ANCRe: Adaptive Neural Connection Reassignment for Efficient Depth Scaling

Yilang Zhang*
ML Research
Morgan Stanley
New York, NY 10019
yilang.zhang@morganstanley.com

Bingcong Li*
Department of CS
ETH Zürich
8092 Zürich, Switzerland
bingcong.li@inf.ethz.ch

Niao He
Department of CS
ETH Zürich
8092 Zürich, Switzerland
niao.he@inf.ethz.ch

Georgios B. Giannakis
Department of ECE
University of Minnesota
Minneapolis, MN 55455
georgios@umn.edu

Abstract

Scaling network depth has been a central driver behind the success of modern foundation models, yet recent investigations suggest that deep layers are often underutilized. This paper revisits the default mechanism for deepening neural networks, namely residual connections, from an optimization perspective. On a motivating example, rigorous analysis proves that the layout of residual connections can fundamentally shape convergence behavior, and even induces an *exponential* gap in convergence rates. Prompted by this insight, we introduce adaptive neural connection reassignment (ANCRe), a principled and lightweight framework that parameterizes and learns residual connectivities from the data. ANCRe adaptively reassigns residual connections with negligible computational and memory overhead ($< 1\%$), while enabling more effective utilization of network depth. Extensive numerical tests across pre-training of large language models, diffusion models, and deep ResNets demonstrate consistently accelerated convergence, boosted performance, and enhanced depth efficiency over conventional residual connections.

1 Introduction

Foundation models have demonstrated remarkable success across a broad spectrum of domains and impactful applications. For instance, large language models (LLMs) exhibit nearly human-level proficiency in various tasks, including conversational interaction [1], automated code generation [6], and complex mathematical reasoning [24]. Diffusion models have revolutionized vision tasks by enabling high-fidelity and controllable image synthesis [19, 36, 37]. Recent work has further extended foundation models to multimodal settings, where joint representations are learned from heterogeneous data modalities [32, 10].

One of the key factors underlying these advances is the increasing capacity of modern backbone architectures, with a particularly noticeable trend toward greater depth. For instance, the Llama 3.1 family employs 32, 80, and 126 Transformer layers for its 8B, 70B, and 405B variants, respectively [13]. Likewise, Diffusion Transformers (DiTs) scale in depth from 12 (DiT-S) up to 28 layers (DiT-XL) [31]. Moreover, [7] shows a clear positive correlation between depth and performance

*Equal contribution. Work completed while Y. Zhang was a PhD student at University of Minnesota.

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

across 132 open-source LLMs (see their Figure 1). This trend is also supported by theories. It is proved in [40] that polynomially deep networks can express functions that would require exponential width in shallow ones.

Despite the documented success of deep networks, scaling model depth can be *less efficient* than it first appears. For example, [7] shows that skipping an early layer in the Llama 3.1 70B has a remarkably greater impact on the outputs of subsequent layers than omitting a deep layer, and that deeper layers often behave as near-identity mappings. Since the identity function is essentially available "for free", this suggests that late layers are highly underutilized. Complementary evidence also occurs in multimodal foundation models, where the most informative vision embeddings are frequently found in intermediate layers of the Perception Encoder rather than the final layer [5]. Collectively, these observations reveal that the representational potential of depth is not fully exploited yet.

Given that residual (skip) connections are the default strategy and dominant mechanism for scaling model depth, this work revisits their design to enable more efficient depth scaling. Residual connections were proposed in [38, 16] to avoid vanishing and exploding gradients, and they have become almost universal across architectures. For example, the backbone architecture of LLMs, i.e., the Transformer [45], introduces residual connections around each self-attention and feedforward network modules. From an optimization perspective, residual connections are credited with smoothing the loss landscape, which facilitates training by improving the (local) condition number [25].

This work continues on the optimization perspective of residual connections, and demonstrates an intuitive yet often overlooked factor: *where* residual connections are placed within a deep architecture, i.e., the residual *topology*, can play a crucial role in optimization. We provide quantitative theory showing that different topologies can induce an *exponential* gap in convergence for deep linear neural networks. This large gap motivates a principled redesign of residual layout. To this end, we term our approach adaptive neural connection reassignment (ANCRe[2]), which learns an optimal residual topology from data. Our method not only achieves a linear convergence rate for deep linear networks, but also integrates seamlessly into modern architectures including LLMs, DiTs, and ResNets with consistent empirical gains. In a nutshell, our contributions are as follows:

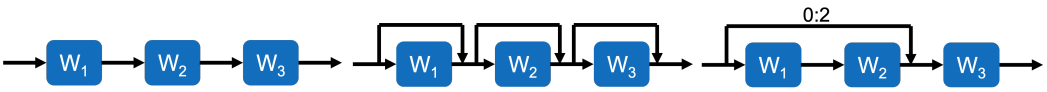
- A theoretical characterization of residual connection topology is established using deep linear neural networks, showing that different layouts can induce exponential gaps in convergence rates.
- ANCRe is proposed to parameterize residual connections and learn a data-driven topology on the fly with appropriate normalization. It incurs negligible computational and memory overhead.
- Extensive numerical evaluations are conducted under varying data modalities and network depths to systematically examine the efficiency of ANCRe. As an illustrative example, ANCRe achieves a 1.85$\times$ training speedup on LLaMA-1B over conventional residual topology.

2 Related work

Residual connections. Residual (skip) connections are a primary mechanism for scaling neural networks to greater depth [38, 16, 17]. They were rapidly popularized in computer vision, with variants such as [3] and DenseNet [21], and have since become a standard component of CNN-based architectures; see e.g., [54, 50]. In contrast, residual connections in large language models (LLMs) have remained relatively stable. Both Transformers and their recent variants adopt the same residual topology [45, 39, 42, 53, 31], while certain shortcuts can be introduced to mitigate over-smoothing [30] or reduce KV cache size [58]. More recently, hyper-connections (HC) [59] and manifold-constrained hyper-connections (mHC) [51] were proposed for foundation models as architectural alternatives. Our work is orthogonal to these lines of research: while they focus on intra-layer designs, we emphasize the *inter-layer topology*.

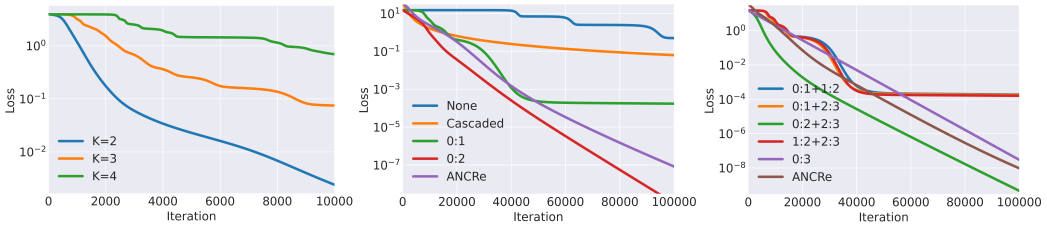
Understanding of residual connections. Residual connections are known to stabilize training by mitigating vanishing and exploding gradients [14], and make gradients in deep networks less "shattered" (i.e., less like white noise) [4]. Empirical visualizations of ResNets also suggest that residual connections lead to a smoother loss landscape [25]. Theoretical understandings are often acquired by contrasting deep linear neural networks [2] with their residual counterparts. For example, the convergence of deep linear networks can degrade exponentially with depth [35], whereas residual

[2]It coincides with "anchor" in French.



(a) No residual connection (b) Cascaded residual connections (c) Residual connection 0:2

Figure 1: Visualization of linear neural network (LNN) of $K = 3$ layers.



(a) Varying depth K (b) $K = 3$ with varying shortcuts (c) $K = 4$ with two shortcuts

Figure 2: Convergence comparison of LNN under varying setups.

connections alleviate this slowdown [49]. Moreover, residual connections can relax network width requirements for global convergence in certain regimes, as by comparing results in [11, 60]. The loss landscape of deep linear residual networks has also been studied in [15]. Our work enriches this line of results by showing that the *topology* of residual connections can affect convergence *exponentially*. More importantly, we translate this insight into a practical design that yields consistent improvements in modern architectures. Due to space limitation, other related work is deferred to Appendix A.

Notation. Bold lowercase (capital) letters denote vectors (matrices); $\|\cdot\|$ and $\|\cdot\|_{\mathrm{F}}$ stand for ℓ_2- and Frobenius-norm.

3 Residual topology matters: a case study

Popularized by ResNet [16], residual connections allow each layer k to learn a residual mapping r_k relative to its input $\mathbf{x}_k$; i.e., $g_k(\mathbf{x}_k) := r_k(\mathbf{x}_k) + \mathbf{x}_k$. By providing an identity shortcut, the loss landscape is more well-behaved [25], thereby enabling training deep networks with hundreds of layers. This architecture has been extensively leveraged in foundation models [45, 48, 32] to train the network at scale. Residual connections are typically arranged in a cascaded structure as in Figure 1b, with each shortcut bypassing a single layer or block.

Despite the popularity of residual connections, their topology, i.e., the shortcut layout, is often fixed by default. This section revisits this simple design, and reveals a somewhat surprising message: in a theoretically convenient case study, *topology alone can provably induce exponentially different optimization behaviors*.

3.1 Linear neural networks with residual connections

Consider a commonly adopted prototype model, deep linear neural network (LNN), $f(\mathbf{x}) := \prod_{k=1}^{K} \mathbf{W}_k \mathbf{x} = \mathbf{W}_K \dots \mathbf{W}_2 \mathbf{W}_1 \mathbf{x}$ as sketched by Figure 1a, where $\mathbf{W}_k \in \mathbb{R}^{d \times d}$ denotes the learnable weights per layer k. This idealized model is often adopted for understanding the optimization and analyzing convergence of deep networks; see [2, 35] and more in Section 2. In addition, consider also LNN augmented with residual connections. For brevity, we henceforth use $i{:}j$ to represent the shortcut bridging the outputs of layers i and j, where $0 \le i < j \le K$, and 0 represents the input of the first layer; see Figure 1c for an example. While NN layers can be highly nonlinear, the residual topology itself remains an additive linear routing mechanism. We therefore study LNN to isolate the role of residual topology from multiple interacting components of modern architectures.

Case Study 3.1. Given a K-layer LNN and input-output pair $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d \times n}$ ($n \ge d$), define regression objective

$$\min_{\{\mathbf{W}_k\}_k} \mathcal{L}(\{\mathbf{W}_k\}_{k=1}^{K}) := \frac{1}{2} \left\| \prod_{k=1}^{K} \mathbf{W}_k \mathbf{X} - \mathbf{Y} \right\|_{\mathrm{F}}^2.$$

Using standard gradient descent (GD) for training, Figure 2a outlines the training loss evolution with depth $K = 2, 3, 4$, with vertical axis in log-scale. Without the aid of residual connection, the convergence slows down when the network grows deeper. Figure 2b depicts the convergence behavior of a 3-layer LNN under different residual connection topologies. While the conventional cascaded layout accelerates the convergence relative to no connection, it is not an optimal choice in this setup. In contrast, incorporating a single residual connection 0:1 or 0:2 yields remarkably faster convergence, where the latter further exhibits an exponential improvement over the former. Figure 2c extends the case study to a 4-layer LNN with 2 residual connections, and compares it with the optimal choice of a single connection 0:3. Notably, the model does not necessarily benefit from the additional residual connections, unless they are appropriately located.

Case Study 3.1 suggests that the configuration of residual connections can induce exponential discrepancies in convergence behaviors, and thus must be chosen cautiously. Nevertheless, there are currently neither rigorous theories supporting these observations, nor a principled approach for selecting an optimal topology. These two challenges call for an analytical investigation and numerical evaluations, which will be provided in the ensuing sections.

3.2 Convergence analysis of exponential discrepancies

This subsection establishes theories characterizing the exponentially different convergence behaviors induced by residual connections. To simplify the setup, our analysis considers the 3-layer LNN with a single residual connection (0:1 or 0:2), as illustrated in Figure 2b, while extension to nonlinear neural networks is left for future work. To eliminate the influence of step size, the analysis is performed under gradient flow (GF), the continuous-time counterpart of GD. Specifically, GF indexes the optimization time with a continuous variable $t \geq 0$, and updates the weight matrix through ordinary differential equation

$$\frac{\mathsf{d}\mathbf{W}_k(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_k}\mathcal{L}(t),\ k = 1, \ldots, K \tag{1}$$

where the dependency of $\mathcal{L}$ on $\{\mathbf{W}_k(t)\}_k$ is omitted for compactness. Clearly, GF can be viewed as an approximation of GD in the limit of infinitely small step sizes. Moreover, we assume the input data matrix $\mathbf{X}$ is whitened, so that it's orthogonal; i.e., $\mathbf{X}\mathbf{X}^\top = \mathbf{I}_d$. This is a standard modeling choice that avoids unnecessarily over-complicating the analysis while still capturing the essential dynamics [2, 49, 27]. When $\mathbf{X}$ is a random matrix, this assumption corresponds to the data vectors being normalized and uncorrelated, which can be readily satisfied in practice without loss of generality.

Under these simplifications, the following two theorems establish respectively a lower bound for the LNN with the 0:1 shortcut, and an upper bound for the 0:2 one.

Theorem 3.2 (Existence lower bound)**.** *Consider a 3-layer LNN with shortcut 0:1, and ℓ_2 loss*

$$\mathcal{L}_1(t) := \frac{1}{2}\left\|\mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t) + \mathbf{I}_d\big)\mathbf{X} - \mathbf{Y}\right\|_{\mathrm{F}}^2. \tag{2}$$

There exists sufficiently small initialization that, GF (1) cannot converge faster than a sublinear rate

$$\mathcal{L}_1(t) \geq \Omega(1/t^2). \tag{3}$$

Theorem 3.3 (Universal upper bound)**.** *Consider a 3-layer LNN with shortcut 0:2, and ℓ_2 loss*

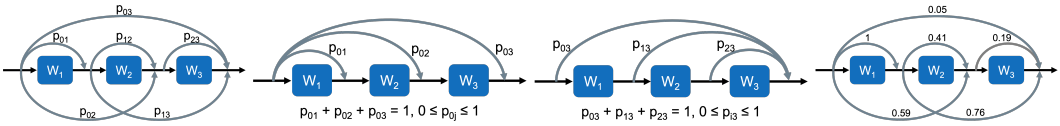
$$\mathcal{L}_2(t) := \frac{1}{2}\left\|\mathbf{W}_3(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t) + \mathbf{I}_d\big)\mathbf{X} - \mathbf{Y}\right\|_{\mathrm{F}}^2. \tag{4}$$

Under any sufficiently small initialization, GF (1) ensures linear convergence

$$\mathcal{L}_2(t) \leq \mathcal{L}_2(0)e^{-2(1-\lambda)^2 t} \tag{5}$$

where $\lambda \in (0, 1)$ is a constant associated with the initialization $\{\mathbf{W}_k(0)\}_{k=1}^3$.

The formal statements of Theorems 3.2 and 3.3 involving all details can be found in Appendix C. We remark that small random initialization is standard in deep networks for numerical stability. While the 0:1 residual connection cannot achieve the universal upper bound faster than a sublinear convergence rate, the 0:2 shortcut yields an exponential improvement. This corroborates our observation in



(a) ANCRe (b) Outgoing normalization (c) Ingoing normalization (d) Learned connectivity

Figure 3: Visualization of ANCRe and two normalization schemes on a 3-layer LNN.

Figure 2b, and validates that proper placements of residual connections can *exponentially* accelerate the training convergence.

It is worth stressing that these results can be readily extended to $K > 3$; see Appendix C.4. To be specific, the 0:1 shortcut always corresponds to the slow case of sublinear convergence rate of $\Omega(1/t^{\frac{K-1}{K-2}}) \geq \Omega(1/t^2)$, whereas the 0:$K-1$ one guarantees fast linear convergence. This suggests: *increasing depth can impede optimization when residual connections are not appropriately placed*. For the simple LNNs, the lower bound approaches $\Omega(1/t)$ as $K \to \infty$. An example of the fast case can be found in the 0:3 (violet) curve of Figure 2c.

Our case study and convergence analysis highlight *where* residual connections are placed could affect the convergence rate exponentially. This pronounced gap suggests that residual topology should be treated as a design lever for faster optimization. In the next section, we turn to the practical question of *how* to find a performant topology.

4 Learning residual connections from data

As observed in the previous section, even with the same network architecture, the residual topology pattern for fast convergence can change with depth (e.g., 0:2 for 3-layer and 0:2+2:3 for 4-layer LNNs in Figures 2b and 2c). More broadly, an optimal layout can depend on other architectural choices. It is thus difficult to prescribe a "panacea" of residual topology, especially for complex foundation models. This motivates learning the desirable topology on the fly.

4.1 ANCRe: Adaptive neural connection reassignment

Instead of exhaustively searching over all the $2^{(K+1)K/2}$ possible topologies, our key idea is to parameterize the layout, and adaptively reassign the connectivity based on the network architecture and data distribution. Thus, we term our approach adaptive neural connection reassignment (ANCRe). In particular, ANCRe considers all possible shortcut connections i:j ($i < j$), each associated with a coefficient p_{ij} that is optimized alongside the model parameters; cf. Figure 3a. We further probe two schemes for normalizing these weights, which we refer to as outgoing and ingoing normalizations, as exemplified in Figures 3b and 3c. This normalization ensures that the coefficients over candidate positions sum to 1. In particular, if some $p_{ij} = 1$, the construction reduces to adding a standard residual connection between layers i and j. As shown later in Section 5.4, this normalization is essential for stabilizing training at scale.

Outgoing normalization restricts the coefficients originating from layer i via a convex combination

$$\sum_{j=i+1}^{K} p_{ij} = 1,\ 0 \leq p_{ij} \leq 1,\ \forall 0 \leq i < j \leq K. \tag{6}$$

That says, the feature embedding from layer i is routed to downstream layers through weighted shortcuts $p_{ij}\mathbf{x}_i$. This ensures that the outgoing information per layer is bounded. Conversely, ingoing normalization imposes the following constraint

$$\sum_{i=0}^{j-1} p_{ij} = 1,\ 0 \leq p_{ij} \leq 1,\ \forall 0 \leq i < j \leq K \tag{7}$$

which preserves the input signal magnitude to layer j and enhances numerical stability. An alternative interpretation is that it forms an *ensemble over network depths*. Letting $f_0(\mathbf{x}) := \mathbf{x}$ denote the initial

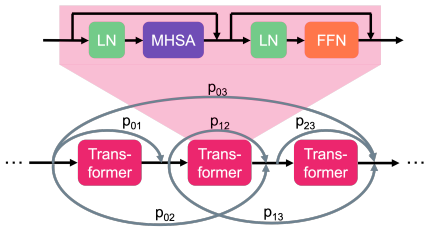


Figure 4: ANCRe applied to standard Transformer comprising LN, MHSA, and FFN modules.

Table 1: Validation perplexity (↓) by adding ANCRe to inputs of different modules.

MHSA	FFN	60M	130M
×	×	30.39	25.07
√	×	**29.62**	24.48
×	√	30.43	24.98
√	√	29.73	**24.41**

input and $f_i(\mathbf{x})$ the output of layer i, the aggregated shortcuts to layer j incur $\sum_{i=0}^{j-1} p_{ij} f_i(\mathbf{x})$, which ensembles subnetworks of depths $0, \ldots, j-1$, with p_{ij} serving as the routing coefficient.

In practice, both normalization constraints can be conveniently enforced via a softmax reparametrization. For example, the ingoing normalization can be formulated as

$$p_{ij} = \frac{\exp(c_{ij}/\tau)}{\sum_{k=0}^{j-1} \exp(c_{kj}/\tau)}, \ \forall 0 \le i < j \le K \tag{8}$$

where $\tau > 0$ is a temperature hyperparameter, with a default value of 0.01 in practice. This softmax allows the model to emphasize beneficial shortcuts by driving $c_{ij} \to +\infty$, while suppressing detrimental ones with $c_{ij} \to -\infty$. Furthermore, if an effective configuration is known a priori, it can be readily incorporated through a regularizer. For instance, adding Tsallis entropy regularization [44] to the loss encourages sparse connectivity for softmax reparameterization. It is worth stressing that ANCRe introduces merely $K(K+1)/2$ additional parameters c_{ij}, which is often fewer than a single feature dimension, and is negligible in modern deep networks. As a result, ANCRe preserves the highly desired computational efficiency; cf. Table 6.

While both normalizations work well in practice, we often find that ingoing normalization slightly outperforms the other at larger scale; see the ablation study in Section 5.4. We conjecture that this is due to the more stable input magnitude to each layer. For this reason, we stick to *ingoing normalization* (7) throughout the paper.

Regarding Case Study 3.1, ANCRe learns a data-driven connectivity, and hence achieves a linear convergence rate on par with the optimal case; cf. Figure 2b and 2c. The learned coefficients for the 3-layer LNN under ingoing normalization are visualized in Figure 3d, where certain shortcuts such as 0:3 and 2:3 are suppressed by the ingoing normalization. Aside from LNNs, the next subsection investigates how to effectively incorporate ANCRe into Transformers.

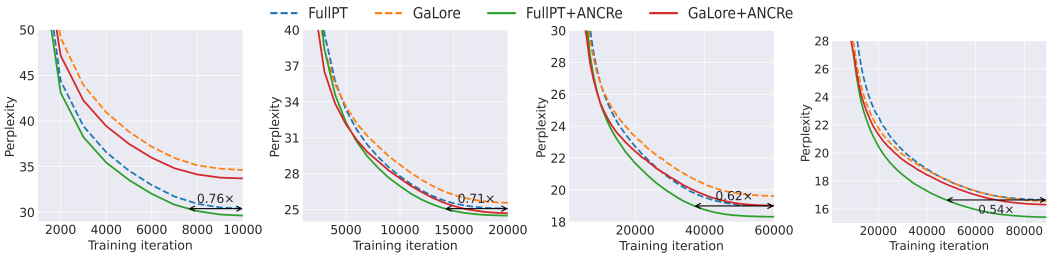
4.2 Applying ANCRe to Transformers

In modern Transformers, residual connections are applied separately to the multi-head self-attention (MHSA) and feedforward network (FFN) modules in a cascaded way, as illustrated in the upper part of Figure 4. We consider two granularities for placing ANCRe: module-level and block-level. The former introduces ANCRe connections from the input of the i-th FFN (resp. MHSA) to the input of the j-th FFN (resp. MHSA), while the latter bridges the inputs of entire Transformer blocks. Note that the block-level granularity coincides with the MHSA module-level one.

To determine the optimal granularity, a lightweight ablation study is performed by pre-training LLaMA-60M and LLaMA-130M [42] on the C4 dataset [33]. As summarized in Table 1, establishing connectivity between MHSA inputs improves the validation perplexity (lower the better), whereas shortcuts originating from FFN inputs are less informative. Moreover, combining both connections yields performance comparable to using MHSA-only connections. For this reason, we adopt the MHSA-only (i.e. block-level) ANCRe for all subsequent tests involving Transformers; cf. Figure 4.

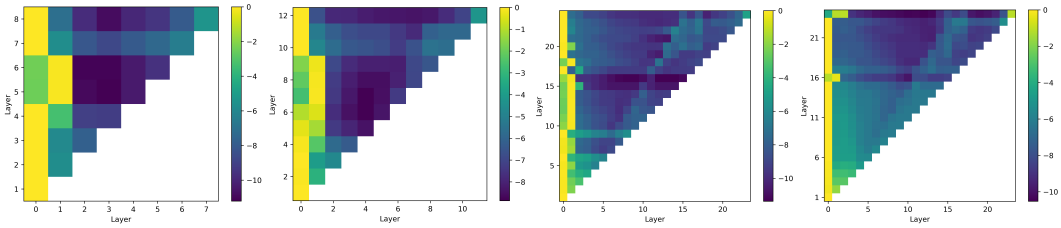
Aside from granularity, another critical consideration for LLM integration is the compatibility with gradient checkpointing and pipeline parallelism, where long-range shortcuts can markedly increase the recomputation and communication costs. To address this challenge, Appendix D presents a practical variant that substantially lowers the associated overhead through sparse residual connectivity.

Next, tests are conducted on foundation models across domains to showcase ANCRe's effectiveness.



(a) 60M (b) 130M (c) 350M (d) 1B

Figure 5: Validation perplexity evolution during pre-training of LLaMA models of different sizes.



(a) 60M (b) 130M (c) 350M (d) 1B

Figure 6: Visualization of residual coefficients $\log(p_{ij})$ learned by ANCRe, where horizontal and vertical axes stand for i and j.

5 Numerical experiments

This section implements numerical tests to assess the performance of ANCRe, and compare it with vanilla cascaded residual connections (i.e., the standard architecture). All tests are conducted on servers with NVIDIA A100 and H100 GPUs. Additional setups including models, hyperparameters, and datasets are detailed in Appendix E. The code is available at https://github.com/zhangyilang/ANCRe.

5.1 Pre-training of LLMs

The evaluation of ANCRe begins with pre-training LLMs. Following the setups in [26, 57], a series of LLaMA models [42] are trained on the widely adopted C4 dataset [33], which consists of cleaned English text collected across the Internet. The models span parameter counts of {60M, 130M, 350M, 1B}, with corresponding depths of {8, 12, 24, 24}. All experiments are run in BF16 precision. Two optimization schemes are considered: standard full pre-training (FullPT) and GaLore [57], which applies low-rank gradient projection for memory efficiency. For both schemes, learning rates are selected from $\{5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}\}$. ANCRe directly adopts the learning rate optimized for cascaded residual connections without additional tuning. HC [59] and mHC [51] with 4 parallel residual streams are included as strong residual-learning baselines.

Figure 5 plots the validation perplexity over pre-training iterations, while Table 2 reports quantitative results evaluated with the pre-trained models. Across all eight combinations of model sizes and optimization schemes, ANCRe consistently converges faster, and reduces perplexities by an average ($\pm$ standard deviation) of 0.72 ± 0.33 over conventional residual topology. Notably, ANCRe matches the perplexity of cascaded residual connections using on average 34.3% fewer training iterations. The advantage grows from 24% to 46% as the network scales deeper. This underscores not only the significance of residual connection topology, but also the effectiveness of ANCRe in leveraging the increased depth. Additional runs with multiple random seeds are reported in Appendix E.5.

In addition, Figure 6 sketches the residual coefficients learned by ANCRe. For clearer visualization, the heatmap is displayed on a logarithmic scale. Rather than relying solely on the shortcut from immediately preceding layer, deeper layers under ANCRe ensemble subnetworks of different depths. The learned connectivity patterns are also consistent across model sizes, with a predominant emphasis

Table 2: Perplexity (PPL↓) comparison by pre-training LLaMA models of varying sizes. The lowest PPLs are marked in solid lines.

Method	60M	130M	350M	1B
FullPT	30.39	25.07	19.00	16.64
FullPT+HC	29.86	24.51	18.58	16.38
FullPT+mHC	30.21	24.62	18.93	16.14
FullPT+ANCRe	**29.62**	**24.48**	**18.32**	**15.41**
GaLore	34.61	25.56	19.62	16.55
GaLore+ANCRe	**33.69**	**24.68**	**19.01**	**16.45**
Training tokens	1.1B	2.2B	6.4B	13.1B

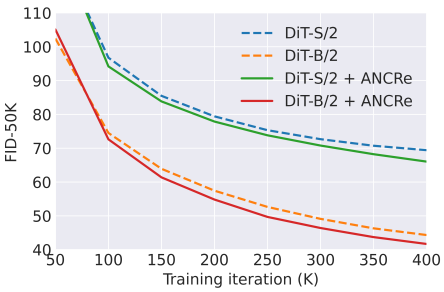


Figure 7: FID-50K evolution when pre-training DiT models.

Table 3: Comparison of ANCRe and cascaded residual connections by pre-training DiT models. The better of the two are marked in bold.

Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
DiT-S/2	69.40	12.45	19.66	35.75%	55.82%
DiT-S/2+ANCRe	**66.01**	**11.68**	**20.70**	**37.90%**	**57.80%**
DiT-S/2 (cfg=1.5)	45.78	9.08	33.48	46.08%	54.25%
DiT-S/2+ANCRe (cfg=1.5)	**42.99**	**8.61**	**35.04**	**48.53%**	**55.29%**
DiT-B/2	44.31	8.42	32.89	47.93%	61.55%
DiT-B/2+ANCRe	**41.66**	**7.89**	**34.40**	**50.42%**	**64.20%**
DiT-B/2 (cfg=1.5)	22.41	6.35	65.27	60.75%	52.41%
DiT-B/2+ANCRe (cfg=1.5)	**20.53**	**5.81**	**70.45**	**65.91%**	**56.51%**

on shortcuts originating from the first two layers. Furthermore, larger models (350M and 1B) exhibit denser connectivity than the smaller ones, indicating more effective exploitation of network depth.

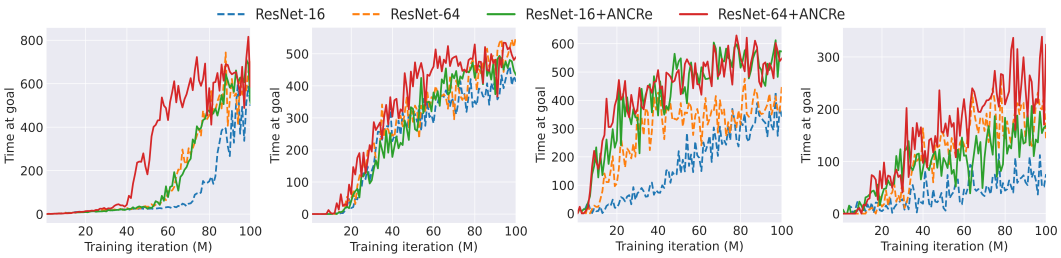
5.2 Pre-training of diffusion models

Beyond LLMs, pre-training of diffusion models is also inspected following the experimental protocols in Diffusion Transformers (DiT) [31]. Specifically, class-conditional latent DiT-S/2 and DiT-B/2 are trained on the ImageNet-1K dataset [9], which contains over 1.3 million images across 1,000 classes, at 256×256 resolution for 400,000 iterations. All hyperparameters including the learning rate are set to the defaults in [31] without tuning. For evaluation, the standard Fréchet inception distance (FID) [18] is calculated using 50,000 randomly sampled images from 250 DDPM steps [19], without classifier-free guidance (cfg) unless otherwise stated. Secondary metrics include spatial FID (sFID) [29], inception score (IS) [34], and improved precision/recall [22].

The evolution of FID-50K over the pre-training iterations is outlined in Figure 7. It is observed that the incorporation of ANCRe remarkably accelerates the convergence, and leads to improved final performance, thanks to the effective utilization of network depth. It is worth noting that the performance gains are relatively smaller than those observed for LLMs, as DiT are generally shallower. In addition to the FID-50K, Table 3 reports auxiliary metrics computed using the pre-trained models for a more comprehensive comparison. Again, ANCRe consistently outperforms cascaded residual connections by an average of approximately 6% among metrics, demonstrating its effectiveness for different data modalities.

5.3 Reinforcement learning with ResNets

The third test examines reinforcement learning (RL) with "actions" as the modality. The test settings follow from [47], which demonstrates that scaling model depth in RL can unlock new goal-reaching capabilities. The test involves an unsupervised goal-conditioned regime, in which rewards are only available upon reaching the commanded goals. Broadly speaking, this setting follows the same spirit as RLVR [23]. Given that the sparse rewards provide rather limited feedback, the problem is considerably more challenging compared to standard RL scenarios supervised by dense rewards or



(a) Humanoid (b) Ant Big Maze (c) Arm Push Hard (d) Arm Binpick Hard

Figure 8: Reinforcement learning using ResNets of 16 and 64 layers.

Table 4: Perplexity under different normalization schemes.

Normalization	130M	350M
None	diverge	diverge
Outgoing (6)	25.03	18.63
Ingoing (7)	**24.48**	**18.32**

Table 5: Ablation study using all or learnable shortcuts.

All $i{:}j$	Learnable	130M	350M
$\surd$	$\times$	diverge	diverge
$\times$	$\surd$	27.18	21.72
$\surd$	$\surd$	**24.48**	**18.32**

expert demonstrations. Consequently, it necessitates deeper models of sufficient representational capacity. The models are ResNets [16] of varying depths, with cascaded residual connections spanning consecutive blocks of four layers. ANCRe is applied at the same granularity of blocks, after removing the original cascaded residual connections. The RL benchmark encompasses four locomotion and manipulation tasks {`Humanoid`, `Ant_Big_Maze`, `Arm_Push_Hard`, `Arm_Binpick_Hard`} from the Gymnasium environments [43]. Training is performed with the simple contrastive RL (CRL) algorithm [12] for 100 million environment steps, and all hyperparameters are consistent with [47].

Figure 8 sketches the time-at-goal metric (↑) averaged across episodes. For all four tasks, ANCRe consistently leads to considerably accelerated convergence than the vanilla ResNet baselines. Notably, when equipped with ANCRe, a 16-layer ResNet can match or even surpass the performance of a $4\times$ deeper ResNet. These results further corroborate that standard ResNet architectures tend to underutilize increased network depth, and demonstrate that ANCRe provides an effective plug-and-play remedy for improving depth efficiency.

5.4 Ablation study

The last group of tests perform ablation studies to justify ANCRe's design choices. The tests employ the setups as in Section 5.1 using the LLaMA models with FullPT.

The first test compares the two normalization schemes discussed in Section 4.1. From Table 4, omitting normalization leads to divergence, since the coefficients can grow unbounded. It is further observed that ingoing normalization yields smaller fluctuations in the training loss and consequently achieves lower perplexity than outgoing normalization. This is because the former normalizes the inputs to each layer, so that computations are numerically stabler.

The second ablation study showcases that the effectiveness of ANCRe does not stem from simply adding more connections or making coefficients learnable. To see this, we consider two variants of ANCRe: one containing all possible shortcuts $i{:}j$ ($i < j$) but with constant coefficients $c_{ij} = 1/j$, and another with learnable cascaded residual connection on each Transformer block. As shown in Table 5, these variants either diverges or achieves less competitive performance. This highlights that ANCRe's gains arise from the synergistic interaction between its connection topology and adaptive coefficient learning rather than either component alone.

Finally, we measure training runtime and GPU memory usage to assess the efficiency of ANCRe. As shown in Table 6, incorporating ANCRe introduces negligible computational and memory overhead across all model scales. In particular, the training throughput drop by $< 1\%$, while the increase in peak GPU memory is consistently below 0.01GB. This efficiency arises because the forward and

Table 6: Training overhead comparison.

Model, depth, GPU, batch	Method	Param	Mem (GB)	Throughput (tok/s)
350M, 24, H100, 64	Cascaded	0	26.60	99.9k
	HC	172k	46.76	43.6k
	mHC	172k	55.55	33.6k
	ANCRe	300	26.60	99.2k
1B, 24, H100, 32	Cascaded	0	30.38	17.3k
	HC	345k	50.52	12.0k
	mHC	345k	58.50	10.3k
	ANCRe	300	30.38	17.3k

Table 7: Ablation study of learned topology pattern.

Shortcuts	130M	350M
ANCRe	24.48	18.32
First and prev.	25.44	19.94
Retraining	24.48	18.49

backward passes of the residual connections involve only a scalar–matrix multiplication and a matrix addition, which are ignorable compared to the dominant computations in Transformers. Together, these results demonstrate that ANCRe delivers performance gains with minimal overheads, ensuring its practical merits of extracting more performance from deep foundation models.

5.5 Analysis of learned topology

Figure 6 suggests that the learned topology exhibits a consistent pattern, underscoring shortcut from the first layer. We next demonstrate that the advantage of ANCRe does not come from a trivial architectural modification. Specifically, we evaluate a fixed shortcut design that incorporates inputs from both the previous layer and the first layer. As shown in Table 7, this fixed topology performs remarkably worse than ANCRe, indicating that the observed gains cannot be attributed merely to the additional shortcut connections from the first layer. Instead, the ability to learn the connectivity pattern is crucial.

To further investigate the role of the learned topology, we froze the final learned coefficients and retrained a new model from scratch. The resulting models retained nearly all of the original performance gains. This result suggests that the learned topology together with its routing coefficients, rather than the mere presence of additional aggregation pathways, captures a transferable and beneficial connectivity structure.

We conjecture that early-layer representations encode reusable low-level features that remain valuable throughout the network when appropriately scaled, whereas intermediate-layer representations become increasingly specialized and task-dependent with the growth of depth.

6 Conclusion and outlook

This paper investigated how the topology of residual connections fundamentally influences the convergence and depth efficiency of deep networks, and introduced adaptive neural connection reassignment (ANCRe) as a principled approach to learn the optimal residual connectivity. One major limitation of this work is that our analysis is limited to LNNs. Additionally, empirical experiments indicate that the benefits are particularly more effective for deep networks. Other potential cases where ANCRe can be less effective include architectures whose residual topology has already been extensively optimized, and tasks where intermediate representations are less reusable across depths.

Looking ahead, several directions have been added to the research agenda. First, to capture the optimization dynamics of nonlinear neural networks, the convergence analysis will be broadened to more general models beyond LNNs. Second, more effective schemes for parameterizing and normalizing residual connections tailored to foundation models such as Transformers will be explored, with an emphasis on improving stability and depth efficiency. Finally, pre-training tests will be scaled to larger models, for which efficient depth utilization becomes increasingly critical.

Acknowledgments

This work is partly supported by NSF grants 2212318, 2220292, and 2312547. B. Li and N. He are supported by Swiss National Science Foundation (SNSF) Starting Grant.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019.

[3] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: fast convergence at large depth. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, volume 161, pages 1352–1361, 27–30 Jul 2021.

[4] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 342–350, 2017.

[5] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[7] Róbert Csordás, Christopher D Manning, and Christopher Potts. Do language models use their depth efficiently? *arXiv preprint arXiv:2505.13898*, 2025.

[8] Soham De and Sam Smith. Batch normalization biases residual blocks towards the identity function in deep networks. *Proc. Neural Information Processing Systems (NeurIPS)*, 33:19964–19975, 2020.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[10] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 202, pages 8469–8488, 23–29 Jul 2023.

[11] Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 1655–1664, 2019.

[12] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 35, pages 35603–35620, 2022.

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

[14] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.

[15] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 630–645. Springer, 2016.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[20] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[22] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[23] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[24] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 35, pages 3843–3857, 2022.

[25] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Proc. Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[26] Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-rank training through low-rank updates. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

[27] Kai Lion, Liang Zhang, Bingcong Li, and Niao He. Polar: Polar-decomposed low-rank adapter representation. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2025.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

[29] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter Battaglia. Generating images with sparse representations. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 139, pages 7958–7968, 18–24 Jul 2021.

[30] Tam Nguyen, Tan Nguyen, and Richard Baraniuk. Mitigating over-smoothing in transformers via regularized nonlocal functionals. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 36, pages 80233–80256, 2023.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 4195–4205, October 2023.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 139, pages 8748–8763, 18–24 Jul 2021.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[34] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[35] Ohad Shamir. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In *Proc. Annual Conf. on Learning Theory (COLT)*, volume 99, pages 2691–2713, 25–28 Jun 2019.

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

[38] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[39] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[40] Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[41] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 32–42, 2021.

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

[43] Mark Towers, Ariel Kwiatkowski, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun KG, Rodrigo De Lazcano Perez-Vicente, J K Terry, Andrea Pierré, Sander V Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In *Proc. Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[44] Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics*, 52(1):479–487, 1988.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[46] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. *IEEE Trans. Pattern Anal. Mach. Intel.*, 46(10):6761–6774, 2024.

[47] Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised RL: Scaling depth can enable new goal-reaching capabilities. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2025.

[48] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 34, pages 11960–11973, 2021.

[49] Lei Wu, Qingcan Wang, and Chao Ma. Global convergence of gradient descent for deep linear residual networks. *Proc. Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[51] Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

[52] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 119, pages 10524–10533, 13–18 Jul 2020.

[53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[54] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[55] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

[56] Yilang Zhang, Bingcong Li, and Georgios B Giannakis. Reflora: Refactored low-rank adaptation for efficient fine-tuning of large models. *arXiv preprint arXiv:2505.18877*, 2025.

[57] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 61121–61143, 21–27 Jul 2024.

[58] Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In *Proc. Conf. Assoc. Comput. Linguist. Meet. (ACL)*, pages 28341–28356, July 2025.

[59] Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[60] Difan Zou, Philip M Long, and Quanquan Gu. On the global convergence of training deep linear resnets. *arXiv preprint arXiv:2003.01094*, 2020.

A Additional related work

Scaling deep. Broadly speaking, our work is also related to mechanisms for scaling neural networks to greater depth. Normalization techniques are often believed to facilitate training at depth. For example, the normalization in pre-activation ResNets [17] and transformers [52] has been shown to improve training stability. Moreover, it is shown in [8] that batch normalization downscales the hidden activations on the residual branch by an order of square root of the network depth at initialization. Proper normalization techniques, in conjunction with residual connections, enable deeper LLMs [46] and vision transformers [41]. At the same time, there are also works suggesting that normalization is not strictly necessary, as some of its early training benefits can be reproduced by carefully designed initialization; see e.g., [55, 8]. Nevertheless, residual connections remain a central ingredient for modern deep neural networks.

Architecture-optimizer co-design. Recently, there has been a growing trend toward opening up the "black box" of neural networks and co-designing architectures and optimizers for more efficient training. For example, [56] studies the gauge invariance underlying the optimization of LoRA [20], while [27] redesigns the architecture so that fine-tuning performance scales more favorably with width (i.e., adapter rank). Our work aligns with this trend by showing that residual connections can substantially reshape the loss landscape, and therefore should be treated as a design choice rather than a fixed architectural default.

B Broader impact

The theoretical analysis and residual topology learning of ANCRe are broadly applicable across diverse large-scale pre-training settings. ANCRe improves the efficiency of scaling network depth in foundation models, yielding accelerated convergence and stronger performance in applications such as LLMs and diffusion transformers. These gains translate into more efficient real-world systems while significantly reducing the total GPU hours required for pre-training, thereby lowering the energy consumption and carbon footprint of AI development. Caution is warranted when applying the method to generative tasks; model outputs should be carefully evaluated, and appropriate safeguards should be implemented to ensure the safety and reliability of the generated content.

C Missing proofs

This appendix offers the missing proofs for theories in the main paper. Our analysis leverages the following notations.

Additional notation. $\langle \cdot, \cdot \rangle_{\mathrm{F}}$, $\mathrm{Tr}(\cdot)$, $\mathrm{rank}(\cdot)$ stand for Frobenius inner product, trace, and rank; $\lambda_i(\cdot)$ and $\sigma_i(\cdot)$ are the i-th largest eigenvalue and singular value. $\mathrm{diag}(\mathbf{v})$ is the diagonal matrix whose diagonal entries are from vector $\mathbf{v}$, while $\mathrm{diag}(\mathbf{M})$ refers to the vector formed by the diagonals of matrix $\mathbf{M}$. For simplicity, the subscripts of $\mathcal{L}_1$ and $\mathcal{L}_2$ are dropped when the objective is clearly specified.

Notably, rather than aiming to analyze the full dynamics of modern architectures, our convergence analysis is intended to support the conjecture that residual topology is an important architectural lever and to motivate the subsequent topology-learning approach. This conjecture is validated through experiments in Section 5. While our current analysis omits nonlinearity in order to better isolate the effects of network depth and residual topology, we view it as a starting point for future extensions to more complex models.

C.1 Proof of Theorem 3.2

Before presenting the proof, we first simplify the optimization objective and look into the associated dynamics.

Recall that $\mathbf{X}$ is orthogonal, so (2) can be equivalently rewritten as

$$\mathcal{L}(t) = \frac{1}{2}\left\|\mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t)+\mathbf{I}_d\big)\mathbf{X}-\mathbf{Y}\right\|_{\mathrm{F}}^2 = \frac{1}{2}\left\|\mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t)+\mathbf{I}_d\big)-\mathbf{Y}\mathbf{X}^\top\right\|_{\mathrm{F}}^2. \quad (9)$$

Defining $\mathbf{A} := \mathbf{Y}\mathbf{X}^\top$, the optimization dynamics then boil down to

$$\begin{aligned}
\mathbf{E}(t) &= \mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t) + \mathbf{I}_d\big) - \mathbf{A}, \\
\nabla_{\mathbf{W}_1}\mathcal{L}(t) &= \mathbf{W}_2(t)^\top \mathbf{W}_3^\top(t)\mathbf{E}(t), \\
\nabla_{\mathbf{W}_2}\mathcal{L}(t) &= \mathbf{W}_3(t)^\top \mathbf{E}(t)\big(\mathbf{W}_1^\top(t) + \mathbf{I}_d\big), \\
\nabla_{\mathbf{W}_3}\mathcal{L}(t) &= \mathbf{E}(t)\big(\mathbf{W}_1^\top(t) + \mathbf{I}_d\big)\mathbf{W}_2^\top(t).
\end{aligned}$$

Given these, the gradient flow (1) can thus be written as

$$\frac{\mathsf{d}\mathbf{W}_1(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_1}\mathcal{L}(t), \ \frac{\mathsf{d}\mathbf{W}_2(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_2}\mathcal{L}(t), \ \frac{\mathsf{d}\mathbf{W}_3(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_3}\mathcal{L}(t). \tag{10}$$

Theorem C.1 (Formal restatement of Theorem 3.2)**.** *There exists a problem instance of* (9) *with diagonally initialized weights, such that if* $\mathbf{W}_1[d,d](0) \in [-0.5, 0]$ *and* $\mathbf{W}_2[d,d](0) = \mathbf{W}_3[d,d](0) \in (0, 0.5]$*, then gradient flow* (10) *can only converge sublinearly at*

$$\mathcal{L}(t) \geq \Omega(1/t^2), \ \forall t > 0.$$

Proof. We construct a specific instance of $\mathbf{A}$ and analyze the resultant gradient flow dynamics. In particular, consider a diagonal matrix $\mathbf{A}$ with rank $r_A < d$. That is, $\mathbf{A} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{r_A}, 0, \ldots, 0)$, where the last singular value $\sigma_d = 0$. Furthermore, since $\mathbf{W}_1(0)$, $\mathbf{W}_2(0)$ and $\mathbf{W}_3(0)$ are diagonally initialized, it can be verified that the dynamics of gradient flow is diagonally separable throughout training. Specifically, letting $w_i(t)$, $u_i(t)$ and $v_i(t)$ be the i-th diagonal of $\mathbf{W}_1(t) + \mathbf{I}_d$, $\mathbf{W}_2(t)$ and $\mathbf{W}_3(t)$, the dynamics for $\forall i \in \{1, 2 \ldots, d\}$ can be written in a decoupled and coordinate-wise form

$$a_i(t) := w_i(t)u_i(t)v_i(t), \tag{11a}$$

$$\frac{\mathsf{d}w_i(t)}{\mathsf{d}t} = -u_i(t)v_i(t)\big(a_i(t) - \sigma_i\big), \tag{11b}$$

$$\frac{\mathsf{d}u_i(t)}{\mathsf{d}t} = -w_i(t)v_i(t)\big(a_i(t) - \sigma_i\big), \tag{11c}$$

$$\frac{\mathsf{d}v_i(t)}{\mathsf{d}t} = -w_i(t)u_i(t)\big(a_i(t) - \sigma_i\big). \tag{11d}$$

We now focus on the coordinate $i = d$, for which $\sigma_d = 0$. Moreover, since we initialize $u_d(0) = v_d(0)$, Lemma C.2 leads to $u_d(t) = v_d(t)$ for $\forall t \geq 0$. Combining this with (11) we obtain

$$\begin{aligned}
\frac{\mathsf{d}a_d(t)}{\mathsf{d}t} &= u_d(t)v_d(t)\frac{\mathsf{d}w_d(t)}{\mathsf{d}t} + w_d(t)v_d(t)\frac{\mathsf{d}u_d(t)}{\mathsf{d}t} + w_d(t)u_d(t)\frac{\mathsf{d}v_d(t)}{\mathsf{d}t} \\
&= -u_d^4(t)a_d(t) - 2w_d^2(t)u_d^2(t)a_d(t) \\
&= -\frac{a_d^3(t)}{w_d^2(t)} - 2w_d(t)a_d^2(t).
\end{aligned}$$

Moreover, the initialization yields $w_d(0) = 1 + \mathbf{W}_1[d,d](0) \in [0.5, 1]$ and $u_d(0) = v_d(0) \in (0, 0.5]$. It then follows from Lemma C.4 that $0 \leq u_d(t) = v_d(t) \leq w_d(t) \leq 1$.

Therefore, it holds

$$a_d(t) = w_d(t)u_d(t)v_d(t) = w_d(t)u_d^2(t) \in [0, 1].$$

Using this, we can further obtain

$$\begin{aligned}
\frac{\mathsf{d}a_d(t)}{\mathsf{d}t} &= -\frac{a_d^3(t)}{w_d^2(t)} - 2w_d(t)a_d^2(t) \\
&\overset{(a)}{\geq} -\frac{a_d^3(t)}{w_d^2(t)} - 2a_d^2(t) \\
&\overset{(b)}{\geq} -a_d^{7/3}(t) - 2a_d^2(t) \\
&\overset{(c)}{\geq} -3a_d^2(t)
\end{aligned}$$

where (a) is by Lemmas C.4; (b) utilizes Lemma C.5 with $a_d(t) > 0$; and (c) follows from $a_d(t) \leq 1$.

The inequality above implies that $a_d(t) \geq \frac{a_d(0)}{1+3a_d(0)t}$ as a result of Lemma C.12. Hence, we arrive at

$$\mathcal{L}(t) = \frac{1}{2}\sum_{i=1}^{d}\big(a_i(t)-\sigma_i\big)^2 \geq \frac{1}{2}\big(a_d(t)\big)^2 \geq \frac{a_d^2(0)}{2(1+3a_d(0)t)^2} = \Omega(1/t^2)$$

where $a_d(0) = w_d(0)u_d^2(0) > 0$. The proof is thus completed. □

We remark that, while the proof is constructed using diagonal matrices, the optimization dynamics are equivalent to $\min_{\mathbf{W}_i} ||(\mathbf{U}\mathbf{W}_3)\mathbf{W}_2(\mathbf{W}_1+\mathbf{I}_d)(\mathbf{X}\mathbf{V}^\top) - \mathbf{U}\mathbf{A}\mathbf{V}^\top|| = ||\tilde{\mathbf{W}}_3\mathbf{W}_2(\mathbf{W}_1+\mathbf{I}_d)\tilde{\mathbf{X}} - \tilde{\mathbf{A}}||$ for any orthogonal $\mathbf{U}$ and $\mathbf{V}$. In fact, this corresponds to a simplified setting where the singular directions of $\tilde{\mathbf{A}}$ are already known. In fact, the experiments in Figure 2 do not use diagonal $\mathbf{A}$ or initialization, yet the sublinear rate is still clearly observed.

Lemma C.2. *If $u_d(0) = v_d(0)$, then $u_d(t) = v_d(t)$, $\forall t \geq 0$.*

Proof. This symmetry follows immediately from the observation that $\frac{\mathsf{d}u_d(t)}{\mathsf{d}t} = \frac{\mathsf{d}v_d(t)}{\mathsf{d}t}$ whenever $u_d(t) = v_d(t)$. As $u_d(0) = v_d(0)$, this symmetry always holds throughout the optimization. □

Lemma C.3. *Under gradient flow (10), it holds that*

$$\frac{\mathsf{d}a_d^2(t)}{\mathsf{d}t} \leq 0.$$

Proof. It is straightforward to see that

$$\begin{aligned}\frac{1}{2}\frac{\mathsf{d}a_d^2(t)}{\mathsf{d}t} &= a_d(t)u_d(t)v_d(t)\frac{\mathsf{d}w_d(t)}{\mathsf{d}t} + a_d(t)w_d(t)v_d(t)\frac{\mathsf{d}u_d(t)}{\mathsf{d}t} + w_d(t)u_d(t)a_d(t)\frac{\mathsf{d}v_d(t)}{\mathsf{d}t}\\ &= -u_d^4(t)a_d^2(t) - 2w_d^2(t)u_d^2(t)a_d^2(t) \leq 0\end{aligned}$$

where we used $u_d(t) = v_d(t)$ given by Lemma C.2, together with the dynamics in (11). □

Lemma C.4. *If $0 < u_d(0) = v_d(0) \leq w_d(0) \leq 1$, then $0 \leq u_d(t) = v_d(t) \leq w_d(t) \leq 1$.*

Proof. Leveraging (11) and Lemma C.2, it can be readily verified that

$$\frac{\mathsf{d}w_d^2(t)}{\mathsf{d}t} = \frac{\mathsf{d}u_d^2(t)}{\mathsf{d}t} = -2w_d(t)u_d(t)v_d(t)a_d(t) = -2a_d^2(t) \leq 0.$$

This implies

$$w_d^2(t) - w_d^2(0) = u_d^2(t) - u_d^2(0) := \Delta(t) \leq 0,$$

which proves $w_d^2(t) \geq u_d^2(t)$ as $w_d(0) \geq u_d(0) > 0$.

Note that

$$\Big|\frac{\mathsf{d}w_d(t)}{\mathsf{d}t}\Big| = u_d^2(t)|a_d(t)| = u_d^4(t)|w_d(t)| \leq u_d^4(0)|w_d(0)| \leq 1.$$

This suggests $w_d(t)$ is a polynomial ODE that is locally Lipschitz, it has no jump according to Picard–Lindelöf theorem. As $w_d(0) \in (0,1]$ and $w_d^2(t)$ is non-increasing, we must have $0 \leq w_d(t) \leq w_d(0) \leq 1$. Likewise, it also holds $0 \leq u_d(t) \leq u_d(0) < 1$.

Combining these results, we conclude that $0 \leq u_d(t) = v_d(t) \leq w_d(t) \leq 1$, which completes the proof. □

Lemma C.5. *If $0 < u_d(0) = v_d(0) \leq w_d(0) \leq 1$, then $w_d(t) \geq a_d^{1/3}(t)$.*

Proof. It directly follows from Lemmas C.2 and C.4 that

$$a_d^2(t) = u_d^4(t)w_d^2(t) \leq w_d^6(t).$$

Given that $w_d(t) \geq 0$ by Lemma C.4, we reach at $w_d(t) \geq a_d^{1/3}(t)$. □

C.2 Proof of Theorem 3.3

Similar to (9), the objective (4) can be simplified as

$$\mathcal{L}(t)=\frac{1}{2}\left\|\mathbf{W}_3(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t)+\mathbf{I}_d\big)\mathbf{X}-\mathbf{Y}\right\|_{\mathrm{F}}^2=\frac{1}{2}\left\|\mathbf{W}_3(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t)+\mathbf{I}_d\big)-\mathbf{A}\right\|_{\mathrm{F}}^2. \tag{12}$$

The associated optimization dynamics are

$$\mathbf{E}(t):=\mathbf{W}_3(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t)+\mathbf{I}_d\big)-\mathbf{A}, \tag{13a}$$

$$\nabla_{\mathbf{W}_1}\mathcal{L}(t)=\mathbf{W}_2(t)^\top\mathbf{W}_3^\top(t)\mathbf{E}(t), \tag{13b}$$

$$\nabla_{\mathbf{W}_2}\mathcal{L}(t)=\mathbf{W}_3(t)^\top\mathbf{E}(t)\mathbf{W}_1^\top(t), \tag{13c}$$

$$\nabla_{\mathbf{W}_3}\mathcal{L}(t)=\mathbf{E}(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t)+\mathbf{I}_d\big)^\top. \tag{13d}$$

And the gradient flow is the same as (10).

Theorem C.6 (Formal restatement of Theorem 3.3)**.** *If* $0<\|\mathbf{W}_i(0)\|_{\mathrm{F}}\le\delta:=\sqrt{\frac{\lambda}{2}}\exp\{-\frac{\sqrt{\lambda\mathcal{L}(0)}}{(1-\lambda)^2}-\frac{\sqrt{2\pi}\mathcal{L}(0)}{2(1-\lambda)^3}\}$, $i=1,2,3$ *for some* $\lambda\in(0,1)$*, then gradient flow* (10) *achieves linear convergence*

$$\mathcal{L}(t)\le\mathcal{L}(0)e^{-2(1-\lambda)^2t},\ \forall t>0.$$

Proof. Recall from the last subsection that the ODEs involved are polynomial and locally Lipschitz, and thus have no jump. Then we can define $T:=\min\{t\ge0\mid\|\mathbf{W}_2(t)\mathbf{W}_1(t)\|\ge\lambda\}$ to be the first time satisfying $\|\mathbf{W}_2\mathbf{W}_1\|\ge\lambda$. Applying Lemma C.8, it follows that

$$\mathcal{L}(t)\le\mathcal{L}(0)e^{-2(1-\lambda)^2t},\ \forall t\le T.$$

Next applying Lemma C.11, we acquire for all $t\le T$ that

$$\begin{aligned}
\|\mathbf{W}_2(t)\mathbf{W}_1(t)\|_2&\le\|\mathbf{W}_1(t)\|_{\mathrm{F}}\|\mathbf{W}_2(t)\|_{\mathrm{F}}\le\frac{1}{2}\Big(\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2+\|\mathbf{W}_2(t)\|_{\mathrm{F}}^2\Big)\\
&\le\frac{1}{2}\Big(\|\mathbf{W}_1(0)\|_{\mathrm{F}}^2+\|\mathbf{W}_2(0)\|_{\mathrm{F}}^2\Big)\exp\left\{\frac{2\delta\sqrt{2\mathcal{L}(0)}}{(1-\lambda)^2}+\frac{\sqrt{2\pi}\mathcal{L}(0)}{(1-\lambda)^3}\right\}\\
&\le\delta^2\exp\left\{\frac{2\delta\sqrt{2\mathcal{L}(0)}}{(1-\lambda)^2}+\frac{\sqrt{2\pi}\mathcal{L}(0)}{(1-\lambda)^3}\right\}\\
&=\frac{\lambda}{2}\exp\left\{\frac{2\sqrt{2\mathcal{L}(0)}}{(1-\lambda)^2}\left(\delta-\sqrt{\frac{\lambda}{2}}\right)\right\}\\
&\overset{(a)}{\le}\frac{\lambda}{2}<\lambda
\end{aligned}$$

where (a) is due to $\delta=\sqrt{\frac{\lambda}{2}}\exp\{-\frac{\sqrt{\lambda\mathcal{L}(0)}}{(1-\lambda)^2}-\frac{\sqrt{2\pi}\mathcal{L}(0)}{2(1-\lambda)^3}\}\le\sqrt{\frac{\lambda}{2}}$. As $\|\mathbf{W}_2(t)\mathbf{W}_1(t)\|_2$ has no jump, this indicates our choice of δ and λ ensures $\|\mathbf{W}_2(t)\mathbf{W}_1(t)\|_2<\lambda$ always holds; that is, $T=+\infty$. □

Lemma C.7. *Under gradient flow* (10)*, we have that*

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}t}=-\sum_{i=1}^3\|\nabla_{\mathbf{W}_i}\mathcal{L}(t)\|_{\mathrm{F}}^2\le0.$$

Proof. It is straightforward to see that

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}t}&=\left\langle\nabla_{\mathbf{W}_1}\mathcal{L}(t),\frac{\mathrm{d}\mathbf{W}_1}{\mathrm{d}t}\right\rangle+\left\langle\nabla_{\mathbf{W}_2}\mathcal{L}(t),\frac{\mathrm{d}\mathbf{W}_2}{\mathrm{d}t}\right\rangle+\left\langle\nabla_{\mathbf{W}_3}\mathcal{L}(t),\frac{\mathrm{d}\mathbf{W}_3}{\mathrm{d}t}\right\rangle\\
&=-\|\nabla_{\mathbf{W}_1}\mathcal{L}(t)\|_{\mathrm{F}}^2-\|\nabla_{\mathbf{W}_2}\mathcal{L}(t)\|_{\mathrm{F}}^2-\|\nabla_{\mathbf{W}_3}\mathcal{L}(t)\|_{\mathrm{F}}^2\le0.
\end{aligned}$$

This concludes the proof. □

Lemma C.8. *For $\lambda \in (0,1)$, defining $T := \min\{t \geq 0 \mid \|\mathbf{W}_2(t)\mathbf{W}_1(t)\| \geq \lambda\}$, then the loss converges linearly at*

$$\mathcal{L}(t) \leq \mathcal{L}(0)e^{-2(1-\lambda)^2 t},\ t \leq T.$$

Proof. We start with bounding

$$\|\nabla_{\mathbf{W}_3}\mathcal{L}(t)\|_{\mathrm{F}}^2 \overset{(a)}{\geq} \|\mathbf{E}(t)\|_{\mathrm{F}}^2 \sigma_d^2\big(\mathbf{I}_d + \mathbf{W}_2(t)\mathbf{W}_1(t)\big) \overset{(b)}{\geq} (1-\lambda)^2\|\mathbf{E}(t)\|_{\mathrm{F}}^2$$

where (a) is by Lemma C.13; and (b) comes from $\sigma_d(\mathbf{I}_d + \mathbf{W}_2(t)\mathbf{W}_1(t)) \geq 1 - \|\mathbf{W}_2(t)\mathbf{W}_1(t)\| \geq 1-\lambda$ when $t \leq T$.

Now based on Lemma C.7, it follows that

$$\frac{\mathsf{d}\mathcal{L}}{\mathsf{d}t} = -\sum_{i=1}^{3}\|\nabla_{\mathbf{W}_i}\mathcal{L}\|_{\mathrm{F}}^2 \leq -\|\nabla_{\mathbf{W}_3}\mathcal{L}\|_{\mathrm{F}}^2 \leq -(1-\lambda)^2\|\mathbf{E}(t)\|_{\mathrm{F}}^2 = -2(1-\lambda)^2\mathcal{L}(t).$$

This directly results in $\mathcal{L}(t) \leq \mathcal{L}(0)e^{-2(1-\lambda)^2 t}$. □

Lemma C.9. *Under gradient flow (10), it holds that*

$$\|\mathbf{W}_3(t)\|_{\mathrm{F}} \leq \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\mathcal{L}(0)}.$$

Proof. From the gradient flow, it can be seen that

$$\mathbf{W}_3(t) = \mathbf{W}_3(0) - \int_0^t \nabla_{\mathbf{W}_3}\mathcal{L}(s)\mathsf{d}s.$$

Further taking norm on both sides, we have that

$$\begin{aligned}
\|\mathbf{W}_3(t)\|_{\mathrm{F}} &\leq \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \Big\|\int_0^t \nabla_{\mathbf{W}_3}\mathcal{L}(s)\mathsf{d}s\Big\| \\
&\leq \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \int_0^t 1 \times \|\nabla_{\mathbf{W}_3}\mathcal{L}(s)\|_{\mathrm{F}}\mathsf{d}s \\
&\overset{(a)}{\leq} \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\int_0^t \|\nabla_{\mathbf{W}_3}\mathcal{L}(s)\|_{\mathrm{F}}^2\mathsf{d}s} \\
&\leq \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\int_0^t \sum_{i=1}^{3}\|\nabla_{\mathbf{W}_i}\mathcal{L}(s)\|_{\mathrm{F}}^2\mathsf{d}s} \\
&\overset{(b)}{=} \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\big(\mathcal{L}(0) - \mathcal{L}(t)\big)} \\
&\overset{(c)}{\leq} \|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\mathcal{L}(0)}
\end{aligned}$$

where (a) comes from Cauchy–Schwarz inequality; (b) uses Lemma C.7; (c) is because the loss is always greater than 0. □

Lemma C.10. *Under gradient flow (10), it follows that*

$$\frac{\mathsf{d}\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2}{\mathsf{d}t} = \frac{\mathsf{d}\|\mathbf{W}_2(t)\|_{\mathrm{F}}^2}{\mathsf{d}t}.$$

Proof. Note that

$$\frac{\mathsf{d}\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2}{\mathsf{d}t} = 2\Big\langle \mathbf{W}_1, \frac{\mathsf{d}\mathbf{W}_1(t)}{\mathsf{d}t}\Big\rangle = -2\mathrm{Tr}\big(\mathbf{W}_1^\top(t)\mathbf{W}_2^\top(t)\mathbf{W}_3^\top(t)\mathbf{E}(t)\big).$$

Similarly, we have

$$\begin{aligned}
\frac{\mathsf{d}\|\mathbf{W}_2(t)\|_{\mathrm{F}}^2}{\mathsf{d}t} &= 2\Big\langle \mathbf{W}_2, \frac{\mathsf{d}\mathbf{W}_2(t)}{\mathsf{d}t}\Big\rangle = -2\mathrm{Tr}\big(\mathbf{W}_2^\top(t)\mathbf{W}_3^\top(t)\mathbf{E}(t)\mathbf{W}_1^\top(t)\big) \\
&= -2\mathrm{Tr}\big(\mathbf{W}_1^\top(t)\mathbf{W}_2^\top(t)\mathbf{W}_3^\top(t)\mathbf{E}(t)\big).
\end{aligned}$$

The proof is thus completed. □

Lemma C.11. *When $t \leq T$, it holds that*

$$\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2 \leq \Big(\|\mathbf{W}_1(0)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(0)\|_{\mathrm{F}}^2\Big) e^M$$

where M is given by

$$M := \frac{2\sqrt{2\mathcal{L}(0)}\|\mathbf{W}_3(0)\|_{\mathrm{F}}}{(1-\lambda)^2} + \frac{\sqrt{2\pi}\mathcal{L}(0)}{(1-\lambda)^3}.$$

Proof. Using Lemma C.10, we obtain

$$\begin{aligned}
\frac{\mathsf{d}\|\mathbf{W}_1\|_{\mathrm{F}}^2 + \|\mathbf{W}_2\|_{\mathrm{F}}^2}{\mathsf{d}t} &= -4\mathrm{Tr}\big(\mathbf{W}_1^\top(t)\mathbf{W}_2^\top(t)\mathbf{W}_3^\top(t)\mathbf{E}(t)\big) \\
&\leq 4\|\mathbf{W}_1(t)\|_{\mathrm{F}}\|\mathbf{W}_2(t)\|_{\mathrm{F}}\|\mathbf{W}_3(t)\|_{\mathrm{F}}\|\mathbf{E}(t)\|_{\mathrm{F}} \\
&\leq 2\|\mathbf{W}_3(t)\|_{\mathrm{F}}\|\mathbf{E}(t)\|_{\mathrm{F}} \cdot \Big(\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2\Big) \\
&\overset{(a)}{\leq} 2\|\mathbf{W}_3(t)\|_{\mathrm{F}}\sqrt{2\mathcal{L}(0)}e^{-(1-\lambda)^2 t} \cdot \Big(\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2\Big) \\
&\overset{(b)}{\leq} 2\Big(\|\mathbf{W}_3(0)\|_{\mathrm{F}} + \sqrt{t\mathcal{L}(0)}\Big) \cdot \sqrt{2\mathcal{L}(0)}e^{-(1-\lambda)^2 t} \cdot \Big(\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2\Big)
\end{aligned}$$

where (a) comes from Lemma C.8, which shows that $\mathcal{L}(t) = \frac{1}{2}\|\mathbf{E}(t)\|_{\mathrm{F}}^2 \leq \mathcal{L}(0)e^{-2(1-\lambda)^2 t}$; and (b) is by Lemma C.9.

Now applying Lemma C.14, with $c_1 = 2\sqrt{2\mathcal{L}(0)}\|\mathbf{W}_3(0)\|_{\mathrm{F}}$, $c_2 = 2\sqrt{2}\mathcal{L}(0)$, $c_3 = (1-\lambda)^2$, and $x(t) = \|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2$, we obtain for $\forall t \leq T$ that

$$\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(t)\|_{\mathrm{F}}^2 \leq \Big(\|\mathbf{W}_1(0)\|_{\mathrm{F}}^2 + \|\mathbf{W}_2(0)\|_{\mathrm{F}}^2\Big) e^M, \;\; M = \frac{2\sqrt{2\mathcal{L}(0)}\|\mathbf{W}_3(0)\|_{\mathrm{F}}}{(1-\lambda)^2} + \frac{\sqrt{2\pi}\mathcal{L}(0)}{(1-\lambda)^3}.$$

The proof is thus completed. □

C.3 Additional auxiliary lemmas

This subsection provides useful lemmas for proving our main results.

Lemma C.12. *If $\frac{\mathsf{d}a(t)}{\mathsf{d}t} \geq -3a^2(t)$, it holds that $a(t) \geq \frac{a(0)}{1+3a(0)t}$.*

Proof. We can rewrite $\frac{\mathsf{d}a(t)}{\mathsf{d}t} \geq -3a^2(t)$ as

$$\frac{1}{a^2(t)}\frac{\mathsf{d}a(t)}{\mathsf{d}t} \geq -3.$$

Integrating both sides renders

$$-\frac{1}{a(s)}\Big|_0^t \geq -3t$$

which concludes the proof. □

Lemma C.13. *Given two matrices $\mathbf{A} \in \mathbb{R}^{r\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times r}$, it holds that $\|\mathbf{AB}\|_{\mathrm{F}}^2 \geq \sigma_r^2(\mathbf{A})\|\mathbf{B}\|_{\mathrm{F}}^2$.*

Proof. Let $\mathbf{b}_i$ be the ith column of $\mathbf{B}$. It then follows that

$$\begin{aligned}
\|\mathbf{AB}\|_{\mathrm{F}}^2 &= \sum_{i=1}^r \|\mathbf{A}\mathbf{b}_i\|_2^2 = \sum_{i=1}^r \Big\|\mathbf{A}\frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}\Big\|_2^2 \|\mathbf{b}_i\|_2^2 \\
&\geq \min_{\|\mathbf{u}\|=1}\|\mathbf{A}\mathbf{u}\|^2 \sum_{i=1}^r \|\mathbf{b}_i\|_2^2 = \sigma_r^2(\mathbf{A})\|\mathbf{B}\|_{\mathrm{F}}^2.
\end{aligned}$$

The proof is thus completed. □

Lemma C.14. *If $x(t) > 0$, $\forall t \le T$, and it holds for constant $c_1 \ge 0, c_2 \ge 0, c_3 > 0$ that*

$$\frac{\mathsf{d}x}{\mathsf{d}t} \le (c_1 + c_2\sqrt{t})e^{-c_3 t}x(t),\ \forall t \le T,$$

then we have

$$x(t) \le x(0)e^M,\ \forall t \le T$$

where $M := \frac{c_1}{c_3} + \frac{\sqrt{\pi}}{2}c_2 c_3^{-3/2}$.

Proof. To start with, since $x(t) > 0$, $\forall t \le T$, it follows

$$\frac{\mathsf{d}\ln x(t)}{\mathsf{d}t} = \frac{1}{x(t)}\frac{\mathsf{d}x(t)}{\mathsf{d}t} \le (c_1 + c_2\sqrt{t})e^{-c_3 t}.$$

Integrating both sides, we acquire

$$\begin{aligned}\ln x(t) - \ln x(0) &\le \int_0^t (c_1 + c_2\sqrt{s})e^{-c_3 s}\mathsf{d}s \\ &\le \int_0^\infty (c_1 + c_2\sqrt{s})e^{-c_3 s}\mathsf{d}s \\ &\le \frac{c_1}{c_3} + c_2\int_0^\infty \sqrt{s}e^{-c_3 s}\mathsf{d}s.\end{aligned}$$

Next, notice that

$$\int_0^\infty \sqrt{s}e^{-c_3 s}\mathsf{d}s \overset{(a)}{=} \frac{1}{c_3^{3/2}}\int_0^\infty \sqrt{u}e^{-u}\mathsf{d}u \overset{(b)}{=} \frac{1}{c_3^{3/2}}\Gamma(\frac{3}{2}) \overset{(c)}{=} c_3^{-3/2}\frac{1}{2}\Gamma(\frac{1}{2}) = \frac{\sqrt{\pi}}{2}c_3^{-3/2}$$

where (a) is by change of variable $u = c_3 s$; (b) uses the definition of Gamma function $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\mathsf{d}t$; and (c) relies on $\Gamma(z+1) = z\Gamma(z)$ and $\Gamma(1/2) = \sqrt{\pi}$.

Combining these together offers

$$\ln x(t) - \ln x(0) \le \frac{c_1}{c_3} + \frac{\sqrt{\pi}}{2}c_2 c_3^{-3/2} := M.$$

The proof is thus completed. □

C.4 Extension to more than 3 layers

This subsection extends Theorems C.1 and C.6 to LNNs with arbitrary depth $K \ge 3$. Following the preceding subsections, we define $\mathbf{A} := \mathbf{Y}\mathbf{X}^\top$ and work with the corresponding reduced objectives after whitening.

First consider the slow-convergence case, namely a K-layer LNN with the 0:1 shortcut

$$\mathcal{L}(t) := \frac{1}{2}\left\|\mathbf{W}_K(t)\cdots\mathbf{W}_2(t)\big(\mathbf{W}_1(t) + \mathbf{I}_d\big) - \mathbf{A}\right\|_{\mathrm{F}}^2. \tag{14}$$

Theorem C.15 (Extension of Theorem C.1)**.** *For any $K \ge 3$, there exists a problem instance of (14) with diagonally initialized weights, such that if*

$$\mathbf{W}_1[d,d](0) \in [-0.5, 0], \qquad \mathbf{W}_2[d,d](0) = \cdots = \mathbf{W}_K[d,d](0) \in (0, 0.5],$$

then gradient flow (1) satisfies

$$\mathcal{L}(t) \ge \Omega\left(1/t^{\frac{K-1}{K-2}}\right),\ \forall t > 0. \tag{15}$$

Proof. Similar to the proof of Theorem C.1, consider a rank-deficient diagonal matrix

$$\mathbf{A} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{r_A}, 0, \ldots, 0), \qquad r_A < d.$$

Since all weights are diagonally initialized, the gradient flow dynamics remain diagonal throughout training.

Likewise, we focus on the last coordinate, for which $\sigma_d = 0$, and define similarly

$$w(t) := 1 + \mathbf{W}_1[d,d](t), \qquad u_k(t) := \mathbf{W}_k[d,d](t), \quad k = 2, \dots, K.$$

The corresponding end-to-end scalar is

$$a(t) := w(t) \prod_{k=2}^{K} u_k(t). \tag{16}$$

The coordinate-wise gradient flow dynamics are

$$\frac{\mathsf{d}w(t)}{\mathsf{d}t} = -w(t) \prod_{k=2}^{K} u_k^2(t), \tag{17a}$$

$$\frac{\mathsf{d}u_j(t)}{\mathsf{d}t} = -w^2(t) u_j(t) \prod_{\substack{k=2 \\ k \neq j}}^{K} u_k^2(t), \qquad j = 2, \dots, K. \tag{17b}$$

Since $u_2(0) = \cdots = u_K(0)$ and the dynamics in (17b) are symmetric with respect to $u_2, \dots, u_K$, uniqueness of the solution gives

$$u_2(t) = \cdots = u_K(t) := u(t), \qquad \forall t \geq 0. \tag{18}$$

Consequently, it holds

$$a(t) = w(t) u^{K-1}(t), \tag{19}$$

and (17a)–(17b) reduce to

$$\frac{\mathsf{d}w(t)}{\mathsf{d}t} = -w(t) u^{2K-2}(t), \qquad \frac{\mathsf{d}u(t)}{\mathsf{d}t} = -w^2(t) u^{2K-3}(t). \tag{20}$$

It follows immediately that

$$\frac{\mathsf{d}w^2(t)}{\mathsf{d}t} = \frac{\mathsf{d}u^2(t)}{\mathsf{d}t} = -2w^2(t) u^{2K-2}(t) = -2a^2(t). \tag{21}$$

As a result, we obtain $w^2(t) - u^2(t) = w^2(0) - u^2(0) \geq 0$, where the last inequality follows from $w(0) = 1 + \mathbf{W}_1[d,d](0) \in [0.5, 1] \geq u(0) \in (0, 0.5]$.

Moreover, following the same process of Lemma C.4, it can be inferred from (20) and (21) that

$$0 < u(t) \leq w(t) \leq 1, \qquad 0 < a(t) = w(t) u^{K-1}(t) \leq 1, \qquad \forall t \geq 0. \tag{22}$$

Next, we derive the dynamics of $a(t)$. Using (19) and (20), we obtain

$$\begin{aligned} \frac{\mathsf{d}a(t)}{\mathsf{d}t} &= u^{K-1}(t) \frac{\mathsf{d}w(t)}{\mathsf{d}t} + (K-1) w(t) u^{K-2}(t) \frac{\mathsf{d}u(t)}{\mathsf{d}t} \\ &= -a(t) \left[u^{2K-2}(t) + (K-1) w^2(t) u^{2K-4}(t) \right]. \end{aligned} \tag{23}$$

Since $u(t) \leq w(t)$, we have

$$a(t) = w(t) u^{K-1}(t) \leq w^K(t),$$

and therefore

$$w(t) \geq a^{1/K}(t). \tag{24}$$

Denote by $q := \frac{2(K-2)}{K-1}$. Using (24), it holds that

$$u^{2K-2}(t) = \frac{a^2(t)}{w^2(t)} \leq a^{2-2/K}(t) \leq a^q(t)$$

where the last inequality follows from $0 < a(t) \leq 1$ and $2 - \frac{2}{K} \geq \frac{2(K-2)}{K-1}$.

Furthermore, it follows from $0 < w(t) \leq 1$ that

$$w^2(t)u^{2K-4}(t) \leq w^{\frac{2(K-2)}{K-1}}(t)u^{2K-4}(t) = \left[w(t)u^{K-1}(t)\right]^{\frac{2(K-2)}{K-1}} = a^q(t).$$

Substituting these two bounds into (23) yields

$$\frac{\mathsf{d}a(t)}{\mathsf{d}t} \geq -Ka^{1+q}(t).$$

Since $a(t) > 0$, this implies

$$\frac{\mathsf{d}a^{-q}(t)}{\mathsf{d}t} = -qa^{-q-1}(t)\frac{\mathsf{d}a(t)}{\mathsf{d}t} \leq Kq.$$

Integrating both sides gives

$$a(t) \geq \left(a^{-q}(0) + Kqt\right)^{-1/q}. \tag{25}$$

Finally, since the last diagonal entry of $\mathbf{A}$ is zero, it holds

$$\begin{aligned}\mathcal{L}(t) &\geq \frac{1}{2}a^2(t) \\ &\geq \frac{1}{2}\left(a^{-q}(0) + Kqt\right)^{-2/q} \\ &= \Omega\left(\frac{1}{t^{(K-1)/(K-2)}}\right)\end{aligned}$$

where we used $\frac{2}{q} = \frac{K-1}{K-2}$. This completes the proof. □

We next consider the fast-convergence case, namely a K-layer LNN with the $0{:}K-1$ shortcut, whose loss is

$$\mathcal{L}(t) := \frac{1}{2}\left\|\mathbf{W}_K(t)\left(\prod_{k=1}^{K-1}\mathbf{W}_k(t) + \mathbf{I}_d\right) - \mathbf{A}\right\|_{\mathrm{F}}^2. \tag{26}$$

For notational convenience, define

$$\mathbf{P}(t) := \prod_{k=1}^{K-1}\mathbf{W}_k(t) = \mathbf{W}_{K-1}(t)\cdots\mathbf{W}_1(t), \tag{27}$$

and error matrix

$$\mathbf{E}(t) := \mathbf{W}_K(t)\big(\mathbf{P}(t) + \mathbf{I}_d\big) - \mathbf{A}.$$

Then, the following theorem establishes the linear convergence of this residual topology.

Theorem C.16 (Extension of Theorem C.6)**.** *Let $K \geq 3$ and $\lambda \in (0,1)$. Define constants*

$$M_K := \frac{2\sqrt{2\mathcal{L}(0)}}{(1-\lambda)^2} + \frac{\sqrt{2\pi}\mathcal{L}(0)}{(1-\lambda)^3}, \tag{28a}$$

$$R_K := \left(\frac{\lambda}{2}\right)^{\frac{2}{K-1}}, \tag{28b}$$

$$\delta_K := \min\left\{1, \sqrt{\frac{R_K}{2}}\exp\left\{-\frac{M_K}{2}R_K^{(K-3)/2}\right\}\right\}. \tag{28c}$$

If $0 < \|\mathbf{W}_k(0)\|_{\mathrm{F}} \leq \delta_K$, $k = 1, \ldots, K$, then gradient flow (1) achieves linear convergence

$$\mathcal{L}(t) \leq \mathcal{L}(0)e^{-2(1-\lambda)^2 t},\ \forall t \geq 0. \tag{29}$$

Proof. We first establish the linear convergence rate as long as $\|\mathbf{P}(t)\|_2 < \lambda$. From (26), it follows

$$\nabla_{\mathbf{W}_K}\mathcal{L}(t) = \mathbf{E}(t)\big(\mathbf{P}(t) + \mathbf{I}_d\big)^\top.$$

Hence, by Lemma C.13, it follows

$$\|\nabla_{\mathbf{W}_K}\mathcal{L}(t)\|_{\mathrm{F}}^2 \geq \sigma_d^2\big(\mathbf{I}_d + \mathbf{P}(t)\big)\|\mathbf{E}(t)\|_{\mathrm{F}}^2 \geq (1-\lambda)^2\|\mathbf{E}(t)\|_{\mathrm{F}}^2$$

where we used $\sigma_d\big(\mathbf{I}_d + \mathbf{P}(t)\big) \geq 1 - \|\mathbf{P}(t)\|_2 > 1 - \lambda$.

Together with the gradient flow descent identity (cf. Lemma C.7)

$$\frac{\mathsf{d}\mathcal{L}(t)}{\mathsf{d}t} = -\sum_{k=1}^{K}\|\nabla_{\mathbf{W}_k}\mathcal{L}(t)\|_{\mathrm{F}}^2,$$

we can establish the linear convergence

$$\mathcal{L}(t) \leq \mathcal{L}(0)e^{-2(1-\lambda)^2 t} \tag{30}$$

whenever $\|\mathbf{P}(t)\|_2 < \lambda$.

We next control the norms of the first $K-1$ layers. For $j \in \{1, \ldots, K-1\}$, define

$$\mathbf{P}_{>j}(t) := \mathbf{W}_{K-1}(t)\cdots\mathbf{W}_{j+1}(t), \qquad \mathbf{P}_{<j}(t) := \mathbf{W}_{j-1}(t)\cdots\mathbf{W}_1(t).$$

Then we can write the gradient as

$$\nabla_{\mathbf{W}_j}\mathcal{L}(t) = \mathbf{P}_{>j}^\top(t)\mathbf{W}_K^\top(t)\mathbf{E}(t)\mathbf{P}_{<j}^\top(t).$$

Therefore, it holds

$$\begin{aligned}\frac{\mathsf{d}\|\mathbf{W}_j(t)\|_{\mathrm{F}}^2}{\mathsf{d}t} &= -2\mathrm{Tr}\left(\mathbf{W}_j^\top(t)\mathbf{P}_{>j}^\top(t)\mathbf{W}_K^\top(t)\mathbf{E}(t)\mathbf{P}_{<j}^\top(t)\right)\\ &= -2\mathrm{Tr}\left(\mathbf{P}^\top(t)\mathbf{W}_K^\top(t)\mathbf{E}(t)\right),\end{aligned} \tag{31}$$

where the last equality follows from the cyclic property of trace.

Thus, we obtain similarly

$$\frac{\mathsf{d}\|\mathbf{W}_1(t)\|_{\mathrm{F}}^2}{\mathsf{d}t} = \cdots = \frac{\mathsf{d}\|\mathbf{W}_{K-1}(t)\|_{\mathrm{F}}^2}{\mathsf{d}t}. \tag{32}$$

Now define $y(t) := \max_{1\leq k\leq K-1}\|\mathbf{W}_k(t)\|_{\mathrm{F}}^2$. Since all squared norms in (32) have the same derivative, their pairwise differences remain constant, and hence $y(t)$ is differentiable. Using (31) leads to

$$\begin{aligned}\frac{\mathsf{d}y(t)}{\mathsf{d}t} &\leq 2\|\mathbf{P}(t)\|_{\mathrm{F}}\|\mathbf{W}_K(t)\|_{\mathrm{F}}\|\mathbf{E}(t)\|_{\mathrm{F}}\\ &\leq 2\|\mathbf{W}_K(t)\|_{\mathrm{F}}\|\mathbf{E}(t)\|_{\mathrm{F}}y^{\frac{K-1}{2}}(t).\end{aligned} \tag{33}$$

Akin to Lemma C.9, the gradient flow descent identity yields

$$\|\mathbf{W}_K(t)\|_{\mathrm{F}} \leq \|\mathbf{W}_K(0)\|_{\mathrm{F}} + \sqrt{t\mathcal{L}(0)} \leq 1 + \sqrt{t\mathcal{L}(0)}, \tag{34}$$

where the last inequality follows from $\delta_K \leq 1$.

We next use a bootstrap argument similar to the proof of Theorem C.6. In particular, let T be the first time at which either $\|\mathbf{P}(t)\|_2 = \lambda$ or $y(t) = R_K$. For all $t \leq T$, combining (30), (33), and (34) gives

$$\begin{aligned}\frac{\mathsf{d}y(t)}{\mathsf{d}t} &\leq 2\big(1+\sqrt{t\mathcal{L}(0)}\big)\sqrt{2\mathcal{L}(0)}e^{-(1-\lambda)^2 t}y^{(K-1)/2}(t)\\ &\leq 2\big(1+\sqrt{t\mathcal{L}(0)}\big)\sqrt{2\mathcal{L}(0)}e^{-(1-\lambda)^2 t}R_K^{(K-3)/2}y(t).\end{aligned}$$

Applying Lemma C.14 yields

$$y(t) \leq y(0)\exp\left\{M_K R_K^{(K-3)/2}\right\},\ t \leq T. \tag{35}$$

Since $y(0) \leq \delta_K^2$, the choice (28c) gives

$$y(t) \leq \frac{R_K}{2},\ t \leq T. \tag{36}$$

Therefore, $y(t)$ cannot reach R_K. Moreover,

$$\begin{aligned}\|\mathbf{P}(t)\|_2 &\le \prod_{k=1}^{K-1} \|\mathbf{W}_k(t)\|_{\mathrm{F}} \\ &\le y^{(K-1)/2}(t) \\ &\le \left(\frac{R_K}{2}\right)^{(K-1)/2} \\ &< R_K^{(K-1)/2} = \frac{\lambda}{2} < \lambda.\end{aligned}$$

Hence $\|\mathbf{P}(t)\|_2$ cannot reach λ either. By continuity, such a finite T cannot exist, and therefore

$$\|\mathbf{P}(t)\|_2 < \lambda, \ \forall t \ge 0.$$

Applying (30) for all $t \ge 0$ proves the desired linear convergence. □

Theorems C.15 and C.16 extend the exponential discrepancy established for the 3-layer case to arbitrary depth. In particular, for every $K \ge 3$, there exist problem instances and sufficiently small initializations for which the 0:1 shortcut exhibits only polynomial convergence, whereas the $0{:}K-1$ shortcut guarantees linear convergence under sufficiently small initialization. Interestingly, the lower bound in (15) becomes slower as the network grows deeper, with exponent

$$\lim_{K\to\infty} (K-1)/(K-2) = 1.$$

D Integration with gradient checkpointing and pipeline parallelism

In practice, gradient checkpointing and pipeline parallelism/model sharding are standard solutions to training large models that do not fit into a single GPU. This appendix describes how to integrate ANCRe with these techniques.

For gradient checkpointing, a practical remedy is to restrict ingoing shortcuts to originate only from checkpointed layers, together with the immediately preceding layer. In our formulation, this corresponds to fixing $p_{ij} = 0$ strategically, thereby reducing the checkpointing burden while preserving the flexibility of learned residual topology. We believe this is more scalable to larger models and could still outperform vanilla training.

Furthermore, one can readily add to the training loss function a Tsallis entropy regularization on the shortcut parameters c_{ij}, which encourages sparse connectivity for softmax. For model sharding, this can be used to reduce the inter-GPU communication by sparsifying the cross-GPU connectivity.

Due to resource limitations, pre-training 1B models is at our full capacity, and we leave larger experiments as a future direction.

E Experimental setups

This appendix offers the detailed setups for reproducing the results reported in the main paper. All our codes are written in python. The LLM pretraining and RL tests are performed on a computing node with 4×A100 GPUs, while the diffusion models are trained on 2×H100 GPUs.

E.1 Case study

The case study is based on synthetic data, where $\mathbf{A}$ is generated from standard Gaussian with a low-rank constraint $r = 4$, $\mathbf{W}_i$ relies on small initialization $\mathcal{N}(0, 10^{-4})$, $d = n = 50$, the optimizer is standard GD with fixed step size of 10^{-3}.

E.2 Datasets

In the following, we provide a brief introduction to the datasets used in our tests.

Table 8: Hyperparameters for pre-training LLMs.

Hyperparameter	FullPT(+ANCRe)				GaLore(+ANCRe)			
	60M	130M	350M	1B	60M	130M	350M	1B
Max seq. len.	256							
Learning rate	5×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-4}	1×10^{-2}	5×10^{-2}	5×10^{-2}	5×10^{-3}
Training steps	10K	20K	60K	100K	10K	20K	60K	100K
Warmup steps	1K	2K	6K	10K	1K	2K	6K	10K
Global batch size	512							
Micro Batch size	256	128	64	16	256	128	64	16
Rank	-	-	-	-	128	256	256	512
GPUs	1×A100	1×A100	4×A100	2×H100	1×A100	1×A100	4×A100	2×H100

Colossal Clean Crawled Corpus (C4) [33] is a large-scale English text dataset derived from the April 2019 Common Crawl snapshot consisting of cleaned web text. It was introduced as part of the development of Google's T5 model [33], and has been widely used as a standard pretraining corpus for LLMs due to its scale, diversity, and high quality. The dataset is available online from TensorFlow datasets[3].

ImageNet-1K [9] is a large-scale image classification dataset containing approximately 1.28 million training images and 50,000 validation images annotated across 1,000 object categories. The dataset is a subset of the full ImageNet (ILSVRC-12) and is organized according to the WordNet hierarchy, with images collected from the web and manually labeled. ImageNet-1K has become a standard benchmark for training and evaluating visual recognition models, particularly in large-scale image classification and representation learning. The dataset is available online[4] for non-commercial research and/or educational purposes.

E.3 Models

Next, we elaborate the models adopted in our evaluation.

LLaMA [42] is a family of decoder-only transformer-based large language models introduced by Meta. In this work, LLaMA is implemented using the Hugging Face `transformers` library, following the official architecture definition[5].

Diffusion Transformer (DiT) [31] is a generative image model that replaces conventional convolutional U-Nets in diffusion models with a vision transformer operating on image patches. Reference implementations of DiT are publicly available[6] and released under the CC-BY-NC license.

Residual Networks (ResNets) [16] are a class of deep convolutional neural networks, characterized by residual connections that enable the effective training of very deep architectures. ResNets have become a foundational backbone for a wide range of tasks. Standard ResNet implementations[7] are released under permissive MIT license. Our test relies on the variants used in [47], where cascaded shortcuts are added to every block of four layers.

E.4 Hyperparameters

We next list the hyperparameters for running the experiments.

Pre-training of LLaMAs. The experimental setup follows [26, 57], with all hyperparameters except the learning rate fixed to their default values. We use the AdamW optimizer [28] with gradient accumulation. The learning rate is tuned over $\{5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}\}$; see Table 8 for further details. For GaLore, the update projection gap and scaling factor are set to

[3] https://www.tensorflow.org/datasets/catalog/c4

[4] https://image-net.org

[5] https://github.com/huggingface/transformers/blob/main/src/transformers/models/llama/modeling_llama.py

[6] https://github.com/facebookresearch/DiT

[7] https://github.com/KaimingHe/deep-residual-networks

Table 9: ANCRe temperature τ for RL tests.

Environment	ResNet-16	ResNet-64
Humanoid	0.01	0.01
Ant Big Maze	0.01	0.01
Arm Push Hard	0.1	0.1
Arm Binpick Hard	0.01	0.1

Table 10: Mean PPL and standard deviation across random seeds.

PPL	60M	130M	350M
FullPT	30.69±0.37	25.04±0.03	19.32±0.40
FullPT+ANCRe	29.58±0.05	24.50±0.03	18.46±0.15

their default values of 200 and 0.25, respectively. For ANCRe, the softmax temperature τ in (8) is fixed to 0.01 without additional tuning.

Pre-training of DiTs. The experimental settings are from [31] without any modifications, including the learning rate. Specifically, we use images of size 256×256, a global batch size of 256, a learning rate of 1×10^{-4}, a patch size of 2×2, and train for 80 epochs (i.e., 400K iterations) using BF16 precision. The ANCRe temperature is set to $\tau = 0.1$ for both DiT-S/2 and DiT-B/2. All experiments are conducted on two H100 GPUs.

RL with ResNets. The numerical experiments follow the experimental setup of [47]. Specifically, the episode length is set to 1000, and training is conducted for 100M environment steps. The number of training epochs and the batch size are 100 and 512, respectively, with 512 parallel environments. The learning rate is fixed at 3×10^{-4}. Both the actor and critic networks use a hidden dimension of 256, while the network depth is selected from $\{16, 64\}$. For ANCRe, the softmax temperature τ is coarsely tuned from $\{10^{-3}, 10^{-2}, 10^{-1}\}$, which can be found in Table 9.

At the start of pre-training, all models are randomly initialized in the default regime. ANCRe parameters c_{ij} are uniformly initialized.

E.5 Runs across random seeds

To assess the robustness of evaluation metric, we train ANCRe and FullPT with three random seeds {0,1,2} for the 60M, 130M, and 350M LLaMA models, and report the mean and standard deviation of the resultant PPLs in Table 10. We observed that ANCRe outperforms FullPT for every size and every seed, and the mean PPLs closely match Table 2. Notably, the error bars between ANCRe and FullPT do not overlap, which confirms that the gain is consistent rather than an effect of seed noise. Moreover, the learned shortcut coefficients are highly correlated across independent runs. For the three random seeds, the mean pairwise Pearson correlation of the learned coefficients is 0.92, 0.75, and 0.81 for the 60M, 130M, and 350M models, respectively. Moreover, the most heavily weighted incoming shortcut for each layer agrees across 73%–81% of layers. These results indicate that the learned topology is stable rather than an artifact of random initialization.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's contributions, including the theoretical analysis of residual topologies and advocated ANCRe method. They are supported by rigorous proofs in Appendix C and extensive numerical experiments in Section 5.

   Guidelines:

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

2. **Limitations**

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

Answer: [Yes]

Justification: We clearly stated in Sections 3 and 6 that our theoretical analysis focuses only on deep LNNs, and future work is needed to take into account the nonlinearities in modern architectures.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Formal theorems and detailed proofs are provided in Appendix C.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

Justification: Models, datasets, hyperparameters, and implementation details are all provided in Appendix E.

Guidelines:

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

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

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

Answer: [Yes]

Justification: Links to models and datasets are included in Appendix E. Codes are uploaded alongside with the submission.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix E specifies training and test setups per experiment, including hyperparameters and type of optimizers.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Standard deviation under multiple random seeds is reported in Appendix E.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The computational and memory efficiency of ANCRe are provided in Table 6. Compute resource details are offered in Section E.

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research is a study on neural network architecture and optimization, which adheres to the NeurIPS ethical code.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The broader impacts of this work, including both positive and negative sides, are explicitly discussed in Appendix B.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This work focuses on generic model architectural learning, rather than releasing specific model checkpoints with risks for misuse.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Datasets and models are properly credited, cited, and used under their corresponding licenses and terms.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper focuses on the analysis and model architecture of pre-training. No new assets are introduced.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor human subjects, so no IRB approval is required.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are experimental subjects rather than core method components.

    Guidelines:

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