ANCRE: ADAPTIVE NEURAL CONNECTION REASSIGNMENT FOR EFFICIENT DEPTH SCALING

Yilang Zhang[1]*, **Bingcong Li**[2]*, **Niao He**[2], **& Georgios B. Giannakis**[1]
[1]Department of ECE, University of Minnesota, Minneapolis, MN 55455, USA
[2]Department of CS, ETH Zürich, 8092 Zürich, Switzerland
{zhan7453,georgios}@umn.edu, {bingcong.li,niao.he}@inf.ethz.ch

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. 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.

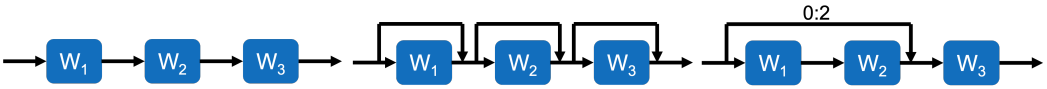
1 INTRODUCTION

Foundation models have demonstrated remarkable success across a broad spectrum of domains and impactful applications (Achiam et al., 2023; Chen et al., 2021; Ho et al., 2020; Radford et al., 2021). 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 (Grattafiori et al., 2024). Moreover, (Csordás et al., 2025) shows a clear positive correlation between depth and performance across 132 open-source LLMs (see their Figure 1). This trend is also supported by theories. It is proved in (Telgarsky, 2015) 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, (Csordás et al., 2025) 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 (Bolya et al., 2025). 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 (Srivastava et al., 2015; He et al., 2016a) 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 (Vaswani et al., 2017), introduces residual connections around each self-attention and feedforward network modules. From an optimization perspective,

*Equal contribution.



(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.

residual connections are credited with smoothing the loss landscape, which facilitates training by improving the (local) condition number (Li et al., 2018).

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[1]), 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 appropriately 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× 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 (Srivastava et al., 2015; He et al., 2016a). They were rapidly popularized in computer vision with plenty of variants (Bachlechner et al., 2021; Huang et al., 2017). This technique has also been widely adopted in modern foundation models (Team et al., 2025; Touvron et al., 2023; Yang et al., 2025; Peebles & Xie, 2023). More recently, hyper-connections (HC) (Zhu et al., 2025) and manifold-constrained hyper-connections (mHC) (Xie et al., 2025) 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*.

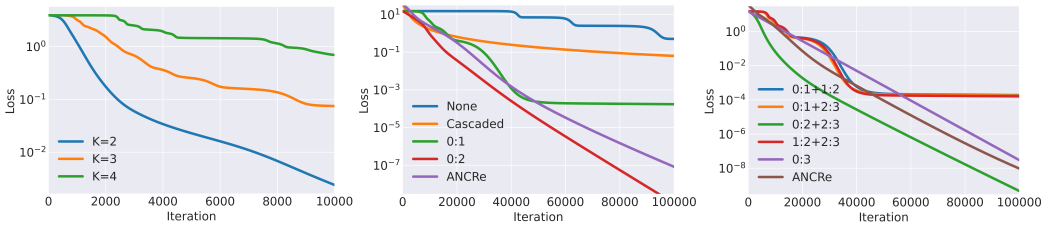
Understanding of residual connections. Residual connections are known to stabilize training by mitigating vanishing and exploding gradients (Haber & Ruthotto, 2017; Balduzzi et al., 2017). Empirical visualizations of ResNets also suggest that residual connections lead to a smoother loss landscape (Li et al., 2018). Theoretical understandings are often acquired by contrasting deep linear neural networks (Arora et al., 2019) with their residual counterparts. For example, the convergence of deep linear networks can degrade exponentially with depth (Shamir, 2019), whereas residual connections alleviate this slowdown (Wu et al., 2019). Moreover, residual connections can relax network width requirements (Du & Hu, 2019; Zou et al., 2020). 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 B.

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 (He et al., 2016a), residual connections allow each layer k to learn a residual mapping r_k relative to its input $\mathbf{x}_k$; i.e., $g_k(\mathbf{x}_k) := r_k(\mathbf{x}_k) + \mathbf{x}_k$. This architecture has been

[1]It coincides with "anchor" in French, which stabilizes a ship.



(a) Varying depth K (b) $K = 3$ (c) $K = 4$

Figure 2: Convergence comparison of LNN under varying setups.

extensively leveraged in foundation models 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 (Arora et al., 2019; Shamir, 2019). 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.

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

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

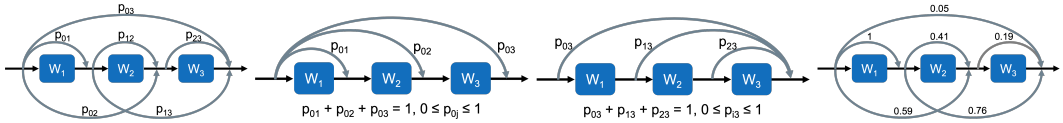
Using standard gradient descent (GD) for training, Figure 2a outlines the loss evolution with depth $K = 2, 3, 4$. Without the aid of residual connection, the convergence slows down when the network grows deeper. Figure 2b depicts the convergence behavior of a 3-layer LNN under different residual connection 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. 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 $t \ge 0$, and updates the weight matrix through ordinary differential equation

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



(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.

where the dependency of $\mathcal{L}$ on $\{\mathbf{W}_k(t)\}_k$ is omitted for compactness. 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 (Arora et al., 2019; Wu et al., 2019; Lion et al., 2025).

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 (Lower bound)**.** *Consider a 3-layer LNN with residual connection 0:1, and loss*

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

For <u>some</u> sufficiently small initialization, GF in (1) *cannot converge faster than a <u>sublinear</u> rate*

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

Theorem 3.3 (Upper bound)**.** *Consider a 3-layer LNN with residual connection 0:2, and 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 <u>any</u> sufficiently small initialization, GF in (1) *ensures <u>linear</u> 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. These theorems corroborate our observation in Figure 2b, and validate 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.

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. More broadly, an optimal layout can also depend on other architectures. 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 $2^{(L+1)L/2}$ 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. Outgoing normalization restricts the coefficients originating from layer i via a convex combination $\sum_{j=i+1}^{K} p_{ij} = 1$, $0 \leq p_{ij} \leq 1$, $\forall 0 \leq i < j \leq K$. That says, the feature embedding from layer i is decomposed as $\mathbf{x}_i = \sum_{j=i+1}^{K} p_{ij}\mathbf{x}_i$, and

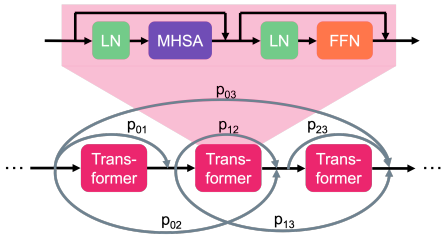


Figure 4: ANCRe applied to the standard Transformers comprising LN, MHSA, and FFN modules.

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**

each component is distributed to the subsequent layers. This ensures that the outgoing information per layer is bounded. Conversely, ingoing normalization imposes the following constraint $\sum_{i=0}^{j-1} p_{ij} = 1$, $0 \leq p_{ij} \leq 1$, $\forall 0 \leq i < j \leq K$, which preserves the input signal magnitude to layer j and enhances numerical stability. An alternative interpretation of this scheme is that it forms an *ensemble over network depths*. Letting $f_0(\mathbf{x}) := \mathbf{x}$ denote the initial 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 $p_{ij} = \exp(c_{ij}/\tau)/\sum_{k=1}^{j-1} \exp(c_{kj}/\tau)$, $\forall 0 \leq i < j \leq K$, where $\tau > 0$ is a temperature hyperparameter. 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* 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.

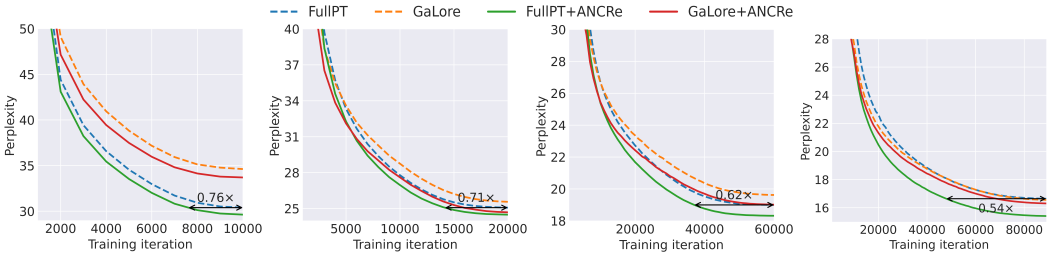
4.2 Applying ANCRe to Transformers

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 (Touvron et al., 2023) on the C4 dataset (Raffel et al., 2020). 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. Next, tests are conducted on foundation models across multiple domains to showcase ANCRe's effectiveness.

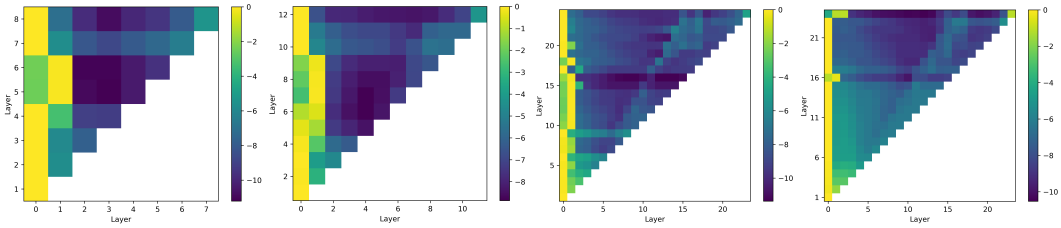
5 Numerical Experiments

This section assesses 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 hyperparameters are detailed in Appendix D.



(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 $\log(c_{ij}/c_{\max,j})$ learned by ANCRe, where horizontal and vertical axes stand for i and j.

5.1 Pre-training of LLMs

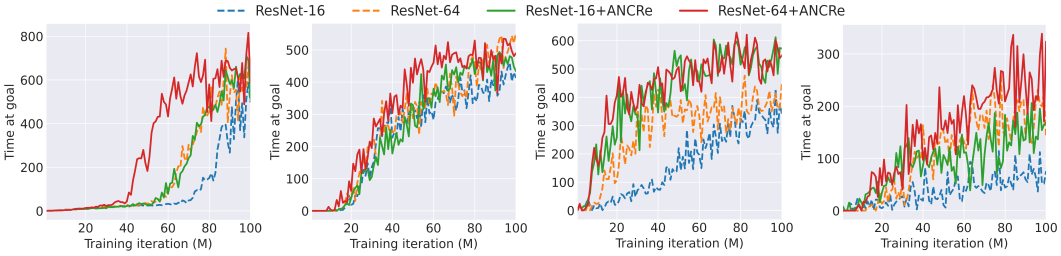
The evaluation of ANCRe begins with pre-training LLMs. Following the setups in (Lialin et al., 2024; Zhao et al., 2024), a series of LLaMA models (Touvron et al., 2023) are trained on the widely adopted C4 dataset (Raffel et al., 2020), which consists of cleaned English text collected across the Internet. 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 (Zhao et al., 2024), which applies low-rank gradient projection for memory efficiency. For both schemes, learning rates are selected from $\{5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$. ANCRe directly adopts the learning rate optimized for cascaded residual connections without additional tuning.

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 0.73 ± 0.33. 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.

Table 2: Perplexity (↓) comparison by pre-training LLaMA models of varying sizes. The better of the two are marked as bold.

Method	60M	130M	350M	1B
FullPT	30.39	25.07	19.00	16.64
+ANCRe	**29.62**	**24.48**	**18.32**	**15.41**
GaLore	34.61	25.56	19.62	16.55
+ANCRe	**33.69**	**24.68**	**19.01**	**16.45**
Tokens	1.1B	2.2B	6.4B	13.1B

In addition, Figure 6 sketches the residual coefficients learned by ANCRe. For clearer visualization, the heatmap is normalized by the largest ingoing coefficient $c_{\max,j} := \max_i c_{ij}$ per row j, and 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 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.



(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.

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) (Peebles & Xie, 2023). Specifically, class-conditional latent DiT-S/2 and DiT-B/2 are trained on the ImageNet-1K dataset (Krizhevsky et al., 2012), which contains over 1.4 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 (Peebles & Xie, 2023) without tuning. For evaluation, the standard Fréchet inception distance (FID) (Heusel et al., 2017) is calculated using 50,000 randomly sampled images from 250 DDPM steps (Ho et al., 2020), without classifier-free guidance (cfg) unless otherwise stated. Secondary metrics include spatial FID (sFID) (Nash et al., 2021), inception score (IS) (Salimans et al., 2016), and improved precision/recall (Kynkäänniemi et al., 2019).

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%
+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%
+ANCRe	**42.99**	**8.61**	**35.04**	**48.53%**	**55.29%**
DiT-B/2	44.31	8.42	32.89	47.93%	61.55%
+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%
+ANCRe	**20.53**	**5.81**	**70.45**	**65.91%**	**56.51%**

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.

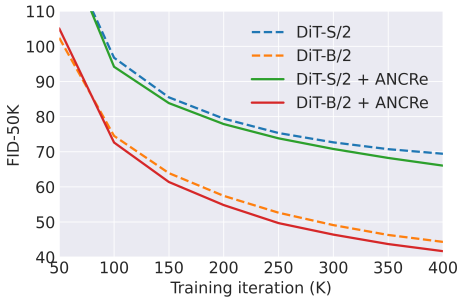


Figure 7: FID-50K evolution when pre-training DiT models.

5.3 Reinforcement learning with ResNets

The third test examines reinforcement learning (RL) with "actions" as the modality. The test settings follow from (Wang et al., 2025), 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. 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 expert demonstrations. Consequently, it necessitates deeper models of sufficient representational capacity. The models are ResNets (He et al., 2016a) 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 Gymnasium (Towers et al., 2025). Training is performed with the contrastive RL (CRL) algorithm (Eysenbach et al., 2022) for 100 million environment steps.

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 of numerical tests perform ablation studies to justify ANCRe's design choices. The setups are as in Section 5.1 using the LLaMA-130M and LLaMA-350M models with FullPT.

The first test compares the two normalization schemes 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 the outgoing one. This is because the former normalizes the inputs to each layer, so that computations are numerically stabler.

Table 4: Perplexity with different normalization schemes.

Normalization	130M	350M
None	diverge	diverge
Outgoing	25.03	18.63
Ingoing	**24.48**	**18.32**

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.

Table 5: Ablation study using all or learnable shortcuts.

All $i{:}j$	Learnable	130M	350M
√	×	diverge	diverge
×	√	27.18	21.72
√	√	**24.48**	**18.32**

Finally, we measure the runtime and GPU memory usage to assess the efficiency of ANCRe. As shown in Table 6, incorporating ANCRe introduces negligible overhead across all model scales. In particular, the additional runtime remains within 1%, while the increase in peak GPU memory is consistently below 0.12GB. This efficiency arises because the forward and 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.

Table 6: Overhead comparison.

Model, GPU, BS	ANCRe	Runtime	Memory
LLaMA-60M,	×	6h55min	29.59GB
1×A100, 256	√	6h57min	29.71GB
LLaMA-130M,	×	14h45min	23.26GB
1×A100, 128	√	14h46min	23.36GB
LLaMA-350M	×	17h25min	27.21GB
4×A100, 64	√	17h32min	27.28GB

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. Looking ahead, several directions have been added to the research agenda. First, the convergence analysis will be broadened to more general models. Second, more effective schemes for parameterizing and normalizing residual connections tailored to foundation models will be explored. Finally, pre-training tests will be scaled to larger models, for which efficient depth utilization becomes increasingly critical.

REFERENCES

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.

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.

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, pp. 1352–1361, 27–30 Jul 2021.

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)*, pp. 342–350, 2017.

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.

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.

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

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.

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

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, pp. 35603–35620, 2022.

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.

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

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 2016a.

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

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.

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

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.

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)*, pp. 4700–4708, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

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.

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.

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.

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

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

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

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

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, pp. 8748–8763, 18–24 Jul 2021.

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.

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.

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, pp. 2691–2713, 25–28 Jun 2019.

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

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.

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

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

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.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

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

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.

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.

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.

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.

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, pp. 10524–10533, 13–18 Jul 2020.

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.

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

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.

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*, pp. 61121–61143, 21–27 Jul 2024.

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.

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

A IMPACT STATEMENT

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. Nevertheless, caution is warranted when deploying the proposed method for generative applications. In such contexts, the outputs of language models should be carefully monitored, and appropriate safeguards such as gating or filtering mechanisms should be employed to ensure the safety, reliability, and trustworthiness of the generated content.

B 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 (He et al., 2016b) and transformers (Xiong et al., 2020) has been shown to improve training stability. Moreover, it is shown in (De & Smith, 2020) 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 (Wang et al., 2024) and vision transformers (Touvron et al., 2021). 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., (Zhang et al., 2019; De & Smith, 2020). 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, (Zhang et al., 2025) studies the gauge invariance underlying the optimization of LoRA (Hu et al., 2022), while (Lion et al., 2025) 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.

C MISSING PROOFS

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

Additional notation. $\langle \cdot, \cdot \rangle_{\mathrm{F}}$, $\mathrm{Tr}(\cdot)$, $\mathrm{rank}(\cdot)$ represent 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.

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

$$\mathcal{L}(t) = \frac{1}{2}\left\|\mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t)+\mathbf{I}_d\big)\mathbf{X}-\mathbf{Y}\right\|_{\mathrm{F}}^2 = \frac{1}{2}\left\|\mathbf{W}_3(t)\mathbf{W}_2(t)\big(\mathbf{W}_1(t)+\mathbf{I}_d\big)-\mathbf{Y}\mathbf{X}^\top\right\|_{\mathrm{F}}^2. \quad (6)$$

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

$$\frac{\mathsf{d}\mathbf{W}_1(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_1}\mathcal{L}(t),\ \frac{\mathsf{d}\mathbf{W}_2(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_2}\mathcal{L}(t),\ \frac{\mathsf{d}\mathbf{W}_3(t)}{\mathsf{d}t} = -\nabla_{\mathbf{W}_3}\mathcal{L}(t). \tag{7}$$

Theorem C.1 (Formal restatement of Theorem 3.2)**.** *There exists a problem instance of* (6) *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* (7) *can only converge sublinearly at*

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

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 as

$$a_i(t) := w_i(t)u_i(t)v_i(t), \tag{8a}$$

$$\frac{\mathsf{d}w_i(t)}{\mathsf{d}t} = -u_i(t)v_i(t)\big(a_i(t) - \sigma_i\big), \tag{8b}$$

$$\frac{\mathsf{d}u_i(t)}{\mathsf{d}t} = -w_i(t)v_i(t)\big(a_i(t) - \sigma_i\big), \tag{8c}$$

$$\frac{\mathsf{d}v_i(t)}{\mathsf{d}t} = -w_i(t)u_i(t)\big(a_i(t) - \sigma_i\big). \tag{8d}$$

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 (8) 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

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

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

□

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* (7)*, it holds that*

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

Proof. It is straightforward to see that

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

where we used $u_d(t) = v_d(t)$ given by Lemma C.2, together with the dynamics in (8). □

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 (8) 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

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

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* $w(t) \geq a_d^{1/3}(t)$.

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 (6), the objective (4) can be simplified as

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

The associated optimization dynamics are

$$\mathbf{E}(t) := \mathbf{W}_3(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t) + \mathbf{I}_d\big) - \mathbf{A}, \quad (10a)$$

$$\nabla_{\mathbf{W}_1}\mathcal{L}(t) = \mathbf{W}_2(t)^\top\mathbf{W}_3^\top(t)\mathbf{E}(t), \quad (10b)$$

$$\nabla_{\mathbf{W}_2}\mathcal{L}(t) = \mathbf{W}_3(t)^\top\mathbf{E}(t)\mathbf{W}_1^\top(t), \quad (10c)$$

$$\nabla_{\mathbf{W}_3}\mathcal{L}(t) = \mathbf{E}(t)\big(\mathbf{W}_2(t)\mathbf{W}_1(t) + \mathbf{I}_d\big)^\top. \quad (10d)$$

And the gradient flow is the same as (7).

Theorem C.6 (Formal restatement of Theorem 3.3)**.** *If* $\|\mathbf{W}_i(0)\|_{\mathrm{F}} \leq \delta := \sqrt{\frac{\lambda}{2}}\exp\{-\frac{\sqrt{\lambda\mathcal{L}(0)}}{(1-\lambda)^2} - \frac{\sqrt{2\pi}\mathcal{L}(0)}{2(1-\lambda)^3}\}$, $i = 1, 2, 3$ *for some* $\lambda \in (0, 1)$*, then gradient flow* (7) *achieves linear convergence*

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

Proof. Recall from the last subsection that the ODEs involved are polynomial and locally Lipschitz, and thus have no jump. Then we can define $T := \min\{t \geq 0 \mid \|\mathbf{W}_1(t)\mathbf{W}_2(t)\| \geq \lambda\}$ to be the first time satisfying $\|\mathbf{W}_1\mathbf{W}_2\| \geq \lambda$. Applying Lemma C.8, it follows that

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

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

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

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

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

$$\frac{\mathsf{d}\mathcal{L}}{\mathsf{d}t} = -\sum_{i=1}^{3}\|\nabla_{\mathbf{W}_i}\mathcal{L}(t)\|_{\mathrm{F}}^2 \leq 0.$$

Proof. It is straightforward to see that

$$\begin{aligned}
\frac{\mathsf{d}\mathcal{L}}{\mathsf{d}t} &= \Big\langle\nabla_{\mathbf{W}_1}\mathcal{L}(t), \frac{\mathsf{d}\mathbf{W}_1}{\mathsf{d}t}\Big\rangle + \Big\langle\nabla_{\mathbf{W}_2}\mathcal{L}(t), \frac{\mathsf{d}\mathbf{W}_2}{\mathsf{d}t}\Big\rangle + \Big\langle\nabla_{\mathbf{W}_3}\mathcal{L}(t), \frac{\mathsf{d}\mathbf{W}_3}{\mathsf{d}t}\Big\rangle \\
&= -\|\nabla_{\mathbf{W}_1}\mathcal{L}(t)\|_{\mathrm{F}}^2 - \|\nabla_{\mathbf{W}_2}\mathcal{L}(t)\|_{\mathrm{F}}^2 - \|\nabla_{\mathbf{W}_3}\mathcal{L}(t)\|_{\mathrm{F}}^2 \leq 0.
\end{aligned}$$

This concludes the proof. □

Lemma C.8. *For* $\lambda \in (0, 1)$*, defining* $T := \min\{t \geq 0 \mid \|\mathbf{W}_1(t)\mathbf{W}_2(t)\| \geq \lambda\}$*, 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 $\sigma_d(\mathbf{I}_d + \mathbf{W}_1(t)\mathbf{W}_2(t)) \geq 1 - \|\mathbf{W}_1(t)\mathbf{W}_2(t)\| \geq 1 - \lambda$ 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* (7)*, 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* (7)*, 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

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

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

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

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

The proof is thus completed. □

C.3 Additional auxiliary lemmas

This subsection provides useful lemmas for proving our main results.

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

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

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

Integrating both sides renders

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

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* $\|\mathbf{AB}\|_{\mathrm{F}}\ge\sigma_r(\mathbf{A})\|\mathbf{B}\|_{\mathrm{F}}^2$.

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

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

The proof is thus completed. □

Lemma C.14. *If* $x(t)\ge0$, $\forall t\le T$*, and it holds for constant* $c_1\ge0,c_2\ge0,c_3>0$ *that*

$$\frac{\mathsf{d}x}{\mathsf{d}t}\le(c_1+c_2\sqrt{t})e^{-c_3t}x(t),\ \forall t\le T,$$

then we have

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

where $M:=\frac{c_1}{c_3}+\frac{\sqrt{\pi}}{2}c_2c_3^{-3/2}$.

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

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

Integrating both sides, we acquire

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

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

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

The proof is thus completed. □

C.4 Extension to more than 3 layers

The proof for LLNs of more than 3 layers follows the same strategy as the proceeding two theorems. To avoid redundancy, we provide a proof sketch and omit the step-by-step details.

For the slow-convergence case (i.e., K-layer LNN with the 0:1 shortcut), one can construct a diagonal initialization with $\mathbf{W}_1[d,d](0) \in [-0.5, 0]$, and $\mathbf{W}_2[d,d](0) = \ldots = \mathbf{W}_K[d,d](0) \in (0, 0.5]$, and likewise a rank-deficient diagonal matrix $\mathbf{A}$. By following the dynamic analysis in Lemma C.2, it can be shown that $\mathbf{W}_2[d,d](t) = \ldots = \mathbf{W}_K[d,d](t)$, $\forall t \geq 0$. As a consequence, Lemma C.4 can be extended to establish that $0 \leq \mathbf{W}_2[d,d](t) = \ldots = \mathbf{W}_K[d,d](t) \leq 1 + \mathbf{W}_1[d,d](t) \leq 1$, $\forall t \geq 0$. To this end, applying the same arguments as in the proof of Theorem C.1 to the dynamics of $a_d(t) := (1 + \mathbf{W}_1[d,d](t))\prod_{k=2}^K \mathbf{W}_k[d,d](t)$ yields the sublinear convergence.

In contrast, K-layer LNN with the $0{:}K-1$ shortcut can be shown to always converge linearly. The core idea is to replace the term $\|\mathbf{W}_1(t)\mathbf{W}_2(t)\|$ in the original proof with the more generic $\|\prod_{k=1}^{K-1}\mathbf{W}_k(t)\|$, and to generalize $\|\mathbf{W}_3\|_{\mathrm{F}}$ in Lemma C.9 to $\|\mathbf{W}_K\|_{\mathrm{F}}$. Accordingly, Lemma C.10 is extended to

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

Finally, with the same steps in the proof of Theorem C.6, linear convergence then follows directly upon choosing an appropriate δ.

D 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.

Table 7: 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
GPUs	1×A100	1×A100	4×A100	2×H100	1×A100	1×A100	4×A100	2×H100

D.1 DATASETS

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

Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020) 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 (Raffel et al., 2020), 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[2].

ImageNet-1K (Krizhevsky et al., 2012) 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[3] for non-commercial research and/or educational purposes.

D.2 MODELS

Nest, we elaborate the models adopted in our evaluation.

LLaMA (Touvron et al., 2023) 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[4].

Diffusion Transformer (DiT) (Peebles & Xie, 2023) 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[5] and released under the CC-BY-NC license.

Residual Networks (ResNets) (He et al., 2016a) 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[6] are released under permissive MIT license. Our test relies on the variants used in (Wang et al., 2025), where cascaded shortcuts are added to every block of four layers.

D.3 HYPERPARAMETERS

We next list the hyperparameters for running the experiments.

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

[3] `https://image-net.org`

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

[5] `https://github.com/facebookresearch/DiT`

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

Table 8: 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

Pre-training of LLaMAs. The experimental setup follows (Lialin et al., 2024; Zhao et al., 2024), with all hyperparameters except the learning rate fixed to their default values. We use the AdamW optimizer (Loshchilov & Hutter, 2017) 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 7 for further details. For GaLore, the rank, update projection gap, and scaling factor are set to their default values of 128, 200, and 0.25, respectively. For ANCRe, the softmax temperature τ is fixed to 0.01 without additional tuning.

Pre-training of DiTs. The experimental settings are from (Peebles & Xie, 2023) 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 (Wang et al., 2025). 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 width 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 8.