How Does Loss Landscape Geometry Transfer Across Scales?

Anonymous authors
Paper under double-blind review

Abstract

Neural scaling limits, such as the maximal update parameterization (μP), have become a powerful tool for zero-shot hyperparameter transfer as model size grows. But which aspects of the loss landscape itself exhibit analogous transfer across scales? We study this question through the lens of *linear mode connectivity* (LMC), which asks whether models are connected in weight space by a linear path of low loss. We find evidence that interpolation barriers approach scale-consistent behavior, can already be stable at practically relevant widths and depths, and that such transfer can persist over long training horizons, beyond the regime of local quadratic approximation. Experiments with convolutional networks and transformers for vision and language show consistent behavior across width and depth under a range of training settings, and demonstrate that this consistency extends to practical weight-space operations such as model merging. As a theoretical foundation for our observations, we establish a scaling limit for loss barriers in a simplified setting. Our analysis offers a unifying perspective on neural scaling theory and weight space geometry, laying a foundation for understanding when interpolation, weight averaging, and model merging behave consistently across scales.

1 Introduction

Much of the recent progress in deep learning has come from training ever larger models on more data. Making effective use of a fixed compute budget requires choosing how large a model to train, for how long, and with which optimizer settings. Empirical scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022) can help allocate compute among model size and data, but searching over all possible training recipes becomes prohibitively expensive at scale.

Neural scaling limits address one axis of this problem by prescribing how initialization, normalization constants, and learning rates should scale with model size (i.e., width N and depth L), with the aim to maintain comparable learning dynamics across scales. In particular, the maximal update parameterization (μP) (Yang & Hu, 2021; Yang et al., 2021) and its depth extensions (Yang et al., 2024; Dey et al., 2025) allow hyperparameters (HPs) to be tuned on small proxy models and transferred to larger ones. Crucially, these parameterizations also preserve *feature learning* in the limit, i.e., internal representations continue to adapt during training, which is widely believed to be central to the empirical success of deep learning. By contrast, training under the neural tangent kernel (NTK) parameterization becomes increasingly *lazy* as width grows, approaching fixed-kernel limiting dynamics (Jacot et al., 2018; Chizat et al., 2019). At practical model sizes, feature-learning parameterizations have also been shown to preserve certain aspects of internal representations (Vyas et al., 2023) and loss curvature (Noci et al., 2024) across scales. This raises a broader question:

How much can we learn from smaller models about the loss landscape of larger ones?

We argue that, from an empirical perspective, scaling limits provide a natural framework for studying the loss landscape. As they control for potential confounders such as signal propagation, update scale, and feature-learning strength, they provide a principled way to understand which properties of the loss landscape are preserved across scales and which depend on model size.

In this work, we study the above question through *linear mode connectivity* (LMC), which asks whether two trained models are connected in weight space by a straight path of low loss. This

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

can occur when models share part of their training history (Frankle et al., 2020), after alignment under parameter symmetries (Singh & Jaggi, 2020; Ainsworth et al., 2023; Theus et al., 2025), or in architectures that break these (Lim et al., 2024; Bürgin et al., 2026). Specifically, we consider interpolation between models that share a training checkpoint and then continue with independent optimizer noise, and compare these pairs across width and depth at a fixed training horizon, while keeping HPs fixed under a scaling limit. Across a range of architectures, modalities, and HPs, we find that the global behavior of loss barriers can already agree at practical widths and depths. For depth, we find that *variance collapse* along the interpolation path remains controlled under proper scaling, i.e., both its magnitude and the benefit of REPAIR (Jordan et al., 2023) are consistent across depths. In some settings, proper depth scaling can even outperform post-hoc application of REPAIR.

Two competing effects determine the interpolation loss: weight averaging can capture some of the benefit of ensembling without its inference cost (Wortsman et al., 2022), but the nonlinear dependence of predictions on weights can also make the average perform poorly even when both endpoints perform well. We connect these terms to the Hessian and its Gauss-Newton decomposition (Martens, 2020), and track how their discrepancies evolve across sizes. In several settings, we observe that the ensemble term accumulates less finite-size discrepancy than the nonlinear interpolation term, helping explain why positive loss barriers often show stronger size dependence than negative ones.

Our findings extend beyond the local curvature studied through *super consistency* (Noci et al., 2024), as they concern pairs of models trained with independent optimization noise and can persist beyond the accuracy of a local quadratic approximation. Furthermore, they also qualify the intuition that landscapes of wider models simply become flatter: while overparameterization introduces flat directions (Simsek et al., 2021), these need not follow the path between independently trained models.

Beyond the foundational interest of understanding the loss landscape, we test whether this scale consistency extends to practical operations in weight space. In particular, we study model merging across scales and find evidence that smaller models can be predictive of merging behavior at larger widths and depths. This suggests that scaling limits can reduce the cost of exploring which models to combine and at which stage of training, extending their value from *training* to *combining* models.

Specifically, we make the following **contributions**:

- Across a range of architectures and modalities (vision and language, up to ∼500M parameters), scaling limits (μP/depth-μP (Yang & Hu, 2021; Yang et al., 2024), completeP (Dey et al., 2025)), and HPs, we find that interpolation losses can transfer across width N and depth L, beyond local quadratic approximation. We disentangle this transfer by separating ensembling gains from the effect of nonlinear parameter interpolation, relate both to loss curvature, and compare how finite-width/depth discrepancies accumulate over time.
- For depth, we find that variance collapse and the benefit of REPAIR (Jordan et al., 2023) remain consistent across scales, and proper scaling can even outperform post-hoc repair.
- On the theoretical side, we formulate the branched training process as a tensor program (Yang & Littwin, 2023) and use this framework to establish a scaling limit for loss barriers.
- We further observe that model merging can exhibit similar cross-scale consistency, which allows smaller models to predict merging behavior at larger widths and depths.

2 Related Work

We use scaling limits to compare models across scales, and linear mode connectivity to study the loss barriers between trained models. We review both topics here and discuss further related work in relation to our contributions, including on model merging, in § A.

Scaling limits. Infinite-width limits are a central theoretical tool for tractably analyzing neural network training. In the neural tangent kernel (NTK) limit, gradient descent reduces to fixed-kernel dynamics (Jacot et al., 2018), whereas mean-field limits retain feature learning (Mei et al., 2018; Chizat & Bach, 2018). The tensor program framework extends feature-learning limits to deep networks and yields the maximal update parameterization (μP) (Yang & Hu, 2021; Yang & Littwin, 2023), which enables HP transfer across widths (Yang et al., 2021). Depth-μP (Yang et al., 2024) and CompleteP (Dey et al., 2025) extend scaling prescriptions to depth. Beyond HP transfer, empirical studies ob-

serve consistency of representations (Vyas et al., 2023) and leading Hessian eigenvalues (Noci et al., 2024), and recent work characterizes evolving weight spectra (Lauditi et al., 2026). We extend this line of work to loss interpolation between models that continue training on separate trajectories.

Linear mode connectivity. The term *mode connectivity* was coined by works that observed low-loss curves between trained networks (Draxler et al., 2018; Garipov et al., 2018). *Linear* mode connectivity (LMC) (Frankle et al., 2020) asks whether straight-line interpolation has low loss. For models sharing the same initialization, prior work (Altıntaş et al., 2023; Singh et al., 2024; Altıntaş et al., 2025) analyzes how training choices affect LMC. For independently initialized models, alignment under *parameter symmetries* can reduce the loss barrier (Entezari et al., 2022; Singh & Jaggi, 2020; Ainsworth et al., 2023; Theus et al., 2025). In addition, Ferbach et al. (2024) prove LMC under permutation alignment in the mean-field limit via Wasserstein convergence of neuron distributions. We study a different question: whether loss barriers remain consistent as models grow wider or deeper.

3 Preliminaries

Model setup and scaling limits. We consider an architecture $f_{\boldsymbol{\theta}}$ of width N and depth L, parameterized by a weight vector $\boldsymbol{\theta}$. Let $\mathcal{F}_0$ denote an embedding, $\boldsymbol{h}_\ell$ the hidden representation and $\mathcal{F}_\ell$ a feature map at each block $\ell \in [L]$, and $\mathcal{F}_{L+1}$ a readout. We write the architecture schematically as

$$\boldsymbol{h}_\ell \;:=\; \tau \boldsymbol{h}_{\ell-1} + \frac{1}{\sqrt{N}L^\alpha}\boldsymbol{W}_\ell \mathcal{F}_\ell(\boldsymbol{h}_{\ell-1}), \qquad f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;:=\; \frac{1}{\gamma N}\boldsymbol{W}_{L+1}\mathcal{F}_{L+1}(\boldsymbol{h}_L(\boldsymbol{x})), \tag{1}$$

where $\boldsymbol{h}_0 := \mathcal{F}_0(\boldsymbol{x})$, and $\boldsymbol{\theta}$ collects all matrices $\boldsymbol{W}_\ell$ as well as potential other weights in $\mathcal{F}_\ell$. Further, τ controls the residual connection, α defines the depth normalization of the residuals, and γ the output scale. In the simplest case of a fixed-depth MLP, $\tau = \alpha = 0$, $\mathcal{F}_0 := \boldsymbol{W}_0/\sqrt{d_{\text{in}}}$, and $\mathcal{F}_\ell$ for $\ell \in [L+1]$ is simply a coordinate-wise nonlinearity $\varphi : \mathbb{R} \to \mathbb{R}$. Initialize the entries of $\boldsymbol{W}_\ell$ independently from $\mathcal{N}(0,1)$. In general, $\mathcal{F}_\ell$ can contain multiple layers, attention, normalization, and so on, whose definition (and scalings) we omit here for simplicity. Given a training set $(\boldsymbol{x}_i, y_i)_{i=1}^M$, we minimize $\mathcal{L}(\boldsymbol{\theta}) := 1/M \sum_{i=1}^M \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$ using a gradient-based optimizer with learning rate (LR) $\eta > 0$, where ℓ measures prediction error. Both the LR η and output scale γ may depend on N and L, which jointly determine the scaling limit. To achieve stable training as model size grows, a reasonable goal is to keep the scale of the update to the predictions $\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\}_i$ comparable across (N, L). Below, we first describe their scaling with width at fixed depth, and discuss the additional scaling required as depth grows (Yang et al., 2024; Bordelon et al., 2024b).

At fixed depth L (and fixed input and output dimensions), setting $\eta = \eta_0 \gamma^2 N$ keeps prediction updates comparable across widths for stochastic gradient descent (SGD), where $\eta_0 > 0$ is independent of N (Bordelon et al., 2024b). The remaining choice of γ controls how much *hidden features* change. For $\gamma_0 > 0$ independent of N, two standard choices are the maximal update parameterization (μP) (Yang & Hu, 2021) and the neural tangent parameterization (NTP) (Jacot et al., 2018):

$$(\gamma, \eta) \;:=\; \begin{cases} (\gamma_0,\, \eta_0 \gamma_0^2 N), & \text{μP}, \\ (\gamma_0 N^{-1/2},\, \eta_0 \gamma_0^2), & \text{NTP}. \end{cases} \tag{2}$$

While both scalings allow predictions to evolve as $N \to \infty$, μP retains changes in hidden features. Under NTP, however, these vanish, approaching fixed-kernel dynamics. We also consider the standard parameterization (SP), which uses fan-in-scaled Gaussian initialization and a width-independent LR. See § B for more details on the different scaling limits for width.

Extending feature-learning limits to additional depth scaling requires scaling the residual blocks and their LRs. Choosing any $\alpha \in [1/2, 1]$ in (1) and scaling the LR in residual blocks by $L^{2\alpha-1}$ under SGD would define a valid feature-learning depth limit, and different choices preserve different properties of the learning dynamics. Depth-μP (Yang et al., 2024; Bordelon et al., 2024b) uses $\alpha = 1/2$, which gives rise to an SDE-like scaling motivated by feature diversity across layers, while completeP (Dey et al., 2025) sets $\alpha = 1$ (a more neural-ODE-like limit; Chen et al., 2018) to retain the nonlinear dependence of the learned representations on parameter updates as $L \to \infty$. In our transformer experiments, we mainly use completeP with AdamW (Loshchilov & Hutter, 2019), and present the exact layerwise scaling rules in § E.1.3. We include comparisons to depth-μP in § F.

Training and interpolation setup. Next, we lay out the setup of the branching experiment used for the theory in § 4 and the LMC experiments, which essentially follows Frankle et al. (2020); Singh

et al. (2024). For each scaling rule, we keep the base HPs and the data budget fixed across model sizes, and we train for a fixed horizon of T steps (or epochs). At step b, where $0 \le b \le T$, we copy the training checkpoint and optimizer state. We then train the two copies separately until step T, drawing minibatches independently and using independent data augmentation when applicable. Write $\boldsymbol{\theta}^{(t)}$ for the shared parameters for $t \le b$, and $\boldsymbol{\theta}_A^{(t)}$ and $\boldsymbol{\theta}_B^{(t)}$ for the two independent trajectories at step $t \ge b$ (i.e., $\boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}_A^{(b)} = \boldsymbol{\theta}_B^{(b)}$). We define the *(signed) chord deviation* at time t as[1]

$$C^{(t)}(s) := \mathcal{L}\left((1-s)\,\boldsymbol{\theta}_A^{(t)} + s\boldsymbol{\theta}_B^{(t)}\right) - \left[(1-s)\,\mathcal{L}\left(\boldsymbol{\theta}_A^{(t)}\right) + s\mathcal{L}\left(\boldsymbol{\theta}_B^{(t)}\right)\right], \qquad s \in [0,1]. \tag{3}$$

In other words, $C^{(t)}(s)$ is the change of evaluation loss incurred by evaluating along the straight line between $\boldsymbol{\theta}_A^{(t)}$ and $\boldsymbol{\theta}_B^{(t)}$, compared to interpolating the endpoint losses. LMC corresponds to $C^{(t)}(s)$ staying small over all $s \in [0,1]$. We will primarily look at the midpoint $C^{(t)} := C^{(t)}(\frac{1}{2})$, which measures the loss change from averaging the two models' weights relative to their average loss.

4 Decomposition and Scaling of Loss Barriers

Motivating example. We compare different parameterizations to illustrate how the chord deviation in (3) changes as width N grows. We train a 3-layer ConvNet on CIFAR-10 with SGD for $T = 100$ epochs, branch at initialization ($b = 0$), and track the midpoint chord deviation throughout training. Fig. 1 demonstrates that: *(i)* under SP, the final chord deviation increases with width, with no apparent stabilization up to $N = 2048$, *(ii)* under NTP, the deviation generally decreases as the width grows, approaching zero or slightly negative values, and *(iii)* under μP, the curves stay close across the tested widths and remain positive at the end of training, suggesting a common nonzero limiting trajectory, i.e., $C_\infty^{(t)} = \lim_{N\to\infty} C_N^{(t)}$.

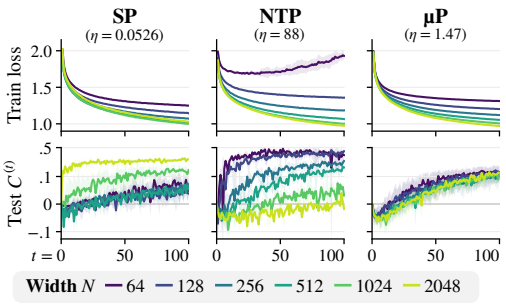


Figure 1: Behavior of training loss and test chord deviation across various widths under different parameterizations: SP, NTP, and μP. We train a 3-layer ConvNet on CIFAR-10 with SGD and branch at initialization ($b = 0$). Colors represent different widths.

To understand these differences, we first decompose (3) to disentangle the effects of weight interpolation from function interpolation, and connect them to the loss curvature (§ 4.1). We then ask whether these quantities approach limits as width grows and establish such limits under μP and NTP (§ 4.2). This analysis provides theoretical support for the scale consistency we observe empirically.

4.1 Decomposition and Quadratic Approximation of Chord Deviation

Fix an evaluation set $(\boldsymbol{x}_i, y_i)_{i=1}^M$ and a training step t, and collect all predictions as $F(\boldsymbol{\theta}) := (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))_{i=1}^M \in \mathbb{R}^{Md_{\text{out}}}$. Write $\ell(F(\boldsymbol{\theta})) = 1/M \sum_{i=1}^M \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$, so that $\mathcal{L}(\boldsymbol{\theta}) = \ell(F(\boldsymbol{\theta}))$. To compare weight and output averaging, let $\overline{\boldsymbol{\theta}} := (\boldsymbol{\theta}_A + \boldsymbol{\theta}_B)/2$ be the weight midpoint and $\overline{F} := (F(\boldsymbol{\theta}_A) + F(\boldsymbol{\theta}_B))/2$ the function midpoint. We can rewrite the chord deviation in (3) as

$$C = \underbrace{\left[\ell(F(\overline{\boldsymbol{\theta}})) - \ell(\overline{F})\right]}_{=: G \text{ (nonlinear interpolation gap)}} - \underbrace{\left[\tfrac{1}{2}\left(\ell(F(\boldsymbol{\theta}_A)) + \ell(F(\boldsymbol{\theta}_B))\right) - \ell(\overline{F})\right]}_{=: J \text{ (Jensen gap / ensembling gain)}}. \tag{4}$$

Here, the *Jensen gap* J measures the loss reduction obtained from averaging the outputs of the two models, which is ≥ 0 for convex ℓ (i.e., cross-entropy or L^2). The remaining *nonlinear interpolation gap* G measures the additional loss change incurred by the map F from parameter to function space not being linear along the chord. While G can have arbitrary signs, under feature learning weight interpolation can disrupt the learned representations and, thus, have higher loss compared to output averaging. Intuitively, as the two branches $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ converge to different solutions, greater prediction diversity increases J, while reduced compatibility of their representations can simultaneously drive up G. Hence, in this case, both effects would contribute in opposite directions to $C = G - J$.

[1] Here, we overload notation by also writing $\mathcal{L}$ for the evaluation loss.

In fact, the quadratic approximation of (4) corresponds precisely to the Gauss-Newton (GN) decomposition (Martens, 2020): If F and ℓ are C^2, at any reference point $\boldsymbol{\theta}_{\text{ref}}$ the chain rule yields

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} \;=\; \underbrace{\left[(\nabla_{\boldsymbol{\theta}} F)^\top \nabla^2 \ell(F) \nabla_{\boldsymbol{\theta}} F\right]}_{=:\, \boldsymbol{H}_{\text{GN}}\ \textbf{(Gauss-Newton)}} \;+\; \underbrace{\left[\textstyle\sum_a [\nabla \ell(F)]_a \nabla^2_{\boldsymbol{\theta}} F_a\right]}_{=:\, \boldsymbol{H}_{\text{rem}}\ \textbf{(remainder)}}, \tag{5}$$

where all derivatives of F and $\mathcal{L}$ are evaluated at $\boldsymbol{\theta}_{\text{ref}}$, and derivatives of ℓ at $F(\boldsymbol{\theta}_{\text{ref}})$. Specifically, $\boldsymbol{H}_{\text{GN}} \succeq 0$ is the loss curvature after linearizing $F(\boldsymbol{\theta}) \approx F(\boldsymbol{\theta}_{\text{ref}}) + \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}_{\text{ref}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{ref}})$, while the remainder $\boldsymbol{H}_{\text{rem}}$ accounts for the curvature of F itself.

To obtain a quadratic approximation of (4), we set $\boldsymbol{\theta}_{\text{ref}} := \overline{\boldsymbol{\theta}}$ and write $\boldsymbol{\Delta} := \boldsymbol{\theta}_B - \boldsymbol{\theta}_A$ for the interpolation direction. If F and ℓ are C^4 near $\overline{\boldsymbol{\theta}}$ and $F(\overline{\boldsymbol{\theta}})$, a symmetric Taylor expansion yields

$$C \;\approx\; -\tfrac{1}{8} \boldsymbol{\Delta}^\top \nabla^2_{\boldsymbol{\theta}} \mathcal{L}\, \boldsymbol{\Delta}, \qquad J \;\approx\; \tfrac{1}{8} \boldsymbol{\Delta}^\top \boldsymbol{H}_{\text{GN}} \boldsymbol{\Delta}, \qquad G \;\approx\; -\tfrac{1}{8} \boldsymbol{\Delta}^\top \boldsymbol{H}_{\text{rem}} \boldsymbol{\Delta}, \tag{6}$$

where each error is $O(\|\boldsymbol{\Delta}\|_2^4)$ as $\|\boldsymbol{\Delta}\|_2 \to 0$. Thus, locally, J and G correspond precisely to the GN and remainder contributions of the directional loss curvature in the interpolation direction. See § C for rigorous derivations of all the above statements.

4.2 Scaling Limits for Chord Deviations

We now ask whether the chord deviation C and its components, i.e., the Jensen gap J and nonlinear interpolation gap G, approach limits as $N \to \infty$. For this, we consider the branching setup from § 3 in the simplified case of a smooth, bias-free MLP trained with SGD at fixed depth over a fixed finite training horizon, and couple the evaluation set and minibatch history across all widths (we make this precise in assumption D.2). For µP, a limit for $(C(s), J(s), G(s))$ on finitely many $s \in [0, 1]$ can be obtained by representing both training branches $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ as a single tensor program (Yang & Littwin, 2023), and one can further obtain a uniform limit for the interpolation curves over all of $[0, 1]$. Namely, the following holds:

Theorem 4.1 (µP, informal). *As width $N \to \infty$, $C_N^{(t)}(s), G_N^{(t)}(s), J_N^{(t)}(s)$ converge jointly a.s. to limits satisfying $C_\infty^{(t)}(s) = G_\infty^{(t)}(s) - J_\infty^{(t)}(s)$, for any horizon t and uniformly in $s \in [0, 1]$.*

In this case, the limits are deterministic given the minibatch histories and do not depend on the initialization randomness. Further, neither $G_\infty^{(t)}$ nor $C_\infty^{(t)}$ is required to vanish, i.e., this convergence is compatible with positive or negative loss barriers. In contrast, the situation for NTP is as follows:

Theorem 4.2 (NTP, informal). *As width $N \to \infty$, $G_N^{(t)}(s) \to 0$ in probability and $(C_N^{(t)}(s), J_N^{(t)}(s)) \to (-J_\infty^{(t)}(s), J_\infty^{(t)}(s))$ in distribution, for any t and uniformly over $s \in [0, 1]$.*

Here, the limits can remain random (that is, depend on the initialization) even for fixed histories. Intuitively, one can also relate the NTP limit to the GN decomposition in (5): under NTP, $\boldsymbol{\Delta}^\top \nabla^2_{\boldsymbol{\theta}} F\, \boldsymbol{\Delta}$ vanishes along the entire chord as $N \to \infty$ (see § D and Liu et al. (2020)), so $F(\boldsymbol{\theta})$ stays close to its linearization. Consequently, the remainder contribution $\boldsymbol{\Delta}^\top \boldsymbol{H}_{\text{rem}} \boldsymbol{\Delta}$ vanishes, which leaves only the GN part, i.e., $C \to -J \leq 0$ for convex ℓ. We remark that the statements of thms. 4.1 and 4.2 are in line with the trends observed in fig. 1, where we see a nontrivial limiting behavior of C under µP, and a drift downwards for NTP consistent with $G \to 0$ (see § F.1.1 for more details). See § D for the formal statements and proofs of thms. 4.1 and 4.2.

Note that while thm. 4.1 establishes the existence of a limit at any fixed horizon t under µP, it makes no statement about how this limit would behave as also $t \to \infty$, or if $C_N^{(t)}$ already stabilizes, i.e., is close to $C_\infty^{(t)}$, at practically relevant widths. In fact, using the chord deviation in (3) to compare the loss landscape across scales is itself a design choice: C stabilizing at finite N would imply that the *additive* interpolation costs agree, even when the endpoint losses still depend significantly on N. Alternatively, one could just as well consider a *relative* version $\tilde{C} := \mathcal{L}\left((\boldsymbol{\theta}_A + \boldsymbol{\theta}_B)/2\right) / [(\mathcal{L}(\boldsymbol{\theta}_A) + \mathcal{L}(\boldsymbol{\theta}_B))/2]$. While reasonable arguments can be made for either choice (e.g., C for L^2 loss or $\tilde{C}$ in some cross-entropy scaling regimes; see § D.5), for simplicity, we use the additive chord deviation C as we empirically observe global consistency of C across realistic model sizes.

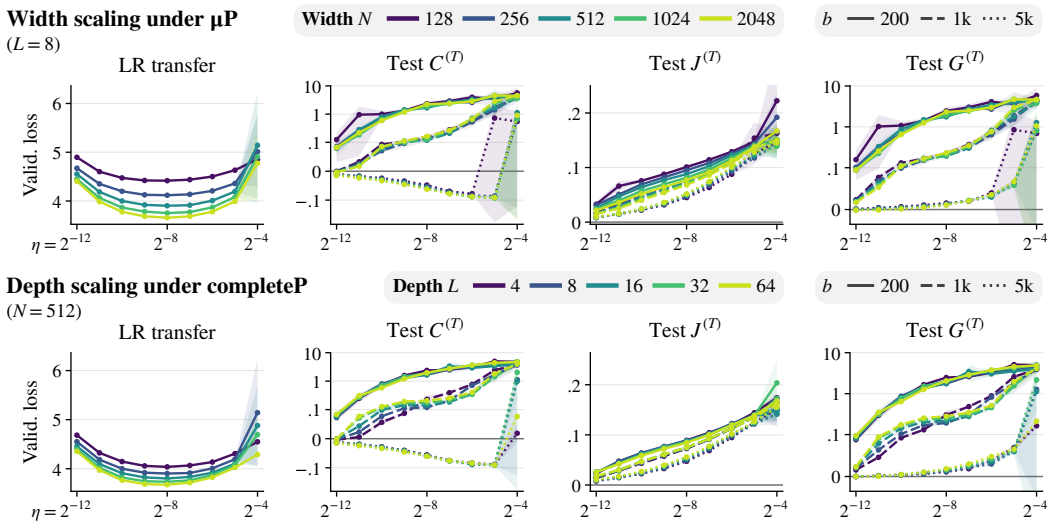


Figure 2: **Width and depth transfer in GPT-2-style transformers under completeP.** *Top:* width scaling at $L = 8$. *Bottom:* depth scaling at $N = 512$. The columns show validation loss and test $C^{(T)}, J^{(T)}, G^{(T)}$ after 300M OpenWebText tokens, curves report mean $\pm$ std. over 3 seeds.

5 Linear Mode Connectivity Across Scales

Having established a scaling limit for chord deviations (§ 4.2) and their decomposition into ensembling gain and nonlinear interpolation gap (§ 4.1), we now study how these quantities behave empirically. We focus on their *scale consistency*, i.e., whether a quantity $X_{N,L}(t)$ is already close to its limit $X_\infty(t)$ at realistic model sizes. When such an agreement persists without accumulation of finite-size discrepancies over training, Noci et al. (2024) call it *super consistency*, which they observe in the leading Hessian eigenvalues. This is not implied by a scaling limit, e.g., the loss can decrease with model size even when other properties of the loss landscape already agree closely.

5.1 Scale Consistency of Chord Deviations

To assess scale consistency at practical model sizes, we consider GPT-2-style transformers trained on OpenWebText (Gokaslan & Cohen, 2019) with a range of ∼8M–500M parameters. Our experiments are organized around the following main questions. **Q1:** *How consistent are chord deviations C across different finite widths N and depths L under scaling limits, at the same training horizon for different LRs?* **Q2:** *How do the Jensen gap J and nonlinear interpolation gap G contribute to the evolution of $C = G - J$ (4) over branching and training times b, t?* **Q3:** *How faithful is the quadratic approximation (6) of C and how does it help us understand its evolution?*

We use NanoGPT under completeP (Dey et al., 2025) width and depth scaling[2], and vary widths $N \in \{128, \dots, 2048\}$ at depth $L = 8$ (which gives a µP limit under the fixed completeP block scaling) as well as depths $L \in \{4, \dots, 64\}$ at width $N = 512$. We train on OpenWebText for 300M tokens (≈9.1k steps) using AdamW with a linear warmup of 200 steps and cosine decay. In this setup, we conduct the branching experiment of § 3 at times $b \in \{200, 1\text{k}, 5\text{k}\}$ and train each pair of models to the training horizon T. Fig. 2 reports final validation losses, as well as test $C^{(T)}, J^{(T)}, G^{(T)}$ across 9 base LRs each. We report further ablations in § F.1.3, including behavior under depth-µP. In § F.1.4, we show that similar empirical behavior extends to vision.

Addressing **Q1**, our experiments reveal the following:

Finding ①: *Under completeP, for realistic N and L the global behavior of $C^{(T)}_{N,T}, J^{(T)}_{N,T}, G^{(T)}_{N,T}$ is already close to its limits $C^{(T)}_\infty$, $J^{(T)}_\infty$, $G^{(T)}_\infty$, across base LRs η and branching times b.*

[2]We adapt the setup of https://github.com/EleutherAI/nanoGPT-mup/tree/completep.

Indeed, fig. 2 shows that while validation losses still change notably under width and depth scaling, $C^{(T)}, J^{(T)}, G^{(T)}$ at the training horizon T collapse across N and L for most peak base LRs η and branching times b, while there is significant variability among different values of η, b. There are some exceptions (e.g., for $J_N^{(T)}$ at $b = 200$ or for $G_L^{(T)}$ at $b = 1$k), which could be in part due to the additive version of $C/J/G$ not being the ideal scale-consistent observable in these cases (§ 4.2), or due to true finite-size effects. Yet, from the perspective of LMC, the loss landscape is already remarkably consistent at practical values of N and L, meaning that different model sizes respond similarly to branching at a certain time with the same base LR. In terms of **Q2**, fig. 2 further shows:

Finding ②a: *At horizon T, variation of the chord deviation C across branching times b is mainly driven by the nonlinear interpolation gap G, while the Jensen gap J depends less on it.*

One can clearly see that $G^{(T)}$ is strongly dependent on the branching time b (and base LR) and positive throughout, meaning that averaging weights after branching early in training has a disruptive effect on features, while after 5k steps it is almost zero, i.e., in that case $F(\boldsymbol{\theta})$ mapping from weight to function space is close to linear along the chord. In contrast, $J^{(T)}$ depends less on b, and we attribute most of the differences in fig. 2 to the LR schedule, as we find that $J^{(T)}$ collapses almost perfectly across different branching times b under a constant LR (see § F.1.2). We further see:

Finding ②b: *The nonlinear interpolation gap G consistently picks up more finite-size discrepancies with increasing time t, while in many cases the discrepancy of J even shrinks.*

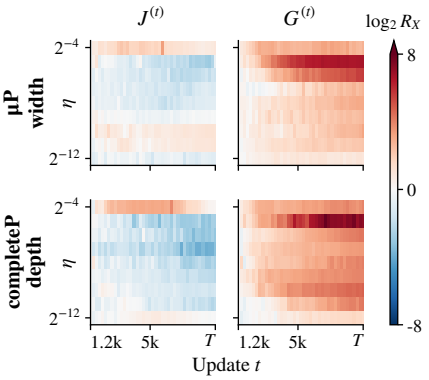


Figure 3: **Discrepancy growth** $\log_2 R_X$ for $X \in \{J, G\}$ relative to $t = 1200$, w/ $b = 1000$.

For $X \in \{J, G\}$, we measure agreement across sizes through the RMS discrepancy

$$D_X^{(t)} := \left[\text{mean}_{n,r}\; n\big(X_{n,r}^{(t)} - X_{2n,r}^{(t)}\big)^2\right]^{1/2}, \quad (7)$$

where the mean is taken over adjacent size pairs $(n, 2n)$ and matched runs r, with $n \in \{N, L\}$. While $O(N^{-1/2})$ fluctuations are expected under μP (Bordelon & Pehlevan, 2023), we use $L^{-1/2}$ only as a heuristic for depth (weakly correlated contributions of scale $1/L$ each would give $O(L^{-1/2})$ fluctuations). We then track $R_X^{(t)} := D_X^{(t)}/D_X^{(b+200)}$ across time t (we track D_X every 200 steps, so $b + 200$ is the first discrepancy after branching). Hence, $R_X > 1$ indicates accumulation of finite-size effects over time, while $R_X < 1$ indicates contraction. From fig. 3, we see that at $b = 1000$, R_G exhibits much stronger growth than R_J across most LRs for both width and depth, while discrepancies in J often shrink. In other words, G is more prone to losing scale consistency during training. Addressing **Q3**, we find:

Finding ③a: *Quadratic approximation works better for small η and late b, but scale consistency of C, J, G often still holds when quadratic approximation fails.*

The final interpolation curves in fig. 4 show that the observed transfer extends along the entire chord. Also, the consistency persists after a quadratic approximation loses quantitative accuracy: in fig. 5, we compare measured C with its midpoint Hessian approximation $-\boldsymbol{\Delta}^\top \nabla_{\boldsymbol{\theta}}^2 \mathcal{L}\, \boldsymbol{\Delta}/8$ from (6). While the approximation quality is high shortly after branching, it deteriorates over time and accumulates significantly more finite-size effects than the true chord deviation C. In § F.1.2, we further show that the quality of

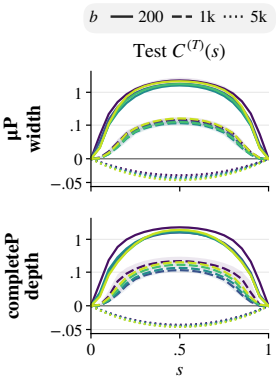


Figure 4: **Full chord deviations** at $T = 9.1$k, $\eta = 2^{-8}$. Mean ± std.; colors indicate width/depth as in fig. 2.

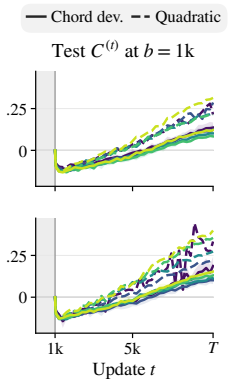


Figure 5: **Quadratic prediction** of $C^{(t)}$ for $\eta = 2^{-8}$, $b = 1$k. Colors as in fig. 2.

the quadratic approximation increases for smaller LRs and later branching times. We also note that the positive interpolation curves in fig. 4 show a clear *non-quadratic* form—thus, scale consistency of chord deviations is likely to be not merely a quadratic phenomenon.

Finding ③b: *The dependence of G on b is not only explained by parameter distance: GN directional curvature is less sensitive to b than remainder curvature.*

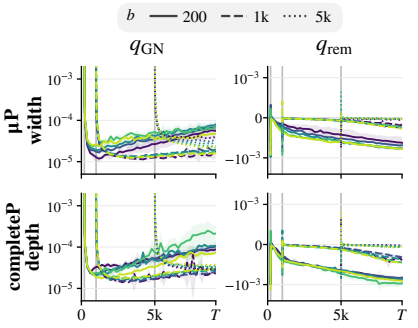


Figure 6: **Directional curvatures** of GN and remainder part in the interpolation direction for $\eta = 2^{-8}$. Mean ± std., colors as in fig. 2.

To separate whether the evolution of the quadratic interpolation terms is driven mainly by the distance between $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$ or by changes in curvature along $\boldsymbol{\Delta} := \boldsymbol{\theta}_B - \boldsymbol{\theta}_A$, we measure parameter separation in the preconditioned norm $\|\boldsymbol{\Delta}\|^2_{\boldsymbol{S}^{-1}} := \boldsymbol{\Delta}^\top \boldsymbol{S}^{-1} \boldsymbol{\Delta}$, where $\boldsymbol{S}$ is a fixed diagonal matrix (from the optimizer preconditioner or scaling rules, see § E.1), and define the directional curvature $q_X := \frac{1}{8} \boldsymbol{\Delta}^\top \boldsymbol{H}_X \boldsymbol{\Delta} / \|\boldsymbol{\Delta}\|^2_{\boldsymbol{S}^{-1}}$ for $X \in \{\mathrm{GN}, \mathrm{rem}\}$, so that the quadratic approximations (6) to J and G are $\|\boldsymbol{\Delta}\|^2_{\boldsymbol{S}^{-1}} q_{\mathrm{GN}}$ and $-\|\boldsymbol{\Delta}\|^2_{\boldsymbol{S}^{-1}} q_{\mathrm{rem}}$, respectively. In fig. 6, we see that q_{GN} is qualitatively more similar across branching times compared to q_{rem}, which becomes significantly weaker for later b. This complements Noci et al. (2024), who find super-consistent sharpness dynamics and that sharpness becomes increasingly GN-dominated over training—however, along LMC directions, non-GN curvature can remain substantial over long horizons (e.g., for $b = 200$) and shrink only for later b.

5.2 Scale Consistency of Variance Collapse

At first glance, the depth transfer from fig. 2 might appear surprising, since prior work found that linear interpolation can become increasingly difficult with depth, and traced much of that effect to *variance collapse* (Jordan et al., 2023): if the two models being interpolated produce hidden activations of comparable variance but correlation $\rho < 1$, their midpoint only has $(1 + \rho)/2 < 1$ of the endpoint variance, and this effect can compound over layers. REPAIR (Jordan et al., 2023) counteracts this by rescaling and shifting midpoint activations to match the endpoint statistics (which is not inherently tied to permutation alignment (Jolicoeur-Martineau et al., 2024)).

Why, then, do the barriers not systematically deteriorate as depth L grows? Under completeP, each block only contributes $O(L^{-1})$ to the residual stream. Heuristically, if interpolation induces an $O(\varepsilon)$ variance distortion per block, its effect on the residual stream is only $O(\varepsilon/L)$, and thus, the retention over L blocks approaches $(1 - \varepsilon/L)^L \to e^{-\varepsilon}$ (vs. $(1 - \varepsilon)^L \to 0$ without rescaling of blocks). Indeed, in fig. 7 we see nontrivial variance collapse that depends strongly on branching time (*left*), and applying REPAIR can reduce the barrier for most LRs. However, the degree of variance collapse and benefit of its correction are quite consistent. In other words, scale consistency of LMC does *not* imply ideal signal propagation, but prevents its exponential accumulation over depth. See § F.2 for further results on ResNets with batch normalization, where proper scaling can even outperform post-hoc application of REPAIR (Jordan et al., 2023).

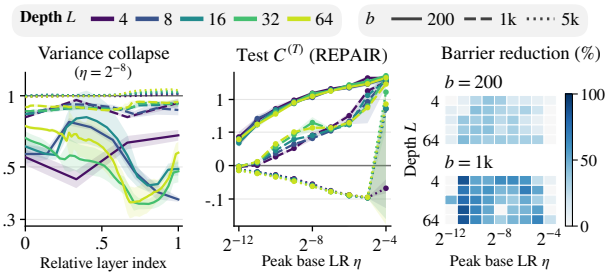


Figure 7: **Variance collapse** in the setup of § 5.1 (completeP). *Left:* Variance collapse by b. *Center:* Difference midpt. vs. endpt. losses w/ REPAIR. *Right:* Barrier reduction vs. w/o REPAIR (fig. 2).

6 Hyperparameter Transfer in Model Merging

We further ask whether the observed scale consistency of loss barriers extends to a transfer-learning setup and can predict a practical weight-space operation. We specifically focus on model merging, i.e., combining task-specific parameter updates from a pretrained model into one, and test whether fine-tuning and merging HPs transfer under scaling limits.

We use the same family of GPT-2-style transformers under completeP as for the LMC experiments in § 5, but now pretrain for 3B OpenWebText tokens at LR $\eta = 2^{-9}$. For fine-tuning, we use the 8 tasks from Zhou et al. (2023a), which cover, e.g., question answering and paraphrase identification, and were used in prior model merging work (Tam et al., 2024; Li et al., 2025). From each pretrained checkpoint, we fine-tune one expert per task for ≈3k steps with AdamW (no warmup or LR schedule) and sweep the constant base LR $\eta \in \{2^{-20}, \dots, 2^{-8}\}$ under the same completeP scaling as during pretraining. Write $\boldsymbol{\theta}_{\text{base}}$ for the pretrained model, and $\boldsymbol{\tau}_i := \boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{base}}$ for the update of expert i. Each merging method returns $\boldsymbol{\theta}_0 + \alpha\boldsymbol{\tau}_{\text{merged}}$, where α is the strength of the update in the merging direction $\boldsymbol{\tau}_{\text{merged}}$. Task arithmetic (TA) (Ilharco et al., 2023) simply uses $\boldsymbol{\tau}_{\text{merged}} := \frac{1}{8}\sum_{i=1}^{8} \boldsymbol{\tau}_i$. TIES (Yadav et al., 2023) retains each expert's largest-magnitude fraction ρ of updates, elects coordinatewise signs, and averages the updates agreeing with these signs. We also evaluate Squisher (Li et al., 2025) as a cheaper proxy for Fisher merging (Matena & Raffel, 2022), which estimates the Fisher diagonal from the Adam optimizer state. See § E.3 for further details.

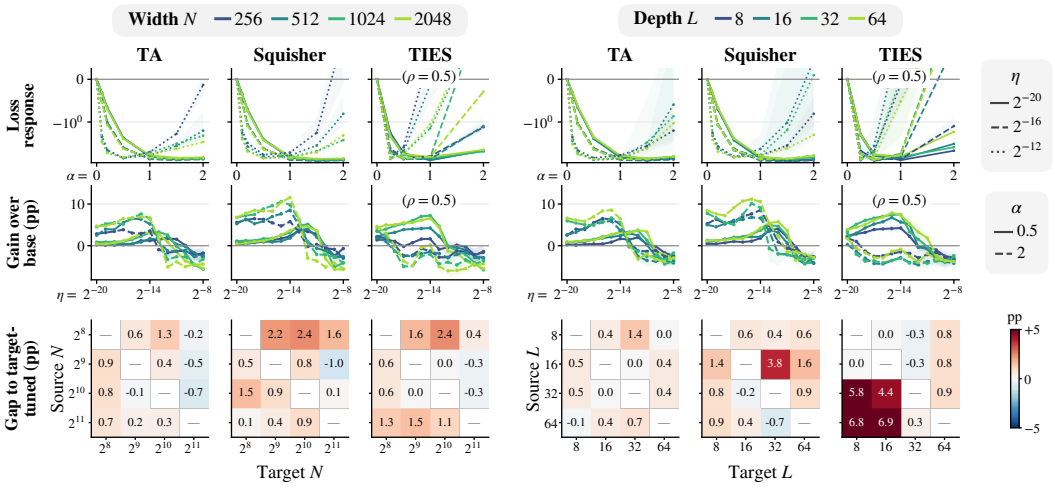


Figure 8: **Model merging HP transfer.** *Top:* Loss change vs. base models across merging strength α. *Middle:* Accuracy gain over base models (in percentage points) across LRs. *Bottom:* Accuracy gap to target-tuned HPs when transferring from source N or L to target (TIES: $\rho \in \{.1, .2, .5, 1\}$).

Fig. 8 shows that the loss landscape responds similarly to the same merging operations across width and depth. In particular, the test loss as a function of merging strength α and the gain over $\boldsymbol{\theta}_{\text{base}}$ follow similar trends across scales, though the former more reliable for TA and Squisher than TIES, and better for smaller LRs and merging strengths. In the bottom row, we transfer HPs selected on a *source* scale to each *target* scale and reports the gap to tuning directly on the target scale. We find that this works most reliably for TA, reasonably for Squisher, but TIES shows some failures when settings selected on the deepest models are transferred back to shallower ones. Overall, we conclude that, to a substantial degree, the scale-consistent geometry we observed for LMC extends to merging directions, so one can use smaller models to predict useful model merging HPs at larger scales.

7 Conclusion

In this work, we studied how loss landscape geometry scales with model size through the lens of linear mode connectivity, and showed that under feature-learning limits, the presence or absence of loss barriers can already be qualitatively consistent at practically relevant widths and depths. We further showed that this transfer extends to model merging and established a finite-horizon width limit for interpolation curves. Interesting theoretical extensions of our analysis could be to characterize leading finite-width fluctuations, depth limits, and the long-time behavior ($t \to \infty$), which could help further explain the phenomena we observe empirically. Another promising future direction is to study alignment under parameter symmetries between independently trained models across scaling limits, which could ultimately inform principled methods for size-heterogenous model merging.

AI USAGE STATEMENT

In this work, we used generative AI tools to assist with writing code and presenting proofs, for local edits to improve presentation, and to identify related work. The research direction, core ideas, theoretical results, and experimental design are independent contributions of the authors. We have reviewed all AI-assisted work and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide formal proofs of all our statements in § C and § D, and state the setup for all experiments in § E. We will release code to reproduce all of our experiments upon acceptance.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

Aditya Kumar Akash, Sixu Li, and Nicolás García Trillos. Wasserstein barycenter-based model fusion and linear mode connectivity of neural networks. *arXiv preprint arXiv:2210.06671*, 2022.

Gül Sena Altıntaş, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. Disentangling linear mode connectivity. In *NeurIPS Workshop on Unifying Representations in Neural Models*, 2023.

Gül Sena Altıntaş, Devin Kwok, Colin Raffel, and David Rolnick. The butterfly effect: Neural network training trajectories are highly sensitive to initial conditions. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 1314–1342, 2025.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 93–104. Association for Computational Linguistics, 2022.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Blake Bordelon, Hamza Tahir Chaudhry, and Cengiz Pehlevan. Infinite limits of multi-head transformer dynamics. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *The Twelfth International Conference on Learning Representations*, 2024b.

Vincent Bürgin, Daniel Herbst, Ya-Wei Eileen Lin, and Stefanie Jegelka. Beyond structural symmetries: Linear mode connectivity via neuron identifiability. In *International Conference on Machine Learning*, 2026.

Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across LoRA ranks and transfer to full finetuning. In *International Conference on Machine Learning*, 2026.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Kenneth R. Davidson and Stanislaw J. Szarek. Local operator theory, random matrices and Banach spaces. In *Handbook of the Geometry of Banach Spaces*, volume 1, pp. 317–366. Elsevier, 2001.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Nolan Dey, Bin Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: CompleteP enables compute-efficient deep transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Luca Di Carlo, Chase Goddard, and David J. Schwab. Entropic confinement and mode connectivity in overparameterized neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1309–1318, 2018.

Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, fifth edition, 2019.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *The Tenth International Conference on Learning Representations*, 2022.

Katie E. Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12666–12700, 2024.

Damien Ferbach, Baptiste Goujaud, Gauthier Gidel, and Aymeric Dieuleveut. Proving linear mode connectivity of neural networks via optimal transport. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3853–3861, 2024.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the Neural Tangent Kernel. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269, 2020.

C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *The Fifth International Conference on Learning Representations*, 2017.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Aaron Gokaslan and Vanya Cohen. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Moritz Haas, Jin Xu, Volkan Cevher, and Leena Chennuru Vankadara. μP^2: Effective sharpness aware minimization requires layerwise perturbation scaling. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Moritz Haas, Sebastian Bordt, Ulrike von Luxburg, and Leena Chennuru Vankadara. On the surprising effectiveness of large learning rates under standard width scaling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Soufiane Hayou and Greg Yang. Width and depth limits commute in residual networks. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12700–12723, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of EMNLP-IJCNLP*, pp. 2391–2401, 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, 2015.

Akira Ito, Masanori Yamada, Daiki Chijiwa, and Atsutoshi Kumagai. Do we really need permutations? Impact of model width on linear mode connectivity. In *The Fourteenth International Conference on Learning Representations*, 2026.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Alexia Jolicoeur-Martineau, Emy Gervais, Kilian Fatras, Yan Zhang, and Simon Lacoste-Julien. PopulAtion parameter averaging (PAPA). *Transactions on Machine Learning Research*, 2024.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: REnormalizing permuted activations for interpolation repair. In *The Eleventh International Conference on Learning Representations*, 2023.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. QASC: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Sanjeev Arora, and Rong Ge. Explaining landscape connectivity of low-cost solutions for multilayer nets. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Clarissa Lauditi, Cengiz Pehlevan, and Blake Bordelon. Spectral dynamics in deep networks: Feature learning, outlier escape, and learning rate transfer. *arXiv preprint arXiv:2605.07870*, 2026.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Tianyi Li and Zhiqiang Shen. Scaling linear mode connectivity and merging to billion parameter pretrained transformers. *arXiv preprint arXiv:2606.23607*, 2026.

Yu Xin Li, Felix Dangel, Derek Tam, and Colin Raffel. Fishers for free? Approximating the Fisher information matrix by recycling the squared gradient accumulator. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34252–34270, 2025.

Derek Lim, Theo (Moe) Putterman, Robin Walters, Haggai Maron, and Stefanie Jegelka. The empirical impact of neural parameter symmetries, or lack thereof. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 58–62, 2019.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: When and why the tangent kernel is constant. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Kangqiao Liu, Liu Ziyin, and Masahito Ueda. Noise and fluctuation of finite learning rate stochastic gradient descent. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7045–7056, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ekdeep Singh Lubana, Eric J. Bigelow, Robert P. Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22965–23004, 2023.

Yuxin Ma, Nan Chen, Mateo Díaz, Soufiane Hayou, Dmitriy Kunisky, and Soledad Villar. μpscaling small models: Principled warm starts and hyperparameter transfer. In *International Conference on Machine Learning*, 2026.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Michael Matena and Colin Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: Dimension-free bounds and kernel limit. In *Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2388–2464, 2019.

Alexandru Meterez, Pranav Ajit Nair, Depen Morwani, Cengiz Pehlevan, Sham Kakade, and Alex Damian. A defense of the quadratic model. *arXiv preprint arXiv:2607.21716*, 2026.

Quynh Nguyen. On connected sublevel sets in deep learning. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4790–4799, 2019.

Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto. Super consistency of neural network landscapes and learning rate transfer. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQA: Commonsense reasoning about social interactions. In *Proceedings of EMNLP-IJCNLP*, pp. 4463–4473, 2019.

Alexander Shevchenko and Marco Mondelli. Landscape connectivity and dropout stability of SGD solutions for over-parameterized neural networks. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8773–8784, 2020.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9722–9732, 2021.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Sidak Pal Singh, Linara Adilova, Michael Kamp, Asja Fischer, Bernhard Schölkopf, and Thomas Hofmann. Landscaping linear mode connectivity. In *ICML Workshop on High-Dimensional Learning Dynamics*, 2024.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of EMNLP-IJCNLP*, pp. 5941–5946, 2019.

Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*, 2024.

N. Joseph Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Alexander Theus, Alessandro Cabodi, Sotiris Anagnostidis, Antonio Orvieto, Sidak Pal Singh, and Valentina Boeva. Generalized linear mode connectivity for transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yuanyi Wang, Yanggan Gu, Yiming Zhang, Qi Zhou, Zhaoyi Yan, Congkai Xie, Xinyao Wang, Jianbo Yuan, and Hongxia Yang. Model merging scaling laws in large language models. In *International Conference on Machine Learning*, 2026.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Greg Yang. Tensor Programs II: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Greg Yang and Edward J. Hu. Tensor Programs IV: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11727–11737, 2021.

Greg Yang and Etai Littwin. Tensor Programs IIb: Architectural universality of neural tangent kernel training dynamics. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11762–11772, 2021.

Greg Yang and Etai Littwin. Tensor Programs IVb: Adaptive optimization in the infinite-width limit. *arXiv preprint arxiv:2308.01814*, 2023.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor Programs V: Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor Programs VI: Feature learning in infinite depth neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, pp. 2013–2018, 2015.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775, 2024.

Keyao Zhan, Puheng Li, and Lei Wu. Analyzing the role of permutation invariance in linear mode connectivity. In *International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2557–2565, 2025.

Chenxiang Zhang, Alexander Theus, Damien Teney, Antonio Orvieto, Jun Pang, and Sjouke Mauw. How does the optimizer implicitly bias the model merging loss landscape? In *The Fourteenth International Conference on Learning Representations*, 2026a.

Fangzhao Zhang, Sungyoon Kim, Erica Zhang, Yiqi Jiang, and Mert Pilanci. Optimizer-induced mode connectivity: From AdamW to Muon. *arXiv preprint arXiv:2605.09991*, 2026b.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of NAACL-HLT*, pp. 1298–1308, 2019.

Jing Zhou, Zongyu Lin, Yanan Zheng, Jian Li, and Zhilin Yang. Not all tasks are born equal: Understanding zero-shot generalization. In *The Eleventh International Conference on Learning Representations*, 2023a.

Zhanpeng Zhou, Yongyi Yang, Xiaojiang Yang, Junchi Yan, and Wei Hu. Going beyond linear mode connectivity: The layerwise linear feature connectivity. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

Liu Ziyin, Kangqiao Liu, Takashi Mori, and Masahito Ueda. Strength of minibatch noise in SGD. In *The Tenth International Conference on Learning Representations*, 2022.

APPENDICES

A Extended Related Work **18**

B Background on Scaling Limits **20**

B.1 *abc* Parameterizations 20

B.2 Lazy and Feature-Learning Limits 21

B.3 Loss-Dependent Stability 24

C Quadratic Approximation of Chord Deviation **25**

D Scaling Limits for Loss Barriers **27**

D.1 Model and Branching Procedure 27

D.2 μP Width Limit on a Finite Grid 28

D.3 Uniform Convergence of μP Barrier Curves 31

D.4 NTP Kernel Limit 32

D.5 Finite-Width Agreement 35

E Experimental Setup **36**

E.1 Linear Mode Connectivity 36

E.2 Variance Collapse 38

E.3 Model Merging 38

F Additional Experimental Results **40**

F.1 Linear Mode Connectivity 40

F.2 Variance Collapse 47

F.3 Model Merging 48

A Extended Related Work

Mode connectivity and loss landscapes. The term *mode connectivity* was coined by early works (Draxler et al., 2018; Garipov et al., 2018) that observed the existence of low-loss curves between trained neural network solutions. Also, the loss landscape has been analyzed through topology or neuron removal (Freeman & Bruna, 2017; Nguyen, 2019; Kuditipudi et al., 2019; Shevchenko & Mondelli, 2020). A special case is *linear* mode connectivity (LMC) (Frankle et al., 2020), which asks if the straight path between two solutions is of low loss. We follow the standard setup of Frankle et al. (2020), who analyze LMC on models trained with different optimization randomness from the same partially trained checkpoint. It has been observed that kernel evolution, the training setup, and sensitivity to initial conditions affect the emergence of LMC (Fort et al., 2020; Altıntaş et al., 2023; Singh et al., 2024; Altıntaş et al., 2025). Other works observed co-occurence of LMC with a stronger notion of feature map connectivity in layers of a neural network (Zhou et al., 2023b), as well as generalization strategies or learned mechanisms (Juneja et al., 2023; Lubana et al., 2023). A non-monotonic dependence on effective optimizer noise (Zhang et al., 2026a), entropic confinement (Di Carlo et al., 2026), and comparisons between different optimizers (Zhang et al., 2026b) further show dependence of LMC on various aspects of the training procedure. Our work addresses a complementary question, namely, instead of asking which training conditions give rise to LMC, we study how the interpolation geometry behaves as width and depth are scaled.

A related line of work observed that alignment under *parameter symmetries*, most importantly hidden-neuron permutations, can significantly lower loss barriers between models trained from independent initializations (Entezari et al., 2022; Singh & Jaggi, 2020; Tatro et al., 2020; Akash et al., 2022; Ainsworth et al., 2023). Jordan et al. (2023) identify variance collapse of interpolated weights as a cause for failure of LMC under permutation alignment in deep models and show that this can be mitigated by renormalizing activations. Ferbach et al. (2024) prove LMC under permutation alignment in the mean-field limit by convergence of neuron distributions in Wasserstein distance. Further theoretical results focus on ReLU teacher-student models (Zhan et al., 2025). Accounting for richer parameter symmetries extends alignment to transformers (Theus et al., 2025; Li & Shen, 2026). Another line of work finds unaligned LMC from independent initializations by removing parameter symmetries (Lim et al., 2024; Bürgin et al., 2026). Further, unaligned LMC from independent initializations can also emerge under growing width (Ito et al., 2026) and softmax temperature calibration, as the interpolated model increasingly behaves like an (implicit) ensemble.

Model merging. Model merging combines trained networks, e.g., fine-tuned from a shared pretrained checkpoint, with the goal to retain complementary capabilities in one model without the inference costs of ensembling. Model soups (Wortsman et al., 2022) average weights of fine-tuned models, while task arithmetic (Ilharco et al., 2023) combines scaled task vectors, i.e. parameter changes from the base model. TIES-Merging (Yadav et al., 2023) reduces interference by pruning small updates and resolving sign conflicts. DARE (Yu et al., 2024) randomly drops task-vector entries and rescales the remainder before merging. Fisher-weighted averaging (Matena & Raffel, 2022) weights parameters by their estimated importance using the diagonal Fisher information. Squisher (Li et al., 2025) estimates the diagonal Fisher information from the Adam optimizer state. RegMean (Jin et al., 2023) minimizes layer-output discrepancies using activation statistics; however, we omit it in § 6 as we focus on operations solely in weight space.

Zhang et al. (2026a) analyze the effect of optimization on model merging and find that merging success varies non-monotonically with the effective optimizer noise scale. This is consistent with our findings, as we keep batch size fixed and vary the learning rate, which is proportional to the effective optimizer noise. We hypothesize that this observed non-monotonic dependence may arise from the balance between the ensembling gain J and nonlinear interpolation gap G (see § 4.1): although both increase with noise, their relative growth rates can change, which would give rise to a local minimum in $C = G - J$. Recently, Wang et al. (2026) study scaling laws for model merging, namely, how the *absolute* loss of a merged model changes as a function of model size and the number of merged experts. They observe that larger base models have a lower loss floor and a smaller *merging tail*, and the merging tail decays as $O(k^{-1})$ in the number of experts k. § 6 of our work addresses a complementary scaling question: whether the weight-space geometry underlying such merging operations itself becomes scale-consistent under scaling limits. At first glance, one might think the model size dependence of the merging tail in Wang et al. (2026) is in mild tension

with scale-consistent merging geometry, as this suggests that the cost associated with combining heterogeneous task updates can decrease with model capacity. However, this does not contradict our results in § 6, as we do not study scaling of expected merge performance, but whether the *loss response* to different merging hyperparameters is preserved across scales, and absolute merging costs can shrink with scale while useful hyperparameter regions still transfer.

Scaling limits. Infinite-width limits are a central tool to provide tractable descriptions of neural-network training. However, the resulting dynamics depend crucially on how models are parameterized. Early work analyzed scaling to infinite width under the *neural tangent kernel* parameterization (Jacot et al., 2018; Lee et al., 2019; Chizat et al., 2019), in which features remain asymptotically fixed and training approaches fixed kernel dynamics. In contrast, *mean-field limits*, which were initially analyzed in two-layer networks (Mei et al., 2018; Chizat & Bach, 2018; Mei et al., 2019), retain *feature learning*. Tensor Programs extend feature-learning limits to deep architectures and identify µP as a parameterization that retains nontrivial feature learning at infinite width (Yang & Hu, 2021). This scaling also enables *µTransfer*, i.e., that hyperparameters tuned on small proxy models can remain close to optimal as width grows (Yang et al., 2021). Subsequent work has characterized and extended these limits from complementary perspectives, such as spectral conditions (Yang et al., 2023), dynamical descriptions of kernel evolution (Bordelon & Pehlevan, 2022), and extensions to adaptive optimizers (Yang & Littwin, 2023). Analogous questions arise when scaling depth. Depth-µP and related dynamical analyses derive residual and update scalings that admit feature-learning infinite-depth limits (Yang et al., 2024; Bordelon et al., 2024b), and Hayou & Yang (2023) establish commuting width and depth limits. For transformers, the limiting dynamics additionally depend on how attention dimensions, width, and depth are scaled (Bordelon et al., 2024a). *CompleteP* (Dey et al., 2025) scales the residual contributions as L^{-1} (vs. $L^{-1/2}$ for depth-µP) motivated by retaining the nonlinear dependence of the learned representations on parameter updates. More broadly, scaling limits have been extended beyond the usual training hyperparameters. For example, Haas et al. (2024) derive perturbation scales for sharpness-aware optimization, Ma et al. (2026) develop a method for principled model upscaling, and Chen et al. (2026) develop a parameterization for LR transfer across LoRA ranks and from LoRA to full fine-tuning.

There is also evidence that scale consistency of certain aspects of training can already emerge at realistic finite model sizes. Vyas et al. (2023) find consistent predictions, representations, and training phenomena across width in feature-learning networks. Noci et al. (2024) show that leading Hessian eigenvalues and directional curvature in the direction of the gradient can exhibit *super consistency* across width and depth. In contrast, we study scale consistency along *paths in weight space* between separately trained models, which extends the existing analysis of local curvature and, as we show, can persist even when a quadratic approximation of the landscape becomes inaccurate.

Loss curvature and stochastic fluctuations. The generalized Gauss-Newton (GN) matrix (Martens, 2020) we use in § 4.1 is a standard PSD approximation to loss curvature. Curvature has also been used to approximate LMC barriers (Singh et al., 2024), and curvature-aware model-merging methods use Fisher or Hessian approximations to combine task-specific solutions (Matena & Raffel, 2022). Recently, Meterez et al. (2026) show that local quadratic models can predict language-model optimization over substantial training windows. The stronger scale consistency of J we observe in § 5.1 may also connect to fluctuations of stochastic gradients. Indeed, analyses of constant-step SGD near stable minima show that its fluctuations can approach a stationary covariance determined by local curvature and gradient-noise covariance (Liu et al., 2021), and Ziyin et al. (2022) characterize the minibatch noise driving these fluctuations. Since J depends only on disagreement between the endpoint predictions, a possible interpretation of our observation is that finite-size discrepancies of J often contract, i.e., that independent post-branch minibatch noise may drive prediction disagreement toward a stable fluctuation scale, while G can continue accumulating finite-size effects.

B Background on Scaling Limits

B.1 *abc* Parameterizations

In this section, we review scaling limits for width through the example of a bias-free MLP in *abc parameterization*, following Yang & Hu (2021). Its layerwise architecture, initialization, and LR scaling exponents are $a_\ell, b_\ell, c_\ell \in \mathbb{R}$, respectively, for $0 \le \ell \le L+1$. Fix input and output dimensions d_{in} and d_{out}, and let $L+1$ denote the number of hidden layers, each of width N, so that L matches the depth in (1). Let $\varphi : \mathbb{R} \to \mathbb{R}$ be a nonlinearity. For an input $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$, the architecture is

$$\boldsymbol{h}_0(\boldsymbol{x}) \;:=\; N^{-a_0}\boldsymbol{W}_0\boldsymbol{x}, \tag{8}$$

$$\boldsymbol{h}_\ell(\boldsymbol{x}) \;:=\; N^{-a_\ell}\boldsymbol{W}_\ell\varphi\big(\boldsymbol{h}_{\ell-1}(\boldsymbol{x})\big), \qquad 1 \le \ell \le L, \tag{9}$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;:=\; N^{-a_{L+1}}\boldsymbol{W}_{L+1}\varphi\big(\boldsymbol{h}_L(\boldsymbol{x})\big) \in \mathbb{R}^{d_{\text{out}}}, \tag{10}$$

where $\boldsymbol{W}_0 \in \mathbb{R}^{N\times d_{\text{in}}}$, $\boldsymbol{W}_\ell \in \mathbb{R}^{N\times N}$ for $1 \le \ell \le L$, and $\boldsymbol{W}_{L+1} \in \mathbb{R}^{d_{\text{out}}\times N}$. For every $0 \le \ell \le L$, we call $\boldsymbol{h}_\ell(\boldsymbol{x}) \in \mathbb{R}^N$ the hidden representations / preactivations, and $\varphi(\boldsymbol{h}_\ell(\boldsymbol{x}))$ the activations. The weights are initialized according to

$$\boldsymbol{W}_\ell^{(0)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(0, N^{-2b_\ell}\big), \qquad 0 \le \ell \le L+1. \tag{11}$$

Given a label space $\mathcal{Y}$, a finite dataset of size M, $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M \subset \mathbb{R}^{d_{\text{in}}} \times \mathcal{Y}$ and a per-example loss $\ell : \mathbb{R}^{d_{\text{out}}} \times \mathcal{Y} \to \mathbb{R}$, empirical risk minimization minimizes $\mathcal{L}(\boldsymbol{\theta}) := M^{-1}\sum_{i=1}^M \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$, where $\boldsymbol{\theta} := (\boldsymbol{W}_0, \dots, \boldsymbol{W}_{L+1})$. SGD with base LR η updates each layer by

$$\boldsymbol{W}_\ell^{(t+1)} \;:=\; \boldsymbol{W}_\ell^{(t)} - \eta N^{-c_\ell}\widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(t)}), \qquad 0 \le \ell \le L+1, \tag{12}$$

where $\widehat{\nabla}$ denotes a minibatch gradient.

Lemma B.1 (Gauge invariance of *abc* parameterizations, Yang & Hu (2021), § 3.2)**.** *Fix $g_0, \dots, g_{L+1} \in \mathbb{R}$, and define a second abc parameterization by*

$$a'_\ell \;:=\; a_\ell + g_\ell, \qquad b'_\ell \;:=\; b_\ell - g_\ell, \qquad c'_\ell \;:=\; c_\ell - 2g_\ell. \tag{13}$$

Couple the initializations by $\boldsymbol{W}_\ell'^{(0)} := N^{g_\ell}\boldsymbol{W}_\ell^{(0)}$, and train both parameterizations with the same base learning rate and minibatch sequence. Then, for every $t \ge 0$ and $0 \le \ell \le L+1$,

$$\boldsymbol{W}_\ell'^{(t)} \;=\; N^{g_\ell}\boldsymbol{W}_\ell^{(t)}, \qquad N^{-a'_\ell}\boldsymbol{W}_\ell'^{(t)} \;=\; N^{-a_\ell}\boldsymbol{W}_\ell^{(t)}. \tag{14}$$

In particular, the two parameterizations have identical functions and training dynamics in effective-weight coordinates.

Proof. The coupled initialization has entries distributed as

$$\boldsymbol{W}_\ell'^{(0)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(0, N^{2g_\ell}N^{-2b_\ell}\big) \;=\; \mathcal{N}\big(0, N^{-2b'_\ell}\big), \tag{15}$$

and $N^{-a'_\ell}\boldsymbol{W}_\ell'^{(0)} = N^{-a_\ell}\boldsymbol{W}_\ell^{(0)}$. Suppose (14) holds at step t. The two networks then have the same effective weights and hence the same minibatch loss. The chain rule gives

$$\widehat{\nabla}_{\boldsymbol{W}'_\ell}\mathcal{L}(\boldsymbol{\theta}'^{(t)}) \;=\; N^{-g_\ell}\widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(t)}). \tag{16}$$

Using $c'_\ell = c_\ell - 2g_\ell$, the update therefore satisfies

$$\begin{aligned}\boldsymbol{W}_\ell'^{(t+1)} \;&=\; N^{g_\ell}\boldsymbol{W}_\ell^{(t)} - \eta N^{-(c_\ell-2g_\ell)}N^{-g_\ell}\widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(t)}) \\ &=\; N^{g_\ell}\Big(\boldsymbol{W}_\ell^{(t)} - \eta N^{-c_\ell}\widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(t)})\Big) \;=\; N^{g_\ell}\boldsymbol{W}_\ell^{(t+1)}.\end{aligned} \tag{17}$$

The result follows by induction. □

By lem. B.1, an *abc* parameterization has multiple equivalent representatives. Table 1 gives the standard representatives of the standard parameterization (SP), neural tangent parameterization (NTP), and maximal update parameterization (μP), followed by equivalent representatives with no forward scaling. The columns I, H, and O denote the input layer $\ell = 0$, hidden layers $1 \le \ell \le L$, and output layer $\ell = L+1$, respectively.

Table 1: Two equivalent abc representations of SP, NTP, and μP.

		SP			NTP			μP		
Gauge	Exponent	I	H	O	I	H	O	I	H	O
Global LR	a_ℓ	0	0	0	0	$1/2$	$1/2$	$-1/2$	0	$1/2$
	b_ℓ	0	$1/2$	$1/2$	0	0	0	$1/2$	$1/2$	$1/2$
	c_ℓ $(=0)$	0	0	0	0	0	0	0	0	0
No arch. scaling	a_ℓ $(=0)$	0	0	0	0	0	0	0	0	0
	b_ℓ	0	$1/2$	$1/2$	0	$1/2$	$1/2$	0	$1/2$	1
	c_ℓ	0	0	0	0	1	1	-1	0	1

B.2 Lazy and Feature-Learning Limits

In this subsection, we review some of the intuitions of Yang et al. (2023). We let $N \to \infty$ while keeping the depth, input and output dimensions, dataset, minibatch size, and number of SGD steps fixed. From now on, we use the representatives in table 1 with $a_\ell = 0$ for every layer. The matrices $\boldsymbol{W}_\ell$ therefore coincide with the effective weights $N^{-a_\ell}\boldsymbol{W}_\ell$. We allow the base LR in (12) to scale as $\eta_N := \eta N^r$. In this gauge, the layerwise update is

$$\boldsymbol{W}_\ell^{(t+1)} \;=\; \boldsymbol{W}_\ell^{(t)} - \eta_{\ell,N}\widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(t)}), \qquad \eta_{\ell,N} \;:=\; \eta N^{r-c_\ell}. \tag{18}$$

Throughout this section, let $\|\boldsymbol{v}\|$ denote the RMS norm of a vector $\boldsymbol{v} \in \mathbb{R}^m$, while $\|\boldsymbol{M}\|$ denotes the induced RMS operator norm of a matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$. Thus,

$$\|\boldsymbol{v}\| \;:=\; m^{-1/2}\|\boldsymbol{v}\|_2, \tag{19}$$

$$\|\boldsymbol{M}\| \;:=\; \sup_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\|\boldsymbol{M}\boldsymbol{v}\|}{\|\boldsymbol{v}\|} \;=\; \left(\frac{n}{m}\right)^{1/2}\|\boldsymbol{M}\|_{2\to 2}. \tag{20}$$

For a fixed input $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$ and time t, write the changes from initialization as

$$\Delta\boldsymbol{h}_\ell^{(t)}(\boldsymbol{x}) \;:=\; \boldsymbol{h}_\ell^{(t)}(\boldsymbol{x}) - \boldsymbol{h}_\ell^{(0)}(\boldsymbol{x}), \qquad 0 \le \ell \le L, \tag{21}$$

$$\Delta\boldsymbol{W}_\ell^{(t)} \;:=\; \boldsymbol{W}_\ell^{(t)} - \boldsymbol{W}_\ell^{(0)}, \qquad 0 \le \ell \le L+1. \tag{22}$$

Definition B.2 (Lazy and feature-learning limits (Yang et al., 2023))**.** Fix $t \ge 1$, $0 \neq \boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$, and $0 \le \ell \le L$. We distinguish:

- **Lazy limit.** The width limit is lazy at $(t, \boldsymbol{x}, \ell)$ if
$$\left\|\boldsymbol{h}_\ell^{(t)}(\boldsymbol{x})\right\| \;=\; \Theta_{\mathbb{P}}(1), \qquad \left\|\Delta\boldsymbol{h}_\ell^{(t)}(\boldsymbol{x})\right\| \;=\; o_{\mathbb{P}}(1). \tag{23}$$
- **Feature-learning limit.** The width limit exhibits feature learning at $(t, \boldsymbol{x}, \ell)$ if
$$\left\|\boldsymbol{h}_\ell^{(t)}(\boldsymbol{x})\right\| \;=\; \Theta_{\mathbb{P}}(1), \qquad \left\|\Delta\boldsymbol{h}_\ell^{(t)}(\boldsymbol{x})\right\| \;=\; \Theta_{\mathbb{P}}(1). \tag{24}$$

We apply either designation to all hidden layers if its corresponding condition holds for every $0 \le \ell \le L$ at the same fixed t and $\boldsymbol{x}$.

Much of the empirical success of deep networks is commonly attributed to feature learning, whereby hidden representations adapt to the task rather than remaining close to random features (Yang & Hu, 2021; Yang et al., 2023). In the lazy limit of def. B.2, the hidden features instead move by only $o_{\mathbb{P}}(1)$ in RMS norm. This motivates linearizing the network around its initialization (Chizat et al., 2019; Lee et al., 2019), in which case training is described by a kernel method governed by the initialization neural tangent kernel (NTK) (Jacot et al., 2018).

Yang et al. (2023) introduce a spectral condition for feature learning that specifies the width scaling of each layer's initial weight and displacement. In this notation, they require

$$\left\|\boldsymbol{W}_\ell^{(0)}\right\| \;=\; \Theta_{\mathbb{P}}(1), \qquad \left\|\Delta\boldsymbol{W}_\ell^{(t)}\right\| \;=\; \Theta_{\mathbb{P}}(1), \qquad 0 \le \ell \le L+1. \tag{25}$$

The spectral condition is motivated by the way changes propagate through the network. At the input layer, $\Delta \boldsymbol{h}_0^{(t)}(\boldsymbol{x}) = \Delta \boldsymbol{W}_0^{(t)} \boldsymbol{x}$, while for $1 \le \ell \le L$ and for the output,

$$\Delta \boldsymbol{h}_\ell^{(t)}(\boldsymbol{x}) \;=\; \Delta \boldsymbol{W}_\ell^{(t)} \varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x})\big) + \boldsymbol{W}_\ell^{(0)} \Big[\varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x})\big) - \varphi\big(\boldsymbol{h}_{\ell-1}^{(0)}(\boldsymbol{x})\big)\Big], \tag{26}$$

$$f_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}^{(0)}}(\boldsymbol{x}) \;=\; \Delta \boldsymbol{W}_{L+1}^{(t)} \varphi\big(\boldsymbol{h}_L^{(t)}(\boldsymbol{x})\big) + \boldsymbol{W}_{L+1}^{(0)} \Big[\varphi\big(\boldsymbol{h}_L^{(t)}(\boldsymbol{x})\big) - \varphi\big(\boldsymbol{h}_L^{(0)}(\boldsymbol{x})\big)\Big]. \tag{27}$$

By the triangle inequality and submultiplicativity of the RMS norm,

$$\big\|\Delta \boldsymbol{h}_\ell^{(t)}(\boldsymbol{x})\big\| \;=\; O_{\mathbb{P}}\Big(\big\|\Delta \boldsymbol{W}_\ell^{(t)}\big\|\big\|\varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x})\big)\big\| + \big\|\boldsymbol{W}_\ell^{(0)}\big\|\big\|\varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x})\big) - \varphi\big(\boldsymbol{h}_{\ell-1}^{(0)}(\boldsymbol{x})\big)\big\|\Big). \tag{28}$$

The same bound holds analogously for the input and output layers. Consequently, order-one RMS operator norms give the correct upper scale for feature changes. To state when the bound is tight, define the alignment of a nonzero matrix $\boldsymbol{M}$ and vector $\boldsymbol{v}$ by

$$\alpha(\boldsymbol{M}, \boldsymbol{v}) \;:=\; \frac{\|\boldsymbol{M}\boldsymbol{v}\|}{\|\boldsymbol{M}\|\|\boldsymbol{v}\|} \;\in\; [0, 1]. \tag{29}$$

The operator-norm bound for $\boldsymbol{M}\boldsymbol{v}$ is order-tight precisely when $\alpha(\boldsymbol{M}, \boldsymbol{v}) = \Theta_{\mathbb{P}}(1)$. To see why SGD can produce such alignment, write $\delta \boldsymbol{W}_\ell^{(t)} := \boldsymbol{W}_\ell^{(t+1)} - \boldsymbol{W}_\ell^{(t)}$. For one example $(\boldsymbol{x}, y)$, abbreviate $\boldsymbol{h}_{\ell-1}^{(t)} := \boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x})$ and let $\boldsymbol{g}_\ell^{(t)} := \nabla_{\boldsymbol{h}_\ell(\boldsymbol{x})} \ell(f_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}), y)$. Then, with $\boldsymbol{x}$ in place of $\varphi(\boldsymbol{h}_{\ell-1}^{(t)})$ for the input layer,

$$\delta \boldsymbol{W}_\ell^{(t)} \;=\; -\eta_{\ell,N} \boldsymbol{g}_\ell^{(t)} \varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}\big)^\top, \qquad \big\|\delta \boldsymbol{W}_\ell^{(t)} \varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}\big)\big\| \;=\; \big\|\delta \boldsymbol{W}_\ell^{(t)}\big\|\big\|\varphi\big(\boldsymbol{h}_{\ell-1}^{(t)}\big)\big\|. \tag{30}$$

Thus the incoming activation is a top right singular vector and $\alpha(\delta \boldsymbol{W}_\ell^{(t)}, \varphi(\boldsymbol{h}_{\ell-1}^{(t)})) = 1$. For a fixed-size minibatch, it is reasonable to assume that $\alpha(\delta \boldsymbol{W}_\ell^{(t)}, \varphi(\boldsymbol{h}_{\ell-1}^{(t)}(\boldsymbol{x}_i))) = \Theta_{\mathbb{P}}(1)$ for each example when the feature Gram matrix is nondegenerate and the contributions from different examples do not cancel (see also Everett et al., 2024). For fixed training times, it is then reasonable to impose the same order-one alignment on the cumulative displacement when the cross-time feature Gram matrix remains nondegenerate and successive update contributions do not cancel, and the nonlinearity preserves the RMS scale of activations and their changes.

The initialization part of the condition already separates the 3 parameterizations from table 1. For the input and hidden matrices,

$$\big\|\boldsymbol{W}_\ell^{(0)}\big\| \;=\; \Theta_{\mathbb{P}}(1), \qquad 0 \le \ell \le L, \tag{31}$$

under SP, NTP, and µP. At the readout,

$$\big\|\boldsymbol{W}_{L+1}^{(0)}\big\| \;=\; \begin{cases} \Theta_{\mathbb{P}}(1), & \text{µP}, \\ \Theta_{\mathbb{P}}(N^{1/2}), & \text{NTP and SP}. \end{cases} \tag{32}$$

More generally, let $\boldsymbol{W} \in \mathbb{R}^{d_{\text{out}} \times N}$ have i.i.d. $\mathcal{N}(0, N^{-2b})$ entries and be independent of $\boldsymbol{h} \in \mathbb{R}^N$, where $\|\boldsymbol{h}\| = \Theta_{\mathbb{P}}(1)$. Since d_{out} is fixed,

$$\begin{aligned} \boldsymbol{W}\boldsymbol{h}|\boldsymbol{h} \;&\sim\; \mathcal{N}\big(\boldsymbol{0}, N^{-2b}\|\boldsymbol{h}\|_2^2 \boldsymbol{I}_{d_{\text{out}}}\big), & \|\boldsymbol{W}\boldsymbol{h}\| \;&=\; \Theta_{\mathbb{P}}(N^{1/2-b}), \\ \|\boldsymbol{W}\| \;&=\; \Theta_{\mathbb{P}}(N^{1-b}), & \alpha(\boldsymbol{W}, \boldsymbol{h}) \;&=\; \Theta_{\mathbb{P}}(N^{-1/2}). \end{aligned} \tag{33}$$

Applying this with $b_{L+1} = 1/2$ for SP and NTP and $b_{L+1} = 1$ for µP gives

$$\big\|f_{\boldsymbol{\theta}^{(0)}}(\boldsymbol{x})\big\| \;=\; \begin{cases} \Theta_{\mathbb{P}}(1), & \text{SP and NTP}, \\ \Theta_{\mathbb{P}}(N^{-1/2}), & \text{µP}. \end{cases} \tag{34}$$

Thus µP satisfies the initialization part of the spectral condition and has vanishing initial outputs. SP and NTP violate the condition at the readout but have order-one initial outputs due to the random $N^{-1/2}$ alignment.

We next compute the initial SGD increment for a minibatch $\mathcal{B} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^B$. Let d_ℓ denote the fan-in of $\boldsymbol{W}_\ell$, i.e., $d_0 := d_{\text{in}}$ and $d_\ell := N$ for $1 \le \ell \le L+1$. For $0 \le \ell \le L$, define $\boldsymbol{h}_{\ell,i}^{(0)} := \boldsymbol{h}_\ell^{(0)}(\boldsymbol{x}_i)$, and set

$$\begin{aligned} \boldsymbol{g}_{\ell,i}^{(0)} \;&:=\; \nabla_{\boldsymbol{h}_\ell(\boldsymbol{x}_i)} \ell\big(f_{\boldsymbol{\theta}^{(0)}}(\boldsymbol{x}_i), y_i\big) \in \mathbb{R}^N, & 0 \le \ell \le L, \\ \boldsymbol{g}_{L+1,i}^{(0)} \;&:=\; \nabla \ell\big(f_{\boldsymbol{\theta}^{(0)}}(\boldsymbol{x}_i), y_i\big) \in \mathbb{R}^{d_{\text{out}}}, \end{aligned} \tag{35}$$

Then,

$$\widehat{\nabla}_{\boldsymbol{W}_0}\mathcal{L}(\boldsymbol{\theta}^{(0)}) \;=\; \frac{1}{B}\sum_{i=1}^{B}\boldsymbol{g}_{0,i}^{(0)}\boldsymbol{x}_i^\top, \quad \widehat{\nabla}_{\boldsymbol{W}_\ell}\mathcal{L}(\boldsymbol{\theta}^{(0)}) \;=\; \frac{1}{B}\sum_{i=1}^{B}\boldsymbol{g}_{\ell,i}^{(0)}\varphi\big(\boldsymbol{h}_{\ell-1,i}^{(0)}\big)^\top, \quad 1\le\ell\le L+1. \tag{36}$$

For every fixed example and hidden layer, assume $\|\boldsymbol{g}_{L+1,i}^{(0)}\|_2 = \Theta_{\mathbb{P}}(1)$ (i.e., initialization is not already fitting the target), $\|\boldsymbol{h}_{\ell,i}^{(0)}\| = \Theta_{\mathbb{P}}(1)$ and $\|\varphi(\boldsymbol{h}_{\ell,i}^{(0)})\| = \Theta_{\mathbb{P}}(1)$ (i.e., stable forward signal propagation), and $\|\varphi'(\boldsymbol{h}_{\ell,i}^{(0)})\| = \Theta_{\mathbb{P}}(1)$ and $\|\varphi'(\boldsymbol{h}_{\ell,i}^{(0)})\|_\infty = O_{\mathbb{P}}(1)$ (i.e., no saturated, dead, or exploding activation derivatives). Backpropagation computes

$$\boldsymbol{g}_{\ell,i}^{(0)} \;=\; (\boldsymbol{W}_{\ell+1}^{(0)})^\top\boldsymbol{g}_{\ell+1,i}^{(0)}\odot\varphi'(\boldsymbol{h}_{\ell,i}^{(0)}), \qquad 0\le\ell\le L. \tag{37}$$

For $0\le\ell<L$, each of the three parameterizations has $[\boldsymbol{W}_{\ell+1}^{(0)}]_{kj}\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,N^{-1})$. Each coordinate of $(\boldsymbol{W}_{\ell+1}^{(0)})^\top\boldsymbol{g}_{\ell+1,i}^{(0)}$ sums N mean-zero terms, so its RMS scale is again $\|\boldsymbol{g}_{\ell+1,i}^{(0)}\|$ (this is rigorous only for the factors being independent which is not true in general, but dependence from the forward pass does not change the width exponent here). Thus,

$$\frac{\|\boldsymbol{g}_{\ell,i}^{(0)}\|}{\|\boldsymbol{g}_{\ell+1,i}^{(0)}\|} \;=\; \Theta_{\mathbb{P}}(1), \qquad 0\le\ell<L. \tag{38}$$

The readout supplies the base case for this backward recursion. Setting $\ell=L$ in (37), each coordinate of $(\boldsymbol{W}_{L+1}^{(0)})^\top\boldsymbol{g}_{L+1,i}^{(0)}$ sums only the fixed number d_{out} of terms. Since $\|\boldsymbol{g}_{L+1,i}^{(0)}\|_2=\Theta_{\mathbb{P}}(1)$, its scale is set by the readout entries, and

$$\big\|\boldsymbol{g}_{L,i}^{(0)}\big\| \;=\; \begin{cases}\Theta_{\mathbb{P}}(N^{-1}), & \text{µP},\\ \Theta_{\mathbb{P}}(N^{-1/2}), & \text{NTP and SP.}\end{cases} \tag{39}$$

(38) then propagates this scale from layer L down to layer 0. Including the order-one output gradient gives

$$\big\|\boldsymbol{g}_{\ell,i}^{(0)}\big\| \;=\; \begin{cases}\Theta_{\mathbb{P}}(N^{-1}), & 0\le\ell\le L,\ \text{µP},\\ \Theta_{\mathbb{P}}(N^{-1/2}), & 0\le\ell\le L,\ \text{NTP or SP},\\ \Theta_{\mathbb{P}}(1), & \ell=L+1,\end{cases} \tag{40}$$

For a single outer product of $\boldsymbol{u}\in\mathbb{R}^m$ and $\boldsymbol{v}\in\mathbb{R}^n$, (19) and (20) give

$$\|\boldsymbol{u}\boldsymbol{v}^\top\| \;=\; \Big(\frac{n}{m}\Big)^{1/2}\|\boldsymbol{u}\boldsymbol{v}^\top\|_{2\to2} \;=\; \Big(\frac{n}{m}\Big)^{1/2}\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 \;=\; n\|\boldsymbol{u}\|\|\boldsymbol{v}\|. \tag{41}$$

For fixed B, under non-cancellation within a batch, the minibatch average has the scale of one representative outer product. Combining (18), (36), and (41) therefore gives, for $1\le\ell\le L+1$ and with $\boldsymbol{x}_i$ in place of $\varphi(\boldsymbol{h}_{\ell-1,i}^{(0)})$ for $\ell=0$,

$$\begin{aligned}\big\|\delta\boldsymbol{W}_\ell^{(0)}\big\| \;&=\; \eta N^{r-c_\ell}\left\|\frac{1}{B}\sum_{i=1}^{B}\boldsymbol{g}_{\ell,i}^{(0)}\varphi\big(\boldsymbol{h}_{\ell-1,i}^{(0)}\big)^\top\right\|\\ &=\; \eta N^{r-c_\ell}\Theta_{\mathbb{P}}\Big(\big\|\boldsymbol{g}_{\ell,i}^{(0)}\varphi\big(\boldsymbol{h}_{\ell-1,i}^{(0)}\big)^\top\big\|\Big)\\ &=\; \eta N^{r-c_\ell}d_\ell\Theta_{\mathbb{P}}\Big(\big\|\boldsymbol{g}_{\ell,i}^{(0)}\big\|\Big).\end{aligned} \tag{42}$$

Substituting $d_0=d_{\text{in}}$, $d_\ell=N$ for $1\le\ell\le L+1$, the scales in (40), and the layerwise c_ℓ from table 1 yields table 2.

At $r=0$, the µP readout increment in table 2 has norm $\eta\,\Theta_{\mathbb{P}}(1)$. Since it is aligned with the training activation, the previous non-cancellation assumption gives $\|f_{\boldsymbol{\theta}^{(1)}}(\boldsymbol{x})\|=\Theta_{\mathbb{P}}(1)$ under µP.

Under the alignment and non-cancellation assumptions above, table 2 also gives the width exponent of the direct preactivation change in I and H and of the direct output change in O. Let r_{I}, r_{H}, and r_{O} denote the largest exponents for which the respective first-step direct changes remain $O_{\mathbb{P}}(1)$. Reading these exponents from table 2 gives $(r_{\text{I}},r_{\text{H}},r_{\text{O}})=(1/2,-1/2,-1)$ for SP, $(1/2,1/2,0)$ for NTP, and $(0,0,0)$ for µP.

Table 2: RMS operator norms of the first SGD increment at global rate $\eta_N = \eta N^r$. Each entry denotes $\Theta_{\mathbb{P}}(\cdot)$ under the stated nondegeneracy assumptions.

	$\|\delta \boldsymbol{W}_0^{(0)}\|$ I	$\|\delta \boldsymbol{W}_\ell^{(0)}\|$ H, $1 \le \ell \le L$	$\|\delta \boldsymbol{W}_{L+1}^{(0)}\|$ O
µP	ηN^r	ηN^r	ηN^r
NTP	$\eta N^{r-1/2}$	$\eta N^{r-1/2}$	ηN^r
SP	$\eta N^{r-1/2}$	$\eta N^{r+1/2}$	ηN^{r+1}

B.3 Loss-Dependent Stability

With respect to the loss, the first-step calculation in § B.2 uses only the order-one scale of the initial output gradient and therefore does not distinguish loss functions. The loss becomes relevant when we ask whether these scales remain stable over subsequent steps. For ℓ_{CE}, d_{out} is the number of classes and $\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{out}}}$ is one-hot. For ℓ_2 loss, $\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{out}}}$ typically has order-one norm. The losses are

$$\ell_{\mathrm{CE}}(\boldsymbol{u}, \boldsymbol{y}) := -\boldsymbol{y}^\top \boldsymbol{u} + \log\left(\sum_{k=1}^{d_{\mathrm{out}}} \exp(u_k)\right), \qquad \nabla \ell_{\mathrm{CE}}(\boldsymbol{u}, \boldsymbol{y}) = \mathrm{softmax}(\boldsymbol{u}) - \boldsymbol{y}, \tag{43}$$

$$\ell_2(\boldsymbol{u}, \boldsymbol{y}) := \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{y}\|_2^2, \qquad \nabla \ell_2(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u} - \boldsymbol{y}. \tag{44}$$

Note that the gradient of ℓ_{CE} is uniformly bounded, $\left\|\nabla \ell_{\mathrm{CE}}(\boldsymbol{u}, \boldsymbol{y})\right\|_2 \le \sqrt{2}$, whereas the gradient of ℓ_2 grows linearly with the output $\boldsymbol{u}$.

Fix T independently of N. To pass from the first-step thresholds to fixed-time stability, assume that the corresponding scales remain valid for $0 \le t \le T$ whenever the output gradient remains $O_{\mathbb{P}}(1)$. Following Haas et al. (2025), for ℓ_{CE}, the uniformly bounded output gradient permits divergent outputs without amplifying the next gradient, so activation stability only requires stable I and H changes. For ℓ_2, a divergent O change enlarges the next residual $\boldsymbol{u} - \boldsymbol{y}$, so O should also remain stable. Consequently,

$$r_{\max}(\ell_{\mathrm{CE}}) = \min\{r_{\mathrm{I}}, r_{\mathrm{H}}\}, \qquad r_{\max}(\ell_2) = \min\{r_{\mathrm{I}}, r_{\mathrm{H}}, r_{\mathrm{O}}\}. \tag{45}$$

Table 3: Behavior at the maximal fixed-time stable global learning rate for $L \ge 1$. FL = feature learning, CD = controlled divergence (Haas et al., 2025).

		SP			**NTP**			**µP**		
Loss		I	H	O	I	H	O	I	H	O
ℓ_{CE}	max. stable η_N	$\Theta(N^{-1/2})$			$\Theta(N^{1/2})$			$\Theta(1)$		
	behavior	lazy	FL	CD	FL	FL	CD	FL	FL	$\Theta_{\mathbb{P}}(1)$
ℓ_2	max. stable η_N	$\Theta(N^{-1})$			$\Theta(1)$			$\Theta(1)$		
	behavior	lazy	lazy	$\Theta_{\mathbb{P}}(1)$	lazy	lazy	$\Theta_{\mathbb{P}}(1)$	FL	FL	$\Theta_{\mathbb{P}}(1)$

C Quadratic Approximation of Chord Deviation

Fix a dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M \subset \mathbb{R}^{d_{\text{in}}} \times \mathcal{Y}$, with $M \geq 1$, and let $\boldsymbol{\theta} \in \mathbb{R}^p$ collect the p parameters. Define

$$F(\boldsymbol{\theta}) := \big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big)_{i=1}^M \in \mathbb{R}^{Md_{\text{out}}}, \tag{46}$$

$$\ell(\boldsymbol{u}) := \frac{1}{M}\sum_{i=1}^M \ell(\boldsymbol{u}_i, y_i), \qquad \boldsymbol{u} = (\boldsymbol{u}_i)_{i=1}^M, \quad \boldsymbol{u}_i \in \mathbb{R}^{d_{\text{out}}}, \tag{47}$$

$$\mathcal{L}(\boldsymbol{\theta}) := \ell\big(F(\boldsymbol{\theta})\big). \tag{48}$$

Assume that F and ℓ are C^2. For two parameter vectors $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$, define the parameter and output displacements and their interpolation paths by

$$\boldsymbol{\Delta} := \boldsymbol{\theta}_B - \boldsymbol{\theta}_A, \qquad \boldsymbol{\theta}(s) := \boldsymbol{\theta}_A + s\boldsymbol{\Delta}, \tag{49}$$

$$\Delta F := F(\boldsymbol{\theta}_B) - F(\boldsymbol{\theta}_A), \qquad \overline{F}(s) := F(\boldsymbol{\theta}_A) + s\Delta F. \tag{50}$$

Here $s \in [0,1]$. Their midpoints are

$$\overline{\boldsymbol{\theta}} := \boldsymbol{\theta}(1/2) = \frac{\boldsymbol{\theta}_A + \boldsymbol{\theta}_B}{2}, \qquad \overline{F}(1/2) = \frac{F(\boldsymbol{\theta}_A) + F(\boldsymbol{\theta}_B)}{2}. \tag{51}$$

In general, $F(\boldsymbol{\theta}(s)) \neq \overline{F}(s)$.

Definition C.1 (Midpoint gaps)**.** The midpoint chord deviation C, Jensen gap J, and interpolation gap G are

$$C := \mathcal{L}(\overline{\boldsymbol{\theta}}) - \frac{\mathcal{L}(\boldsymbol{\theta}_A) + \mathcal{L}(\boldsymbol{\theta}_B)}{2}, \tag{52}$$

$$J := \frac{\mathcal{L}(\boldsymbol{\theta}_A) + \mathcal{L}(\boldsymbol{\theta}_B)}{2} - \ell(\overline{F}(1/2)), \tag{53}$$

$$G := \mathcal{L}(\overline{\boldsymbol{\theta}}) - \ell(\overline{F}(1/2)). \tag{54}$$

They satisfy the exact identity

$$C = G - J. \tag{55}$$

Write $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) \in \mathbb{R}^{Md_{\text{out}} \times p}$ for the Jacobian with output coordinates as rows, and let F_a denote the a-th scalar coordinate of F. The Hessian of $\mathcal{L}$ decomposes as

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{H}_{\text{GN}}(\boldsymbol{\theta}) + \boldsymbol{H}_{\text{rem}}(\boldsymbol{\theta}), \tag{56}$$

where the Gauss-Newton (GN) (Martens, 2020) and its remainder are

$$\boldsymbol{H}_{\text{GN}}(\boldsymbol{\theta}) := \big(\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\big)^\top \nabla^2 \ell\big(F(\boldsymbol{\theta})\big) \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}), \tag{57}$$

$$\boldsymbol{H}_{\text{rem}}(\boldsymbol{\theta}) := \sum_{a=1}^{Md_{\text{out}}} \big[\nabla \ell\big(F(\boldsymbol{\theta})\big)\big]_a \nabla^2_{\boldsymbol{\theta}} F_a(\boldsymbol{\theta}). \tag{58}$$

Proposition C.2 (Exact GN decomposition)**.** *Define the integrated non-GN contribution by*

$$R := -\frac{1}{4}\int_0^1 \big(1 - |2s-1|\big)\boldsymbol{\Delta}^\top \boldsymbol{H}_{\text{rem}}(\boldsymbol{\theta}(s))\boldsymbol{\Delta}\, \mathrm{d}s, \tag{59}$$

and the GN path mismatch by

$$E := \frac{1}{4}\int_0^1 \big(1 - |2s-1|\big)\Big(\Delta F^\top \nabla^2 \ell(\overline{F}(s))\Delta F - \boldsymbol{\Delta}^\top \boldsymbol{H}_{\text{GN}}(\boldsymbol{\theta}(s))\boldsymbol{\Delta}\Big)\, \mathrm{d}s. \tag{60}$$

Then, the midpoint quantities satisfy

$$C = -J + R + E, \qquad G = R + E. \tag{61}$$

Proof. The midpoint Peano-kernel identity gives, for every $g \in C^2([0,1])$,

$$g(1/2) - \frac{g(0) + g(1)}{2} = -\frac{1}{4}\int_0^1 \big(1 - |2s-1|\big) g''(s)\, \mathrm{d}s. \tag{62}$$

Applying (62) to $g(s) := \mathcal{L}(\boldsymbol{\theta}(s))$, followed by (56), yields

$$C = -\frac{1}{4}\int_0^1 \big(1-|2s-1|\big)\Big(\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{GN}}(\boldsymbol{\theta}(s))\boldsymbol{\Delta} + \boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{rem}}(\boldsymbol{\theta}(s))\boldsymbol{\Delta}\Big)\,\mathrm{d}s. \tag{63}$$

Applying the same identity to $g(s) := \ell(\overline{F}(s))$ gives

$$J = \frac{1}{4}\int_0^1 \big(1-|2s-1|\big)\Delta F^\top \nabla^2 \ell(\overline{F}(s))\Delta F\,\mathrm{d}s. \tag{64}$$

Adding and subtracting (64) in (63) proves the identity for C. The identity for G then follows from (55). □

Thus, the interpolation gap G equals the integrated non-GN contribution only up to the finite-distance path mismatch E. If parameter interpolation maps exactly to the output chord, then $F(\boldsymbol{\theta}(s)) = \overline{F}(s)$ for every $s \in [0,1]$. In that case, $R = E = G = 0$, and $C = -J$.

Proposition C.3 (Local GN correspondence)**.** *Suppose that F and ℓ are C^4 near $\overline{\boldsymbol{\theta}}$ and $F(\overline{\boldsymbol{\theta}})$, respectively, and let $\|\boldsymbol{\Delta}\|_2 \to 0$ with $\overline{\boldsymbol{\theta}}$ fixed. Then,*

$$J = \frac{1}{8}\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{GN}}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big), \tag{65}$$

$$G = -\frac{1}{8}\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{rem}}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big), \tag{66}$$

$$C = -\frac{1}{8}\boldsymbol{\Delta}^\top \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big), \tag{67}$$

$$E = O(\|\boldsymbol{\Delta}\|_2^4). \tag{68}$$

Proof. A symmetric Taylor expansion about $\overline{\boldsymbol{\theta}}$ gives

$$\overline{F}(1/2) - F(\overline{\boldsymbol{\theta}}) = \frac{1}{8}\nabla^2_{\boldsymbol{\theta}} F(\overline{\boldsymbol{\theta}})[\boldsymbol{\Delta},\boldsymbol{\Delta}] + O\big(\|\boldsymbol{\Delta}\|_2^4\big), \tag{69}$$

$$\Delta F = \nabla_{\boldsymbol{\theta}} F(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^3\big). \tag{70}$$

Expanding the symmetric Jensen gap around $\overline{F}(1/2)$ and using (69) and (70) yields

$$J = \frac{1}{8}\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{GN}}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big). \tag{71}$$

Analogously, expanding $\ell(\overline{F}(1/2))$ about $F(\overline{\boldsymbol{\theta}})$ gives

$$G = -\frac{1}{8}\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{rem}}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big). \tag{72}$$

Their difference and (56) give the expansion for C. From a symmetric expansion of (59) we obtain

$$R = -\frac{1}{8}\boldsymbol{\Delta}^\top \boldsymbol{H}_{\mathrm{rem}}(\overline{\boldsymbol{\theta}})\boldsymbol{\Delta} + O\big(\|\boldsymbol{\Delta}\|_2^4\big). \tag{73}$$

Finally, combining (73) with (61) shows (68). □

Note that the term $1-|2s-1|$ in (62) is maximal at $s = 1/2$. Writing $g(s) := \mathcal{L}(\boldsymbol{\theta}(s))$, one can compute

$$C - [-\tfrac{1}{8}g''(\tfrac{1}{2})] = -\tfrac{1}{384}g^{(4)}(\tfrac{1}{2}) + o(\|\boldsymbol{\Delta}\|_2^4), \tag{74}$$

whereas the Hessian at any other point of the chord incurs an error of order $\|\boldsymbol{\Delta}\|_2^3$, and the average of the endpoint Hessians would have leading error $\frac{5}{384}g^{(4)}(1/2)$. Singh et al. (2024) obtain a related quadratic barrier approximation from second-order expansions around the endpoints, which requires the endpoints to be stationary points so that the gradient terms vanish. In prop. C.3, all odd terms cancel from the Taylor approximation and, thus, we do not need to assume a stationary point.

D SCALING LIMITS FOR LOSS BARRIERS

In this section, we will show that for the procedure of sharing initialization, training a fixed prefix of b steps, and then branching into two procedures that get independent minibatch randomness, the joint evolution of the two models can be represented in one tensor program (Yang & Hu, 2021; Yang et al., 2021; Yang & Littwin, 2023). By appending forward passes along the chord, we are then able to obtain a joint width limit for its predictions and losses, and obtain control for the maximum over the entire chord as well. For simplicity, we state this result for smooth MLPs trained with SGD under μP at fixed depth and a finite training horizon, though the finite-grid statement likely extends to much broader architectures and adaptive optimizers (e.g., using the framework of Yang & Littwin (2023)). The uniform control along the chord below uses that SGD updates are of low rank.

D.1 MODEL AND BRANCHING PROCEDURE

We use the model and RMS norm notation of § B with $a_\ell = 0$ and $r = 0$, which is the representation of μP in table 1. Fix the input and output dimensions $d_{\text{in}}, d_{\text{out}}$, depth L, minibatch size B. We consider a fixed training horizon of $T < \infty$ steps, and branch the model into two copies at time $0 \le b \le T$, independent of the width N.

Fix a constant base LR $\eta > 0$, shared by both branches and all widths. For a minibatch $\mathcal{B}$, let $\boldsymbol{\theta} \mapsto U_{N,\eta}(\boldsymbol{\theta}; \mathcal{B})$ be the function mapping a parameter vector $\boldsymbol{\theta}$ to the updated parameter vector after one SGD step in (18). We write $\mathcal{H} := (\mathcal{H}_A, \mathcal{H}_B)$, where

$$\mathcal{H}_R := \big(\mathcal{B}_R^{(t)}\big)_{t=0}^{T-1}, \qquad \mathcal{B}_R^{(t)} := \{(\boldsymbol{x}_{R,i}^{(t)}, y_{R,i}^{(t)})\}_{i=1}^{B}, \qquad R \in \{A, B\}. \tag{75}$$

Definition D.1 (SGD branched at a checkpoint)**.** Suppose the histories have the common prefix

$$\mathcal{B}_A^{(t)} = \mathcal{B}_B^{(t)}, \qquad 0 \le t < b. \tag{76}$$

Draw one initialization $\boldsymbol{\theta}_N^{(0)}$. The coupled trajectories are defined by

$$\begin{aligned} \boldsymbol{\theta}_{A,N}^{(0)} &:= \boldsymbol{\theta}_N^{(0)}, \qquad \boldsymbol{\theta}_{B,N}^{(0)} := \boldsymbol{\theta}_N^{(0)}, \\ \boldsymbol{\theta}_{R,N}^{(t+1)} &:= U_{N,\eta}\big(\boldsymbol{\theta}_{R,N}^{(t)}; \mathcal{B}_R^{(t)}\big), \qquad R \in \{A, B\}, \quad 0 \le t < T. \end{aligned} \tag{77}$$

Clearly, $\boldsymbol{\theta}_{A,N}^{(t)} = \boldsymbol{\theta}_{B,N}^{(t)}$ for $0 \le t \le b$. For independent continuations from a common data distribution, the post-branch histories satisfy

$$\begin{gathered} \mathcal{H}^- := \big(\mathcal{B}_A^{(t)}\big)_{0 \le t < b}, \qquad \mathcal{H}_R^+ := \big(\mathcal{B}_R^{(t)}\big)_{b \le t < T}, \\ \mathcal{H}_A^+ \perp\!\!\!\perp \mathcal{H}_B^+ \mid \mathcal{H}^-. \end{gathered} \tag{78}$$

We will need the following assumptions:

Assumption D.2 (Regularity and external randomness)**.** For the activation φ, let $\varphi \in C^2$, $\|\varphi'\|_\infty < \infty$, and let φ'' have polynomial growth. Let the loss $\ell : \mathbb{R}^{d_{\text{out}}} \times \mathcal{Y} \to \mathbb{R}$ be jointly measurable. For every $y \in \mathcal{Y}$, $\ell(\cdot, y) : \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}$ is C^2, and its first two derivatives have polynomial growth. The label space $\mathcal{Y}$ is a standard Borel space. The distribution of the history $\mathcal{H}$ is width-independent, and $\mathcal{H} \perp\!\!\!\perp \boldsymbol{\theta}_N^{(0)}$. Use the same realization of $\mathcal{H}$ at every width. $\{\boldsymbol{\theta}_N^{(0)}\}_N$ may have any coupling with the marginal laws prescribed by (11).

For example, tanh, GELU, and the identity satisfy these conditions, as do squared loss ℓ_2 and cross-entropy ℓ_{CE}. The smoothness assumption excludes ReLU.

We now fix a deterministic evaluation dataset $\mathcal{D}_{\text{eval}} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M_{\text{eval}}} \subset \mathbb{R}^{d_{\text{in}}} \times \mathcal{Y}$, with $M_{\text{eval}} \ge 1$. For $s \in [0, 1]$, define the parameter chord and its predictions by

$$\boldsymbol{\theta}_N^{(t)}(s) := (1 - s)\boldsymbol{\theta}_{A,N}^{(t)} + s\boldsymbol{\theta}_{B,N}^{(t)}, \qquad f_{N,i}^{(t)}(s) := f_{N,\boldsymbol{\theta}_N^{(t)}(s)}(\boldsymbol{x}_i). \tag{79}$$

The output ensemble and the two evaluation losses are

$$\begin{aligned} \overline{f}^{(t)}_{N,i}(s) &:= (1-s)f^{(t)}_{N,i}(0) + sf^{(t)}_{N,i}(1), \\ \mathcal{L}^{(t)}_{N}(s) &:= \frac{1}{M_{\text{eval}}}\sum_{i=1}^{M_{\text{eval}}} \ell(f^{(t)}_{N,i}(s), y_i), \qquad \mathcal{L}^{(t)}_{\text{ens},N}(s) := \frac{1}{M_{\text{eval}}}\sum_{i=1}^{M_{\text{eval}}} \ell(\overline{f}^{(t)}_{N,i}(s), y_i). \end{aligned} \tag{80}$$

This yields the chord deviation and its decomposition

$$C^{(t)}_N(s) := \mathcal{L}^{(t)}_N(s) - (1-s)\mathcal{L}^{(t)}_N(0) - s\mathcal{L}^{(t)}_N(1), \tag{81}$$

$$J^{(t)}_N(s) := (1-s)\mathcal{L}^{(t)}_N(0) + s\mathcal{L}^{(t)}_N(1) - \mathcal{L}^{(t)}_{\text{ens},N}(s), \tag{82}$$

$$G^{(t)}_N(s) := \mathcal{L}^{(t)}_N(s) - \mathcal{L}^{(t)}_{\text{ens},N}(s), \tag{83}$$

$$C^{(t)}_N(s) = G^{(t)}_N(s) - J^{(t)}_N(s). \tag{84}$$

Convexity of $\ell(\cdot, y)$ additionally gives $J^{(t)}_N(s) \geq 0$, but we do not need it for convergence. For a vector-valued curve $\boldsymbol{v}$, we write $\|\boldsymbol{v}\|_\infty := \sup_{s\in[0,1]}\|\boldsymbol{v}(s)\|$ in the following.

D.2 µP Width Limit on a Finite Grid

The following theorem is the main result of this section and establishes joint convergence of the predictions and losses at finitely many interpolation points.

Theorem D.3 (Width limit at finitely many interpolation points)**.** *Under assumption D.2, the trajectories of def. D.1 admit limiting predictions $f^{(t)}_{\infty,i}(s)$ and loss observables $C^{(t)}_\infty(s)$, $J^{(t)}_\infty(s)$, and $G^{(t)}_\infty(s)$, measurable in $\mathcal{H}$ for each fixed $s \in [0,1]$. The loss limits are given by (80)–(83) with $f^{(t)}_{N,i}(\cdot)$ replaced by $f^{(t)}_{\infty,i}(\cdot)$. For every fixed nonempty finite $\mathcal{S} \subset [0,1]$, convergence is almost sure and joint over $s \in \mathcal{S}$, $b \leq t \leq T$, and $1 \leq i \leq M_{\text{eval}}$. Further, for every $\varepsilon > 0$, almost surely,*

$$N^{1/2-\varepsilon} \max_{b\leq t\leq T} \max_{s\in\mathcal{S}} \max_{X\in\{C,J,G\}} |X^{(t)}_N(s) - X^{(t)}_\infty(s)| \to 0. \tag{85}$$

The prediction coordinates satisfy the same rate. These statements also hold with conditional probability one for almost every history $\mathcal{H}$.

The limit is deterministic when the training history is fixed (and shared by widths), though random histories can still produce a random limit (in which initialization randomness disappears). With the history coupling across widths, the unconditional statement is also almost sure convergence to the random values $X^{(t)}_\infty(s)$, $X \in \{C, J, G\}$. Note that the theorem only concerns finite training horizons and the fixed finite evaluation dataset $\mathcal{D}_{\text{eval}}$, and the width at which the limit is approached may deteriorate with the horizon T. While the existence of this limit provides basic justification for our empirical findings, it makes no statement about what that limit actually is, i.e., if the limiting chord deviation is small or the approximation is accurate/stable at a certain finite width.

To prove thm. D.3, we encode both training branches and their interpolated forward passes in a single tensor program (Yang & Littwin, 2023, Definition 2.6.1). The master theorem (Yang & Littwin, 2023, Theorem 2.6.10) then directly yields joint convergence at any finite set of interpolation points. Throughout this construction, we condition on a realization of $\mathcal{H}$ and keep the minibatches and labels fixed.

We use the following subset of the language in Yang & Littwin (2023, Definition 2.6.1). Starting from independent Gaussian vectors and matrices, each instruction creates a vector in $\mathbb{R}^N$ or a scalar:

$$\texttt{[matrix multiplication]} \qquad \boldsymbol{v} := \boldsymbol{W}^{(0)}_{\ell,N}\boldsymbol{u} \quad \text{or} \quad \boldsymbol{v} := (\boldsymbol{W}^{(0)}_{\ell,N})^\top\boldsymbol{u}, \tag{86}$$

$$\texttt{[coordinatewise map]} \qquad v_k := \psi(u^1_k, \ldots, u^m_k; c_1, \ldots, c_p),\ 1 \leq k \leq N, \tag{87}$$

$$\texttt{[average]} \qquad c := \texttt{Avg}_N(\boldsymbol{v}) := \frac{1}{N}\sum_{k=1}^{N} v_k. \tag{88}$$

Here the operands are previously available objects, and matrix multiplication uses an initial hidden matrix, $1 \le \ell \le L$. For example, $N^{-1}\boldsymbol{u}^\top \boldsymbol{v} = \mathtt{Avg}_N(\boldsymbol{u} \odot \boldsymbol{v})$, where $\odot$ denotes coordinatewise multiplication. A function of finitely many scalars can be evaluated by broadcasting its value to a constant vector and averaging that vector (Yang & Littwin, 2023, Lemma 2.6.2).

Lemma D.4 (Tensor program representation)**.** *Under assumption D.2, fix a history and a finite nonempty $\mathcal{S} \subset [0,1]$. The joint collection*

$$\left\{ f_{N,i}^{(t)}(s) \;\middle|\; b \le t \le T, \quad 1 \le i \le M_{\mathrm{eval}}, \quad s \in \mathcal{S} \right\} \tag{89}$$

is obtained by grouping designated scalar outputs of one finite tensor program satisfying the Gaussian Setup 2.6.3 of Yang & Littwin (2023) and having no initial scalars. Its instructions and their coefficients are independent of width, apart from the normalization in $\mathtt{Avg}_N$ and the Gaussian matrix variances.

Proof. **Step ①: Input and output layers.** For one branch at a time, temporarily suppress indices and write $\boldsymbol{W}_\ell$ for its current weights. Store the input columns and rescaled output rows as

$$\boldsymbol{u}_j := (\boldsymbol{W}_0)_{:j} \in \mathbb{R}^N, \qquad \boldsymbol{v}_c := N((\boldsymbol{W}_{L+1})_{c:})^\top \in \mathbb{R}^N, \qquad 1 \le j \le d_{\mathrm{in}}, \quad 1 \le c \le d_{\mathrm{out}}. \tag{90}$$

Under the μP initialization, these vectors initially have independent standard Gaussian entries. The initial hidden matrices have independent $\mathcal{N}(0, N^{-1})$ entries and are independent of these vectors. Their rectangular input and output counterparts act by

$$\boldsymbol{h}_0(\boldsymbol{x}) = \sum_{j=1}^{d_{\mathrm{in}}} x_j \boldsymbol{u}_j, \qquad f_c(\boldsymbol{x}) = \mathtt{Avg}_N\big(\boldsymbol{v}_c \odot \varphi(\boldsymbol{h}_L(\boldsymbol{x}))\big). \tag{91}$$

Thus, the input layer uses coordinatewise linear combinations, and each output coordinate is an average. The factor N in $\boldsymbol{v}_c$ expresses the μP readout as an average of vectors whose initial coordinate scales are independent of width.

Step ②: Backpropagation and updates. We use the per-example Euclidean gradients $\boldsymbol{g}_\ell$ defined as in (35), evaluated at the current parameters, and set $\widetilde{\boldsymbol{g}}_\ell := N\boldsymbol{g}_\ell$ for $0 \le \ell \le L$. The output gradient $\boldsymbol{g}_{L+1} := \nabla \ell(f, y)$ has a fixed number of scalar coordinates. With $i \in \{1, \dots, B\}$ indexing the minibatch, one can compute

$$\widetilde{\boldsymbol{g}}_L = \varphi'(\boldsymbol{h}_L) \odot \sum_{c=1}^{d_{\mathrm{out}}} [\boldsymbol{g}_{L+1}]_c \boldsymbol{v}_c, \tag{92}$$

$$\widetilde{\boldsymbol{g}}_\ell = \varphi'(\boldsymbol{h}_\ell) \odot (\boldsymbol{W}_{\ell+1})^\top \widetilde{\boldsymbol{g}}_{\ell+1}, \qquad 0 \le \ell < L, \tag{93}$$

$$\delta \boldsymbol{W}_0 = -\frac{\eta}{B} \sum_{i=1}^{B} \widetilde{\boldsymbol{g}}_{0,i} \boldsymbol{x}_i^\top, \tag{94}$$

$$\delta \boldsymbol{W}_\ell = -\frac{\eta}{NB} \sum_{i=1}^{B} \widetilde{\boldsymbol{g}}_{\ell,i} \varphi(\boldsymbol{h}_{\ell-1,i})^\top, \qquad 1 \le \ell \le L, \tag{95}$$

$$\delta \boldsymbol{W}_{L+1} = -\frac{\eta}{NB} \sum_{i=1}^{B} \boldsymbol{g}_{L+1,i} \varphi(\boldsymbol{h}_{L,i})^\top. \tag{96}$$

The input and output vector updates are therefore $\delta \boldsymbol{u}_j = -(\eta/B) \sum_i x_{i,j} \widetilde{\boldsymbol{g}}_{0,i}$ and $\delta \boldsymbol{v}_c = -(\eta/B) \sum_i [\boldsymbol{g}_{L+1,i}]_c \varphi(\boldsymbol{h}_{L,i})$. Both are coordinatewise operations whose coefficients are fixed data or previously computed program scalars. All gradients in this update use the parameters at its start.

Step ③: Represent a trained hidden matrix. We now reattach branch and time indices and suppress N on the stored vectors. In particular, $\boldsymbol{h}_{\ell,R,i}^{(t)} := \boldsymbol{h}_\ell(\boldsymbol{x}_{R,i}^{(t)}; \boldsymbol{\theta}_{R,N}^{(t)})$, and index the gradients analogously. Summing the updates gives, for $1 \le \ell \le L$,

$$\boldsymbol{W}_{\ell,R,N}^{(t)} = \boldsymbol{W}_{\ell,N}^{(0)} - \frac{\eta}{NB} \sum_{t'=0}^{t-1} \sum_{i=1}^{B} \widetilde{\boldsymbol{g}}_{\ell,R,i}^{(t')} \varphi(\boldsymbol{h}_{\ell-1,R,i}^{(t')})^\top. \tag{97}$$

At most TB outer products are stored per branch and hidden layer. Consequently, its action on any available program vector $\boldsymbol{a} \in \mathbb{R}^N$, including a vector depending on these same weights, is exactly

$$\boldsymbol{W}_{\ell,R,N}^{(t)}\boldsymbol{a} \;=\; \boldsymbol{W}_{\ell,N}^{(0)}\boldsymbol{a} - \frac{\eta}{B}\sum_{t'=0}^{t-1}\sum_{i=1}^{B}\widetilde{\boldsymbol{g}}_{\ell,R,i}^{(t')}\,\mathtt{Avg}_N\big(\varphi(\boldsymbol{h}_{\ell-1,R,i}^{(t')})\odot\boldsymbol{a}\big), \tag{98}$$

$$(\boldsymbol{W}_{\ell,R,N}^{(t)})^\top\boldsymbol{a} \;=\; (\boldsymbol{W}_{\ell,N}^{(0)})^\top\boldsymbol{a} - \frac{\eta}{B}\sum_{t'=0}^{t-1}\sum_{i=1}^{B}\varphi(\boldsymbol{h}_{\ell-1,R,i}^{(t')})\,\mathtt{Avg}_N\big(\widetilde{\boldsymbol{g}}_{\ell,R,i}^{(t')}\odot\boldsymbol{a}\big). \tag{99}$$

Each term uses only (86)–(88). At time t, the sums in (98) and (99) use only vectors from times $t' < t$. Induction over t proves that every training step is a finite sequence of the allowed instructions.

Step ④: Branching and interpolation. Execute the first b updates once. Both continuations start from the resulting input and output vectors and the same stored update lists. Continue each with its own fixed minibatch sequence as two parts of one tensor program. For an interpolation point s, form $\boldsymbol{u}_j^{(t)}(s) := (1-s)\boldsymbol{u}_{A,j}^{(t)} + s\boldsymbol{u}_{B,j}^{(t)}$ and $\boldsymbol{v}_c^{(t)}(s) := (1-s)\boldsymbol{v}_{A,c}^{(t)} + s\boldsymbol{v}_{B,c}^{(t)}$. For a hidden layer, define its interpolated matrix action by

$$\boldsymbol{W}_{\ell,N}^{(t)}(s)\boldsymbol{a} \;:=\; (1-s)\boldsymbol{W}_{\ell,A,N}^{(t)}\boldsymbol{a} + s\boldsymbol{W}_{\ell,B,N}^{(t)}\boldsymbol{a}, \qquad 1 \le \ell \le L, \tag{100}$$

using (98) for both terms. The shared initial matrix and all prefix updates contribute once, because their coefficients sum to $(1-s)+s=1$. For each evaluation input, append the forward recursion

$$\boldsymbol{h}_{0,i}^{(t)}(s) \;:=\; \sum_{j=1}^{d_{\text{in}}} x_{i,j}\boldsymbol{u}_j^{(t)}(s), \tag{101}$$

$$\boldsymbol{h}_{\ell,i}^{(t)}(s) \;:=\; \boldsymbol{W}_{\ell,N}^{(t)}(s)\varphi\big(\boldsymbol{h}_{\ell-1,i}^{(t)}(s)\big), \qquad 1 \le \ell \le L, \tag{102}$$

$$[f_{N,i}^{(t)}(s)]_c \;=\; \mathtt{Avg}_N\big(\boldsymbol{v}_c^{(t)}(s)\odot\varphi\big(\boldsymbol{h}_{L,i}^{(t)}(s)\big)\big). \tag{103}$$

The hidden activations are recomputed from the interpolated weights. This is the parameter-interpolated network in (79). Interpolating endpoint predictions would instead give the output ensemble. Appending this recursion for each $b \le t \le T$, $s \in \mathcal{S}$, and evaluation input produces all the claimed outputs in a single finite program.

Step ⑤: Applicability of the master theorem (Yang & Littwin, 2023, Theorem 2.6.10). A map $\psi : \mathbb{R}^m \to \mathbb{R}$ is pseudo-Lipschitz if, for some $C > 0$ and $p \ge 0$,

$$|\psi(\boldsymbol{a}) - \psi(\boldsymbol{b})| \;\le\; C\|\boldsymbol{a}-\boldsymbol{b}\|_2\big(1 + \|\boldsymbol{a}\|_2^p + \|\boldsymbol{b}\|_2^p\big), \qquad \boldsymbol{a},\boldsymbol{b} \in \mathbb{R}^m. \tag{104}$$

Analogously to Yang & Littwin (2023, Assumption 2.3.2 and footnote 3), we verify this regularity for the activation derivative, loss-gradient components, and update maps. A C^1 map with $\|\nabla\psi(\boldsymbol{z})\|_2 \le C(1+\|\boldsymbol{z}\|_2^p)$ satisfies (104) by the mean-value theorem, since the segment joining $\boldsymbol{a}$ and $\boldsymbol{b}$ lies in the ball of radius $\max\{\|\boldsymbol{a}\|_2, \|\boldsymbol{b}\|_2\}$. Thus the bounded derivative of φ, polynomial bound on φ'', and polynomial bounds on the loss Hessian ensure this condition for φ, φ', and each component of a fixed-label loss gradient, respectively. Sums and products of these maps retain polynomially bounded derivatives. Fixed inputs, labels, η, and interpolation coefficients are absorbed into these maps, so the set of initial scalars can be empty. The initialization and all instructions therefore satisfy the setup of Yang & Littwin (2023). Clearly, the number of instructions is finite, since $d_{\text{in}}, d_{\text{out}}, L, B, T, M_{\text{eval}}$, and $|\mathcal{S}|$ are fixed. □

With lem. D.4 in place, thm. D.3 now follows by a straightforward application of the master theorem (Yang & Littwin, 2023, Theorem 2.6.10).

Proof of thm. D.3. Fix a history and apply lem. D.4 to $\mathcal{S} \cup \{0,1\}$. The scalar master theorem (Yang & Littwin, 2023, Theorem 2.6.10) gives joint a.s. convergence of the prediction coordinates to limits that are deterministic given the history, with the rate in (85) because the program has no initial scalars. Local Lipschitz continuity of the finite evaluation losses transfers this convergence and rate to C, J, G. Adding interpolation points leaves the finite-width predictions at existing points unchanged, so their limits agree. For measurability in $\mathcal{H}$, note that $f_{N,i}^{(t)}(s)$ is jointly measurable in $(\mathcal{H}, \boldsymbol{\theta}_N^{(0)})$, as $\nabla\ell$ is jointly measurable as a limit of difference quotients. Integrating over $\mathcal{H}$ gives the unconditional statement. □

D.3 Uniform Convergence of μP Barrier Curves

We now extend thm. D.3 to the entire chord by bounding prediction variations uniformly in width. We see the barrier curves $C_N^{(t)}(\cdot)$ as elements of $C([0,1],\mathbb{R})$ equipped with $\|h\|_\infty := \sup_{s\in[0,1]} |h(s)|$.

Theorem D.5 (Uniform width limit of barrier curves)**.** *Under assumption D.2, the limits in thm. D.3 have Lipschitz versions $f_{\infty,i}^{(t)}(\cdot)$, $C_\infty^{(t)}(\cdot)$, $J_\infty^{(t)}(\cdot)$, and $G_\infty^{(t)}(\cdot)$, measurable in $\mathcal{H}$. As $N\to\infty$,*

$$\max_{b\le t\le T} \|C_N^{(t)}(\cdot) - C_\infty^{(t)}(\cdot)\|_\infty \;\to\; 0 \tag{105}$$

almost surely. The same convergence holds for $J_N^{(t)}(\cdot)$, $G_N^{(t)}(\cdot)$, and the prediction curves $f_{N,i}^{(t)}(\cdot)$, jointly over the evaluation inputs. These statements also hold with conditional probability 1 for almost every history $\mathcal{H}$.

In particular, the loss barrier $\max_{s\in[0,1]} C_N^{(t)}(s)$ and the minimum $\min_{s\in[0,1]} C_N^{(t)}(s)$ converge almost surely to those of $C_\infty^{(t)}$, since both are 1-Lipschitz functionals in the supremum norm.

Proof. We fix a history.

Step ①: Uniform control along the chord. We show that there is a finite constant $K(\mathcal{H})$, independent of N, such that almost surely, for all sufficiently large N,

$$\max_{b\le t\le T}\ \max_{1\le i\le M_{\text{eval}}}\ \sup_{s\in[0,1]} \left(\|f_{N,i}^{(t)}(s)\| + \|\partial_s f_{N,i}^{(t)}(s)\|\right) \;\le\; K(\mathcal{H}). \tag{106}$$

The curves $C_N^{(t)}(\cdot), J_N^{(t)}(\cdot), G_N^{(t)}(\cdot)$ consequently have uniformly bounded supremum norms and Lipschitz constants. For a matrix $\boldsymbol{Z}\in\mathbb{R}^{m\times n}$ with independent $\mathcal{N}(0,1)$ entries, the Gaussian singular-value bound of Davidson & Szarek (2001, Theorem II.13) gives, after rescaling,

$$\mathbb{P}\big(\|\boldsymbol{Z}\|_{2\to2} > \sqrt{m}+\sqrt{n}+u\big) \;\le\; \exp(-u^2/2), \qquad u>0. \tag{107}$$

Define $\boldsymbol{Z}_{0,N} := \boldsymbol{W}_{0,N}^{(0)}$, $\boldsymbol{Z}_{\ell,N} := \sqrt{N}\boldsymbol{W}_{\ell,N}^{(0)}$ for $1\le\ell\le L$, and $\boldsymbol{Z}_{L+1,N} := N\boldsymbol{W}_{L+1,N}^{(0)}$. Under the μP initialization, each $\boldsymbol{Z}_{\ell,N}$ has independent standard Gaussian entries and the same dimensions as its weight matrix. Apply (107) with $u=\sqrt{N}$ and use the RMS normalization in (20). For $N\ge N_0 := \max\{d_{\text{in}}, d_{\text{out}}\}$, this gives

$$\|\boldsymbol{W}_{0,N}^{(0)}\| \;=\; \sqrt{\frac{d_{\text{in}}}{N}}\|\boldsymbol{Z}_{0,N}\|_{2\to2} \;\le\; 3\sqrt{d_{\text{in}}}, \tag{108}$$

$$\|\boldsymbol{W}_{\ell,N}^{(0)}\| \;=\; N^{-1/2}\|\boldsymbol{Z}_{\ell,N}\|_{2\to2} \;\le\; 3, \qquad 1\le\ell\le L, \tag{109}$$

$$\|\boldsymbol{W}_{L+1,N}^{(0)}\| \;=\; \frac{\|\boldsymbol{Z}_{L+1,N}\|_{2\to2}}{\sqrt{d_{\text{out}}N}} \;\le\; \frac{3}{\sqrt{d_{\text{out}}}}, \tag{110}$$

with failure probability at most $e^{-N/2}$ for each individual layer. Let $\mathcal{E}_N$ be the event that all bounds in (108)–(110) hold. Independence of the initialization from $\mathcal{H}$ and a union bound over the $L+2$ layers yield

$$\sum_{N=N_0}^{\infty} \mathbb{P}(\mathcal{E}_N^c \mid \mathcal{H}) \;\le\; (L+2)\sum_{N=N_0}^{\infty} e^{-N/2} \;<\; \infty. \tag{111}$$

The Borel-Cantelli lemma (see, e.g., Durrett (2019, Theorem 2.3.1)), applied under the conditional law given $\mathcal{H}$, implies that only finitely many $\mathcal{E}_N^c$ occur, with conditional probability 1. Consequently, for all sufficiently large N,

$$\sum_{\ell=0}^{L+1}\|\boldsymbol{W}_{\ell,N}^{(0)}\| \;\le\; K_0 \;:=\; 3\left(\sqrt{d_{\text{in}}} + L + \frac{1}{\sqrt{d_{\text{out}}}}\right). \tag{112}$$

Set $K_\varphi := \sup_u |\varphi'(u)|$. If all weight norms are bounded at a given update, forward propagation gives

$$\begin{aligned} \|\boldsymbol{h}_0\| &\le \|\boldsymbol{W}_0\|\,\|\boldsymbol{x}\|, && (113)\\ \|\boldsymbol{h}_\ell\| &\le \|\boldsymbol{W}_\ell\|\,\|\varphi(\boldsymbol{h}_{\ell-1})\|, \qquad 1 \le \ell \le L, && (114)\\ \|\varphi(\boldsymbol{h}_\ell)\| &\le |\varphi(0)| + K_\varphi\|\boldsymbol{h}_\ell\|, && (115)\\ \|f_N\| &\le \|\boldsymbol{W}_{L+1}\|\,\|\varphi(\boldsymbol{h}_L)\|. && (116) \end{aligned}$$

On the resulting bounded prediction set, the output loss gradients $\boldsymbol{g}_{L+1}$ for the finitely many labels are bounded. For the rectangular readout, (20) gives $N\|(\boldsymbol{W}_{L+1})^\top\| = d_{\text{out}}\|\boldsymbol{W}_{L+1}\|$, whereas transposition preserves the norm of a square hidden matrix. Hence backpropagation satisfies

$$\begin{aligned} \|\widetilde{\boldsymbol{g}}_L\| &\le K_\varphi d_{\text{out}}\|\boldsymbol{W}_{L+1}\|\,\|\boldsymbol{g}_{L+1}\|, && (117)\\ \|\widetilde{\boldsymbol{g}}_\ell\| &\le K_\varphi\|\boldsymbol{W}_{\ell+1}\|\,\|\widetilde{\boldsymbol{g}}_{\ell+1}\|, \qquad 0 \le \ell < L. && (118) \end{aligned}$$

For $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$, the induced RMS norm satisfies $\|\boldsymbol{u}\boldsymbol{v}^\top\| = n\|\boldsymbol{u}\|\,\|\boldsymbol{v}\|$. Applying this identity to (94)–(96) gives

$$\begin{aligned} \|\delta\boldsymbol{W}_0\| &\le \frac{|\eta| d_{\text{in}}}{B}\sum_{i=1}^{B}\|\widetilde{\boldsymbol{g}}_{0,i}\|\,\|\boldsymbol{x}_i\|, && (119)\\ \|\delta\boldsymbol{W}_\ell\| &\le \frac{|\eta|}{B}\sum_{i=1}^{B}\|\widetilde{\boldsymbol{g}}_{\ell,i}\|\,\|\varphi(\boldsymbol{h}_{\ell-1,i})\|, \qquad 1 \le \ell \le L, && (120)\\ \|\delta\boldsymbol{W}_{L+1}\| &\le \frac{|\eta|}{B}\sum_{i=1}^{B}\|\boldsymbol{g}_{L+1,i}\|\,\|\varphi(\boldsymbol{h}_{L,i})\|. && (121) \end{aligned}$$

All right-hand sides are bounded independently of width. Induction over the finite number of updates therefore bounds all weight norms on both branches by a constant depending only on the history and the fixed model. Convex combinations of the endpoint matrices obey the same bounds, and their derivatives in s are endpoint differences and obey twice those bounds. Differentiating the forward pass gives $\partial_s\boldsymbol{h}_0 = (\partial_s\boldsymbol{W}_0)\boldsymbol{x}$ and, for $1 \le \ell \le L$,

$$\partial_s\boldsymbol{h}_\ell = (\partial_s\boldsymbol{W}_\ell)\varphi(\boldsymbol{h}_{\ell-1}) + \boldsymbol{W}_\ell\big[\varphi'(\boldsymbol{h}_{\ell-1}) \odot \partial_s\boldsymbol{h}_{\ell-1}\big]. \tag{122}$$

Using (114)–(116) and $|\varphi'| \le K_\varphi$ bounds $\|\partial_s\boldsymbol{h}_\ell\|$ at each layer and then $\|\partial_s f_N\|$, uniformly in s. The evaluation loss gradients are bounded on the resulting compact prediction set. Applying the chain rule to (80)–(83) proves the last claim.

Step ②: Convergence in the supremum norm. The convergence in thm. D.3 holds simultaneously at every rational $s \in [0,1]$ (on a countable intersection of almost sure events). By (106), these deterministic rational-point limits share a finite Lipschitz constant. They therefore extend uniquely to Lipschitz curves on $[0,1]$. For a finite rational δ-net $\mathcal{S}_\delta$, the prediction error is bounded by its maximum on that net plus $2K(\mathcal{H})\delta$. First let $N \to \infty$, and then $\delta \to 0$, to obtain uniform convergence. Continuity of the losses on the bounded prediction set gives the uniform limits of C, J, G. Their limiting curves inherit the Lipschitz bounds from **Step ①**. At each fixed s, these curves agree almost surely with the limits in thm. D.3. All arguments hold jointly over the finitely many times and examples.

Step ③: Dependence on the training history. The rational-point limits are measurable in $\mathcal{H}$ by thm. D.3. Since a continuous curve is determined by its values on this countable dense set, the limiting curves are measurable in $\mathcal{H}$. Integrating over $\mathcal{H}$ proves the unconditional statement. □

D.4 NTP Kernel Limit

We will now see that the branching procedure also has a width limit under the neural tangent parameterization (NTP), but the nonlinear interpolation gap G vanishes. We retain the setup of § D.1 and assumption D.2, but use the NTP representative with $a_\ell = 0$ in table 1 and keep $r = 0$ (i.e., a width-independent base LR).

Proposition D.6 (NTP limit of branched interpolation)**.** *Under the assumptions above, additionally suppose $\|\varphi''\|_\infty < \infty$. Let $f^{(t)}_{\infty,i}(s) := (1-s)f^{(t)}_{A,\infty}(\boldsymbol{x}_i) + sf^{(t)}_{B,\infty}(\boldsymbol{x}_i)$, where $f^{(t)}_{A,\infty}, f^{(t)}_{B,\infty}$ are the kernel SGD trajectories with shared Gaussian initialization defined in (136) below, define $J^{(t)}_\infty(\cdot)$ from these endpoints by (80) and (82), and set $C^{(t)}_\infty := -J^{(t)}_\infty$ and $G^{(t)}_\infty := 0$. Then, jointly with $f^{(t)}_{N,i}(\cdot) \xrightarrow{d} f^{(t)}_{\infty,i}(\cdot)$ over the evaluation inputs,*

$$\max_{b\le t\le T}\|G^{(t)}_N(\cdot)\|_\infty \xrightarrow{\mathbb{P}} 0, \qquad (C^{(t)}_N(\cdot), J^{(t)}_N(\cdot), G^{(t)}_N(\cdot))^T_{t=b} \xrightarrow{d} (C^{(t)}_\infty(\cdot), J^{(t)}_\infty(\cdot), G^{(t)}_\infty(\cdot))^T_{t=b}. \tag{123}$$

The curve convergence is in $C([0,1],\mathbb{R}^3)^{T-b+1}$ (and in $C([0,1],\mathbb{R}^{d_{\text{out}}})$ for each prediction curve), with the maximum of the component supremum norms. For convex evaluation loss ℓ, additionally,

$$C^{(t)}_\infty(s) \;=\; -J^{(t)}_\infty(s) \;\le\; 0, \qquad \max_{b\le t\le T}\max_{s\in[0,1]} C^{(t)}_N(s) \xrightarrow{\mathbb{P}} 0. \tag{124}$$

All statements hold conditionally on almost every history $\mathcal{H}$.

Proof. Fix a history. Let $\mathcal{X}_\mathcal{H}$, $\mathcal{Y}_\mathcal{H}$ be the sets of inputs and labels in the training histories and $\mathcal{D}_{\text{eval}}$. All probabilities are conditional on this history.

Step ①: Affine approximation. Use the gauge with a global LR, $\widetilde{\boldsymbol{W}}_0 := \boldsymbol{W}_0$ and $\widetilde{\boldsymbol{W}}_\ell := N^{1/2}\boldsymbol{W}_\ell$ for $1 \le \ell \le L+1$ (i.e., $g_0 = 0$ and $g_\ell = 1/2$ in lem. B.1), and collect their entries in $\boldsymbol{\theta}$. By lem. B.1, these coordinates have i.i.d. initialization from $\mathcal{N}(0,1)$ and LR η under SGD. Write the predictions as $f_{N,\boldsymbol{\theta}}(\boldsymbol{x})$ and let $\nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^{d_{\text{out}}\times\dim(\boldsymbol{\theta})}$ be the output Jacobian.

Let $\boldsymbol{\theta}^{(t)}_{R,N}$ denote the parameters in this gauge from now on. For $\rho \ge 0$, write $B_N(\rho) := \left\{\boldsymbol{\theta} \;\middle|\; \|\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}_N\|_2 \le \rho\right\}$. The uniform Hessian bound of Liu et al. (2020, Theorem 3.2) applies to each output coordinate $[f_{N,\boldsymbol{\theta}}(\boldsymbol{x})]_c$, so a union bound over $\mathcal{X}_\mathcal{H}$ gives for each fixed $\rho > 0$

$$H_N(\rho) \;:=\; \max_{\substack{\boldsymbol{x}\in\mathcal{X}_\mathcal{H}\\ 1\le c\le d_{\text{out}}}} \sup_{\boldsymbol{\theta}\in B_N(\rho)} \left\|\nabla^2_{\boldsymbol{\theta}}[f_{N,\boldsymbol{\theta}}(\boldsymbol{x})]_c\right\|_{2\to2} \xrightarrow{\mathbb{P}} 0, \tag{125}$$

and

$$\begin{aligned}\varepsilon_N(\rho) \;&:=\; \max_{\boldsymbol{x}\in\mathcal{X}_\mathcal{H}} \sup_{\boldsymbol{\theta}\in B_N(\rho)} \left\|\nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x})\right\|_{2\to2}\\ &\le\; \sqrt{d_{\text{out}}}\,\rho H_N(\rho) \xrightarrow{\mathbb{P}} 0.\end{aligned} \tag{126}$$

Further, by the Gaussian process and NTK limits at initialization of Yang (2019; 2020) (see also Jacot et al., 2018; Yang & Littwin, 2021), we have, jointly for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_\mathcal{H}$ and by Slutsky's theorem since $\boldsymbol{K}_\infty$ is deterministic,

$$\begin{aligned}\boldsymbol{K}_N(\boldsymbol{x},\boldsymbol{x}') \;&:=\; \nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x})\big(\nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x}')\big)^\top \xrightarrow{\mathbb{P}} \boldsymbol{K}_\infty(\boldsymbol{x},\boldsymbol{x}'),\\ f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x}) \;&\xrightarrow{d}\; f^{(0)}_\infty(\boldsymbol{x}).\end{aligned} \tag{127}$$

Here $\boldsymbol{K}_\infty$ is deterministic and $f^{(0)}_\infty$ is a centered Gaussian process. We further define

$$\begin{aligned}F_N \;&:=\; \max_{\boldsymbol{x}\in\mathcal{X}_\mathcal{H}} \|f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x})\|_2 = O_\mathbb{P}(1),\\ A_N \;&:=\; \max_{\boldsymbol{x}\in\mathcal{X}_\mathcal{H}} \|\nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}^{(0)}_N}(\boldsymbol{x})\|_{2\to2} = O_\mathbb{P}(1), \qquad M_N(\rho) := A_N + \varepsilon_N(\rho),\\ g(a) \;&:=\; \max_{y\in\mathcal{Y}_\mathcal{H}} \sup_{\|\boldsymbol{z}\|_2\le a} \|\nabla\ell(\boldsymbol{z},y)\|_2 < \infty.\end{aligned} \tag{128}$$

Applying the mean value inequality gives

$$\begin{aligned}\max_{\boldsymbol{x}\in\mathcal{X}_\mathcal{H}} \sup_{\boldsymbol{\theta}\in B_N(\rho)} \|f_{N,\boldsymbol{\theta}}(\boldsymbol{x})\|_2 \;&\le\; F_N + \rho M_N(\rho),\\ D^{(t)}_N \;&:=\; \max_{R\in\{A,B\}} \|\boldsymbol{\theta}^{(t)}_{R,N} - \boldsymbol{\theta}^{(0)}_N\|_2,\\ D^{(t+1)}_N \;&\le\; D^{(t)}_N + \eta M_N(D^{(t)}_N) g\big(F_N + D^{(t)}_N M_N(D^{(t)}_N)\big).\end{aligned} \tag{129}$$

By monotonicity, for $u, \rho > 0$,

$$\mathbb{P}\big(\varepsilon_N(D_N^{(t)}) > u\big) \;\le\; \mathbb{P}(D_N^{(t)} > \rho) + \mathbb{P}\big(\varepsilon_N(\rho) > u\big). \tag{130}$$

Hence $D_N^{(t)} = O_{\mathbb{P}}(1)$ implies $\varepsilon_N(D_N^{(t)}) = o_{\mathbb{P}}(1)$ and, by (129), $D_N^{(t+1)} = O_{\mathbb{P}}(1)$. Starting from $D_N^{(0)} = 0$, induction over fixed T gives $D_N := \max_{0\le t\le T} D_N^{(t)} = O_{\mathbb{P}}(1)$. Define the affine approximation

$$f^{\mathrm{lin}}_{N,\boldsymbol{\theta}}(\boldsymbol{x}) \;:=\; f_{N,\boldsymbol{\theta}_N^{(0)}}(\boldsymbol{x}) + \nabla_{\boldsymbol{\theta}} f_{N,\boldsymbol{\theta}_N^{(0)}}(\boldsymbol{x})(\boldsymbol{\theta} - \boldsymbol{\theta}_N^{(0)}). \tag{131}$$

Then, integration along the segment from $\boldsymbol{\theta}_N^{(0)}$ to $\boldsymbol{\theta}$ yields

$$E_N(\rho) := \max_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{H}}} \sup_{\boldsymbol{\theta}\in B_N(\rho)} \|f_{N,\boldsymbol{\theta}}(\boldsymbol{x}) - f^{\mathrm{lin}}_{N,\boldsymbol{\theta}}(\boldsymbol{x})\|_2 \;\le\; \rho\varepsilon_N(\rho). \tag{132}$$

Both endpoints and their chords lie in $B_N(D_N)$. As f^{lin} commutes with interpolation in weight space,

$$\begin{aligned}\delta_N \;&:=\; \max_{b\le t\le T} \max_{1\le i\le M_{\mathrm{eval}}} \sup_{s\in[0,1]} \big\|f_{N,i}^{(t)}(s) - \overline{f}_{N,i}^{(t)}(s)\big\|_2 \\ &\le\; 2E_N(D_N) \le 2D_N\varepsilon_N(D_N) \;\xrightarrow{\mathbb{P}}\; 0.\end{aligned} \tag{133}$$

Step ②: Joint kernel dynamics. Write $\boldsymbol{v}_{R,N,j}^{(t)} := \nabla\ell(f_{N,\boldsymbol{\theta}_{R,N}^{(t)}}(\boldsymbol{x}_{R,j}^{(t)}), y_{R,j}^{(t)})$. Applying (131) to each SGD update and replacing its Jacobian by the initial one gives

$$f_{N,\boldsymbol{\theta}_{R,N}^{(t+1)}}(\boldsymbol{x}) \;=\; f_{N,\boldsymbol{\theta}_{R,N}^{(t)}}(\boldsymbol{x}) - \frac{\eta}{B}\sum_{j=1}^{B} \boldsymbol{K}_N(\boldsymbol{x}, \boldsymbol{x}_{R,j}^{(t)})\boldsymbol{v}_{R,N,j}^{(t)} + \boldsymbol{r}_{R,N}^{(t)}(\boldsymbol{x}), \tag{134}$$

where, uniformly over $R, t < T, \boldsymbol{x} \in \mathcal{X}_{\mathcal{H}}$,

$$\|\boldsymbol{r}_{R,N}^{(t)}(\boldsymbol{x})\|_2 \;\le\; 2E_N(D_N) + \eta A_N\varepsilon_N(D_N)g\big(F_N + D_N M_N(D_N)\big) \;=\; o_{\mathbb{P}}(1). \tag{135}$$

Define the limiting trajectories for $R \in \{A, B\}$ by

$$\begin{aligned} f_{R,\infty}^{(0)}(\boldsymbol{x}) \;&:=\; f_{\infty}^{(0)}(\boldsymbol{x}), \\ f_{R,\infty}^{(t+1)}(\boldsymbol{x}) \;&:=\; f_{R,\infty}^{(t)}(\boldsymbol{x}) - \frac{\eta}{B}\sum_{j=1}^{B} \boldsymbol{K}_{\infty}(\boldsymbol{x}, \boldsymbol{x}_{R,j}^{(t)})\nabla\ell\big(f_{R,\infty}^{(t)}(\boldsymbol{x}_{R,j}^{(t)}), y_{R,j}^{(t)}\big). \end{aligned} \tag{136}$$

Continuity of the loss gradients, (127), (135), and induction in (134) yield

$$\big(f_{N,\boldsymbol{\theta}_{R,N}^{(t)}}(\boldsymbol{x})\big)_{R,t,\boldsymbol{x}} \;\xrightarrow{d}\; \big(f_{R,\infty}^{(t)}(\boldsymbol{x})\big)_{R,t,\boldsymbol{x}}. \tag{137}$$

Together with (133), this gives joint convergence of the prediction curves in the supremum norm to

$$f_{\infty,i}^{(t)}(s) \;=\; (1-s)f_{A,\infty}^{(t)}(\boldsymbol{x}_i) + sf_{B,\infty}^{(t)}(\boldsymbol{x}_i), \qquad s \in [0,1]. \tag{138}$$

Step ③: Loss curves. All chord predictions and output ensembles have norm $\le P_N := F_N + D_N M_N(D_N) = O_{\mathbb{P}}(1)$, and thus,

$$\max_{b\le t\le T} \|G_N^{(t)}(\cdot)\|_{\infty} \;\le\; g(P_N)\delta_N \;\xrightarrow{\mathbb{P}}\; 0. \tag{139}$$

Recall that $J_{\infty}^{(t)}(\cdot)$ is defined from the endpoints in (138) using (82). On any prediction ball of radius a, changing every endpoint by $\le u$ changes its Jensen-gap curve by $\le 2g(a)u$, so the map from endpoints to Jensen-gap curves is continuous. Thus, by the continuous mapping theorem and (137),

$$(J_N^{(t)}(\cdot))_{t=b}^{T} \;\xrightarrow{d}\; (J_{\infty}^{(t)}(\cdot))_{t=b}^{T}, \qquad (C_N^{(t)}(\cdot), J_N^{(t)}(\cdot), G_N^{(t)}(\cdot))_{t=b}^{T} \;\xrightarrow{d}\; (-J_{\infty}^{(t)}(\cdot), J_{\infty}^{(t)}(\cdot), 0)_{t=b}^{T}, \tag{140}$$

where the r.h.s. uses Slutsky's theorem. For convex ℓ, $J_N^{(t)}(s) \ge 0$ and $C_N^{(t)}(0) = C_N^{(t)}(1) = 0$, so

$$0 \;\le\; \max_{b\le t\le T} \max_{s\in[0,1]} C_N^{(t)}(s) \;\le\; \max_{b\le t\le T} \|G_N^{(t)}(\cdot)\|_{\infty} \;\xrightarrow{\mathbb{P}}\; 0, \qquad C_{\infty}^{(t)}(\cdot) = -J_{\infty}^{(t)}(\cdot) \le 0, \tag{141}$$

which completes the proof. □

D.5 Finite-Width Agreement

The existence of the limits from § D.3 and D.4 does not make any statement about agreement at practical widths. As mentioned in § 4.2, to assess effective agreement already at finite widths, the normalization of the chord deviation C w.r.t. the endpoint losses is already a design choice itself, and instead of additive, one could also define a loss ratio[3]

$$\overline{\mathcal{L}}_N(s) := (1-s)\mathcal{L}_N(0) + s\mathcal{L}_N(1), \qquad \tilde{C}_N(s) := \frac{\mathcal{L}_N(s)}{\overline{\mathcal{L}}_N(s)}, \qquad \overline{\mathcal{L}}_N(s) > 0. \tag{142}$$

Effectively, the additive quantity $C_N(s) = \mathcal{L}_N(s) - \overline{\mathcal{L}}_N(s)$ removes a common loss *offset* and $\tilde{C}_N(s)$ removes a common loss *factor*. Write $\mathcal{L}_\infty$ and $\overline{\mathcal{L}}_\infty$ for the limits of $\mathcal{L}_N(\cdot)$ and $\overline{\mathcal{L}}_N(\cdot)$, which exist by thm. D.5 and prop. D.6, and note that $\inf_s \overline{\mathcal{L}}_\infty(s) = \min\{\mathcal{L}_\infty(0), \mathcal{L}_\infty(1)\}$. If this is positive (a.s. in the NTP case), we get the same modes of convergence as in § D.3 and D.4 for $\tilde{C}_N(\cdot)$, with limits $\mathcal{L}_\infty/\overline{\mathcal{L}}_\infty$ under µP and $1 - J_\infty/\overline{\mathcal{L}}_\infty$ under NTP.

L^2 loss. Under ℓ_2, the additive normalization would be natural for the Jensen gap J. Namely, one can compute

$$J_N(s) = \frac{s(1-s)}{2M_{\text{eval}}} \sum_{i=1}^{M_{\text{eval}}} \left\| f_{N,i}(1) - f_{N,i}(0) \right\|_2^2. \tag{143}$$

Thus, if prediction disagreement stays fixed across widths, so does the absolute ensembling gain $J_N(\cdot)$, even if a shared prediction error changes the endpoint losses.

Cross-entropy loss. For ℓ_{CE}, a common increase in correct-class margins can instead multiply all losses by the same factor. Consider a binary classification example, with margin m (i.e. difference between correct and incorrect logit). Its loss is $\ell_{\text{CE}}(m) := \log(1 + e^{-m}) \sim e^{-m}$ as $m \to \infty$. Suppose the chord has margins $m_N(s) := a_N + u(s)$, where $a_N \to \infty$ and $u : [0,1] \to \mathbb{R}$ is continuous and does not depend on N. Logit averaging has margin $a_N + (1-s)u(0) + su(1)$, so

$$\begin{aligned} \mathcal{L}_N(s) &\sim e^{-a_N} e^{-u(s)}, \\ \mathcal{L}_{\text{ens},N}(s) &\sim e^{-a_N} e^{-(1-s)u(0) - su(1)}, \\ \tilde{C}_N(s) &\to \frac{e^{-u(s)}}{(1-s)e^{-u(0)} + se^{-u(1)}}. \end{aligned} \tag{144}$$

In this case, $J_N(\cdot) \to 0$, while

$$\tilde{J}_N := \frac{J_N}{\overline{\mathcal{L}}_N} \to 1 - \frac{e^{-(1-s)u(0)-su(1)}}{\left((1-s)e^{-u(0)} + se^{-u(1)}\right)} \geq 0. \tag{145}$$

The two examples above show that, while a certain normalization of the chord deviation across scales can be motivated in some settings, these depend on rather specific assumptions (similar prediction dispersion for ℓ_2 and similar differences up to a shifting margin for ℓ_{CE}), which may not hold in practice. For simplicity, we work with the additive chord deviation $C_N(\cdot)$, for which we qualitatively observe scale consistency over a range of architectures and HPs. We leave a systematic comparison of such normalization choices at realistic large-scale models for future work.

[3] We suppress the time index t in this subsection.

E EXPERIMENTAL SETUP

In this section, we provide further details on the setup for the experiments regarding LMC (§ E.1), variance collapse (§ E.2), and model merging (§ E.3).

E.1 LINEAR MODE CONNECTIVITY

For each LMC experiment, we train for a total horizon of T steps/epochs. The general setup is that a model $\boldsymbol{\theta}^{(t)}$ is trained until the *branching time* $b \in \{0, \dots, T\}$, the parameters and optimizer state are copied into two replicates $\boldsymbol{\theta}_A^{(t)}$ and $\boldsymbol{\theta}_B^{(t)}$. Then, the two branches continue training with independent minibatches and (for CIFAR-10) independent augmentations until T. We evaluate the chord deviation $C^{(T)}$ and decomposition into $J^{(T)}$ and $G^{(T)}$ from def. C.1, as well as the quadratic approximation in prop. C.3. Unless stated otherwise, curves always show means $\pm$ sample standard deviations over all seeds we evaluated.

E.1.1 CONVNET ON CIFAR-10

We follow the setup of Noci et al. (2024, Figure 1). Namely, the architecture consists of 2 bias-free 3×3 convolutions of width N with strides $(2, 4)$ and padding $(1, 1)$, respectively, each followed by ReLU and 2×2 max pooling, and a bias-free linear readout. In total, this gives $27N + 9N^2 + 10N = 9N^2 + 37N$ parameters. We vary $N \in \{64, 128, 256, 512, 1024, 2048\}$. Initialization, forward multipliers, and learning-rate scalings for μP, neural tangent parameterization (NTP), and standard parameterization (SP) are reported in table 4. We train on all 50k CIFAR-10 training examples for $T = 50$ epochs using batch size 128, cross-entropy loss, and constant-rate SGD without any momentum or weight decay. We augment using random crops with padding 4, horizontal flips, and rotations of up to 10 degrees, followed by channelwise normalization with mean and std. 0.5. Evaluation uses all 10k test examples without augmentation. For μP, we use 9 logarithmically spaced base rates $\eta = 10^{-1/2+2k/9}$, $k \in \{-2, \dots, 6\}$, roughly 0.114 to 6.81.

Table 4: ConvNet parameterizations in stored coordinates per layer.

	Init. std.	**Forward mult.**	**LR**
μP	$(1, 1, 1)$	$(27^{-1/2}, (9N)^{-1/2}, N^{-1})$	ηN
NTP	$(1, 1, 1)$	$(27^{-1/2}, (9N)^{-1/2}, N^{-1/2})$	η
SP	$(27^{-1/2}, (9N)^{-1/2}, N^{-1/2})$	$(1, 1, 1)$	η

For fig. 27, we branch at $b \in \{0, 10\}$ epochs and report mean $\pm$ std. over 10 pairs of $(\boldsymbol{\theta}_A^{(t)}, \boldsymbol{\theta}_B^{(t)})$. For the additional results in § F.1.1, we branch at $b \in \{0, 5, 10\}$ over 3 seeds. For the alignment from independent initializations in fig. 27, we perform activation matching (Ainsworth et al., 2023). Specifically, we match post-pooling channels by maximizing Pearson activation correlation on all 50k training examples using the Hungarian algorithm, and then interpolate linearly between the weights aligned with the found permutations. We evaluate 5 endpoint pairs each.

E.1.2 VIT ON IMAGENET

As a proof of concept that the main claims of § 5 extend to vision, we adapt the ViT setup of Bordelon et al. (2024b, Appendix A.3–A.4) to ImageNet training with cosine LR decay for one epoch under depth-µP. We use a pre-LayerNorm ViT (Dosovitskiy et al., 2021) with 8×8 patches, a class token, learned positional embeddings, 8 attention heads, GELU MLPs of width $4N$, and no dropout. We train on ImageNet-1k (Deng et al., 2009; Russakovsky et al., 2015) at 64×64 resolution without replacement ($T = 20{,}019$ updates at batch size 64), and use the standard augmentation choices (bilinear random resized crops (area ([0.08,1]), aspect ratio ([3/4,4/3])), horizontal flips, and channel normalization). We use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\varepsilon = 10^{-8}$ without any weight decay or gradient clipping, and perform a 1,000-step linear warmup followed by cosine decay to 10%. We sweep $N \in \{128, 256, 512, 1024, 2048\}$ at $L = 12$ and $L \in \{3, 6, 12, 24, 48\}$ at $N = 512$, with $\eta \in \{2^{-13}, \dots, 2^{-7}\}$ and 3 seeds. At $b \in \{1\text{k}, 2\text{k}, 10\text{k}\}$, each fork copies the shared

reference parameters and Adam state, independently permutes the remaining training images, and uses independent augmentations.

E.1.3 NanoGPT on OpenWebText

We use NanoGPT in the completeP (Dey et al., 2025) parameterization for width/depth scaling[4], with reference width $N_0 = 256$ and depth $L_0 = 2$. We retain the GPT-2 tokenizer and context length 1,024, set dropout to zero, and keep the attention head dimension fixed at 64 as width varies. Write $m_N := N/N_0$ and $m_L := L/L_0$. We set $\sigma_{\text{init}} = 0.02$. The readout has a forward multiplier m_N^{-1} and each residual branch is multiplied by m_L^{-1} (completeP) or $m_L^{-1/2}$ (depth-μP). We collect the initialization and optimizer scalings in table 5.

Table 5: Init. and optimizer scalings for the NanoGPT experiments. Normalization layers are initialized with one.

Parameterization	Parameter group	Init. std.	LR	Adam ε	Weight decay
completeP (Dey et al., 2025)	Hidden matrices	$\sigma_{\text{init}}/\sqrt{m_N}$	η/m_N	$\varepsilon/(m_N m_L)$	λm_N
	Hidden normalization	–	η	$\varepsilon/(m_N m_L)$	0
	Embeddings/readout	σ_{init}	η	ε/m_N	λ
	Final normalization	–	η	ε/m_N	0
Depth-μP (Yang et al., 2024)	Hidden matrices	$\sigma_{\text{init}}/\sqrt{m_N}$	$\eta/(m_N\sqrt{m_L})$	$\varepsilon/(m_N\sqrt{m_L})$	λm_N
	Hidden normalization	–	$\eta/\sqrt{m_L}$	$\varepsilon/(m_N\sqrt{m_L})$	0
	Embeddings/readout	σ_{init}	η	ε/m_N	λ
	Final normalization	–	η	ε/m_N	0

We train on OpenWebText[5]. At each update, we draw contiguous 1,024-token chunks with replacement from the tokenized training stream, until the respective token budget is reached. We reserve 0.05% of documents and split them equally into validation and test sets.

All NanoGPT experiments use AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.95)$, gradient clipping at 1, and BF16 computation with FP32 parameters and optimizer state. At every size, we use a batch size of 32 (i.e., 32,768 tokens per update). Training budgets and schedules for all pretraining sweeps are listed in table 6, and parameter counts in table 7. Cosine schedules decay to 10% of the peak LR and continue across branching without restarting.

Table 6: NanoGPT pretraining setups for *LMC* (100M and 300M tokens) and *model merging* (3B tokens). Unless stated otherwise, all sweeps use completeP.

Sweep	Fixed size	Varied sizes	η	Updates	Warmup	Schedule
300M (width)	$L = 8$	$N \in \{2^7, \ldots, 2^{11}\}$	$\{2^{-12}, \ldots, 2^{-4}\}$	9,156	200	Cosine
300M (depth)	$N = 512$	$L \in \{2^2, \ldots, 2^6\}$	$\{2^{-12}, \ldots, 2^{-4}\}$	9,156	200	Cosine
100M (*depth-μP*)	$N = 256$	$L \in \{2^2, \ldots, 2^5\}$	$\{2^{-12}, \ldots, 2^{-6}\}$	3,052	0	Constant
100M (depth)	$N = 256$	$L \in \{2^2, \ldots, 2^5\}$	$\{2^{-12}, \ldots, 2^{-6}\}$	3,052	0	Constant
3B (width)	$L = 8$	$N \in \{2^7, \ldots, 2^{11}\}$	2^{-9}	91,553	1,000	Cosine
3B (depth)	$N = 512$	$L \in \{2^2, \ldots, 2^6\}$	2^{-9}	91,553	1,000	Cosine

The 300M experiments branch at $b \in \{200, 1\text{k}, 5\text{k}\}$ updates and share a reference trajectory across branch times. The 100M ablations use $b \in \{0, 100, 500, 1\text{k}\}$. We evaluate each branch on 3 seeds, and pair initialization and minibatch streams across sizes and learning rates. We measure interpolation behavior every 200 updates for the 300M runs and every 100 for the 100M runs. For LR selection and to report the losses and $C/J/G$ trajectories, we evaluate on ∼131k fixed validation/test tokens, respectively. Furthermore, we use the full test split (∼2.23M tokens) to report the final interpolation curves for the 300M runs. Throughout, we use token-averaged cross-entropy loss. For the 3B token pretraining experiment for model merging, we choose $\eta = 2^{-9}$.

[4] We adapt the setup of https://github.com/EleutherAI/nanoGPT-mup/tree/completep.
[5] https://huggingface.co/datasets/Skylion007/openwebtext

Table 7: Param. counts (in millions) for the architectures in table 6.

$N \backslash L$	4	8	16	32	64
128	—	8.1M	—	—	—
256	16.3M	19.4M	25.7M	38.3M	—
512	38.9M	51.5M	76.6M	127.0M	227.7M
1024	—	153.2M	—	—	—
2048	—	507.8M	—	—	—

We evaluate the Hessian and GN term at $\overline{\boldsymbol{\theta}} := (\boldsymbol{\theta}_A + \boldsymbol{\theta}_B)/2$ along the direction $\boldsymbol{\Delta} := \boldsymbol{\theta}_B - \boldsymbol{\theta}_A$. We further set $s_L := m_L^{2\alpha-1}$ ($\alpha = 1$ for completeP and $\alpha = 1/2$ for depth-µP). We use the fixed diagonal matrix $\boldsymbol{S} := \mathrm{diag}\big(s_L \boldsymbol{I}_h,\, m_N s_L \boldsymbol{I}_n,\, m_N \boldsymbol{I}_e\big)$, where the identity blocks correspond to hidden matrices (h), normalization parameters (n), and embeddings/readout (e).

E.2 Variance Collapse

At each block output, we compare the midpoint activation variance with the squared average endpoint standard deviation. Writing $\sigma_{A,\ell c}$, $\sigma_{B,\ell c}$, and $\sigma_{\mathrm{mid},\ell c}$ for channel standard deviations, the ratio in fig. 7 is $\sum_c \sigma^2_{\mathrm{mid},\ell c} / \sum_c [(\sigma_{A,\ell c} + \sigma_{B,\ell c})/2]^2$.

NanoGPT on OpenWebText (fig. 7). We use final endpoint pairs from the 300M-token completeP depth setup in § E.1.3. REPAIR is calibrated sequentially at the QKV, attention-output, both MLP linear outputs, and block outputs, with earlier corrections active. We use calibration, validation, and test sets of 131,072 tokens each, where the validation set is used to plot the variance profiles over depth and the test set to report raw and repaired midpoint chord deviations.

ResNet on CIFAR-10 (§ F.2). We adapt the training pipeline of Lim et al. (2024) for postactivation ResNets (He et al., 2016) using BatchNorm (Ioffe & Szegedy, 2015) with 3 stages of widths $(16, 32, 64)$, two 3×3 convolutions per block, projection shortcuts at stage transitions, and global average pooling. Each stage contains $K \in \{2, 8, 16, 32\}$ blocks, which gives nominal depths $6K + 2 \in \{14, 50, 98, 194\}$. We compare unscaled residual branches with completeP depth scaling (Dey et al., 2025) adapted to this architecture (i.e., residual multiplier K^{-1}, depth-independent initialization and Adam LR, residual Adam $\varepsilon = 10^{-8}/K$). We train for 10 epochs with batch size 128 and use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, zero weight decay, and $\eta \in \{0.003, 0.01\}$, with a 100-step warmup and cosine decay to 5%. We use 2,048 unaugmented training images to calibrate REPAIR, and evaluate accuracy on all 10k test images.

E.3 Model Merging

We use the 8-task subset identified by Zhou et al. (2023a) as important for zero-shot generalization in the T0 training mixture (Sanh et al., 2022). The subset covers question answering and paraphrase identification and was subsequently adopted for model merging by Tam et al. (2024); Li et al. (2025). See table 8 for an overview of the tasks. We work with the 3B-token pretrained models (from table 6), using the pretraining setup described in § E.1.3. This means that, in total, we evaluate merging on 9 unique models (5 widths/depths, with $N = 512, L = 8$ shared between both).

E.3.1 Fine-Tuning Setup and Tasks

From each pretrained checkpoint, we fine-tune one full-model expert per task, resetting the optimizer at the start of fine-tuning. We train for 3,072 updates, with 32 examples per update, using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and no weight decay. We use no warmup or LR schedule and sweep the constant base LR $\eta \in \{2^{-20}, \ldots, 2^{-8}\}$ under the same completeP scaling as during pretraining. We fix 3 fine-tuning seeds for determining data order for training the task expert, and couple these across models and LRs.

Following Tam et al. (2024); Li et al. (2025), we use natural-language templates from PromptSource (Bach et al., 2022), which turn each dataset example into a prompt and target answer. In our experiments, each example uses the same template across fine-tuning seeds. We remove duplicates and

Table 8: Tasks used for fine-tuning in the model merging experiment.

Task	Training exs.	Eval. exs.	Epochs	Score
CosmosQA (Huang et al., 2019)	24,241	2,985	4.06	4-choice accuracy
SocialIQA (Sap et al., 2019)	32,386	1,935	3.04	3-choice accuracy
PAWS (Zhang et al., 2019)	48,364	7,994	2.03	Binary accuracy
QuAIL (Rogers et al., 2020)	9,247	2,164	10.63	4-choice accuracy
WikiQA (Yang et al., 2015)	19,358	2,732	5.08	Balanced binary accuracy
QuaRTz (Tafjord et al., 2019)	2,157	384	45.57	2-choice accuracy
QASC (Khot et al., 2020)	7,134	926	13.78	8-choice accuracy
ROPES (Lin et al., 2019)	9,924	1,688	9.91	Exact match

overlaps between training and evaluation. For classification tasks, we score each possible answer by its conditional log-probability and select the highest-scoring answer. For ROPES, the model generates an answer using greedy decoding with a limit of 64 tokens, and we check for an exact match (after lowercasing, removing punctuation, articles, and excess whitespace). As WikiQA is label-imbalanced, we report balanced accuracy, i.e., the average recall for its two classes.

E.3.2 Merging Methods

Write $\boldsymbol{\tau}_i := \boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{base}}$ for the update of expert i from the pretrained parameters. Each merging method returns $\boldsymbol{\theta}_0 + \alpha \boldsymbol{\tau}_{\text{merged}}$, where α is the strength of the update in the merging direction $\boldsymbol{\tau}_{\text{merged}}$. Task arithmetic (TA) (Ilharco et al., 2023) simply uses $\boldsymbol{\tau}_{\text{merged}} := \frac{1}{8} \sum_{i=1}^{8} \boldsymbol{\tau}_i$. TIES (Yadav et al., 2023) retains each expert's largest-magnitude fraction ρ of updates, elects coordinatewise signs, and averages the retained updates agreeing with those signs. We use Squisher (Li et al., 2025) as a cheaper proxy for Fisher merging (Matena & Raffel, 2022), which estimates the Fisher diagonal via the squared-gradient accumulator of the Adam optimizer state.

Table 9: Model merging HP grid per method.

Merging method	Settings
TA (Ilharco et al., 2023)	$\alpha \in \{0.1, 0.2, 0.5, 1, 1.5, 2\}$
Squisher (Li et al., 2025)	$\alpha \in \{0.1, 0.2, 0.5, 1, 1.5, 2\}$
TIES (Yadav et al., 2023)	$\alpha \in \{0.25, 0.5, 1, 2\}, \rho \in \{0.1, 0.2, 0.5, 1\}$

F Additional Experimental Results

In this section, we collect additional results for the experiments on LMC (§ F.1), variance collapse (§ F.2), and model merging (§ F.3).

F.1 Linear Mode Connectivity

F.1.1 ConvNet on CIFAR-10

Here, we plot further diagnostics for the ConvNet experiment on CIFAR-10 from fig. 1 and § E.1.1.

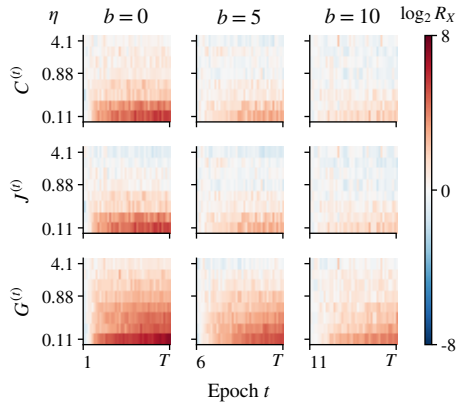


Figure 9: **Discrepancy growth** for $X \in \{J, G\}$ relative to 5 steps after branching time b (cf. Finding 2b in § 5.1).

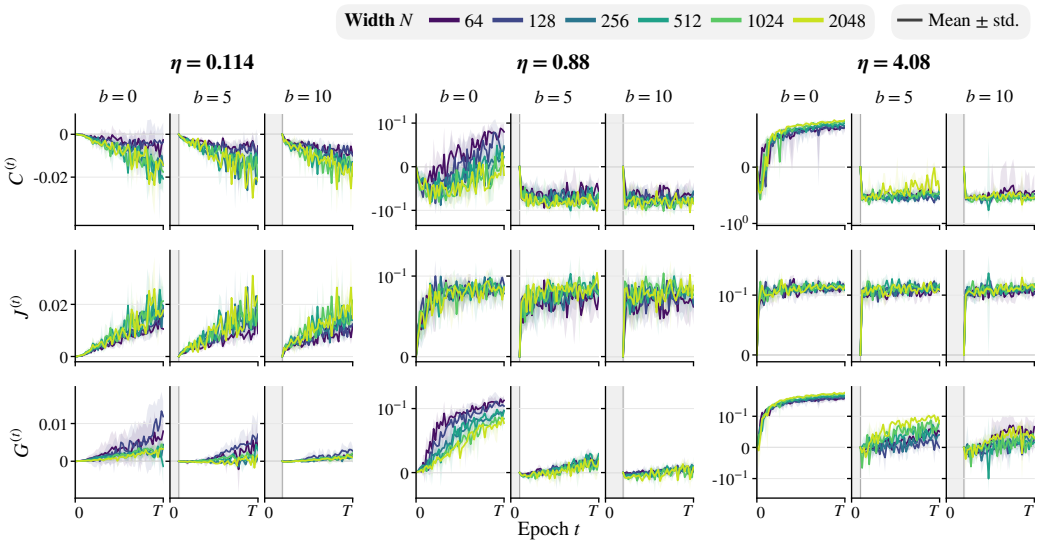


Figure 10: **Trajectories of chord deviations** and their decomposition (4) for 3 LR representatives.

F.1.2 NANOGPT ON OPENWEBTEXT

We collect additional diagnostics for the main experiment of § 5.1.

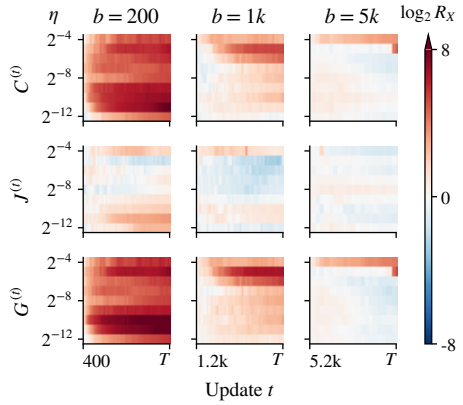


Figure 11: **µP width (300M tokens)** discrepancy growth relative to $t = b + 200$.

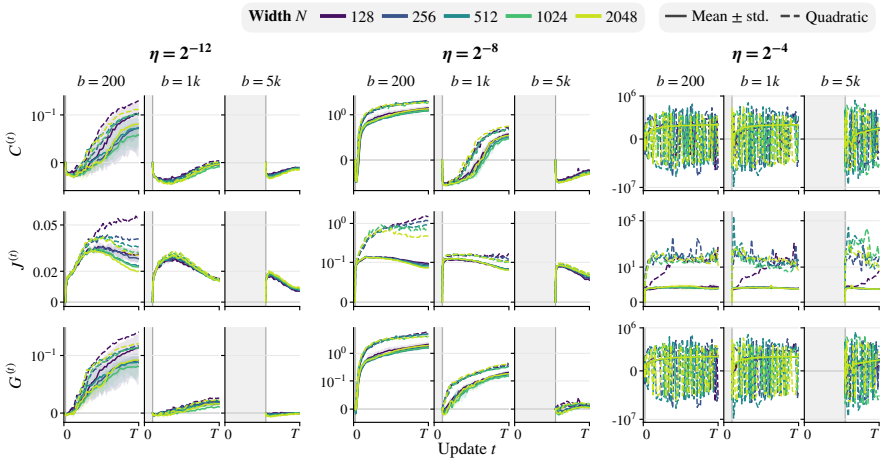


Figure 12: **µP width (300M tokens)** trajectories of chord deviations and quadratic approximations.

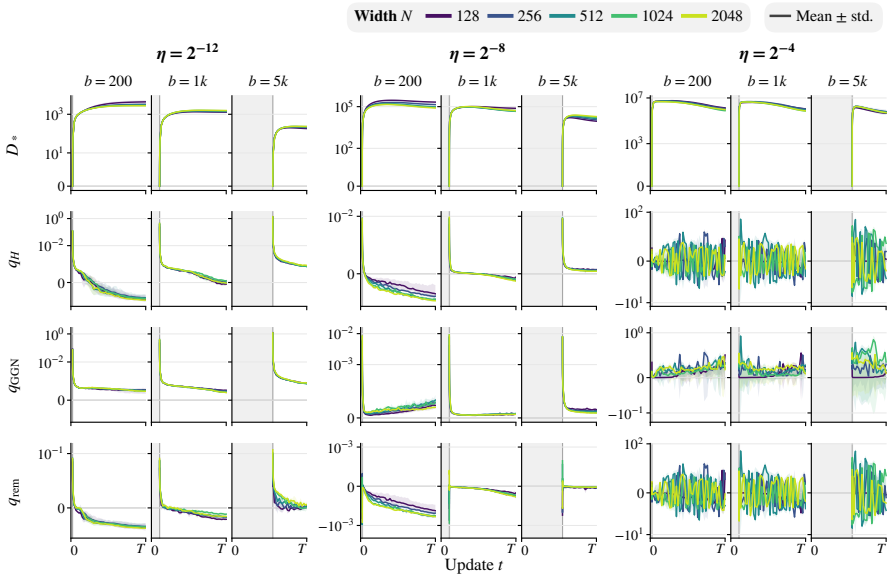


Figure 13: **µP width (300M tokens)** directional curvatures and $D_* := \|\boldsymbol{\Delta}\|_{\boldsymbol{S}^{-1}}$.

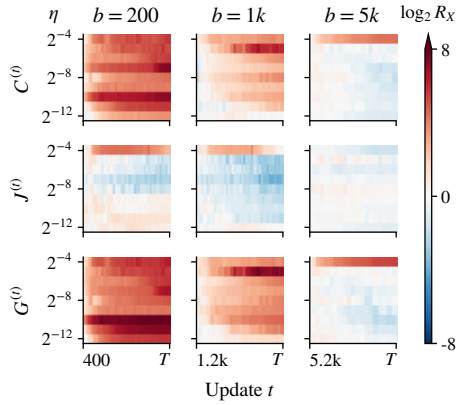


Figure 14: **completeP depth (300M tokens)** discrepancy growth relative to $t = b + 200$.

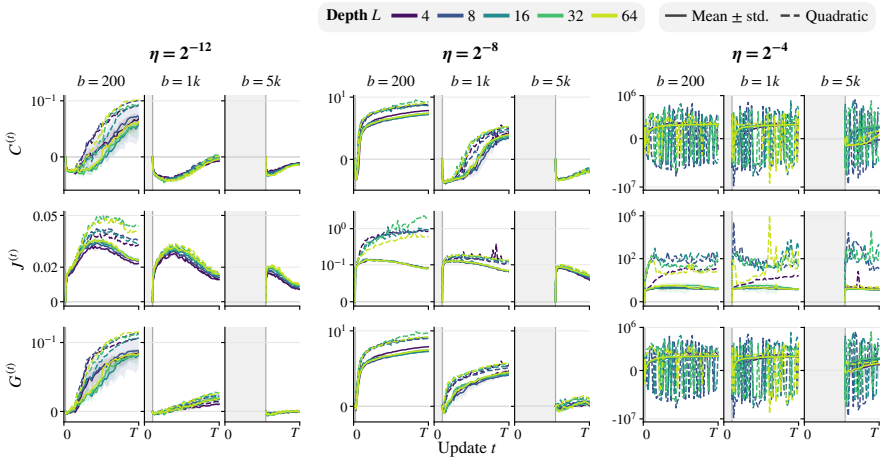


Figure 15: **completeP depth (300M tokens)** trajectories of chord deviations and quadratic approximations.

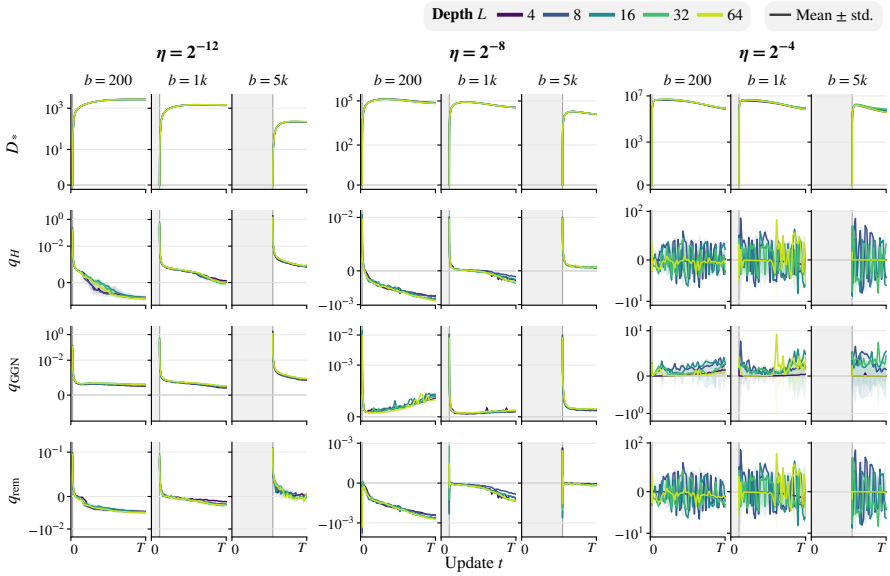


Figure 16: **completeP depth (300M tokens)** directional curvatures and $D_* := \|\mathbf{\Delta}\|_{S^{-1}}$.

F.1.3 Effect of Depth Scaling and LR Schedule

Depth-μP (weight decay 0.1, 100M tokens, constant base LR).

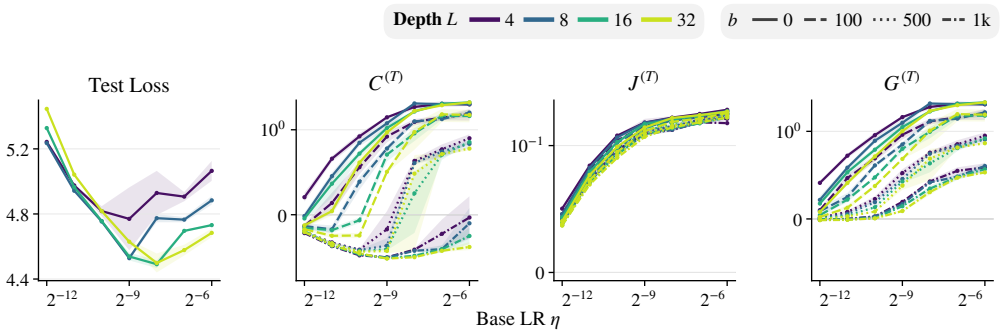


Figure 17: **Depth transfer under depth-μP.** The columns show test loss and test $C^{(T)}, J^{(T)}, G^{(T)}$ after 100M OpenWebText tokens, curves report mean ± std. over 3 seeds.

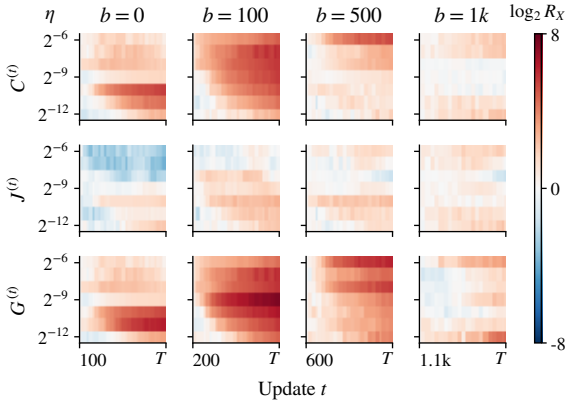


Figure 18: **Discrepancy growth** for $X \in \{J, G\}$ relative to 100 steps after branching time b (cf. Finding 2b in § 5.1).

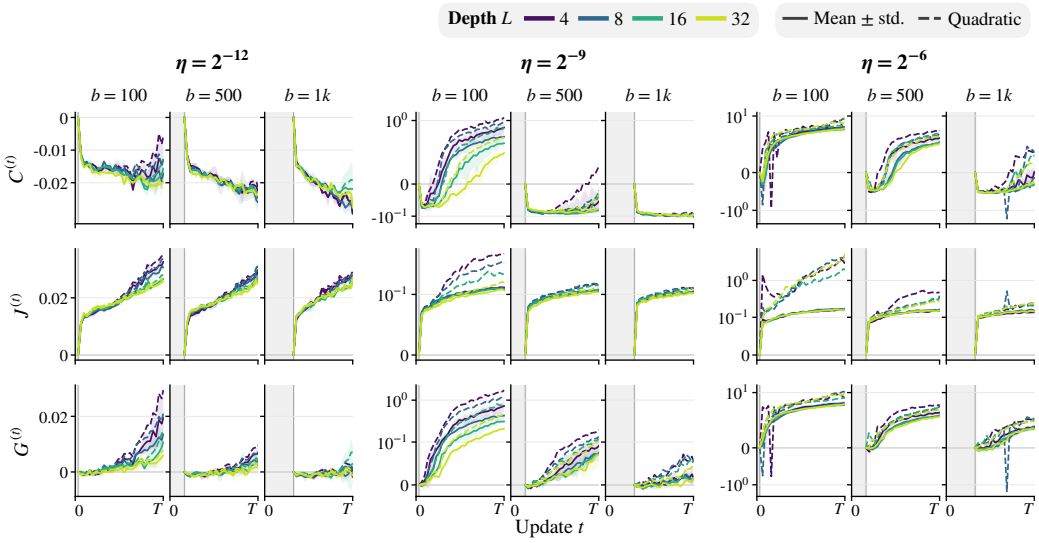


Figure 19: **Trajectories of chord deviations** and their decomposition (4) for 3 LR representatives.

completeP (weight decay 0.1, 100M tokens, constant base LR).

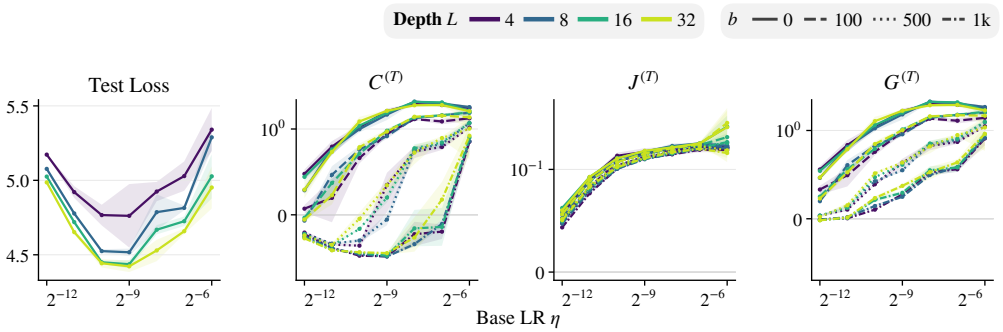


Figure 20: **Depth transfer under completeP.** The columns show test loss and test $C^{(T)}$, $J^{(T)}$, $G^{(T)}$ after 100M OpenWebText tokens, curves report mean $\pm$ std. over 3 seeds.

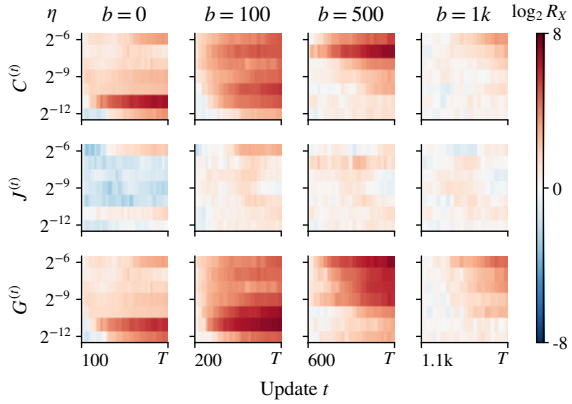


Figure 21: **Discrepancy growth** for $X \in \{J, G\}$ relative to 100 steps after branching time b (cf. Finding 2b in § 5.1).

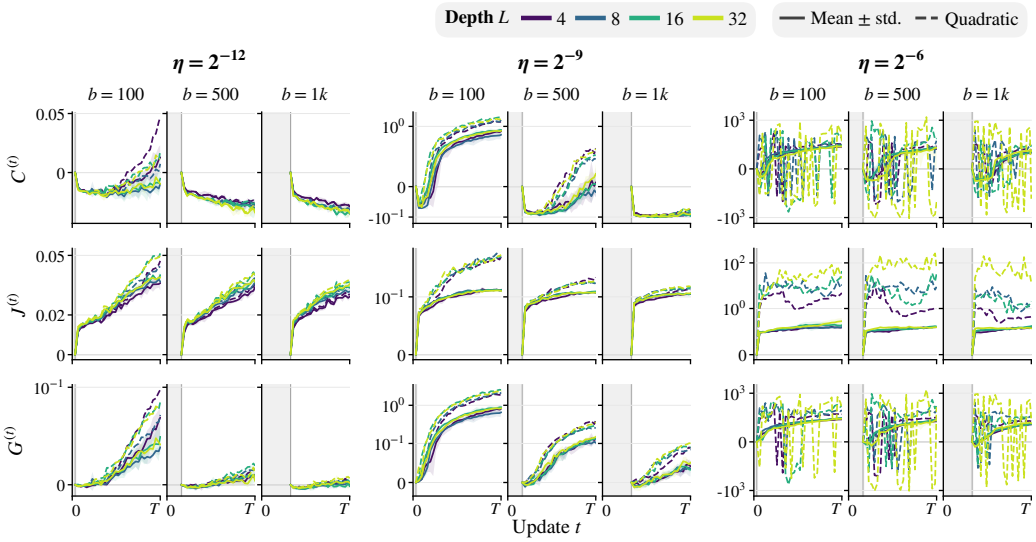


Figure 22: **Trajectories of chord deviations** and their decomposition (4) for 3 LR representatives.

F.1.4 ViT on ImageNet

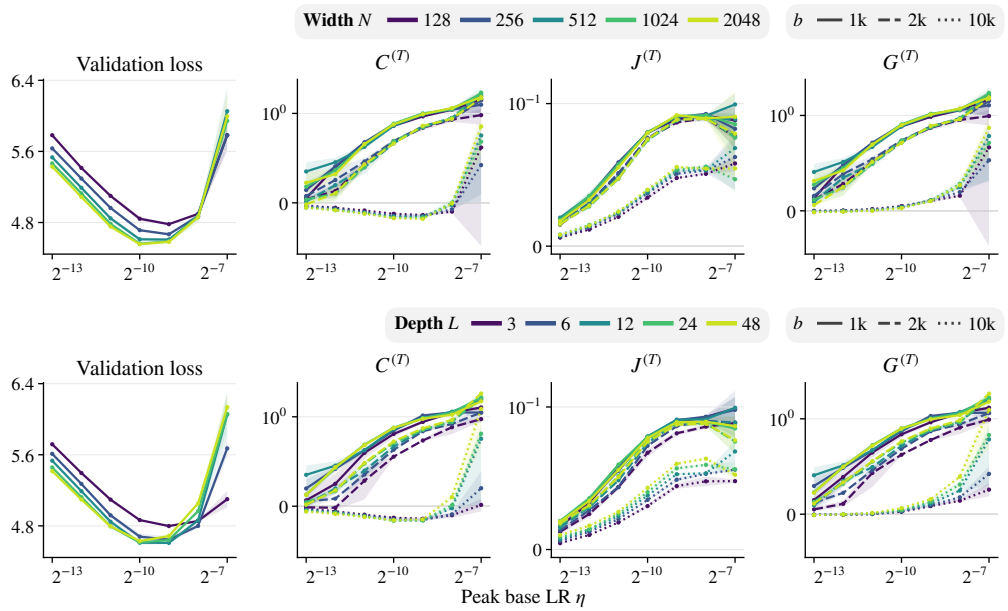


Figure 23: **Width and depth transfer in vision transformers under depth-µP.** *Top:* width scaling at $L = 12$. *Bottom:* depth scaling at $N = 512$. The columns show validation loss and test $C^{(T)}, J^{(T)}, G^{(T)}$ after one epoch on ImageNet, curves report mean $\pm$ std. over 3 seeds.

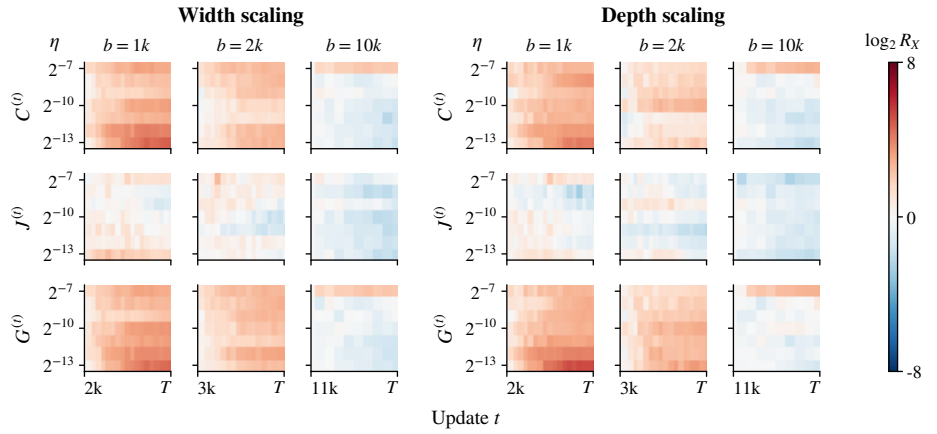


Figure 24: **Discrepancy growth** for $X \in \{J, G\}$ relative to 1k steps after branching time b (cf. Finding 2b in § 5.1).

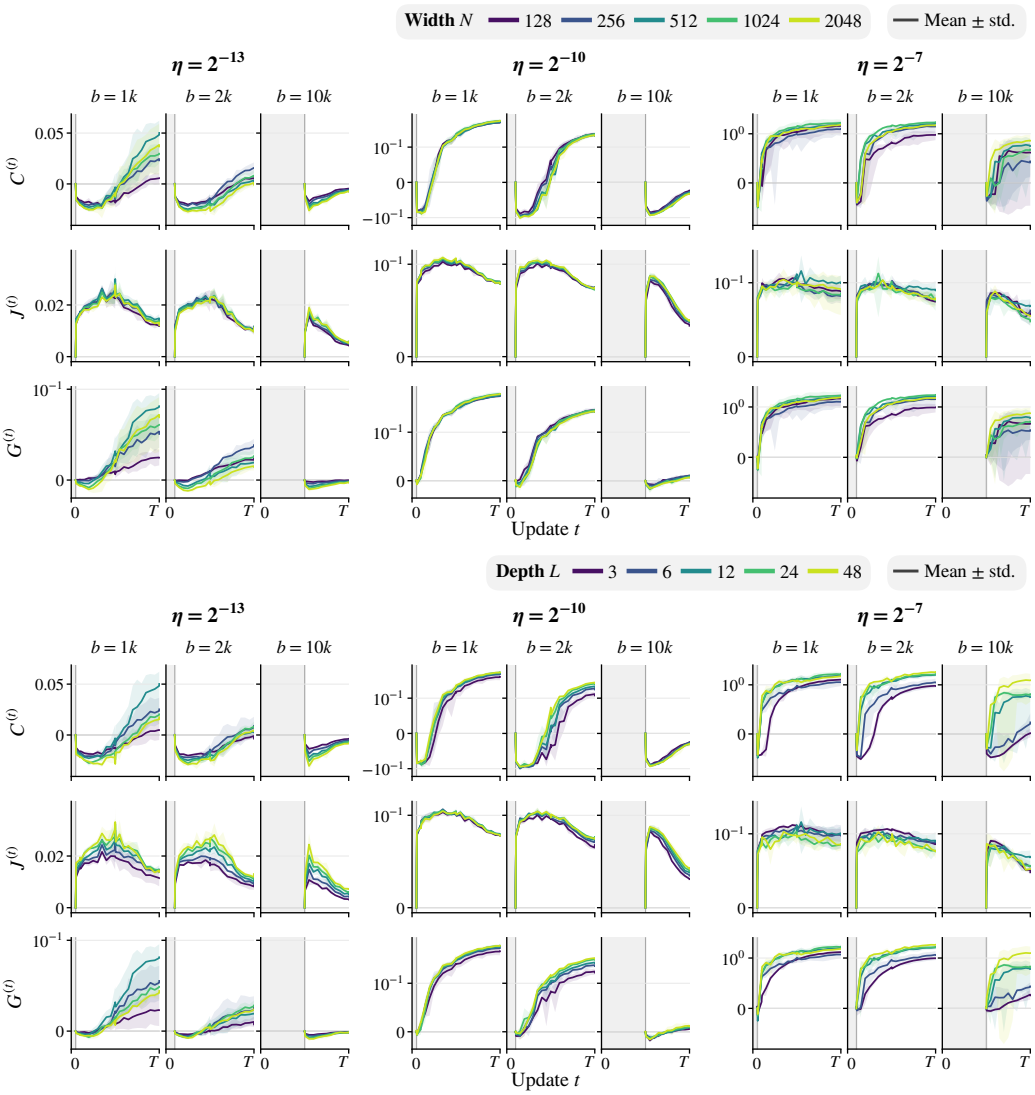


Figure 25: **Trajectories of chord deviations** and their decomposition (4) for 3 LR representatives.

F.2 Variance Collapse

We also test variance collapse and the effectiveness of REPAIR (Jordan et al., 2023) for a ResNet under BatchNorm trained on CIFAR-10. Indeed, fig. 26 confirms the finding in § 5.2: Under completeP depth scaling, the amount of variance collapse is roughly stable across depths, while it degrades substantially under the (unscaled) standard parameterization (*right*). This also shows clearly in the midpoint accuracy barriers, which are stable under completeP (or even slightly shrinking), whereas they rise up to almost 60 % in the unscaled version. Interestingly, applying the proper K^{-1} scaling even consistently outperforms repairing the activations of the models in standard parameterization.

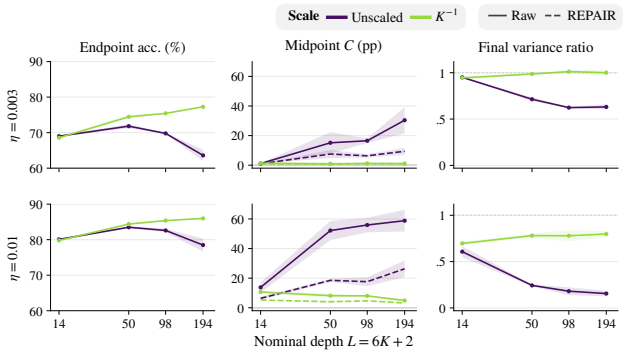


Figure 26: **Variance collapse in ResNets with BatchNorm on CIFAR-10.** SP (unscaled) vs. completeP (K^{-1}) residual scaling. *Left:* Endpoint acc., *center:* midpoint acc. barriers, *right:* variance ratios in the final block.

F.2.1 Behavior Under Permutation Alignment

A related question to ours is how loss barriers behave for models trained from independent initializations under alignment with parameter symmetries (Singh & Jaggi, 2020; Ainsworth et al., 2023). One could therefore ask if chord deviations for aligned models from independent initializations admit a scaling limit as well, and, e.g., whether they approach a width limit differently from unaligned branches of a shared initialization. There is theoretical evidence for the existence of a scaling limit for aligned loss barriers as well: Ferbach et al. (2024) establish a vanishing-barrier limit for two-layer mean-field networks. Intuitively, as width grows, two empirical neuron measures approach a common distribution in Wasserstein distance, and their optimal matching cost vanishes in the limit. We compare unaligned vs. aligned barriers in fig. 27 for the ConvNet on CIFAR-10 from § E.1.1. In this setting, we find that for most LRs, branches with independent training noise still show substantial width dependence after 50 epochs for $b = 0$, but stabilize considerably for $b = 10$ (*middle*). In contrast, independently initialized models generally become better alignable with increasing width (*right*) which is consistent with Ainsworth et al. (2023). We hypothesize that these different finite-width trends reflect the matching problem itself, i.e., increasing width refines the empirical measures and changes the optimal bijection between two growing sets of particles, whereas shared initialization fixes their correspondence in advance which can leave a nonzero limiting barrier. We view a systematic study of how large architectures approach scaling limits under alignment by parameter symmetries, and how this compares with our results for shared initialization, as an interesting direction for future work.

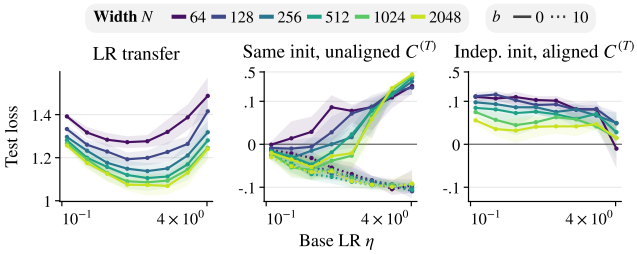


Figure 27: **Comparison to LMC under permutation alignment in ConvNet on CIFAR-10.** *Left:* LR transfer. *Center:* Chord deviations for unaligned LMC from the same initialization, with the branching setup of § 3 at two branching times. *Right:* Chord deviations for *aligned* LMC from *different* initializations.

F.3 Model Merging

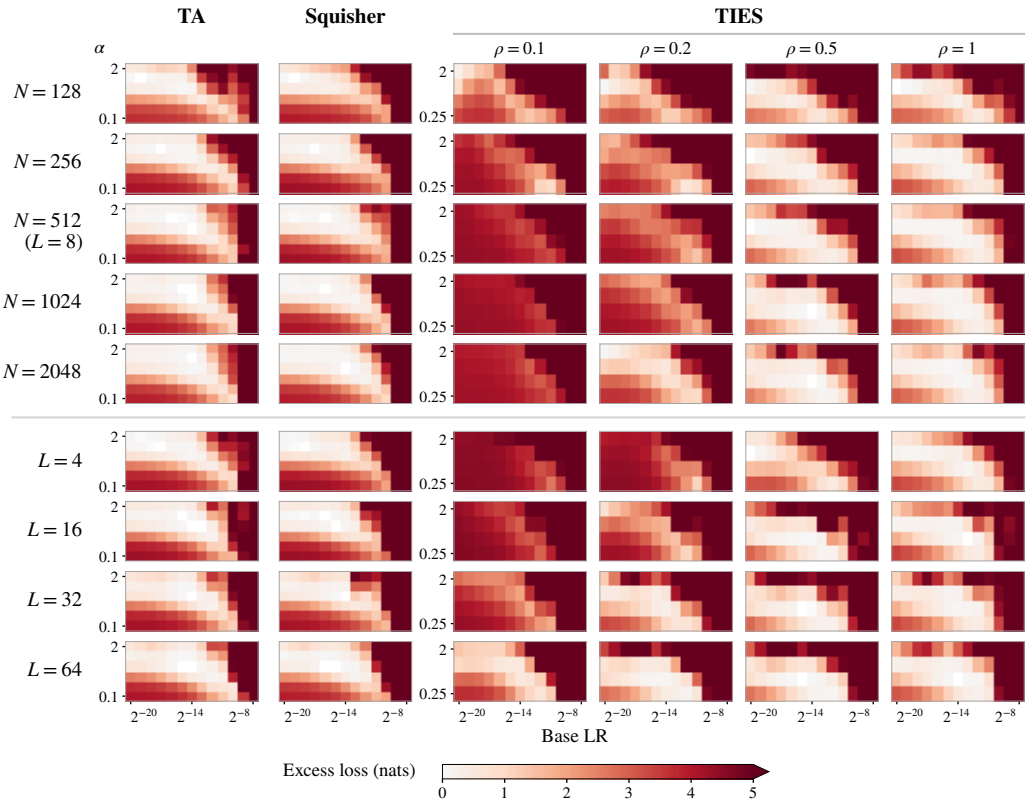


Figure 28: Excess loss relative to the best setting for each model and merging method, minimized over base LR η, α, and TIES density ρ. Losses are averaged over 8 tasks and 3 seeds each.