^l)^2 y_{\text{in}_i}^2}{2\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)})dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty} y_{\text{in}_i}(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp(\frac{-y_{\text{in}_i}^2}{2}))$$

$$\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1-(r_{x_{\text{in}}}^l)^2)}{2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)}\big)dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty} y_{\text{in}_i}\big(\frac{1}{2}(1+\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}}))+\frac{y_{\text{in}_i}}{\sqrt{2\pi}}\exp\big(\frac{-y_{\text{in}_i}^2}{2}\big)\big)$$

$$\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

$$I_{3,1} = \frac{r_{x_{\text{in}}}^l}{4\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty} y_{\text{in}_i}\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i} = 0 \quad \text{(Integral of odd function)}$$

$$I_{3,2} = \frac{r_{x_{\text{in}}}^l}{4\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty} y_{\text{in}_i}\text{erf}(\frac{y_{\text{in}_i}}{\sqrt{2}})\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

From 2.6.1.4 of Korotkov & Korotkov (2020), $\int_{-\infty}^{\infty} z\text{erf}(az)\exp(-a_1z^2)dz = \frac{a}{a_1\sqrt{a^2+a_1}}$

Substituting, $a = \frac{1}{\sqrt{2}}, a_1 = \frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}$, we have

$$I_{3,2} = \frac{r_{x_{\text{in}}}^l}{4\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\frac{\frac{1}{\sqrt{2}}}{\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\sqrt{\frac{1}{2}+\frac{(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}}$$

$$= \frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{2\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{\sigma_{x_{\text{in}}}^4+2\sigma_{x_{\text{in}}}^2+1-(r_{x_{\text{in}}}^l)^2\sigma_{x_{\text{in}}}^4}}$$

$$I_{3,2} = \frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{2\pi(\sigma_{x_{\text{in}}}^2+1)\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2}}$$

$$I_{3,3} = \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\cdot$$

$$\int_{-\infty}^{\infty}\frac{y_{\text{in}_i}^2}{\sqrt{2\pi}}\exp\big(\frac{-y_{\text{in}_i}^2}{2}\big)\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^2+1)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\cdot$$

$$\int_{-\infty}^{\infty}\frac{y_{\text{in}_i}^2}{\sqrt{2\pi}}\exp\big(\frac{-y_{\text{in}_i}^2(\sigma_{x_{\text{in}}}^4+2\sigma_{x_{\text{in}}}^2+1-(r_{x_{\text{in}}}^l)^2\sigma_{x_{\text{in}}}^4)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\int_{-\infty}^{\infty}\frac{y_{\text{in}_i}^2}{\sqrt{2\pi}}\exp\big(\frac{-y_{\text{in}_i}^2((\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\frac{\sigma_{x_{\text{in}}}\sqrt{(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}{\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2}}$$

$$\int_{-\infty}^{\infty}\frac{y_{\text{in}_i}^2}{\sqrt{2\pi}\frac{\sigma_{x_{\text{in}}}\sqrt{(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}}{\sqrt{(\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2}}}\exp\big(\frac{-y_{\text{in}_i}^2((\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2)}{2\sigma_{x_{\text{in}}}^2(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)}\big)dy_{\text{in}_i}$$

$$= \frac{r_{x_{\text{in}}}^l}{2\pi\sigma_{x_{\text{in}}}(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}\frac{\sigma_{x_{\text{in}}}^3(\sigma_{x_{\text{in}}}^2(1-(r_{x_{\text{in}}}^l)^2)+1)^{\frac{3}{2}}}{((\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2)^{\frac{3}{2}}}$$

$$I_{3,3} = \frac{r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2}{2\pi((\sigma_{x_{\text{in}}}^2+1)^2-(r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2)^2)^{\frac{3}{2}}}$$

$$
\begin{aligned}
I &= I_1 + I_2 + I_3 \\
&= I_{1,1} + I_{1,2} + I_{1,3} + I_{2,1} + I_{2,2} + I_{2,3} + I_{3,1} + I_{3,2} + I_{3,3}
\end{aligned}
$$

$$
\begin{aligned}
I = \frac{1}{4} + \frac{1}{2\pi}\sin^{-1}\left(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}\right) + \\
\frac{2r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2\pi(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}} + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{2\pi((\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)^{\frac{3}{2}}}
\end{aligned}
$$

$$
I = \frac{1}{4} + \frac{1}{2\pi}\sin^{-1}\left(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}\right) + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}((2\sigma^2_{x_{\text{in}}}+3)(\sigma^2_{x_{\text{in}}}+1)-2(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)((\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)^{\frac{3}{2}}}
$$

We defined $\text{Cov}(g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}})$, as

$$
\text{Cov}(g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}}) = I r^l_{g_{\text{out}}}\sigma^2_{g_{\text{out}}}
$$

$$
\boxed{
\begin{aligned}
&\text{Cov}(g_{\text{in}_{x_i}}, g_{\text{in}_{y_i}}) = \\
&\left[\frac{1}{4} + \frac{1}{2\pi}\sin^{-1}\left(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}\right) + \frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}((2\sigma^2_{x_{\text{in}}}+3)(\sigma^2_{x_{\text{in}}}+1)-2(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)((\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)^{\frac{3}{2}}}\right] r^l_{g_{\text{out}}}\sigma^2_{g_{\text{out}}}
\end{aligned}
}
$$

A.6 LayerNorm

The affine transformation for layernorm are typically initialized with 1 scale and 0 bias, so they do not change any of our derivations below and are ignored henceforth. For an input $\mathbf{x}_{\text{in}}$ the forward pass of LayerNorm is,

$$
\begin{aligned}
\mathbf{x}_{\text{out}} &= \text{LayerNorm}(\mathbf{x}_{\text{in}}) \\
\implies x_{\text{out}_i} &= \frac{x_{\text{in}_i} - \bar{x}_{\text{in}}}{\hat{\sigma}_{x_{\text{in}}}}
\end{aligned}
$$

Where

$$
\begin{aligned}
\bar{x}_{\text{in}} &= \frac{\sum_{i=1}^{d_{\text{in}}} x_{\text{in}_i}}{d_{\text{in}}} \\
\hat{\sigma}_{x_{\text{in}}} &= \sqrt{\frac{\sum_{i=1}^{d_{\text{in}}}(x_{\text{in}_i} - \bar{x}_{\text{in}})^2}{d_{\text{in}}}}
\end{aligned}
$$

To get expectation of output of LayerNorm,

$$
\begin{aligned}
\mathbb{E}[x_{\text{out}_i}] &= \mathbb{E}[\frac{x_{\text{in}_i} - \bar{x}_{\text{in}}}{\hat{\sigma}_{x_{\text{in}}}}] \\
\sum_{i=1}^{d_{\text{in}}}\mathbb{E}[x_{\text{out}_i}] &= \sum_{i=1}^{d_{\text{in}}}\mathbb{E}[\frac{x_{\text{in}_i} - \bar{x}_{\text{in}}}{\hat{\sigma}_{x_{\text{in}}}}] \\
&= \mathbb{E}[\sum_{i=1}^{d_{\text{in}}}\frac{x_{\text{in}_i} - \bar{x}_{\text{in}}}{\hat{\sigma}_{x_{\text{in}}}}] \\
&= \mathbb{E}[\frac{\sum_{i=1}^{d_{\text{in}}}(x_{\text{in}_i} - \bar{x}_{\text{in}})}{\hat{\sigma}_{x_{\text{in}}}}] \\
\sum_{i=1}^{d_{\text{in}}}\mathbb{E}[x_{\text{out}_i}] &= 0
\end{aligned}
$$

By symmetry for any i, j and $i \neq j$ we have $\mathbb{E}[x_{\text{out}_i}] = \mathbb{E}[x_{\text{out}_j}] = \mu_{x_{\text{out}}}$

$$
\implies d_{\text{in}}\mu_{x_{\text{out}}} = 0
$$

$$\boxed{\mu_{x_{\text{out}}} = 0}$$

Similarly we calculate variance of output by,

$$\begin{aligned}
\text{Var}(x_{\text{out}_i}) &= \mathbb{E}[x_{\text{out}_i}^2] - \mathbb{E}[x_{\text{out}_i}]^2 = \mathbb{E}[x_{\text{out}_i}^2] \\
\mathbb{E}[x_{\text{out}_i}^2] &= \mathbb{E}[\frac{(x_{\text{in}_i} - \bar{x}_{\text{in}})^2}{\hat{\sigma}_{x_{\text{in}}}^2}] \\
\sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{out}_i}^2] &= \sum_{i=1}^{d_{\text{in}}} \mathbb{E}[\frac{(x_{\text{in}_i} - \bar{x}_{\text{in}})^2}{\hat{\sigma}_{x_{\text{in}}}^2}] \\
&= \mathbb{E}[\sum_{i=1}^{d_{\text{in}}} \frac{(x_{\text{in}_i} - \bar{x}_{\text{in}})^2}{\hat{\sigma}_{x_{\text{in}}}^2}] \\
&= \mathbb{E}[\frac{\sum_{i=1}^{d_{\text{in}}}(x_{\text{in}_i} - \bar{x}_{\text{in}})^2}{\hat{\sigma}_{x_{\text{in}}}^2}] \\
\sum_{i=1}^{d_{\text{in}}} \mathbb{E}[x_{\text{out}_i}^2] &= d_{\text{in}}
\end{aligned}$$

By symmetry for any i, j and $i \neq j$ we have $\mathbb{E}[x_{\text{out}_i}^2] = \mathbb{E}[x_{\text{out}_j}^2] = \sigma_{x_{\text{out}}}^2$

$$\begin{aligned}
\implies d_{\text{in}} \sigma_{x_{\text{out}}}^2 &= d_{\text{in}} \\
\boxed{\sigma_{x_{\text{out}}}^2 = 1}
\end{aligned}$$

Now we have $\hat{\sigma}_{x_{\text{in}}} \xrightarrow{a.s} \sigma_{x_{\text{in}}}$ for large d_{in}. So for large values of d_{in} we can treat $\hat{\sigma}_{x_{\text{in}}}$ as a constant which has value $\sigma_{x_{\text{in}}}$. We use this approximation to get the following results. For two inputs $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ such that for all i, $\text{Corr}(x_{\text{in}_i}, y_{\text{in}_i}) = r_{x_{\text{in}}}^l$. For all j we have,

$$\begin{aligned}
\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) &= \frac{\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] - \mathbb{E}[x_{\text{out}_j}]\mathbb{E}[y_{\text{out}_j}]}{\sqrt{\text{Var}(x_{\text{out}_j})\text{Var}(y_{\text{out}_j})}} \\
&= \frac{\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] - \mu_{x_{\text{out}}}\mu_{x_{\text{out}}}}{\sqrt{\sigma_{x_{\text{out}}}^2 \sigma_{x_{\text{out}}}^2}} \\
&= \frac{\mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] - 0}{\sqrt{1}} \\
&= \mathbb{E}[x_{\text{out}_j} y_{\text{out}_j}] \\
&= \mathbb{E}[\frac{(x_{\text{in}_j} - \bar{x}_{\text{in}})(y_{\text{in}_j} - \bar{y}_{\text{in}})}{\hat{\sigma}_{x_{\text{in}}}\hat{\sigma}_{y_{\text{in}}}}] \\
&\approx \mathbb{E}[\frac{(x_{\text{in}_j} - \bar{x}_{\text{in}})(y_{\text{in}_j} - \bar{y}_{\text{in}})}{\sigma_{x_{\text{in}}}\sigma_{x_{\text{in}}}}] \\
&= \frac{\mathbb{E}[(x_{\text{in}_j} - \bar{x}_{\text{in}})(y_{\text{in}_j} - \bar{y}_{\text{in}})]}{\sigma_{x_{\text{in}}}^2} \\
&= \frac{\mathbb{E}[(x_{\text{in}_j} - \frac{\sum_{k=1}^{d_{\text{in}}} x_{\text{in}_k}}{d_{\text{in}}})(y_{\text{in}_j} - \frac{\sum_{l=1}^{d_{\text{in}}} y_{\text{in}_l}}{d_{\text{in}}})]}{\sigma_{x_{\text{in}}}^2} \\
&= \frac{\mathbb{E}[x_{\text{in}_j} y_{\text{in}_j} - y_{\text{in}_j}\frac{\sum_{k=1}^{d_{\text{in}}} x_{\text{in}_k}}{d_{\text{in}}} - x_{\text{in}_j}\frac{\sum_{l=1}^{d_{\text{in}}} y_{\text{in}_l}}{d_{\text{in}}} + \frac{\sum_{k=1}^{d_{\text{in}}} x_{\text{in}_k}}{d_{\text{in}}}\frac{\sum_{l=1}^{d_{\text{in}}} y_{\text{in}_l}}{d_{\text{in}}}]}{\sigma_{x_{\text{in}}}^2}
\end{aligned}$$

Elements belonging to different dimensions from $\mathbf{x}_{\text{in}}$ and $\mathbf{y}_{\text{in}}$ are independent of each other and hence for i, j and $i \neq j$ we

have $\mathbb{E}[x_{\text{in}_i} y_{\text{in}_j}] = \mu^2_{x_{\text{in}}}$.

$$
\begin{aligned}
&= \frac{\mathbb{E}[x_{\text{in}_j} y_{\text{in}_j}] - \mathbb{E}[y_{\text{in}_j} \frac{\sum_{k=1}^{d_{\text{in}}} x_{\text{in}_k}}{d_{\text{in}}}] - \mathbb{E}[x_{\text{in}_j} \frac{\sum_{l=1}^{d_{\text{in}}} y_{\text{in}_l}}{d_{\text{in}}}] + \mathbb{E}[\frac{\sum_{k=1}^{d_{\text{in}}} x_{\text{in}_k}}{d_{\text{in}}} \frac{\sum_{l=1}^{d_{\text{in}}} y_{\text{in}_l}}{d_{\text{in}}}]}{\sigma^2_{x_{\text{in}}}} \\
&= \frac{r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} + \mu^2_{x_{\text{in}}} - \frac{r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} + d_{\text{in}} \mu^2_{x_{\text{in}}}}{d_{\text{in}}} - \frac{r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} + d_{\text{in}} \mu^2_{x_{\text{in}}}}{d_{\text{in}}} + \frac{r^l_{x_{\text{in}}} d_{\text{in}} \sigma^2_{x_{\text{in}}} + d^2_{\text{in}} \mu^2_{x_{\text{in}}}}{d^2_{\text{in}}}}{\sigma^2_{x_{\text{in}}}} \\
&= \frac{r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} (1 - \frac{1}{d_{\text{in}}})}{\sigma^2_{x_{\text{in}}}}
\end{aligned}
$$

$$
\boxed{\text{Corr}(x_{\text{out}_j}, y_{\text{out}_j}) = r^l_{x_{\text{in}}} (1 - \frac{1}{d_{\text{in}}}) \approx r^l_{x_{\text{in}}} = r^l_{x_{\text{out}}}}
$$

From Xu et al. (2019) (Eq. 17), the backward pass through LayerNorm is,

$$
\begin{aligned}
\mathbf{g}_{\text{in}} &= \frac{\mathbf{g}_{\text{out}}}{\hat{\sigma}_{x_{\text{in}}}} (\mathbf{I}_{\mathbf{d}_{\text{in}}} - \frac{\mathbf{1}^{\mathbf{T}}_{\mathbf{d}_{\text{in}}} \mathbf{1}_{\mathbf{d}_{\text{in}}} + \mathbf{x}^{\mathbf{T}}_{\text{out}} \mathbf{x}_{\text{out}}}{d_{\text{in}}}) \\
&\approx \frac{\mathbf{g}_{\text{out}}}{\sigma_{x_{\text{in}}}} (\mathbf{I}_{\mathbf{d}_{\text{in}}} - \frac{\mathbf{1}^{\mathbf{T}}_{\mathbf{d}_{\text{in}}} \mathbf{1}_{\mathbf{d}_{\text{in}}} + \mathbf{x}^{\mathbf{T}}_{\text{out}} \mathbf{x}_{\text{out}}}{d_{\text{in}}})
\end{aligned}
$$

We have $\lim_{d_{\text{in}} \to \infty} \frac{\mathbf{1}^{\mathbf{T}}_{\mathbf{d}_{\text{in}}} \mathbf{1}_{\mathbf{d}_{\text{in}}} + \mathbf{x}^{\mathbf{T}}_{\text{out}} \mathbf{x}_{\text{out}}}{d_{\text{in}}} = \mathbf{O}_{\mathbf{d}_{\text{in}}, \mathbf{d}_{\text{in}}}$ where $\mathbf{O}_{\mathbf{d}_{\text{in}}, \mathbf{d}_{\text{in}}}$ is zero matrix with shape $d_{\text{in}} \times d_{\text{in}}$

$$
\begin{aligned}
\mathbf{g}_{\text{in}} &\approx \frac{\mathbf{g}_{\text{out}}}{\sigma_{x_{\text{in}}}} (\mathbf{I}_{\mathbf{d}_{\text{in}}}) \\
&= \frac{\mathbf{g}_{\text{out}}}{\sigma_{x_{\text{in}}}} \\
\implies g_{\text{in}_i} &= \frac{g_{\text{out}_i}}{\sigma_{x_{\text{in}}}}
\end{aligned}
$$

If $\mu_{g_{\text{out}}} = 0$,

$$
\boxed{
\begin{aligned}
\mu_{g_{\text{in}}} &= 0 \\
\sigma^2_{g_{\text{in}}} &= \frac{\sigma^2_{g_{\text{out}}}}{\sigma^2_{x_{\text{in}}}}
\end{aligned}
}
$$

A.7 Softmax

Assumption: Other than assuming normally distributed inputs, we also assume that L is large $L >> 1$ to derive softmax variance.

The forward pass of Softmax can be defined as

$$
\begin{aligned}
\mathbf{x}_{\text{out}} &= \text{Softmax}(\mathbf{x}_{\text{in}}) \\
x_{\text{out}_i} &= \frac{e^{x_{\text{in}_i}}}{\sum_{j=1}^{L} e^{x_{\text{in}_j}}}
\end{aligned}
$$

For calculating mean we can easily see that,

$$
\sum_{i=1}^{L} x_{\text{out}_i} = 1
$$

Taking expectation both sides, we get

$$\mathbb{E}[\sum_{i=1}^{L} x_{\text{out}_i}] = 1$$
$$\sum_{i=1}^{L} \mathbb{E}[x_{\text{out}_i}] = 1$$

By symmetry we can assume that for any $i, j, i \neq j$, we have $\mathbb{E}[x_{\text{out}_i}] = \mathbb{E}[x_{\text{out}_j}]$

$$L\mathbb{E}[x_{\text{out}_i}] = 1$$
$$\boxed{\mu_{x_{\text{out}}} = \frac{1}{L}}$$

Let us define $z = \sum_j e^{y_j}$ where $y_j = x_j - x_i$ is normally distributed $\mathcal{N}(0, \sigma_j)$. Hence, each e^{y_j} is log-normally distributed, and z is a sum of correlated log-normals. Following (Lo, 2013), this sum of log-normals can be approximated as another log-normal random variable, $Log\mathcal{N}(\mu_z, \sigma_z)$, where μ_z and σ_z are as follows -

$$S_+ = E[\sum_j y_j] = \sum_j e^{\frac{\sigma_j^2}{2}}$$
$$\sigma_z^2 = \frac{1}{S_+^2} \sum_{j,k} corr_{j,k} \sigma_j \sigma_k e^{\frac{1}{2}(\sigma_j^2 + \sigma_k^2)}$$
$$\mu_z = ln(S_+) - \frac{\sigma_z^2}{2}$$

Since the difference of two normals x_j and x_i is also normal, from the M.G.F. of normal distribution, we have $\sigma_j^2 = 2\sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}})$ if $j \neq i$, and $\sigma_j^2 = 0$ if $j = i$.

Also, $corr_{j,k} = 0$ if $j = i$ or $k = i$, else $corr_{j,k} = \frac{1}{2}$.

We can substitute these values in the above equations, to get

$$S_+ = (L-1)e^{\sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}})} + 1$$
$$\sigma_z^2 = \sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}}) \frac{L}{L-1}$$
$$\mu_z = ln(S_+) - \frac{\sigma_z^2}{2}$$

Since z is log-normal, $x_{\text{out}} = \frac{1}{z}$ is also log-normal with $Log\mathcal{N}(-\mu_z, \sigma_z)$. The variance of log-normal distribution can be obtained from standard formulae for log-normal distribution as $(e^{\sigma_z^2} - 1)e^{\sigma_z^2 - 2\mu_z}$.

Substituting the values of μ_z and σ_z from above, we get

$$\sigma_{x_{\text{out}}}^2 = \frac{(e^{\sigma_z^2} - 1)e^{2*\sigma_z^2}}{S_+^2}$$
$$= \frac{(e^{\sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}})\frac{L}{L-1}} - 1)e^{2\sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}})\frac{L}{L-1}}}{((L-1)e^{\sigma_{x_{\text{in}}}^2(1 - r_{x_{\text{in}}})} + 1)^2}$$

For large L, we can ignore the 1 in the denominator -

$$\sigma^2_{x_{\text{out}}} = \frac{(e^{\sigma^2_{x_{\text{in}}}(1-r_{x_{\text{in}}})\frac{L}{L-1}} - 1)}{(L-1)^2}$$

If $L >> 1$ and $\sigma^2_{x_{\text{in}}}$ is small, we get the more simplified formula as -

$$\boxed{\sigma^2_{x_{\text{out}}} \approx \frac{(e^{(1-r^d_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}} - 1)}{L^2}} \qquad \text{(Assuming } L >> 1\text{)}$$

Using the mean and variances, we can calculate the scale of softmax output as follows-

$$\begin{aligned} E[x^2_{\text{out}}] &= \sigma^2_{x_{\text{out}}} + \mu^2_{x_{\text{out}}} \\ &= \frac{(e^{(1-r^d_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}})}{L^2} \end{aligned}$$

The Jacobian of Softmax can be calculated as ((Kim et al., 2021)):

$$J_{i,j} = \begin{cases} x_{\text{out}_i}(1 - x_{\text{out}_i}) & \text{if } i = j \\ -x_{\text{out}_i} x_{\text{out}_j} & \text{else} \end{cases}$$

For large values of L this approximately becomes

$$\begin{aligned} \mathbf{J} &\approx \text{diag}(\mathbf{x}_{\text{out}}) \\ \mathbf{g}_{\text{in}} &= \mathbf{g}_{\text{out}}\mathbf{J} \\ g_{\text{in}_i} &\approx g_{\text{out}_i} x_{\text{out}_i} \\ \mathbb{E}[g_{\text{in}_i}] &\approx \mathbb{E}[g_{\text{out}_i} x_{\text{out}_i}] \\ &= \mathbb{E}[g_{\text{out}_i}]\mathbb{E}[x_{\text{out}_i}] = 0 = \mu_{g_{\text{in}}} \\ \mathbb{E}[g^2_{\text{in}_i}] &\approx \mathbb{E}[g^2_{\text{out}_i} x^2_{\text{out}_i}] \\ &= \mathbb{E}[g^2_{\text{out}_i}]\mathbb{E}[x^2_{\text{out}_i}] \\ \boxed{\sigma^2_{g_{\text{in}}} &= \sigma^2_{g_{\text{out}}} \frac{(e^{(1-r^d_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}})}{L^2}} \end{aligned}$$

A.8 Scaled Dot-Product Attention

Inapplicability of Direct Usage of Softmax Derivations for SHA: One may be tempted to assume attention scores to be independent of values. This then enables the use of our previous LogNormal-based softmax derivation, to easily derive the forward variances.

But the theoretically calculated moments strongly disagree with empirical simulations. This is because SHA is $\mathbf{X}_{\text{out}} = \text{Dropout}(\text{SoftMax}(\frac{\mathbf{X}_{\text{in}}\mathbf{W_Q}\mathbf{W_K}^\mathbf{T}\mathbf{X}^\mathbf{T}_{\text{in}}}{\sqrt{d_k}}))\mathbf{X}_{\text{in}}\mathbf{W_V}$, and the $\mathbf{W_K}^\mathbf{T}\mathbf{X}^\mathbf{T}_{\text{in}}$ term cannot be treated independently of the $\mathbf{X}_{\text{in}}\mathbf{W_V}$ term. A simple verification of this can be checked by simply simulating $(\mathbf{X}\mathbf{W}^\mathbf{T})X$, and verifying that the variances of the results do not match that of $L * \sigma^2((\mathbf{X}\mathbf{W}))$, but do if the second $\mathbf{X}$ is replaced by another random tensor.

This necessitates an alternate methodology to derive SHA, where the components are treated as a unified whole.

Assumption: We assume that L and d_{in} are very large when compared to scale of scores being passed to the Softmax. These approximations hold true for small values of σ_q and σ_k, and the resulting formulae are fairly accurate, as shown in the numerical verification section.

The forward pass of Scaled Dot-Product Attention is

$$\mathbf{X}_{\text{out}} = \text{Dropout}(\text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d_{i,k}}}))\mathbf{V}$$

Where,

$$\begin{aligned}
\mathbf{Q} &= \mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{Q}} \\
\mathbf{K} &= \mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{K}} \\
\mathbf{V} &= \mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{V}} \\
\mathbf{X}_{\text{out}} &= \text{Dropout}(\text{SoftMax}(\frac{\mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{Q}}\mathbf{W}_{\mathbf{K}}{}^{\mathbf{T}}\mathbf{X}_{\text{in}}^{\mathbf{T}}}{\sqrt{d_{i,k}}}))\mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{V}}
\end{aligned}$$

Let,

$$\begin{aligned}
\mathbf{O} &= \text{Dropout}(\text{SoftMax}(\frac{\mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{Q}}\mathbf{W}_{\mathbf{K}}{}^{\mathbf{T}}\mathbf{X}_{\text{in}}^{\mathbf{T}}}{\sqrt{d_{i,k}}}))\mathbf{X}_{\text{in}} \\
\mathbf{W} &= \frac{\mathbf{X}_{\text{in}}\mathbf{W}_{\mathbf{Q}}\mathbf{W}_{\mathbf{K}}{}^{\mathbf{T}}}{\sqrt{d_{i,k}}} \\
\mathbf{O} &= \text{Dropout}(\text{SoftMax}(\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}}))\mathbf{X}_{\text{in}}
\end{aligned}$$

Using results from Linear Layer we have $\sigma_w^2 = d_{\text{in}}\sigma_{x_{\text{in}}}^2\sigma_q^2\sigma_k^2 = d_{\text{in}}\sigma_{x_{\text{in}}}^2\sigma_{qk}^2$

$$\begin{aligned}
O_{i,j} &= \sum_{k=1}^{L} \text{Dropout}(\text{SoftMax}(\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}}))_{i,k}X_{\text{in}_{k,j}} \\
&= \sum_{k=1}^{L} \text{Dropout}(\frac{\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,k}\right)}{\sum_{m=1}^{L}\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,m}\right)})X_{\text{in}_{k,j}} \\
&= \sum_{k=1}^{L} \frac{\text{Dropout}(\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,k}\right))}{\sum_{m=1}^{L}\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,m}\right)}X_{\text{in}_{k,j}} \\
&= \frac{\sum_{k=1}^{L}\text{Dropout}(\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,k}\right))X_{\text{in}_{k,j}}}{\sum_{m=1}^{L}\exp\left((\mathbf{W}\mathbf{X}_{\text{in}}^{\mathbf{T}})_{i,m}\right)} \\
&= \frac{\sum_{k=1}^{L}\text{Dropout}(\exp(\sum_{l=1}^{d_{\text{in}}}W_{i,l}X_{\text{in}_{k,l}}))X_{\text{in}_{k,j}}}{\sum_{m=1}^{L}\exp(\sum_{n=1}^{d_{\text{in}}}W_{i,n}X_{\text{in}_{m,n}})} \\
&= \frac{\sum_{k=1}^{L}\text{Dropout}(\exp(\sum_{l=1}^{d_{\text{in}}}W_{i,l}X_{\text{in}_{k,l}})X_{\text{in}_{k,j}})}{\sum_{m=1}^{L}\exp(\sum_{n=1}^{d_{\text{in}}}W_{i,n}X_{\text{in}_{m,n}})}
\end{aligned}$$

Each $X_{\text{in}_{i,j}}$ can be written as:

$$X_{\text{in}_{i,j}} = \epsilon_j + \delta_{i,j}$$

Where ϵ_j and $\delta_{i,j}$ are all independent and defined as

$$\begin{aligned}
\epsilon_j &\sim \mathcal{N}(0, r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}) \\
\delta_{i,j} &\sim \mathcal{N}(0, (1 - r^l_{x_{\text{in}}})\sigma^2_{x_{\text{in}}}) \\
O_{i,j} &= \frac{\sum_{k=1}^{L} \text{Dropout}(\exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l} X_{\text{in}_{k,l}}) X_{\text{in}_{k,j}})}{\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l} X_{\text{in}_{k,l}})} \\
&= \frac{\sum_{k=1}^{L} (1 - d_{i,k})(\exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l} X_{\text{in}_{k,l}}) X_{\text{in}_{k,j}})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l} X_{\text{in}_{k,l}})}
\end{aligned}$$

Where $d_{i,k}$ is Bernoulli random variable which is 1 with probability p

$$\begin{aligned}
&= \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}(\epsilon_l + \delta_{k,l}))(\epsilon_j + \delta_{k,j})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}(\epsilon_l + \delta_{k,l}))} \\
&= \epsilon_j \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\epsilon_l) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\epsilon_l) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})} \\
&\quad + \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\epsilon_l) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})\delta_{k,j}}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\epsilon_l) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})} \\
&= \epsilon_j \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})} + \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})\delta_{k,j}}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}
\end{aligned}$$

Let $v_1 = \epsilon_j \frac{\sum_{k=1}^{L} (1-d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}$ and $v_2 = \frac{\sum_{k=1}^{L} (1-d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})\delta_{k,j}}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}$. We have,

$$O_{i,j} = v_1 + v_2$$

Given a fixed ϵ, W, we have

$$\begin{aligned}
v_1|\epsilon, W &= \epsilon_j \frac{\sum_{k=1}^{L} (1 - d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{(1-p)\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})} \\
&= \epsilon_j \frac{\frac{\sum_{k=1}^{L} (1-d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{L}}{(1-p)\frac{\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{L}}
\end{aligned}$$

By WLLN, $\frac{\sum_{k=1}^{L} (1-d_{i,k}) \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{L} \xrightarrow{p} (1-p)\mathbb{E}_\delta[\exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})]$, and

$(1-p)\frac{\sum_{k=1}^{L} \exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})}{L} \xrightarrow{p} (1-p)\mathbb{E}_\delta[\exp(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l})]$

Thus, we have $v_1|\epsilon, W \xrightarrow{p} \epsilon_j$

$$
\begin{aligned}
v_2|\epsilon, W &= \frac{\sum_{k=1}^{L}(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{(1-p)\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)} \\
&= \frac{\frac{1}{\sqrt{L}}\sqrt{L}\sum_{k=1}^{L}\frac{(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L}}{(1-p)\sum_{k=1}^{L}\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)}{L}}
\end{aligned}
$$

Let $\mu_{\text{num}} = \mathbb{E}_{\delta,d}[(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}]$, $\sigma^2_{\text{num}} = \text{Var}_{\delta,d}((1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j})$. By central limit theorem for large L,

$$
\begin{aligned}
\sqrt{L}\frac{\sum_{k=1}^{L}(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L} &= \sqrt{L}\frac{\sum_{k=1}^{L}(1-d_{i,k})(\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}-\mu_{\text{num}})}{L} + \sqrt{L}\mu_{\text{num}} \\
\sqrt{L}\frac{\sum_{k=1}^{L}(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L} &\xrightarrow{d} \mathcal{N}(0,\sigma^2_{\text{num}}) + \sqrt{L}\mu_{\text{num}} \\
\frac{\sum_{k=1}^{L}(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L} &\xrightarrow{d} \mathcal{N}(\mu_{\text{num}}, \frac{\sigma^2_{\text{num}}}{L}) \\
\mu_{\text{num}} &= \mathbb{E}_d[1-d_{i,k}](\prod_{l=1,l\neq j}^{l=d}\mathbb{E}_\delta[\exp\left(W_{i,l}\delta_{k,l}\right)])\mathbb{E}_\delta[\exp\left(W_{i,j}\delta_{k,j}\right)\delta_{k,j}] \\
\mathbb{E}_\delta[\exp\left(W_{i,l}\delta_{k,l}\right)] &= \exp\left(\frac{W^2_{i,l}\sigma^2_\delta}{2}\right) \qquad \text{(MGF of gaussian)} \\
\mathbb{E}_\delta[\exp\left(W_{i,j}\delta_{k,j}\right)\delta_{k,j}] &= \int_{-\infty}^{\infty}\frac{\exp\left(W_{i,j}\delta_{k,j}\right)\delta_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{\delta^2_{k,j}}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \int_{-\infty}^{\infty}\exp\left(\frac{W^2_{i,j}\sigma^2_\delta}{2}\right)\frac{\delta_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{(\delta_{k,j}-W_{i,j}\sigma^2_\delta)^2}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \exp\left(\frac{W^2_{i,j}\sigma^2_\delta}{2}\right)\int_{-\infty}^{\infty}\frac{\delta_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{(\delta_{k,j}-W_{i,j}\sigma^2_\delta)^2}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \exp\left(\frac{W^2_{i,j}\sigma^2_\delta}{2}\right)W_{i,j}\sigma^2_\delta \\
\mu_{\text{num}} &= (1-p)\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}} W^2_{i,l}\sigma^2_\delta}{2}\right)W_{i,j}\sigma^2_\delta \\
\sigma^2_{\text{num}} &= \mathbb{E}_d[(1-d_{i,k})^2](\prod_{l=1,l\neq j}^{l=d}\mathbb{E}_\delta[\exp\left(2W_{i,l}\delta_{k,l}\right)]\mathbb{E}_\delta[\exp\left(2W_{i,j}\delta_{k,j}\right)\delta^2_{k,j}] - \mu^2_{\text{num}}) \\
\mathbb{E}_{\delta_{k,l}}[\exp\left(2W_{i,l}\delta_{k,l}\right)] &= \exp\left(2W^2_{i,l}\sigma^2_\delta\right) \qquad \text{(MGF of gaussian)} \\
\mathbb{E}_{\delta_{k,j}}[\exp\left(2W_{i,j}\delta_{k,j}\right)\delta^2_{k,j}] &= \int_{-\infty}^{\infty}\frac{\exp\left(2W_{i,j}\delta_{k,j}\right)\delta^2_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{\delta^2_{k,j}}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \int_{-\infty}^{\infty}\exp\left(2W^2_{i,j}\sigma^2_\delta\right)\frac{\delta^2_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{(\delta_{k,j}-2W_{i,j}\sigma^2_\delta)^2}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \exp\left(2W^2_{i,j}\sigma^2_\delta\right)\int_{-\infty}^{\infty}\frac{\delta^2_{k,j}}{\sqrt{2\pi}\sigma_\delta}\exp\left(-\frac{(\delta_{k,j}-2W_{i,j}\sigma^2_\delta)^2}{2\sigma^2_\delta}\right)d\delta_{k,j} \\
&= \exp\left(2W^2_{i,j}\sigma^2_\delta\right)(4W^2_{i,j}\sigma^4_\delta+\sigma^2_\delta) \\
\sigma^2_{\text{num}} &= (1-p)\exp\left(2\sum_{l=1}^{d_{\text{in}}} W^2_{i,l}\sigma^2_\delta\right)(4W^2_{i,j}\sigma^4_\delta+\sigma^2_\delta) - (1-p)^2\exp\left(\sum_{l=1}^{d_{\text{in}}} W^2_{i,l}\sigma^2_\delta\right)W^2_{i,j}\sigma^4_\delta
\end{aligned}
$$

Similarly, $\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)$ is also a sum of L i.i.d. random variables for fixed W. By WLLN we have,

$$(1-p)\frac{\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)}{L}\xrightarrow{p}(1-p)\mathbb{E}_{\delta}[\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)]$$

$$\xrightarrow{p}(1-p)(\prod_{l=1}^{l=d}\mathbb{E}_{\delta}[\exp\left(W_{i,l}\delta_{k,l}\right)])$$

$$(1-p)\frac{\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)}{L}\xrightarrow{p}(1-p)\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2}{2}\right)$$

$$v_2=\frac{\frac{\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L}}{\frac{\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)}{L}}$$

As for a given W,ϵ, both the numerator and denominator converge in distribution and denominator is converging to a constant by Slutskys theorem,

$$v_2|W,\epsilon\xrightarrow{d}\mathcal{N}\left(\frac{\mu_{\text{num}}}{(1-p)\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2}{2}\right)},\frac{\sigma_{\text{num}}^2}{L(1-p)^2\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)}\right)$$

$$v_2|W,\epsilon\xrightarrow{d}\mathcal{N}\left(W_{i,j}\sigma_{\delta}^2,\frac{\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)(4W_{i,j}^2\sigma_{\delta}^4+\sigma_{\delta}^2)}{(1-p)}-W_{i,j}^2\sigma_{\delta}^4}{L}\right)$$

Thus we have,

$$O_{i,j}|W,\epsilon\sim\mathcal{N}\left(W_{i,j}\sigma_{\delta}^2,\frac{\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)(4W_{i,j}^2\sigma_{\delta}^4+\sigma_{\delta}^2)}{(1-p)}-W_{i,j}^2\sigma_{\delta}^4}{L}\right)+\epsilon_j$$

We have,

$$\mathbb{E}[O_{i,j}|W]=W_{i,j}\sigma_{\delta}^2+0=W_{i,j}\sigma_{\delta}^2$$

$$\mathbb{E}[O_{i,j}^2|W]=\frac{\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)(4W_{i,j}^2\sigma_{\delta}^4+\sigma_{\delta}^2)}{(1-p)}-W_{i,j}^2\sigma_{\delta}^4}{L}+\sigma_{\epsilon}^2$$

$$\mathbb{E}[O_{i,j}]=\mathbb{E}_W[O_{i,j}|W]=\mathbb{E}_W[W_{i,j}\sigma_{\delta}^2]=0$$

$$\mathbb{E}[O_{i,j}^2]=\mathbb{E}_W[O_{i,j}^2|W]$$

$$=\mathbb{E}_W[W_{i,j}^2\sigma_{\delta}^4+\frac{\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)(4W_{i,j}^2\sigma_{\delta}^4+\sigma_{\delta}^2)}{(1-p)}-W_{i,j}^2\sigma_{\delta}^4}{L}+\sigma_{\epsilon}^2]$$

For large d_{in} by WLLN and continuous mapping theorem $\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_{\delta}^2\right)\approx\exp\left(d_{\text{in}}\sigma_w^2\sigma_{\delta}^2\right)$

$$=\frac{(L-1)\sigma_w^2\sigma_{\delta}^4+\frac{\exp\left(d_{\text{in}}\sigma_w^2\sigma_{\delta}^2\right)(4\sigma_w^2\sigma_{\delta}^4+\sigma_{\delta}^2)}{(1-p)}}{L}+\sigma_{\epsilon}^2$$

$$=\frac{(1-r_{x_{\text{in}}}^l)^2(L-1)d_{\text{in}}\sigma_{x_{\text{in}}}^6\sigma_{qk}^2+\frac{\exp\left((1-r_{x_{\text{in}}}^l)d_{\text{in}}^2\sigma_{x_{\text{in}}}^4\sigma_{qk}^2\right)(4(1-r_{x_{\text{in}}}^l)^2d_{\text{in}}\sigma_{x_{\text{in}}}^6\sigma_{qk}^2+(1-r_{x_{\text{in}}}^l)\sigma_{x_{\text{in}}}^2)}{(1-p)}}{L}+r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2$$

Hence,

$$\boxed{\begin{aligned}&\mu_{x_{out}}=0\\&\sigma_{x_{out}}^2=\frac{(1-r_{x_{\text{in}}}^l)^2(L-1)d_{\text{in}}\sigma_{x_{\text{in}}}^6\sigma_{qk}^2+\frac{\exp\left((1-r_{x_{\text{in}}}^l)d_{\text{in}}^2\sigma_{x_{\text{in}}}^4\sigma_{qk}^2\right)(4(1-r_{x_{\text{in}}}^l)^2d_{\text{in}}\sigma_{x_{\text{in}}}^6\sigma_{qk}^2+(1-r_{x_{\text{in}}}^l)\sigma_{x_{\text{in}}}^2)}{(1-p)}}{L}+r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2\end{aligned}}$$

Now to get covariance we make two approximations. As the term $\frac{\sum_{k=1}^{L}(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)}{(1-p)\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)}$ converges to 1, we approximate $v_{1_{i,j}} \approx \epsilon_j$. Also we will treat $\sum_{k=1}^{L}\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right) \approx \exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_\delta^2}{2}\right)$. Then, we have

$$
\begin{aligned}
v_{1_{i,j}} &\approx \epsilon_j \\
v_{2_{i,j}} &\approx \frac{\sum_{k=1}^{L}\frac{(1-d_{i,k})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k,l}\right)\delta_{k,j}}{L}}{(1-p)\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}W_{i,l}^2\sigma_\delta^2}{2}\right)}
\end{aligned}
$$

This makes $v_{1_{i,j}}$ and $v_{2_{i,j}}$ independent. For covariance

$$
\begin{aligned}
\mathbb{E}[O_{i,j}O_{m,j}] &= \mathbb{E}_W[\mathbb{E}[O_{i,j}O_{m,j}|W]] \\
O_{i,j}O_{m,j}|W &= (v_{1_{i,j}} + v_{2_{i,j}})(v_{1_{m,j}} + v_{2_{m,j}}) \\
&= v_{1_{i,j}}v_{1_{m,j}} + v_{1_{i,j}}v_{2_{m,j}} + v_{2_{i,j}}v_{1_{m,j}} + v_{2_{i,j}}v_{2_{m,j}} \\
v_{1_{i,j}}v_{1_{m,j}} &= \epsilon_j^2 \\
\mathbb{E}[v_{1_{i,j}}v_{1_{m,j}}|W] &= \sigma_\epsilon^2
\end{aligned}
$$

As $v_{1_{i,j}} = v_{1_{m,j}} = \epsilon_j, v_{1_{i,j}}v_{2_{m,j}} + v_{2_{i,j}}v_{1_{m,j}} = \epsilon_j(v_{2_{i,j}} + v_{2_{m,j}})$, and ϵ_j is independent of $(v_{2_{i,j}} + v_{2_{m,j}})$. Thus, we have

$$
\begin{aligned}
\mathbb{E}[v_{1_{i,j}}v_{2_{m,j}} + v_{2_{i,j}}v_{1_{m,j}}|W] &= \mathbb{E}[\epsilon_j|W]\mathbb{E}[(v_{2_{i,j}} + v_{2_{m,j}})|W] = 0 * \mathbb{E}[(v_{2_{i,j}} + v_{2_{m,j}})|W] = 0 \\
v_{2_{i,j}}v_{2_{m,j}} &= \frac{\sum_{k_1=1}^{L}\frac{(1-d_{i,k_1})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k_1,l}\right)\delta_{k_1,j}}{L}\sum_{k_2=1}^{L}\frac{(1-d_{m,k_2})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{m,l}\delta_{k_2,l}\right)\delta_{k_2,j}}{L}}{(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)}
\end{aligned}
$$

$$
\mathbb{E}[v_{2_{i,j}}v_{2_{m,j}}|W] = \frac{\mathbb{E}[\sum_{k_1=1}^{L}(1-d_{i,k_1})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k_1,l}\right)\delta_{k_1,j}\sum_{k_2=1}^{L}(1-d_{m,k_2})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{m,l}\delta_{k_2,l}\right)\delta_{k_2,j}]}{L^2(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)}
$$

Breaking summation into two parts: $k_1 = k_2 = k$ and $k_1 \neq k_2$, we get

$$
\begin{aligned}
&= \frac{\mathbb{E}[\sum_{k=1}^{L}(1-d_{i,k})(1-d_{m,k})\exp\left(\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})\delta_{k,l}\right)\delta_{k,j}^2]}{L^2(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)} \\
&\quad + \frac{\mathbb{E}[\sum_{k_1=1}^{L}\sum_{k_2=1,k_2\neq k_1}^{L}(1-d_{i,k_1})(1-d_{m,k_2})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k_1,l}\right)\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{m,l}\delta_{k_2,l}\right)\delta_{k_1,j}\delta_{k_2,j}]}{L^2(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)} \\
&= \frac{\sum_{k=1}^{L}\mathbb{E}[(1-d_{i,k})(1-d_{m,k})\exp\left(\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})\delta_{k,l}\right)\delta_{k,j}^2]}{L^2(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)} \\
&\quad + \frac{\sum_{k_1=1}^{L}\sum_{k_2=1,k_2\neq k_1}^{L}\mathbb{E}[(1-d_{i,k_1})(1-d_{m,k_2})\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{i,l}\delta_{k_1,l}\right)\exp\left(\sum_{l=1}^{d_{\text{in}}}W_{m,l}\delta_{k_2,l}\right)\delta_{k_1,j}\delta_{k_2,j}]}{L^2(1-p)^2\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)}
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}[(1-d_{i,k})(1-d_{m,k})\exp\left(\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})\delta_{k,l}\right)\delta_{k,j}^2] = \\
&\qquad = \mathbb{E}[(1-d_{i,k})]\mathbb{E}[(1-d_{m,k})]\mathbb{E}[\exp\left(\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})\delta_{k,l}\right)\delta_{k,j}^2]
\end{aligned}
$$

$$= (1-p)^2 \exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})^2\sigma_\delta^2}{2}\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)$$

$$\mathbb{E}[(1-d_{i,k_1})(1-d_{m,k_2})\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k_1,l}\right)\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{m,l}\delta_{k_2,l}\right)\delta_{k_1,j}\delta_{k_2,j}] =$$

$$\mathbb{E}[(1-d_{i,k_1})]\mathbb{E}[(1-d_{m,k_2})]\mathbb{E}[\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}\delta_{k_1,l}\right)\delta_{k_1,j}]\mathbb{E}[\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{m,l}\delta_{k_2,l}\right)\delta_{k_2,j}]$$

$$= (1-p)^2 \exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)W_{i,j}W_{m,j}\sigma_\delta^4$$

$$\mathbb{E}[v_{2_{i,j}}v_{2_{m,j}}|W] = \frac{\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}+W_{m,l})^2\sigma_\delta^2}{2}\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)}{L\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)}$$

$$+ \frac{(L-1)\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)W_{i,j}W_{m,j}\sigma_\delta^4}{L\exp\left(\frac{\sum_{l=1}^{d_{\text{in}}}(W_{i,l}^2+W_{m,l}^2)\sigma_\delta^2}{2}\right)}$$

$$= \frac{\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}W_{m,l}\sigma_\delta^2\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)}{L} + \frac{(L-1)W_{i,j}W_{m,j}\sigma_\delta^4}{L}$$

So, we have

$$\mathbb{E}[O_{i,j}O_{m,j}|W] = \sigma_\epsilon^2 + \frac{\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}W_{m,l}\sigma_\delta^2\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)}{L} + \frac{(L-1)W_{i,j}W_{m,j}\sigma_\delta^4}{L}$$

$$\mathbb{E}[O_{i,j}O_{m,j}] = \mathbb{E}_W[\mathbb{E}[O_{i,j}O_{m,j}|W]]$$

$$= \mathbb{E}_W[\sigma_\epsilon^2 + \frac{\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}W_{m,l}\sigma_\delta^2\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)}{L} + \frac{(L-1)W_{i,j}W_{m,j}\sigma_\delta^4}{L}]$$

$$= \sigma_\epsilon^2 + \mathbb{E}_W[\frac{\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}W_{m,l}\sigma_\delta^2\right)((W_{i,j}+W_{m,j})^2\sigma_\delta^4+\sigma_\delta^2)}{L}]$$

For large values of d_{in} by WLLN and continuous mapping theorem we have $\exp\left(\sum_{l=1}^{d_{\text{in}}} W_{i,l}W_{m,l}\sigma_\delta^2\right) \approx 1$. Thus, we have

$$\mathbb{E}[O_{i,j}O_{m,j}] = \sigma_\epsilon^2 + \frac{(2\sigma_w^2\sigma_\delta^4+\sigma_\delta^2)}{L}$$

$$\mathbb{E}[O_{i,j}O_{m,j}] = r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2 + \frac{(2(1-r_{x_{\text{in}}}^l)^2 d_{\text{in}}\sigma_{x_{\text{in}}}^6\sigma_{qk}^2 + (1-r_{x_{\text{in}}}^l)\sigma_{x_{\text{in}}}^2)}{L}$$

The convergence arguments we have made require the scale of the variables to be small when compared to L and d_{in}. The growth in scale can be controlled easily by controlling σ_{qk}, and we observe that if we let σ_{qk} become arbitrarily large the scores passed to Softmax diverge leading to degenerate attention only attending to one token which has the highest score. To avoid this degenerate attention, we choose smaller values of σ_q, σ_k and in that scenario, the approximate value for variance and covariance are,

$$\boxed{\begin{aligned} \sigma_{x_{out}}^2 &\approx r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2 \\ \text{Cov}_{x_{out}}^l &\approx r_{x_{\text{in}}}^l\sigma_{x_{\text{in}}}^2 \end{aligned}}$$

To get the final variance and covariance we can use results of Linear layer to account for $\mathbf{W_V}$. If we initialize σ_q and σ_k to be small, in initial phase of training the output of Softmax layer can be treated as being a constant $= \frac{\mathbf{1_L^T 1_L}}{\mathbf{L}}$. Using this assumption we have,

$$\mathbf{X}_{\text{out}} \approx \text{Dropout}(\frac{\mathbf{1_L^T 1_L}}{\mathbf{L}})\mathbf{X}_{\text{in}}\mathbf{W_V}$$

$$\implies \mathbf{g_{X_{in}}} \approx \text{Dropout}(\frac{\mathbf{1_L^T 1_L}}{\mathbf{L}})^T \mathbf{g_{X_{out}}} \mathbf{W_V}^T$$
$$= \text{Dropout}(\frac{\mathbf{1_L^T 1_L}}{\mathbf{L}}) \mathbf{g_{X_{out}}} \mathbf{W_V}^T$$

$$\boxed{\begin{aligned} \mu_{g_{\text{in}}} &= 0 \\ \sigma^2_{g_{\text{in}}} &= \frac{\sigma^2_{g_{\text{out}}} d \sigma_v^2}{L(1-p)}(1 + (L-1) r^l_{g_{\text{out}}}(1-p)) \\ \text{Cov}^l_{g_{\text{in}}} &= \frac{\sigma^2_{g_{\text{out}}} d \sigma_v^2}{L}(1 + (L-1) r^l_{g_{\text{out}}}) \end{aligned}}$$

B Moment Propagation through Transformer Blocks

B.1 Transformer Attention Block

A forward pass through attention block consists of LayerNorm, followed by Scaled Dot-Product Attention, followed by an output projection layer (a Linear Layer), and finally a Dropout. Using the results from above we get,

$$\mu_{x_{\text{out}}} = 0 * 0 * 0 * 0 = 0$$

$$\sigma^2_{x_{\text{out}}}$$
$$= \left(\frac{(1 - r^l_{x_{\text{in}}})^2 (L-1) d_{\text{in}} \sigma^6_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + \frac{\exp\left((1-r^l_{x_{\text{in}}}) d^2_{\text{in}} \sigma^4_{x_{\text{in}}} \sigma_q^2 \sigma_k^2\right)(4(1-r^l_{x_{\text{in}}})^2 d_{\text{in}} \sigma^6_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + (1-r^l_{x_{\text{in}}}) \sigma^2_{x_{\text{in}}})}{(1-p)}}{L} + r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} \right) . d_{\text{in}} \sigma_v^2 . \frac{d_{\text{in}} \sigma_o^2}{(1-p)}$$
$$= \frac{d^2_{\text{in}} \sigma_o^2 \sigma_v^2 \sigma^2_{x_{\text{in}}}}{(1-p)} \left(\frac{(1 - r^l_{x_{\text{in}}})^2 (L-1) d_{\text{in}} \sigma^4_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + \frac{\exp\left((1-r^l_{x_{\text{in}}}) d^2_{\text{in}} \sigma^4_{x_{\text{in}}} \sigma_q^2 \sigma_k^2\right)(4(1-r^l_{x_{\text{in}}})^2 d_{\text{in}} \sigma^4_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + (1-r^l_{x_{\text{in}}}))}{(1-p)}}{L} + r^l_{x_{\text{in}}} \right)$$

$$\text{Cov}^l_{x_{\text{out}}}$$
$$= \left(r^l_{x_{\text{in}}} \sigma^2_{x_{\text{in}}} + \frac{(2(1 - r^l_{x_{\text{in}}})^2 d_{\text{in}} \sigma^6_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + (1 - r^l_{x_{\text{in}}}) \sigma^2_{x_{\text{in}}})}{L} \right) . d_{\text{in}} \sigma_v^2 . d_{\text{in}} \sigma_o^2 . 1$$
$$= d^2_{\text{in}} \sigma_o^2 \sigma_v^2 \sigma^2_{x_{\text{in}}} \left(r^l_{x_{\text{in}}} + \frac{(2(1 - r^l_{x_{\text{in}}})^2 d_{\text{in}} \sigma^4_{x_{\text{in}}} \sigma_q^2 \sigma_k^2 + (1 - r^l_{x_{\text{in}}}))}{L} \right)$$

$$\sigma^2_{g_{\text{in}}} = \sigma^2_{g_{\text{out}}} * \frac{1}{(1-p)} * d_{\text{in}} \sigma_o^2 * \frac{d_{\text{in}} \sigma_v^2}{L(1-p)}(1 + (L-1) r^l_{g_{\text{out}}}(1-p))$$
$$= \frac{d^2_{\text{in}} \sigma^2_{g_{\text{out}}} \sigma_v^2 \sigma_o^2}{L(1-p)^2}(1 + (L-1) r^l_{g_{\text{out}}}(1-p))$$

$$\text{Cov}^l_{g_{\text{in}}} = \sigma^2_{g_{\text{out}}} * 1 * d_{\text{in}} \sigma_o^2 * \frac{d_{\text{in}} \sigma_v^2}{L}(1 + (L-1) r^l_{g_{\text{out}}})$$
$$= \frac{d^2_{\text{in}} \sigma^2_{g_{\text{out}}} \sigma_v^2 \sigma_o^2}{L}(1 + (L-1) r^l_{g_{\text{out}}})$$

B.2 Transformer FFN Block

A forward pass through the FFN block of a transfer has a LayerNorm, then a Linear layer from d to $4d$, which is then passed through a ReLU gate, the output of which is the projected back to d dimension using another Linear layer, and eventually

passed through a Dropout. Again using the results from above we get,

$$
\begin{aligned}
\mu_{x_{\text{out}}} &= 0 && \text{(Last Linear Layer makes it 0)} \\
\sigma^2_{x_{\text{out}}} &= 1 * d_{\text{in}}\sigma^2_{w_1} * (\frac{\pi-1}{2\pi} + \frac{1}{2\pi}) * 4d_{\text{in}}\sigma^2_{w_2} * \frac{1}{(1-p)} * \sigma^2_{x_{\text{in}}} \\
&= \frac{2d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}}{(1-p)}\sigma^2_{x_{\text{in}}} \\
\text{Cov}^l_{x_{\text{out}}} &= d_{\text{in}}\sigma^2_{w_1} * (\frac{r^l_{x_{\text{in}}}}{4} + \frac{(1-(r^l_{x_{\text{in}}})^2)^{0.5}}{2\pi} + \frac{r^l_{x_{\text{in}}}\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi} - \frac{1}{2\pi} + \frac{1}{2\pi}) * 4d_{\text{in}}\sigma^2_{w_2} * \sigma^2_{x_{\text{in}}} \\
&= 4d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{x_{\text{in}}}(\frac{r^l_{x_{\text{in}}}}{4} + \frac{(1-(r^l_{x_{\text{in}}})^2)^{0.5}}{2\pi} + \frac{r^l_{x_{\text{in}}}\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi}) \\
\text{r}^l_{x_{\text{out}}} &= 2*(1-p)*(\frac{r^l_{x_{\text{in}}}}{4} + \frac{(1-(r^l_{x_{\text{in}}})^2)^{0.5}}{2\pi} + \frac{r^l_{x_{\text{in}}}\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi}) \\
&\approx (1-p)*(\frac{r^l_{x_{\text{in}}}}{2} + \frac{1}{\pi} + (\frac{1}{2} - \frac{1}{\pi}){r^l_{x_{\text{in}}}}^2) && \text{(Fitting a 2-nd order polynomial)} \\
\sigma^2_{g_{\text{in}}} &= \sigma^2_{g_{\text{out}}} * \frac{1}{(1-p)} * d_{\text{in}}\sigma^2_{w_2} * \frac{1}{2} * 4d_{\text{in}}\sigma^2_{w_1} \\
&= \frac{2d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{g_{\text{out}}}}{(1-p)} \\
\text{Cov}^l_{g_{\text{in}}} &= \text{Cov}^l_{g_{\text{out}}} * 1 * d_{\text{in}}\sigma^2_{w_2} * (\frac{1}{4} + \frac{\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi}) * 4d_{\text{in}}\sigma^2_{w_1} \\
&= 4d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\text{Cov}^l_{g_{\text{out}}}(\frac{1}{4} + \frac{\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi})
\end{aligned}
$$

C Summary Table of Moment Propagation through Transformer Components

In Table 15, Table 16, Table 17, Table 18, Table 19 and Table 20, we summarize the signal propagation formulae for all the transformer components.

Table 15. Moment Propagation (mean) during forward pass through components of transformer model.

Component	$\mu_{x_{\text{out}}}$
Embeddings	0
FFN ($d_1.d_2$)	0
ReLU	$\frac{\sigma_{x_{\text{in}}}}{\sqrt{(2\pi)}}$
GeLU	$\frac{\sigma^2_{x_{\text{in}}}}{\sqrt{2\pi(\sigma^2_{x_{\text{in}}}+1)}}$
LayerNorm (d)	0
Dropout (p)	$\mu_{x_{\text{in}}}$
Softmax	$\frac{1}{L}$
SHA Block (without V)	0
Attn Block	0
FFN Block	0

Table 16. Moment Propagation (variance) during forward pass through components of transformer model.

Component	$\sigma^2_{x_{\text{out}}}$
Embeddings	0
FFN ($d_1.d_2$)	$d_1\sigma_w^2(\sigma^2_{x_{\text{in}}}+\mu^2_{x_{\text{in}}})$
ReLU	$\frac{(\pi-1)}{(2\pi)}\sigma^2_{x_{\text{in}}}$
GeLU	$\frac{\sigma^2_{x_{\text{in}}}}{2\pi}(\frac{\pi}{2}-\frac{\sigma^2_{x_{\text{in}}}}{1+\sigma^2_{x_{\text{in}}}}+\sin^{-1}(\frac{\sigma^2_{x_{\text{in}}}}{1+\sigma^2_{x_{\text{in}}}})+\frac{2\sigma^2_{x_{\text{in}}}}{(1+\sigma^2_{x_{\text{in}}})\sqrt{1+2\sigma^2_{x_{\text{in}}}}})$
Layer Norm (d)	1
Dropout (p)	$\frac{\sigma^2_{x_{\text{in}}}+p\mu^2_{x_{\text{in}}}}{1-p}$
Softmax	$\frac{(e^{\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})\frac{L}{L-1}}-1)e^{2\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})\frac{L}{L-1}}}{((L-1)e^{\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})}+1)^2}$
SHA (without V)	$\frac{d_{\text{in}}\sigma^2_{x_{\text{in}}}}{(1-p)}\left(\frac{(1-r^l_{x_{\text{in}}})^2(L-1)d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k+\frac{\exp((1-r^l_{x_{\text{in}}})d^2_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k)(4(1-r^l_{x_{\text{in}}})^2d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k+(1-r^l_{x_{\text{in}}}))}{(1-p)}}{L}+r^l_{x_{\text{in}}}\right)$
Attn Block (Approx)	$\frac{d^2_{\text{in}}\sigma^2_o\sigma^2_v\sigma^2_{x_{\text{in}}}}{(1-p)}\left(\frac{(1-r^l_{x_{\text{in}}})^2(L-1)d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k+\frac{\exp((1-r^l_{x_{\text{in}}})d^2_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k)(4(1-r^l_{x_{\text{in}}})^2d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma^2_q\sigma^2_k+(1-r^l_{x_{\text{in}}}))}{(1-p)}}{L}+r^l_{x_{\text{in}}}\right)$
FFN Block	$\frac{2d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{x_{\text{in}}}}{(1-p)}$

Table 17. Moment Propagation (variance) during backwards pass through components of transformer model.

Component	$\sigma^2_{g_{\text{in}}}$
Embeddings	-
FFN ($d_1.d_2$)	$d_2\sigma_w^2\sigma^2_{g_{\text{out}}}$
ReLU	$\frac{1}{2}\sigma^2_{g_{\text{out}}}$
GeLU	$\left[\frac{1}{4}+\frac{1}{2\pi}\sin^{-1}\left(\frac{\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}\right)+\frac{\sigma^2_{x_{\text{in}}}(5\sigma^2_{x_{\text{in}}}+3)}{2\pi(\sigma^2_{x_{\text{in}}}+1)(2\sigma^2_{x_{\text{in}}}+1)^{\frac{3}{2}}}\right]\sigma^2_{g_{\text{out}}}$
LayerNorm (d)	$\frac{\sigma^2_{g_{\text{out}}}}{\sigma^2_{x_{\text{in}}}}$
Dropout (p)	$\frac{1}{1-p}\sigma^2_{g_{\text{out}}}$
Softmax	$\left(\frac{(e^{\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})\frac{L}{L-1}}-1)e^{2\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})\frac{L}{L-1}}}{((L-1)e^{\sigma^2_{x_{\text{in}}}(1-r^l_{x_{\text{in}}})}+1)^2}+\frac{1}{L^2}\right)\sigma^2_{g_{\text{out}}}$
SHA Block (without V)	$\frac{d_{\text{in}}\sigma^2_{g_{\text{out}}}}{L(1-p)^2}(1+(L-1)r^l_{g_{\text{out}}}(1-p))$
Attn Block (Approx)	$\frac{d^2_{\text{in}}\sigma^2_{g_{\text{out}}}\sigma^2_v\sigma^2_o}{L(1-p)^2}(1+(L-1)r^l_{g_{\text{out}}}(1-p))$
FFN Block	$\frac{2d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{g_{\text{out}}}}{(1-p)}$

Table 18. Covariance (along sequence length) propagation through the components of transformer model.

Component	$\mathrm{Cov}^l_{x_{\text{out}}}$
Embeddings	$\sum \frac{N_i * (N_i - 1)}{L * (L-1))} * \sigma^2_{w_{\text{embd}}}$
FFN ($d_1.d_2$)	$d_1 \sigma_w^2 (\mathrm{Cov}^l_{x_{\text{in}}} + \mu^2_{x_{\text{in}}})$
ReLU	$(\frac{1}{4} + \frac{\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi})\mathrm{Cov}^l_{x_{\text{in}}} - (1 - \sqrt{(1-(r^l_{x_{\text{in}}})^2)})\frac{\sigma^2_{x_{\text{in}}}}{2\pi}$
GeLU	$\frac{\sigma^2_{x_{\text{in}}}}{4\pi}(\pi r^l_{x_{\text{in}}} + 2r^l_{x_{\text{in}}} \sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1}) + \frac{2\sigma^2_{x_{\text{in}}}(\sigma^2_{x_{\text{in}}}(1-(r^l_{x_{\text{in}}})^2)+1+(r^l_{x_{\text{in}}})^2)}{(\sigma^2_{x_{\text{in}}}+1)\sqrt{(\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2}} - \frac{2\sigma^2_{x_{\text{in}}}}{(\sigma^2_{x_{\text{in}}}+1)})$
LayerNorm (d)	$(1 - \frac{1}{d})\frac{\mathrm{Cov}^l_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}}$
Dropout (p)	$\mathrm{Cov}^l_{x_{\text{in}}}$
SHA (without V)	$d_{\text{in}}\sigma^2_{x_{\text{in}}}\left(r^l_{x_{\text{in}}} + \frac{(2(1-r^l_{x_{\text{in}}})^2 d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma_q^2\sigma_k^2 + (1-r^l_{x_{\text{in}}}))}{L}\right)$
Attn Block (Approx)	$d^2_{\text{in}}\sigma_o^2\sigma_v^2\sigma^2_{x_{\text{in}}}\left(r^l_{x_{\text{in}}} + \frac{(2(1-r^l_{x_{\text{in}}})^2 d_{\text{in}}\sigma^4_{x_{\text{in}}}\sigma_q^2\sigma_k^2 + (1-r^l_{x_{\text{in}}}))}{L}\right)$
FFN Block	$4d_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\sigma^2_{x_{\text{in}}}(\frac{r^l_{x_{\text{in}}}}{4} + \frac{\sqrt{(1-(r^l_{x_{\text{in}}})^2}}{2\pi} + \frac{r^l_{x_{\text{in}}}\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi})$

Table 19. Covariance (hidden dimension) propagation through the components of transformer model.

Component	$\mathrm{Cov}^d_{x_{\text{out}}}$
Embeddings	0
FFN ($d_1.d_2$)	0
ReLU	$(\frac{1}{4} + \frac{\sin^{-1}(r^d_{x_{\text{in}}})}{2\pi})\mathrm{Cov}^d_{x_{\text{in}}} - (1 - \sqrt{(1-(r^d_{x_{\text{in}}})^2)})\frac{\sigma^2_{x_{\text{in}}}}{2\pi}$
GeLU	
LayerNorm (d)	$-\frac{1}{d-1}$
Dropout (p)	$\mathrm{Cov}^d_{x_{\text{in}}}$
SHA Block(without V)	0
Attn Block	0
FFN Block	0

Table 20. Gradient covariance (along sequence length) propagation through the components of transformer model.

Component	$\text{Cov}^l_{g_{\text{in}}}$
Embeddings	-
FFN ($d_1.d_2$)	$d_2\sigma_w^2\text{Cov}^l_{g_{\text{out}}}$
ReLU	$(\frac{1}{4}+\frac{\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi})\text{Cov}^l_{g_{\text{out}}}$
GeLU	$\left[\frac{1}{4}+\frac{1}{2\pi}\sin^{-1}(\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}}{\sigma^2_{x_{\text{in}}}+1})+\frac{r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}}((2\sigma^2_{x_{\text{in}}}+3)(\sigma^2_{x_{\text{in}}}+1)-2(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)}{2\pi(\sigma^2_{x_{\text{in}}}+1)((\sigma^2_{x_{\text{in}}}+1)^2-(r^l_{x_{\text{in}}}\sigma^2_{x_{\text{in}}})^2)^{\frac{3}{2}}}\right] r^l_{g_{\text{out}}}\sigma^2_{g_{\text{out}}}$
LayerNorm (d)	$\frac{\text{Cov}^l_{g_{\text{out}}}}{\sigma^2_{x_{\text{in}}}}$
Dropout (p)	$\text{Cov}^l_{g_{\text{out}}}$
SHA Block (without V)	$\frac{d_{\text{in}}\sigma^2_{g_{\text{out}}}}{L}(1+(L-1)r^l_{g_{\text{out}}})$
Attn Block (Approx)	$\frac{d^2_{\text{in}}\sigma^2_{g_{\text{out}}}\sigma^2_v\sigma^2_o}{L}(1+(L-1)r^l_{g_{\text{out}}})$
FFN Block	$4d^2_{\text{in}}\sigma^2_{w_1}\sigma^2_{w_2}\text{Cov}^l_{g_{\text{out}}}(\frac{1}{4}+\frac{\sin^{-1}(r^l_{x_{\text{in}}})}{2\pi})$

D Numerical Verification

We perform numerical verification for the formulae reported in Table 15, Table 16, Table 17, Table 18, Table 19 and Table 20. The parameter ranges have been provided in Table 22. For each parameter, 3-5 values were sampled uniformly (or log uniformly) across the range for numerical simulation. Table 21 provides the percentage error corresponding to the 50_{th}, 90_{th} and 99_{th} percentile. These simulation results are all fully reproducible using our released code. Even at 99 percentile, no error (other than SHA backwards) is larger than 10%, verifying our assumptions.

Table 21. Percentage Errors [50th, 90th, 99th percentile] for the theoretical formulas corresponding to forward and backward pass through components of the transformer model.

Component	$\mu_{x_{\text{out}}}$	$\sigma^2_{x_{\text{out}}}$	$\sigma^2_{g_{\text{in}}}$	$\text{Cov}^l_{x_{\text{out}}}$	$\text{Cov}^l_{g_{\text{in}}}$
FFN	[0.0, 0.4, 1.3]	[0.4, 1.4, 2.8]	[0.2, 1.0, 2.2]	[0.4, 1.4, 2.8]	[0.2, 1.0, 2.2]
ReLU	[0.3, 1.3, 2.3]	[0.5, 1.9, 3.4]	[0.6, 1.5, 2.6]	[0.3, 1.6, 3.1]	[0.2, 1.1, 2.3]
GeLU	[0.1, 1.0, 2.4]	[0.2, 0.6, 1.3]	[0.2, 0.6, 1.1]	[0.1, 0.5, 1.2]	[0.1, 0.4, 0.9]
LayerNorm	[0.0 , 0.0, 0.0]	[0.0, 0.0, 0.0]	[0.4, 1.5, 3.2]	[0.1, 0.5, 1.0]	[0.2, 0.9, 2.2]
Dropout	[0.0, 0.1, 0.5]	[0.1, 0.5, 1.5]	[0.1, 0.7, 1.5]	[0.0, 0.4, 1.3]	[0.1, 0.5, 1.2]
Softmax	[0.0 , 0.0, 0.0]	[0.2, 0.9, 4.0]	[0.1, 0.6, 4.5]	-	-
Single-Head Atten.	[0.2, 1.0, 2.5]	[1.4, 4.1, 7.8]	[2.2, 13.3, 44.5]	[1.3, 3.9, 7.4]	[1.6, 4.5, 8.2]

Table 22. Range of input variance/correlations used for theoretical formula verification reported in Table 21 for the theoretical formulas corresponding to forward and backward pass through components of the transformer model. The dropout probability range was $[0, 1)$ for Dropout and Single-Head Attention, and σ_w^2 for FFN was $[10^{-2}, 10^2]/d_{\text{in}}$.

Component	$\mu_{x_{\text{in}}}$	$\sigma^2_{x_{\text{in}}}$	$\sigma^2_{g_{\text{out}}}$	$\text{Corr}^l_{x_{\text{in}}}$	$\text{Corr}^l_{g_{\text{out}}}$	d_{in}	d_{out}	L
FFN	[-10, 10]	[0.1, 10]	[0.1, 10]	[0, 1.0)	[0, 1.0)	$[10^1, 10^3]$	$[10^1, 10^3]$	$[10^2, 10^3]$
ReLU	[0]	[0.1, 10]	[0.1, 10]	[0, 1.0)	[0, 1.0)	-	-	$[10^2, 10^3]$
GeLU	[0]	[0.1, 10]	[0.1, 10]	[0, 1.0)	[0, 1.0)	-	-	$[10^2, 10^3]$
LayerNorm	[-10, 10]	[0.1, 10]	[0.1, 10]	[0, 1.0)	[0, 1.0)	$[10^2, 10^3]$	-	$[10^2, 10^3]$
Dropout	[-10, 10]	[0.1, 10]	[0.1, 10]	[0, 1.0)	[0, 1.0)	$[10^2, 10^3]$	-	$[10^2, 10^3]$
Softmax	[0]	$[10^{-4}, 1]$	[0.1, 10]	[0, 1.0)	-	-	-	$[300, 10^4]$
Single-Head Atten.	[0]	[1]	[0.1, 10]	[0, 1.0)	[0, 1.0)	$[10^2, 10^3]$	[32, 64, 128, 256]	$[300, 10^4]$

E Moment Propagation through the Entire Transformer Model

E.1 Vanilla Pre-LN

We will use the approximations listed in Table 2 here.

E.1.1 Forward Pass

For forward pass, a Transformer Pre-LN has LayerNorm followed by the Attention block, residual connection, LayerNorm, and then the FFN block. Let σ^2_{layer} be the output variance after 1 such layer, and σ^2_{model} be the output variance after the entire model of N layers.

$$
\begin{aligned}
\sigma^2_{x_{\text{attn}}} &= \frac{d^2\sigma_o^2\sigma_v^2 * r^l_{x_{\text{in}}}}{(1-p)} \\
\sigma^2_{x_{\text{ffn}}} &= \frac{2d^2\sigma^2_{w_1}\sigma^2_{w_2}}{(1-p)} \\
\sigma^2_{x_{\text{layer}}} &= \sigma^2_{x_{\text{in}}} + \sigma^2_{x_{\text{attn}}} + \sigma^2_{x_{\text{ffn}}} \\
&= \sigma^2_{x_{\text{in}}} + \frac{d^2\sigma_o^2\sigma_v^2 * r^l_{x_{\text{in}}}}{(1-p)} + \frac{2d^2\sigma^2_{w_1}\sigma^2_{w_2}}{(1-p)} \\
\text{Let, } C_1 &= \frac{d^2\sigma_o^2\sigma_v^2}{(1-p)}, C_2 = \frac{2d^2\sigma^2_{w_1}\sigma^2_{w_2}}{(1-p)bu}, \\
\text{Then, } \sigma^2_{x_{\text{layer}}} &= \sigma^2_{x_{\text{in}}} + C_1 * r^l_{x_{\text{in}}} + C_2
\end{aligned}
$$

As we discuss in Section 3.4, the correlation $r^l_{x_{\text{in}}}$ quickly reaches a stable constant maximum value $r^l_{x_{\text{max}}}$, which can be found using the calculations in Appendix F. Let $r^l_{x_{\text{min}}} > 0$ be the minimum value of this correlation, let $C_3 = C_1 * r^l_{x_{\text{max}}} + C_2$, and $C_4 = C_1 * r^l_{x_{\text{min}}} + C_2$. Then,

$$
\sigma^2_{x_{\text{in}}} + C_4 \leq \sigma^2_{x_{\text{layer}}} \leq \sigma^2_{x_{x_{\text{in}}}} + C_3
$$

Hence after N layers,

$$
\sigma^2_{x_{\text{in}}} + N * C_4 \leq \sigma^2_{x_{\text{model}}} \leq \sigma^2_{x_{\text{in}}} + N * C_3
$$

$$\implies \sigma^2_{x_{\text{model}}} = \Theta(N) \tag{2}$$

This shows that output variance of Pre-LN will increase linearly with number of layers N.

In practice, because the correlation quickly reaches $r^l_{x_{\text{max}}}$, the variance of the entire model $\sigma^2_{x_{\text{model}}} \approx \sigma^2_{x_{\text{in}}} + N * C_3$.

Discussion: This has the effect that transformer blocks near the output can affect the model output much less, as the skip connection variance increases but block output variance is constant. We conjecture that parameters in these are hence not being utilized to their full potential. Specifically in case of Xavier initialization, $C_1 = 2.2, C_2 = 0.4, r^l_{x_{\text{max}}} = 0.85$. For large d, $\sigma^2_{x_{\text{in}}}$ will be negligibly small compared to $\sigma^2_{x_{\text{layer}}}$, so we have -

$$\sigma^2_{x_{\text{model}}} \approx C_3 * N \approx (2.2 * 0.85 + 0.4)N \approx 2.2N$$

E.1.2 Backward Pass

For the backward pass, a Transformer Pre-LN gradient will first backpropagate through the FFN block, then gets rescaled by Layernorm, and added with the skip connection. It then backpropagates through the Attention block, gets rescaled by Layernorm, and finally added with the skip connection. Let $\sigma^2_{g,n}$ be the gradient variance backpropagating from the n^{th} layer, and $\sigma^2_{g_{\text{model}}}$ be the gradient variance after the entire model of N layers.

For the Attention block, let $\sigma^2_{g_{\text{attn}},n-1}$ be the gradient backpropagating from the block. Then for long sequence length L we have -

$$\begin{aligned}
\sigma^2_{g_{\text{attn}},n-1} &= \frac{d^2\sigma_o^2\sigma_v^2 * \sigma^2_{g_{\text{out},n}}}{L(1-p)}(1 + (L-1)r^l_{g_{\text{out},n}}) \\
&\approx \frac{d^2\sigma_o^2\sigma_v^2 * r^l_{g_{\text{out},l}} * \sigma^2_{g_{\text{out},n}}}{(1-p)}
\end{aligned}$$

$\sigma^2_{g_{\text{attn}},n-1}$ is then rescaled by the Layernorm to give $\sigma^2_{g_{\text{attn-layernorm}},n-1}$. As Layernorm scales gradient by the inverse of the input variance $\sigma^2_{x_{\text{in}},n-1}$, which from the section above, we know is approximately $\sigma^2_{x_{\text{in}},n-1} = C_3 * (n-1)$. Then

$$\begin{aligned}
\sigma^2_{g_{\text{attn}},n-1} &= C_1 * r^l_{g_{\text{out}},n} * \sigma^2_{g_{\text{out}},n} \\
\sigma^2_{g_{\text{attn-layernorm}},n-1} &= \frac{C_1 * r^l_{g_{\text{out}},n} * \sigma^2_{g_{\text{out}},n}}{\sigma^2_{x_{\text{in}},n-1}} \\
&\approx \frac{C_1 * r^l_{g_{\text{out}},n} * \sigma^2_{g_{\text{out}},n}}{C_3 * (n-1)}
\end{aligned}$$

Therefore, the final gradient $\sigma^2_{g_{\text{attn-layer}},n-1}$ after addition with the skip connection is

$$\sigma^2_{g_{\text{attn-layer}},n-1} = (1 + \frac{C_1 * r^l_{g_{\text{out}},n}}{C_3 * (n-1)})\sigma^2_{g_{\text{out}},n}$$

Similarly, we can get $\sigma^2_{g_{\text{ffn-layer}},n-1}$ for the ffn block. Then to get the gradient backpropagated through the entire layer $\sigma^2_{g_{\text{out}},n-1}$, we have,

$$\sigma^2_{g_{\text{ffn-layer}},n-1} = (1 + \frac{C_2}{C_3 * (n-1)})\sigma^2_{g_{\text{out}},n}$$

$$\sigma^2_{g_{\text{out}},n-1} = (1 + \frac{C_1 * r^l_{g_{\text{out}},n}}{C_3 * (n-1)})(1 + \frac{C_2}{C_3 * (n-1)})\sigma^2_{g_{\text{out}},n}$$

$$\sigma^2_{g_{\text{out}},n-1} \approx (1 + \frac{C_1 * r^l_{g_{\text{out}},n}}{C_3 * (n-1)} + \frac{C_2}{C_3 * (n-1)})\sigma^2_{g_{\text{out}},n}$$

$$= (1 + \frac{C_1 * r^l_{g_{\text{out}},n} + C_2}{C_3 * (n-1)})\sigma^2_{g_{\text{out}},n}$$

$$= (1 + \frac{C_1 * r^l_{g_{\text{out}},n} + C_2}{(C_1 * r^l_{x_{\text{in}},n} + C_2) * (n-1)})\sigma^2_{g_{\text{out}},n}$$

$$= (1 + \frac{C_{g_{pre},n}}{n-1})\sigma^2_{g_{\text{out}},n}$$

Where we ignore higher order terms for large n, and define $C_{g_{pre},n} = \frac{C_1 * r^l_{g_{\text{out}},n} + C_2}{C_1 * r^l_{x_{\text{in}},n} + C_2}$.

Since $C_{g_{pre},n} > 0$, we will witness an increase in gradient going backward, and this increase is inversely proportional to the current layer n, matching with empirically observed growth (Figure 2).

Let $C_{g_{pre},min} = \frac{C_2}{C_1 + C_2} = 0.15$ be the minimum value of $C_{g_{pre},n}$, and $C_{g_{pre},max} = \frac{C_1 + C_2}{C_2} = 6.5$ be the maximum. Then the above equation is bounded by:

$$(1 + \frac{C_{g_{pre},min}}{n-1})\sigma^2_{g_{\text{out}},n} \leq \sigma^2_{g_{\text{out}},n-1} \leq (1 + \frac{C_{g_{pre},min}}{n-1})\sigma^2_{g_{\text{out}},max}$$

Applying the above equation repeatedly until the final layer N, this recurrence can be approximately solved by treating $\sigma^2_{g_{\text{out}},n}$ as a continuous function of n, taking logarithm of both sides, and integrating. This gives the following solution for $\sigma^2_{g_{\text{out}},n}$:

$$\sigma^2_{g_{\text{out}},N} * (\frac{N}{n})^{C_{g_{pre},min}} \leq \sigma^2_{g_{\text{out}},n} \leq \sigma^2_{g_{\text{out}},N} * (\frac{N}{n})^{C_{g_{pre},max}}$$

If the correlation $r^l_{g_{\text{out}},n}$ quickly reaches a stable constant maximum value $r^l_{g_{\max}}$ (approximately equal to but slightly less than $r^l_{x_{\max}}$ (Appendix F)), $C_{g_{pre}} \approx 1$, and we get exactly hyperbolic growth as shown below:

$$\sigma^2_{g_{\text{out}},n} = \sigma^2_{g_{\text{out}},N} * (\frac{N}{n})$$

The gradient variance will increase hyberbolically with number of layers N while going backwards.

Discussion: This has the effect that much lower learning rate is required for the entire model, because the gradients near the input layers are much higher, slowing down learning and making the model unstable.

E.2 Vanilla Post-LN

E.2.1 Forward Pass

The forward pass of Post-LN is trivially always 1 at initialization, because the skip connection does not cross the LayerNorm.

E.2.2 Backward Pass

Following an analysis similar to that for Pre-LN, we get

$$\sigma^2_{g_{\text{ffn-layer}},n-1} = \frac{1 + C_2}{1 + C_1 * r^l_{x_{\text{out}},n-1}}\sigma^2_{g_{\text{out}},n}$$

$$\sigma^2_{g_{\text{attn-layer}},n-1} = \frac{1 + C_1 * r^l_{g_{\text{out}},n}}{1 + C_2}\sigma^2_{g_{\text{out}},n}$$

$$\sigma^2_{g_{\text{out}},n-1} = \frac{1 + C_1 * r^l_{g_{\text{out}},n}}{1 + C_2} * \frac{1 + C_2}{1 + C_1 * r^l_{x_{\text{out}},n-1}} * \sigma^2_{g_{\text{out}},n}$$

$$= \frac{1 + C_1 * r^l_{g_{\text{out}},n}}{1 + C_1 * r^l_{x_{\text{out}},n-1}}\sigma^2_{g_{\text{out}},n}$$

Let $C_{5,n} = \frac{1+C_1*r^l_{g_{\text{out}},n}}{1+C_1*r^l_{x_{\text{out}},n-1}}$. As we discuss in Appendix F, the correlations both quickly reach a maximum stable value. But the $r^l_{g_{\text{out}},n}$'s maximum value $r^l_{g_{\text{max}}}$ is slightly different than $r^l_{x_{\text{max}}}$. Let $C_5 = \frac{1+C_1*r^l_{g_{\text{max}}}}{1+C_1*r^l_{x_{\text{max}}}}$, then C_5 can be either greater or smaller than 1. Hence, we get

$$\sigma^2_{g_{\text{attn-layer}},n-1} = C_{5,n}\sigma^2_{g_{\text{out}},n}$$

$$= \prod_{i=n}^{N} C_{5,i}\sigma^2_{g_{\text{out}},N}$$

$$\approx C_5^{(N-n)}\sigma^2_{g_{\text{out}},N}$$

$$\sigma^2_{g_{\text{attn-layer}},n-1} = C_5^{(N-n)}\sigma^2_{g_{\text{out}},N} \tag{3}$$

This shows that gradient variance of Post-LN will decrease/increase exponentially with number of layers N while going backwards. Even very slightly different value of C_5 from 1, such as 0.96, will cause a $2000x$ fall in gradient after 200 layers.

Discussion: This shows why Post-LN transformer is much more difficult to train for deeper models than Pre-LN. While for Pre-LN the backwards gradient increases hyber-bolically to a maximum of N, in Post-LN the gradient can increase or decrease exponentially, stopping the model from converging.

E.3 DeepScaleLM Pre-LN

E.3.1 Forward Pass

In DeepScaleLM, the weight initialization are chosen specifically so that $\sigma^2_{x_{\text{attn}}}$ and $\sigma^2_{x_{\text{ffn}}}$ are both equal to 1 for all layers, by iteratively calculating $r^l_{x_{\text{in}}}$ as detailed in Appendix M. Also, the embeddings are initialized so that $\sigma^2_{x_{\text{in}}}$ is also 1. Hence,

$$\sigma^2_{\text{layer}} = \lambda^2 * \sigma^2_{\text{skip}} + \beta^2 * \sigma^2_{\text{block}}$$

$$= \lambda^2 + \beta^2 = 1$$

Hence the forward pass variance remains 1 throughout the model.

E.3.2 Backward Pass

For the FFN-block, we have $\sigma^2_{x_{\text{in}},n-1} = \sigma^2_{x_{\text{out}},n-1} = 1$, as per equations in Table 2 of the main paper.

Similar to Vanilla-PreLN, we arrive at

$$\sigma^2_{g_{\text{attn-layernorm}},n-1} = \frac{C_1 * r^l_{g_{\text{out}},n} * \sigma^2_{g_{\text{out}},n}}{\sigma^2_{x_{\text{in}},n-1}}$$

Here, $\sigma^2_{x_{\text{in}},n-1} = 1$ as shown above, and since weights are initialized so that $C1 * r^l_{x_{\text{in}}} = 1$. Let $C_{6,n} = \frac{r^l_{g_{\text{out}},n}}{r^l_{x_{\text{out}},n-1}}$:

$$\sigma^2_{g_{\text{attn-layernorm}},n-1} = \frac{r^l_{g_{\text{out},n}}}{r^l_{x_{\text{in},n-1}}} * \sigma^2_{g_{\text{out}},n}$$
$$= C_{6,n} * \sigma^2_{g_{\text{out}},n}$$

Therefore, assuming no covariance between block gradients and skip connection (which will be true at initialization), the final gradient $\sigma^2_{g_{\text{attn-layer}},n-1}$ after addition with the skip connection is

$$\begin{aligned}\sigma^2_{g_{\text{attn-layer}},n-1} &= \lambda^2\sigma^2_{g_{\text{out}},n} + \beta^2\sigma^2_{g_{\text{attn-layernorm}},n-1} \\ &= \lambda^2\sigma^2_{g_{\text{out}},n} + \beta^2 C_{6,n}\sigma^2_{g_{\text{out}},n} \\ &= (\lambda^2 + \beta^2 C_{6,n}) * \sigma^2_{g_{\text{out}},n} \\ &= (1 + \frac{C_{6,n}-1}{N}) * \sigma^2_{g_{\text{out}},n}\end{aligned}$$

Similarly for the FFN layer, $\sigma^2_{g_{\text{ffn-layer}},n-1} = \sigma^2_{g_{\text{out}},n}$, as $\sigma^2_{x_{\text{in}},n-1} = \sigma^2_{x_{\text{out}},n-1} = 1$.

Hence,

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &= (1 + \frac{C_{6,n}-1}{N}) * \sigma^2_{g_{\text{out}},n}, \\ \sigma^2_{g_{\text{out}},1} &= \prod_{i=1}^{N}(1 + \frac{C_{6,n}-1}{N}) * \sigma^2_{g_{\text{out}},N}, \\ &\approx \prod_{i=1}^{N}(1 + \frac{C_6-1}{N}) * \sigma^2_{g_{\text{out}},N}, \\ &\approx (1 + \frac{C_6-1}{N})^{N-1} * \sigma^2_{g_{\text{out}},N}, \\ &= e^{C_6-1} * \sigma^2_{g_{\text{out}},N} \\ &\approx \sigma^2_{g_{\text{out}},N}\end{aligned}$$

, where we applied $(1 - \frac{k}{N})^N \approx e^{-k}$, and $C_6 \approx 1$.

Discussion: Hence for DeepScaleLM, the backward variance of gradient remains constant (bounded by a constant) across all layers.

E.4 DeepScaleLM Post-LN

E.4.1 Forward Pass

Same as vanilla Post-LN, this will remain preserved at 1.

E.4.2 Backward Pass

Following an analysis similar to that for Vanilla Post-LN, we get

$$\sigma^2_{g_{\text{ffn-layer}},n-1} = \sigma^2_{g_{\text{out}},n}$$
$$\sigma^2_{g_{\text{attn-layer}},n-1} = (\lambda^2 * 1 + \beta^2 * C_1 * r^l_{g_{\text{out}},n})\sigma^2_{g_{\text{out}},n}$$

$$= (\lambda^2 + \beta^2 * \frac{r^l_{g_{\text{out}},n}}{r^l_{x_{\text{in}},n}})\sigma^2_{g_{\text{out}},n}$$

$$\sigma^2_{g_{\text{out}},n-1} = (\lambda^2 + \beta^2 * \frac{r^l_{g_{\text{out}},n}}{r^l_{x_{\text{in}},n}})\sigma^2_{g_{\text{out}},n}$$

Similar to Pre-LN, we use the maximum value of these correlations, and assume $C_6 = 1$. We get

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &= (\lambda^2 + \beta^2 * \frac{r^l_{g_{\text{max}}}}{r^l_{x_{\text{max}}}})\sigma^2_{g_{\text{out}},n} \\ &= (\lambda^2 + \beta^2 C_6)\sigma^2_{g_{\text{out}},n} \\ &\approx (\lambda^2 + \beta^2)\sigma^2_{g_{\text{out}},n} \\ &= \sigma^2_{g_{\text{out}},n}\end{aligned}$$

Hence for DeepScaleLM, the backward variance of gradient remains constant across all layers.

Discussion: Similar to DeepScale-LM Pre-LN, the assumption $C_6 = 1$ is not required, and yields the same constant bound if we do not assume it to be 1.

E.5 DeepScaleLM (Simplified) Pre-LN

E.5.1 Forward Pass

For simplified DeepScaleLM, the initialization for the FFN block does not change, so its output remains 1 same as DeepScaleLM. For the Attention block, we changed its initialization to mimic that of the FFN block. We will show that initially, simplified DeepScaleLM's forward pass is bounded.

$\sigma^2_{x_{\text{ffn}}} = 1$ as DeepScaleLM, $\sigma^2_{x_{\text{attn}}} = \frac{r^l_{x_{\text{in}}}}{2}$. Therefore, the output variance after layer n will be

$$\begin{aligned}\sigma^2_{x_{\text{attn-skip}},n} &= \lambda^2 * \sigma^2_{x_{\text{layer},n-1}} + \beta^2 * \sigma^2_{x_{\text{attn}}} \\ &= (1 - \frac{2}{N}) * \sigma^2_{x_{\text{layer},n-1}} + \frac{1}{N} * r^l_{x_{\text{in}}}\end{aligned}$$

Similarly after the FFN block, the output skip will be -

$$\begin{aligned}\sigma^2_{x_{\text{layer}},n} &= \lambda^2 * \sigma^2_{x_{\text{attn-skip}},n} + \beta^2 * \sigma^2_{x_{\text{ffn}}} \\ &= (1 - \frac{2}{N}) * ((1 - \frac{2}{N}) * \sigma^2_{x_{\text{layer},n-1}} + \frac{1}{N} * r^l_{x_{\text{in}}}) + \frac{2}{N} * 1 \\ &= (1 - \frac{2}{N})^2 * \sigma^2_{x_{\text{layer},n-1}} + (1 - \frac{2}{N}) * \frac{1}{N} * r^l_{x_{\text{in}}} + \frac{2}{N}\end{aligned}$$

As correlation coefficient $r^l_{x_{\text{in}}} \leq 1$, we get,

$$\begin{aligned}\sigma^2_{x_{\text{layer}},n} &\leq (1 - \frac{2}{N})^2 * \sigma^2_{x_{\text{layer},n-1}} + (1 - \frac{2}{N}) * \frac{1}{N} * 1 + \frac{2}{N} \\ &= (1 - \frac{2}{N})^2 * \sigma^2_{x_{\text{layer},n-1}} + \frac{3}{N} - \frac{2}{N^2} \\ &\leq (1 - \frac{2}{N})^2 * \sigma^2_{x_{\text{layer},n-1}} + \frac{3}{N}\end{aligned}$$

Applying the above recurrence equation N times, we get

$$\sigma^2_{x_{\text{layer},N}} \leq (1 - \frac{2}{N})^{2N} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{N} * \sum_{i=0}^{N} (1 - \frac{2}{N})^{2i}$$

$$= (1 - \frac{2}{N})^{2N} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{N} * \frac{1 - (1 - \frac{2}{N})^{2N}}{1 - (1 - \frac{2}{N})^2}$$

Since $\lambda^2 + \beta^2 = 1$ and β^2 is small for large N. We can rewrite the above equations completely in terms of β as follows

$$\sigma^2_{x_{\text{layer},N}} = (1 - \beta^2)^{2N} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{2}\beta^2 * \frac{1 - (1 - \beta^2)^{2N}}{1 - (1 - \beta^2)^2} \tag{4}$$

$$\approx (1 - \beta^2)^{2N} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{4}(1 - (1 - \beta^2)^{2N}) \tag{5}$$

For large N, we know $(1 - \frac{k}{N})^N \approx e^{-k}$. So the above becomes -

$$\begin{aligned}
\sigma^2_{x_{\text{layer},N}} &\approx e^{-4} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{N} * \frac{1 - e^{-4}}{\frac{4}{N} - \frac{4}{N^2}} \\
&\leq e^{-4} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{N} * \frac{1 - e^{-4}}{\frac{4}{N}} \\
&= e^{-4} * 1 + \frac{3}{4} * (1 - e^{-4}) \\
&= \frac{3}{4} + \frac{1}{4e^4}
\end{aligned}$$

This gives us an upper bound on the output variance after N layers. By setting $r^l_{x_{\text{in}}} = 0$ instead of 1 in the equation above, and proceeding similarly, we can also arrive at a lower bound of $\frac{1}{2} + \frac{1}{2e^4}$.

$$\frac{1}{2} + \frac{1}{2e^4} \leq \sigma^2_{x_{\text{layer},N}} \leq \frac{3}{4} + \frac{1}{4e^4} \tag{6}$$

Discussion Informally, this is because the attention block output variance will be between 0 and 0.5, and ffn block output always 1. Because of our λ, β scaling, the output will slowly converge to be in between the two outputs.

Note that the above derivation assumes no correlation between the block output and the skip connection. As we mentioned in our main paper, we do observe correlation between the input and the output. As such, theoretically, after every block, the variance $\sigma^2_{x_{\text{layer},n}}$ can increase by $\sigma^2_{x_{\text{block}}} + \sqrt{\sigma^2_{x_{\text{layer},n}}}$. This will cause the final output variance to increase by factors of $2 * \sqrt{N}$. In practice however, we observe the output variances to not grow too large.

E.5.2 Backward Pass

Similar to DeepScaleLM Pre-LN, we arrive at

$$\begin{aligned}
\sigma^2_{g_{\text{attn-layernorm}},n-1} &= \frac{C_1 * r^l_{g_{\text{out}},n} * \sigma^2_{g_{\text{out}},n}}{\sigma^2_{x_{\text{in}},n-1}} \\
&\approx \frac{0.5 * C_6}{\sigma^2_{x_{\text{in}},n-1}} * \sigma^2_{g_{\text{out}},n}
\end{aligned}$$

$$\begin{aligned}
\sigma^2_{g_{\text{attn-layer}},n-1} &= \lambda^2 \sigma^2_{g_{\text{out}},n} + \beta^2 \sigma^2_{g_{\text{attn-layernorm}},n-1} \\
&= (\lambda^2 + \beta^2 * \frac{0.5 * C_6}{\sigma^2_{x_{\text{in}},n-1}}) * \sigma^2_{g_{\text{out}},n} \\
&= (1 + \frac{2}{N} * (\frac{0.5 * C_6}{\sigma^2_{x_{\text{in}},n-1}} - 1)) * \sigma^2_{g_{\text{out}},n}
\end{aligned}$$

Similarly, for the FFN layer, we get

$$\sigma^2_{g_{\text{ffn-layer}},n-1} = (1 + \frac{2}{N} * (\frac{1}{\sigma^2_{x_{\text{in}},n-1}} - 1)) * \sigma^2_{g_{\text{out}},n}$$

Multiplying these, we get

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &= (1 + \frac{2}{N} * (\frac{0.5 * C_6}{\sigma^2_{x_{\text{in}},n-1}} - 1)) * (1 + \frac{2}{N} * (\frac{1}{\sigma^2_{x_{\text{in}},n-1}} - 1)) * \sigma^2_{g_{\text{out}},n} \\ &\approx (1 + \frac{2}{N} * (\frac{0.5 * C_6}{\sigma^2_{x_{\text{in}},n-1}} + \frac{1}{\sigma^2_{x_{\text{in}},n-1}} - 2)) * \sigma^2_{g_{\text{out}},n}\end{aligned}$$

As $0.5 \leq \sigma^2_{x_{\text{in}},n-1}$, we get $-4 \leq (\frac{C_6}{\sigma^2_{x_{\text{in}},n-1}} + \frac{2}{\sigma^2_{x_{\text{in}},n-1}} - 4) \leq 2C_6 + 2$. Hence, on applying the above recurrence N times, we get

$$e^{-4} * \sigma^2_{g_{\text{out}},N} \leq \sigma^2_{g_{\text{out}},n-1} \leq e^{2C_6+2} * \sigma^2_{g_{\text{out}},N}$$

Hence, we show that even for simplified DeepScaleLM Pre-LN, the maximum relative increase/fall in gradient variance is bounded across layers.

Discussion: The above derivations will also be valid if there is correlation in the input. Correlation will cause $\sigma^2_{x_{\text{in}},n-1}$ to increase, effectively decreasing the backpropagated gradient through the block to decrease (as Layernorm will scale by inverse of $\sigma^2_{x_{\text{in}},n-1}$). However, even in that case, our gradient will still be bounded by the above lower-bound.

Intuitively, as the gradient can flow freely through the skip connection, hence, $\sigma^2_{g_{\text{out}},n-1} \geq \lambda^4 * \sigma^2_{g_{\text{out}},n}$, which when applied N times, yields $\sigma^2_{g_{\text{out}},1} \geq e^{-4} * \sigma^2_{g_{\text{out}},N}$

E.6 DeepScaleLM (Simplified) Post-LN

E.6.1 Forward Pass

The forward pass variance for Post-LN is trivially bounded.

E.6.2 Backward Pass

Following an analysis similar to that for DeepScaleLM Post-LN, we get

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &= \frac{\lambda^2 + 0.5 * \beta^2 * r^l_{g_{\text{out}},n}}{\lambda^2 + 0.5 * \beta^2 * r^l_{x_{\text{in}},n}} \sigma^2_{g_{\text{out}},n} \\ &= \frac{1 + \frac{2}{N}(0.5 r^l_{g_{\text{out}},n} - 1)}{1 + \frac{2}{N}(0.5 r^l_{x_{\text{in}},n} - 1)} \sigma^2_{g_{\text{out}},n}\end{aligned}$$

Applying taylor expansion, we get,

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &\approx (1 + \frac{2}{N}((0.5 r^l_{g_{\text{out}},n} - 1) - (0.5 r^l_{x_{\text{in}},n} - 1)))\sigma^2_{g_{\text{out}},n} \\ &= (1 + \frac{1}{N}(r^l_{g_{\text{out}},n} - r^l_{x_{\text{in}},n}))\sigma^2_{g_{\text{out}},n}\end{aligned}$$

The above equation can be rewritten in terms of β as follows

$$\sigma^2_{g_{\text{out}},n-1} = (1 + \frac{\beta^2}{2}(r^l_{g_{\text{out}},n} - r^l_{x_{\text{in}},n}))\sigma^2_{g_{\text{out}},n} \tag{7}$$

As $-2 \leq (r^l_{g_{\text{out}},n} - r^l_{x_{\text{in}},n}) \leq 2$, applying the above recurrence N times we get

$$e^{-2} * \sigma^2_{g_{\text{out}},N} \leq \sigma^2_{g_{\text{out}},n-1} \leq e^2 * \sigma^2_{g_{\text{out}},N}$$

Discussion: The above derivations assume no correlation in the input, and hence is only correct at initialization. However, if there is correlation between the block output and skip connection (r_x), the layernorm will cause $\sigma^2_{g_{\text{out}},n-1}$ to be down-scaled by a factor of $1 + \frac{2*r_x}{\sqrt{N}}$, where c is some constant, as opposed to $1 + \frac{2}{N}$ above. However, if there is also correlation in the gradients of the block and skip connection (r_g), the numerator in the equations above for $\sigma^2_{g_{\text{out}},n-1}$ will also be increased, by a factor of $1 + \frac{2*r_g}{\sqrt{N}}$. Hence if the correlations among the gradients and among the output are similar, the above bounds will remain. If β^2 is set as $\frac{1}{N^2}$, then even if input correlations exist, the backward gradient will be bounded, following a similar derivation as above. However, we conjecture that this decreases the ability of the transformer layers to modify the skip connection too strongly, decreasing the "expressivity" of the model. This is similar to the approach of DSInit, which we show in our main paper does indeed decrease model performance.

F Rank Collapse and Correlation Analysis

In the previous sections, we derived the formulas that determine how the correlation will change through the Attention and FFN blocks both for forward and backward pass. Both attention and FFN blocks modify the correlation as shown in the Table 2.

Simplifying the formulae in the table above, we rewrite the output variance for the attention block as $\sigma^2_{x_{\text{attn}}} = C_1 * r^l_{x_{\text{in}}} * \sigma^2_{x_{\text{in}}}$, and the output of the FFN block is $\sigma^2_{x_{\text{ffn}}} = C_2 * \sigma^2_{x_{\text{in}}}$, where C_1 and C_2 are defined as follows.

$$C_1 = \frac{d^2\sigma_o^2\sigma_v^2}{(1-p)}, C_2 = \frac{2d^2\sigma^2_{w_1}\sigma^2_{w_2}}{(1-p)},$$

This also helps us to rewrite the backward pass as the $\sigma^2_{g_{\text{attn}}} = C_1 * r^l_{g_{\text{out}}} * \sigma^2_{g_{\text{out}}}$ and $\sigma^2_{g_{\text{ffn}}} = C_2 * \sigma^2_{g_{\text{out}}}$.

Specifically in case of Xavier initialization with 0.1 dropout, $C_1 = 2.2, C_2 = 0.4$.

Assuming a dropout of 0.1, the FFN block (with the ReLU) will reduce the correlation if it rises above 0.64 (where $r^l_{x_{\text{out}}} < r^l_{x_{\text{in}}}$ for FFN block). And the attention block will never output a correlation higher than 0.9. Hence correlation will never reach 1, but rather a steady, stable value between ReLU's maximum correlation and that of the attention block. Dropout's effect in preventing rank collapse was also observed in (Rong et al., 2020).

We can approximate the stable value of correlation after many layers based on the weightage average of the correlation in the Attention output and FFN output. When the attention output is added to the skip connection, the new correlation will be a weighted (by variance) average of the correlation among the tokens of attention output and among the tokens in the skip connection. And the same will happen after the FFN block.

A weighted average of the correlations of FFN and attention blocks gives the stable asymptotic correlation $r^l_{x_{\text{max}}}$

$$r^l_{x_{\text{max}}} = \frac{C_1 * (1-p) + C_2 * (1-p)(\frac{1}{\pi} + \frac{r^l_{x_{\text{max}}}}{2} + (\frac{1}{2} - \frac{1}{\pi}){r^l_{x_{\text{max}}}}^2)}{C_1 + C_2}$$

Specifically for the case of xavier initialization, solving the above equation with $C_1 = 2.2, C_2 = 0.4$, gives $r^l_{x_{\text{max}}} \approx 0.88$, as empirically verified in Figure 8.

Similarly, the correlation for backward gradient will also converge at a stable value $r^l_{g_{\text{max}}}$, obtained by solving the below equation -

$$r^l_{g_{\text{max}}} = \frac{C_1 * (1-p) + C_2 * (1-p)(\frac{1}{2} + \frac{\sin^{-1}(r^l_{x_{\text{max}}})}{\pi})r^l_{g_{\text{max}}}}{C_1 + C_2}$$

Specifically for the case of xavier initialization, this gives $r^l_{g_{\text{max}}} = 0.87$. Note how $r^l_{g_{\text{max}}} \approx r^l_{x_{\text{max}}}$.

Discussion on rank collapse observed in Noci et al. (2022) Noci et al. (2022) focuses primarily on linear activation, we theoretically analyze the change in output correlation caused by ReLU. We find that ReLU (or any asymmetric non-linearity in general) critically affects correlation. As our closed form expressions suggest, both FFN block (because of ReLU) and dropout reduce the correlation. While Noci et al. (2022) mentions the use of dropout, as we show above and observe empirically in Figure 8, rank will not collapse with dropout, and perhaps Noci et al. (2022) did not use dropout.

We replicated the experimental settings of Noci et al. (2022) without dropout, and observed that the rank collapse occurs due to incorrect initialization. They use a rather non-standard version of xavier initialization - instead of $\frac{2}{fan_{in}+fan_{out}}$, they use $\frac{1}{fan_{out}}$. Hence, they initialize a much higher value for V as fan_{in} is much greater than fan_{out} ("Number of heads" times greater), and this results in variance of the output of the attention block $C1$ being much higher than FFN $C2$. As attention block outputs a much higher correlation than the FFN block, increasing its output variance without using dropout will result in rank collapse. This highlights the criticality of correct initialization, as well as the explainability power of our theoretical framework proposed in the paper.

G Discussion of Relative Strength

In Equation 4, we discussed that the backward recurrence equation for PreLN can be written as

$$\sigma^2_{x_{\text{layer},N}} \approx (1-\beta^2)^{2N} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{4}(1-(1-\beta^2)^{2N})$$

Replacing $\beta^2 = \frac{k}{N^\alpha}$ and using $(1+\frac{k}{N^\alpha})^N = e^{kN^{1-\alpha}}$, we get

$$\begin{aligned}\sigma^2_{x_{\text{layer},N}} &\approx e^{2cN^{1-\alpha}} * \sigma^2_{x_{\text{layer},0}} + \frac{3}{4}(1-e^{2cN^{1-\alpha}}) \\ &= e^{2cN^{1-\alpha}} * (\sigma^2_{x_{\text{layer},0}} - \frac{3}{4}) + \frac{3}{4}\end{aligned}$$

Hence, the fall in gradient for $\beta^2 = \frac{k}{N^\alpha}$ is $\mathcal{O}(e^{kN^{1-\alpha}})$.

Similarly for PostLN, we can use Equation 7

$$\begin{aligned}\sigma^2_{g_{\text{out}},n-1} &= (1+\frac{\beta^2}{2}(r^l_{g_{\text{out}},n} - r^l_{x_{\text{in}},n}))\sigma^2_{g_{\text{out}},n} \\ (1-\beta^2) * \sigma^2_{g_{\text{out}},N} \leq \sigma^2_{g_{\text{out}},n-1} &\leq (1+\beta^2) * \sigma^2_{g_{\text{out}},N}\end{aligned}$$

Hence, for N layers, the gradient fall/growth is again $\mathcal{O}(e^{\pm kN^{1-\alpha}})$.

H Applying DeepscaleLM to Vision Transformers

Applying our method to vision transformers (for eg. ViT (Dosovitskiy et al., 2021) or DeiT (Touvron et al., 2021a)) will only require handling the input embeddings section Appendix A.1 - For ViT, this is a simple linear projection. Given normalized image inputs, our Linear section Appendix A.2 provides formulae to calculate the variance and correlation of the embeddings which are input to the model.

We empirically verified that for images from ImageNet, the embeddings after the linear projection do indeed follow the normal distribution, with an R^2 of 0.95. Furthermore, normalizing images to have approximately unit variance, given linear weights initialized by $\sqrt{\frac{1}{d}}$, the output variance was observed as 1.02 (within 2% error). While we used Zipf's law to estimate input embedding correlation for text, this could simply be empirically measured for vision after the embedding layer – we measured this to be 0.46 using the code provided by Beyer et al. (2022).

Using this measured value of input correlation, we can apply our DSLM method to ViT. As we show in Figure 11, our method successfully controls both the forward and backward moments for the ViT model with $100s$ of layers.

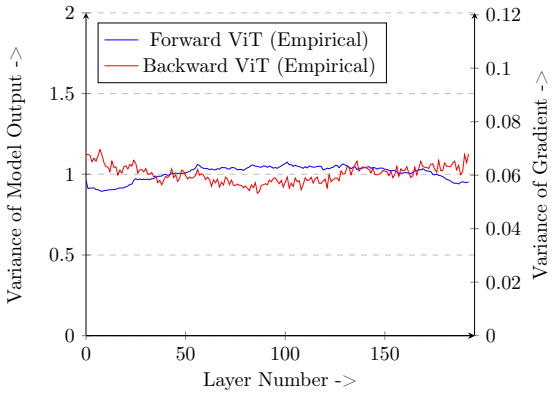

Figure 11. DeepScaleLM: The variances remain conserved for both backward and forwards for ViT, using ImageNet data, after even 192 transformer layers

I Compute

I.1 Theoretical compute

Table 23 provides the exact compute for the models reported in Table 4. We follow the code provided by Electra (Clark et al., 2020) to calculate the each model's compute (FLOPs). We observe that up to 200 layers, the extra compute is within $6-7\%$ of the original shallow model.

Table 23. Model compute with increasing depth (keeping Nd^2 constant).

Layers (N)	Hidden Dim (d)	Params	Compute (Flops)	% Extra
12	1024	185M	1.06e20	-
48	512	168M	1.03e20	-2.5%
192	256	160M	1.12e20	6.3%
784	128	156M	1.38e20	30.6%
24	1024	336M	1.92e20	-
96	512	319M	1.96e20	2.3%
384	128	311M	2.19e20	14.5%

I.2 Wall Clock times

We also compared wall clock time overheads, and found them to not be too large. For example, the 48-layer-512-d model has only 9.8% overhead in wall clock time compared to 12-layer-1024-d model. Even when larger number of layers, such as 96-layer-512-d, the overhead is only 14.9% compared to 24-layer-1024-d model. Profiling revealed majority of the overhead was due to extra latency of added GPU kernel launches. Hence, approaches such as cudaGraphs (which batches kernel launches together) or graph compilation techniques may decrease this overhead further.

This overhead will decrease the bigger the original model size, and become much smaller. For example, for a 5B params model with 24-Layers-4096d (a reasonable shape in contemporary models, for example, LLaMA 7B has 32L-4096D) has much less compute overhead - only 6.6% overhead at 96 layers, and 13.6% overhead at 192 layers.

Despite this wall-clock time overhead, due to large performance gains from increasing depth, the 160M params 192-L model from Table 4 outperforms the vanilla 336M BERT-large 24-L model with $2x$ more params, even at equal wall times.

Furthermore, a large fraction of the performance improvements mentioned happen when increasing the number of model layers by $4x$ - and as shown above, the wall clock time overhead is minimal. Making standard models $4x$ more deep to $50-100$ layers, will provide a large fraction of performance gains without much overhead.

Note that this performance overhead seems to be dependent on the framework used – some frameworks may be less optimized for such deeper models and may incur additional overhead for small but deep models.

J Statistical Significance

J.1 Error Bars for Pre-Training Experiments

In our initial experiments, we observed very little variation in performance across different runs – we conjecture that the model is trained on a large enough number of tokens for differences in initialization/data seed to not matter. We provide mean and standard error for the 12L-1024D Post-LN and DSLM models from Table 4 below:

Table 24. Standard error across runs for pre-training.

Model	Mean	Standard Error
Post-LN Baseline	14.33	0.14
DSLM	15.56	0.08

As the variation was so small, and due to compute limitations, we did not run multiple runs for other experiments thereafter. We also reported the best score for Baseline Post-LN, and the worst score for DSLM for the 12L-1024D models Table 4 for a conservative comparison.

J.2 Statistical Significance for Fine-tuning Experiments

Mean and standard errors for all downstream fine-tuning experiments were reported in Table 6. The differences are statistically significant at $p < 5\%$ for all datasets except QQP.

K Related Works

K.1 Initialization

Several works, such as Glorot & Bengio (2010); He et al. (2015); Brock et al. (2021a) improved the initialization of ResNets/ReLU networks, but crucially these works do not consider the impact of correlation in the input, which is large in Transformer models. Poole et al. (2016) takes correlation into account, and Schoenholz et al. (2017) initializes weights for networks with bounded activations so that correlation reaches 1 asymptotically.

Some works, such as Mishkin & Matas (2016), sequentially profile each layer empirically by running forward passes through the model, and scaling the weights and/or output to achieve unit variance, and Liu et al. (2020a;b) applied the same method for Transformers. Blake et al. (2023) also tries to achieve unit variance, but does not consider correlation in input or across tokens, and ignores the non-zero mean of ReLU. Bachlechner et al. (2021) shows unit variance leads to faster convergence at the start of the training.

We demonstrate that this profiling is unnecessary, and can instead be done theoretically in DeepScaleLM. Furthermore, where output or gradient increases in some prior works with more layers (eg. for ADMIN (Liu et al., 2020a), grad decreases by $\mathcal{O}(N)$ (increases by $\mathcal{O}(log(N))$ for Pre-LN)), our method allows maintaining both unit output and equal gradient across all layers at initialization, and bounded during training.

Yang et al. (2021) proposed μP initialization such that updates to a layer are of the same order regardless of width. Their work was focused on enabling transfer of hyper-parameters across model widtd, and does not target solving pathologies inherent in deeper architectures – they do not model the impact of ReLU and Attention on correlation, and hence are unable to prevent rank-collapse at large depths. When applied to 100s of layers, μP diverges with rank collapse at initialization.

K.2 Signal Propagation

Signal propagation in Neural Networks has a long history, such as Neal (1995); LeCun et al. (1996). More recently, several works have focused on signal propagation for ResNets, such as He et al. (2015); De & Smith (2020); Brock et al. (2021a); Schoenholz et al. (2017); Hoedt et al. (2022); Labatie et al. (2021); Marion et al. (2022); Klambauer et al. (2017); Balduzzi

et al. (2017).

For transformers, signal propagation was studied in Xu et al. (2019); Dong et al. (2021); Davis et al. (2021); Noci et al. (2022). Our work also considers previously neglected effects of dropout, input correlation between tokens, non-linearity, QK initialization, and provides closed forms with verifiable correctness of this signal propagation. Ours is the first work to theoretically constrain the output and gradient to almost exactly unit without any profiling passes, showing the validity of our formulae and of our assumptions.

He et al. (2023) extends neural kernel methods of DKS (Martens et al., 2021) to Transformers to model network behaviour, assuming the MLP to be linear in its effect on attention with respect to correlation. Q/C maps in kernel methods are similar to signal propagation, as expected moments are equivalent to q and m values of kernels (Martens et al., 2021). Our method relaxes these assumptions, and we show that considering the impact of ReLU/GeLU on correlation is critical to correctly modelling attention. In particular, our formulae show that an MLP block with GeLU will also increase correlation in the absence of dropout (the same setting as used in He et al. (2023)). At large depths, He et al. (2023)'s method suffers from rank collapse (with their deeper models under-performing shallower ones), which our method successfully prevents.

We also account for cases with non-IID inputs that may occur due to segment/position embeddings or due to non-uniform token distributions in real data (that are distributed approximately per Zipf's law Kingsley (1935)) – and find that this strongly affects output variance of the attention block.

K.3 Moment Control & Residual Scaling

Bounded gradients, or normalizing per-layer gradients, have been shown to results in better/faster convergence (Shen et al., 2020; Yu et al., 2017; You et al., 2017; 2020). Woks such as Takase et al. (2022); Shleifer et al. (2021); Hayou et al. (2019) also achieved improved training by empirically mitigating the gradient explosion.

Scaling with $\lambda^2 + \beta^2 = 1$ to control moments have often been used for ResNets (Balduzzi et al., 2017; Szegedy et al., 2017; Hanin & Rolnick, 2018; Arpit et al., 2019; Zhang et al., 2019b; Hoedt et al., 2022). Szegedy et al. (2017) proposed to use any small β, Balduzzi et al. (2017) proposed to set $\beta^2 = 0.5$, Bachlechner et al. (2021) sets $\beta = 0$ and learnable. De & Smith (2020) showed that $\lambda^2 = 0.5$ is not sufficient to solve vanishing gradients.

$\beta^2 = \frac{k}{N}$ was used to control growth of moments in Arpit et al. (2019); Brock et al. (2021a); Marion et al. (2022); Zhang et al. (2022b); Noci et al. (2022); He et al. (2023); Yang et al. (2024)) . $\beta^2 = \frac{k}{n}$, where n is the current layer, was used in De & Smith (2020); Liu et al. (2020a;b); Davis et al. (2021); Blake et al. (2023), but this results in logarithmic bounds instead of constant for forward propagation if $\lambda = 1$ is used, and vanishing gradient for backward propagation otherwise.

Values of $\beta^2 < \frac{k}{N}$, such as (effectively) $\frac{1}{N^2}$ for DSInit (Zhang et al., 2019a) or $\frac{1}{N^{1.5}}$ for DeepNet (Wang et al., 2024) decrease sensitivity of the model, and may result in the model becoming "too linear". DeepNet shows performance improvements by making the model deeper, but keeping the hidden dimension constant. Our setting is much more strict – we keep the number of parameters (and hence compute) constant, and our method still show performance improves on making the model deeper. For example, DeepNet's 200 layer model is 3.2B params, whereas our 192 layer model is 160M params (20x smaller).

Sometimes, these β values are used in conjunction with $\lambda = 1$, such as in Liu et al. (2020a;b), but as shown in He et al. (2023), fully normalized residual connections with $\lambda^2 + \beta^2 = 1$ often perform better than those with $\lambda = 1$. We also observed lower performance with $\lambda = 1$ in our initial experiments, and hence we fully normalize the residual connections.

Our contribution goes beyond providing an optimal scaling scheme. Using the theoretical framework and closed-form expressions for moment propagation through both Pre-LN and Post-LN developed in this work, practitioners can make informed choices about using any of the scaling factors above based on the stability-performance tradeoffs, such as using a lower β for scenarios with high correlation, or using higher β with uncorrelated inputs.

K.4 Other Network modifications for Deep Networks

Shi et al. (2022); Zhou et al. (2021); Wang et al. (2022); Dong et al. (2021) showed that attention causes rank collapse in deeper models, and Chen et al. (2020); Zhao et al. (2023) showed the same for graphs. Takase et al. (2022) added some extra skip connections from the input of the model, Nguyen & Salazar (2019) modified layernorm slightly, Zhai et al. (2023) normalized all linear layers by their spectral norm, and Shleifer et al. (2021) added extra layer norms. Some works in particular, such as Zhai et al. (2023); Zhou et al. (2021) can only prevent attention entropy collapse later during training,

but our work will also prevent rank collapse at initialization caused by the very structure of the transformer model, in particular increase in correlation caused by both attention and ReLU/GeLU. The methods in these works are orthogonal to our approach, and our equations can be easily extended to cover the architectural modifications suggested in these.

L Discussion of Approximations and Assumptions

L.1 Illustrative Approximations of Full Formulae in Main Paper

Some values listed in Table 1 are approximations/illustrative simplifications of their full closed forms in Appendix C and Appendix A. We discuss all of these below.

- For ReLU forward correlation, we used a simple polynomial regression of the closed form formula. This simple regression is a remarkably good fit, as shown in figure Figure 12, and can be reproduced in using our released code.
- For layernorm, we ignored the factor of 1 compared to d, or $1/d$ compared to 1, assuming large enough hidden dimension d.
- For SHA without V, we used the final simplified formulae for $\sigma^2_{x_{\text{out}}}$ and output correlation from Appendix A.8. For the gradient, we further simplified the formulae in Appendix A.8, assuming $L \approx L - 1$.

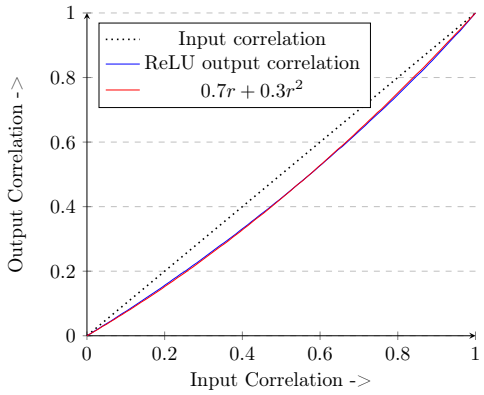

Figure 12. Approximation of the Relu forward correlation formula

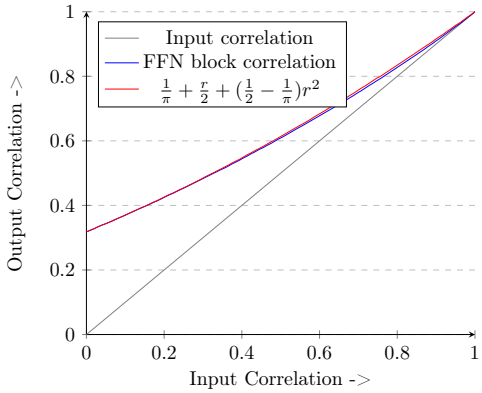

Figure 13. Approximation of the FFN forward correlation formula, without dropout. Dropout will reduce the above correlation by $1 - p$.

Furthermore, the formulae provided in Table 2 are approximate versions of the full formulae provided in Appendix C. In Table 2, we applied a similar approximation as done in Table 1 for ReLU, from the full formula in Appendix C for output correlation. This polynomial approximation is also a very good fit, as shown in Figure 13, and can be reproduced using our released code.

Our exact formulae for blocks and components also account for IID cases - as can be verified by our simulations, in which we do cover cases IID inputs with exactly 0 correlation, as noted in $\text{Corr}^l_{x_{\text{in}}}$ column in Table 22. In the simplified formulae, and in DeepScaleLM initialization and model, we simplified our formulae so that they only remain accurate for non-IID inputs. This was because of three considerations:

1. In NLP domain, most text will inevitably be non-IID due to repeated common words. This was encountered in all our experiments.
2. In Vision domain, for ViT in particular, there will be correlation among pixel intensities across patch embeddings, as discussed in common response section.
3. In Speech domain, similar to text, most speech will inevitably be non-IID due to repeated common sounds.

4. Lastly, even if there is exactly 0 correlation in input, the very first attention layer and the first FFN layer in particular, will add correlations to the output, ensuring our simplified formulae hold reasonably accurately.

L.2 Assumptions and Approximations in Derivations

- Except for attention, softmax and LayerNorm all other derivations of transformer components – Embeddings, FFN, ReLU/GeLU, Dropout, FFN Block are fully exact, assuming only normal distribution of inputs, weights and gradients. We justify this normality assumption below:

  1. **Inputs:** As the embeddings are lookup tables of token-ids, and embedding weights are initialized from Normal distribution in Xavier, the inputs to the transformer are normally distributed.
  2. **Gradients:** As the model outputs are Normal, the softmax of the classification head results in a Log-Normal distribution for probabilities p, as shown in Appendix A.7. Since the cross-entropy loss is $-log(p)$, we expect the loss (and hence the final gradient being back-propagated) being log(Log-Normal distribution), to be a Normal distribution. We also verify this empirically by checking the normality of the backpropagated gradients to the deepest transformer layer, and the gradients match the best-fit Normal distribution with an R^2 of 0.999, showing that the gradients are indeed Normally distributed.
  3. **Weights:** Weights are initialized from Normal distribution in Xavier, and are hence Normal.

- For attention, softmax and LayerNorm, we assume the sequence length L and the hidden dimension d are large.

- For embeddings, we assumed Zipf's law to calculate initial input correlation in tokens, as well as assumed uniform distribution for segment lengths for next sentence prediction task of BERT. Note that this assumption is not strictly required, and can also be empirically observed and given as input to our method.

M DeepScaleLM Pseudocode

```
## Define constants of DeepScaleLM
λ² = 1 − 2/N ; β² = 2/N
σ_e² = (1−p)/3 ; σ_qk² = 1/d ; σ_f² = 1/d * sqrt((1−p)/2)

## Scale skip connection and block output
def add_skip(x, f(x)):
    return λ * x + β * f(x)

## Stable initialization of weights
def init(w):
    if w is ['ffn', 'v_proj', 'out_proj']:
        nn.init.normal_(w, gain = σ_f)
    elif w is ['q_proj', 'k_proj']:
        nn.init.normal_(w, gain = σ_qk)
    elif w is ['embd']:
        nn.init.normal_(w, gain = σ_e)
```

Figure 14. Pseudo-code for simplified version of our DeepScaleLM method.

```
## Define constants for scaling residual and output
λ² = 1 − 2/N ; β² = 2/N
## Define constants for embedding and FFN block
σ_e² = (1−p)/3 ; σ_f² = (1/d) * sqrt((1−p)/2)

## Scale skip connection and block output
def add_skip(x, f(x)):
    return λ * x + β * f(x)

## Find layerwise input correlation upto N layers
def corr_input_layerwise(r, N):
    r_N = []
    for i in range(N):
        r = λ² . r + β²(1 − p)
        r = λ² . r + β²(1 − p)(r^l_{x_in} + ((1 − (r^l_{x_in})²)^0.5)/π − (r^l_{x_in} cos⁻¹(r^l_{x_in}))/π)
        r_N.append(r)
    return r_N

## Define constants for attention block
σ²_{l,o} = (1/d) * sqrt((1−p)/r^{l,n}_{x_in}) ; σ²_{qk} = 1/d ; r = r^l_{x_in}
where r^{l,n}_{x_in} = corr_input_layerwise(r, N)[n]


## Stable initialization of weights
def dslm_init(w, l):
    if w is ['ffn']:
        nn.init.normal_(w, gain = σ_f)
    elif w is ['v_proj', 'out_proj']:
        nn.init.normal_(w, gain = σ_{l,o})
    elif w is ['q_proj', 'k_proj']:
        nn.init.normal_(w, gain = σ_qk)
    elif w is ['embd']:
        nn.init.normal_(w, gain = σ_e)
```

Figure 15. Pseudo-code for our proposed method DeepScaleLM: We scale the block output and the skip connection before adding, and keep track of correlation across layers. We appropriately initialize the weights. (N: num of layers, d: model hidden dimension, p: dropout probability, $r^l_{x_{in}}$ is calculated based on expressions provided in subsection A.1.)

N Hyper-parameters

BERT Pretraining We used Megatron-LM's default BertWordPieceLowerCase tokenizer, with the original BERT lower-cased vocab, and with trainable position embeddings. The same hyper-parameters (including LR schedule, warmup) were used for all models, and LR search over the range below was performed for all models. The final best models always had optimal LR within the range and not at the boundary of the LR range for all of our experiments. Detailed hyper-params are provided in Table 25.

Table 25. Training Hyper-Parameters. We use all original hyper-parameters of BERT, except for learning-rate(LR).

Parameters	**Values**
Optimizer	Adam
β_1, β_2	0.9, 0.999
Effective Batch Size	256
Drop-out (p)	0.1
Sequence Length	256
Train Iters	100,000
Num GPUs	8
Learning rate	[1, 3, 5, 7, 10]*10^{-4}
Schedule	Linear
LR Decay Iterations	98%
Warmup steps	1%
Min LR	$1 * 10^{-5}$
Gradient clipping	1.0
Batch Size / GPU	2
Grad Accum Steps	16

Reproducible Longer Pre-training and Finetuning Our released code provides exact scripts for both pre-training and all fine-tuning. We used all original/official hyper-params of BERT, except LR was increased for DSLM as mentioned previously.

Downstream Low Rank Finetuning Following QLoRA (Dettmers et al., 2023), we apply LoRA on all linear modules, with $r = 32$, $\alpha = 16$, and searched for LR. All other hyper-parameters were kept the same as finetuning. We used the same number of epochs as finetuning for LoRA, but perhaps more epochs may result in even better scores – Hu et al. (2022) used 30 epochs for LoRA.

Vision ViT Training We used ViT-S Baseline from (Beyer et al., 2022) for ImageNet-1k along with its default hyper-parameters. It uses an MLP head, a Global AvgPool and a fixed 2D sin-cos position embeddings. The same hyper-parameters were used for all the models. Detailed hyper-parameters are provided in Table 26

Table 26. Training Hyper-Parameters for ViT Training. We use all original hyper-parameters of (Beyer et al., 2022), except for learning-rate LR.

Parameters	Values
Optimizer	Adam
β_1, β_2	0.9, 0.999
Weight Decay	10^{-4}
Effective Batch Size	1024
Drop-out (p)	0.0
Patch Size	16
Training Image Size	224x224
Evaluation Image Size	224x224
Train Epochs	[90, 300]
Num GPUs	8
Learning rate	[1, 2, 3.5, 4]*10^{-3}
Schedule	Linear
LR Decay Schedule	Cosine
Warmup steps	10000
Min LR	0.0
Gradient clipping	1.0
Batch Size / GPU	128
Augmentation	RandAug(n=2,mag=10)+MixUp(p=0.2)

Speech Fairseq Training Table 27 provides the hyperparameters used to train the Speech translation models, following those of official fairseq. The same hyper-parameters were used for all the models. We report the BLEU by averaging the weights of the last 10 checkpoints at the end of training.

Table 27. Training Hyper-Parameters for speech-to-text translation. We use all original hyper-parameters in Fairseq, except for effective batch size and learning-rate(LR).

Parameters	Values
Optimizer	Adam
β_1, β_2	0.9, 0.999
Source tokens per Batch	[30k, 40k]
Drop-out (p)	0.1
Text Sequence Length	1024
Speech Sequence Length	6000
Train Iters	[66k, 100k]
Num GPUs	1
Learning rate	[3, 5]*10^{-4}, [1, 2, 3, 4]*10^{-3}
Schedule	Inverse Square-root
Warmup steps	20%
Gradient clipping	10.0
Batch Size / GPU	[52, 80]
Grad Accum Steps	8,16

O Notations

Helpful definitions for notations used in this Manuscript.

N - Number of layers in the transformer network

L - Maximum sequence length for the transformer network

d/d_{in} - Hidden dimension used to represent a token

$\mu_{x_{\text{in}}}$ - Expected value of single element in the input tensor to a layer/block

$\sigma^2_{x_{\text{in}}}$ - Variance of single element in the input tensor to a layer/block

$\mu_{x_{\text{out}}}$ - Expected value of single element in the output tensor of a layer/block

$\sigma^2_{x_{\text{out}}}$ - Variance of single element in the output tensor of a layer/block

$\mu_{g_{\text{in}}}$ - Expected value of single element in the gradient of input tensor to a layer/block

$\sigma^2_{g_{\text{in}}}$ - Variance of single element in the gradient of input tensor to a layer/block

$\mu_{g_{\text{out}}}$ - Expected value of single element in the gradient of output tensor of a layer/block

$\sigma^2_{g_{\text{out}}}$ - Variance of single element in the gradient of output tensor of a layer/block

$r^l_{x_{\text{in}}}$ - Correlation between two elements in the input tensor to a layer/block having same hidden dimension index but corresponding to different tokens

$r^l_{x_{\text{out}}}$ - Correlation between two elements in the output tensor of a layer/block having same hidden dimension index but corresponding to different tokens

$r^l_{g_{\text{in}}}$ - Correlation between two elements in the gradient of input tensor to a layer/block having same hidden dimension index but corresponding to different tokens

$r^l_{g_{\text{out}}}$ - Correlation between two elements in the gradient of output tensor of a layer/block having same hidden dimension index but corresponding to different tokens

$r^d_{x_{\text{in}}}$ - Correlation between two elements in the input tensor to a layer/block having different hidden dimension indices but corresponding to same token

$r^d_{x_{\text{out}}}$ - Correlation between two elements in the output tensor of a layer/block having different hidden dimension indices but corresponding to same token

$r^d_{g_{\text{in}}}$ - Correlation between two elements in the gradient of input tensor to a layer/block having different hidden dimension indices but corresponding to same token

$r^d_{g_{\text{out}}}$ - Correlation between two elements in the gradient of output tensor of a layer/block having different hidden dimension indices but corresponding to same token