Revisiting the Goldilocks Zone in Inhomogeneous Networks

Zacharie Garnier Cuchet[2,3,5], **Sarath Chandar**[1,2,3,6], **Ekaterina Lobacheva**[1,2,4]
[1]*Chandar Research Lab* [2]*Mila - Quebec AI Institute* [3]*Polytechnique Montréal* [4]*Université de Montréal*
[5]*Université Technologique de Compiègne* [6]*Canada CIFAR AI Chair*
{zacharie.garnier-cuchet,sarath.chandar,ekaterina.lobacheva}@mila.quebec

Abstract

We investigate how architectural inhomogeneities, such as biases, layer normalization, and residual connections, affect the curvature of the loss landscape at initialization and its link to trainability. We focus on the Goldilocks zone, a region in parameter space with excess positive curvature, previously associated with improved optimization in homogeneous networks. To extend this analysis, we compare two scaling strategies: weight scaling and softmax temperature scaling. Our results show that in networks with biases or residual connections, both strategies identify a Goldilocks zone aligned with better training. In contrast, layer normalization leads to lower or negative curvature, yet stable optimization, revealing a disconnect between curvature and trainability. Softmax temperature scaling behaves more consistently across models, making it a more robust probe. Overall, the Goldilocks zone remains relevant in inhomogeneous networks, but its geometry and predictive power depend on architectural choices, particularly normalization.

1. Introduction

Understanding what makes neural networks easy or hard to train remains a core challenge in deep learning. Among the many contributing factors, the geometry of the loss landscape, often captured via the Hessian's curvature, plays a key role in shaping optimization dynamics. A particularly relevant region is the *Goldilocks zone* [3], located at a roughly constant radius in parameter space where the loss shows excess positive curvature and training tends to be most effective.

This phenomenon has been primarily studied in *homogeneous* architectures, where scaling all weights by a positive scalar α scales the output as $f_{\alpha\theta}(x) = \alpha^L f_\theta(x)$ for depth L. In this setting, Fort and Scherlis [3] proposed probing the loss by adjusting α, effectively moving radially and revealing a curvature peak at intermediate values. More recently, Vysogorets et al. [13] introduced softmax temperature scaling (T), which alters output confidence without changing the weight norm. While both are equivalent in homogeneous networks when $T = \alpha^L$, this breaks down under architectural inhomogeneities.

Practical networks are rarely homogeneous. Elements like non-zero biases, residual connections, and LayerNorm break the scaling property and reshape the loss landscape in nontrivial ways. While recent work has questioned the predictive value of the Goldilocks zone even in idealized settings [13], its behavior in realistic inhomogeneous architectures remains poorly understood.

In this work, we revisit the Goldilocks zone in inhomogeneous networks. We compare two strategies for probing curvature at initialization: weight scaling (α) and softmax temperature scaling (T). While α-scaling modifies all trainable parameters (weights and biases), altering both norm and output, its interaction with inhomogeneous components is often inconsistent. In contrast, T-scaling adjusts model confidence at the output level, offering a cleaner control signal across architectures.

© Z.G. Cuchet, S. Chandar & E. Lobacheva.

We address two questions: (1) Can the Goldilocks zone be identified in inhomogeneous networks? (2) Does it correlate with trainability? For networks with biases or residual connections, both scaling methods reveal a zone of high positive curvature aligned with improved optimization. With layer normalization, however, curvature is lower or negative, yet training remains stable, suggesting a more complex relationship between curvature and trainability. Overall, softmax temperature scaling behaves more consistently across architectures and proves to be a more robust tool for probing initialization geometry.

2. Methodology

We study how the Goldilocks zone behaves in the presence of architectural inhomogeneities, and whether it remains predictive of trainability. Our approach combines curvature analysis at initialization with optimization evaluations. We test two hypotheses: (1) temperature scaling is a more robust and interpretable probe of the Goldilocks zone than weight scaling, and (2) excess positive curvature at initialization does not consistently predict trainability in inhomogeneous networks.

Curvature Estimation. Following Vysogorets et al. [13], we approximate the Hessian spectrum via a low-rank projection using a sparse orthogonal matrix $R \in \mathbb{R}^{P\times 50}$. The resulting matrix $H_d = R^\top HR$ captures essential curvature information. We quantify curvature using the **excess of positive curvature** [3], where λ_i are the Hessian eigenvalues:

$$\frac{\mathrm{Tr}(H)}{\|H\|_F} = \frac{\sum_i \lambda_i}{\sqrt{\sum_i \lambda_i^2}}.$$

Architectures and Perturbations. We conduct experiments with LeNet-300-100 on Fashion-MNIST [14], LeNet-5 [11] on CIFAR-10 [9], and a minimalist ResNet-4 on both datasets. To break homogeneity, we introduce non-zero biases, layer normalization [2], and residual connections [7]. Each disrupts the standard scaling $f_{\alpha\theta}(x) = \alpha^L f_\theta(x)$ differently: biases alter intermediate scales, LayerNorm nullifies scaling in some layers, and residuals introduce asymmetric pathways. Formal analyses are in the Appendix. All networks use Kaiming normal initialization [6] for linear and convolutional weights. Biases are zero-initialized by default, or sampled from $\mathcal{U}[-0.1, 0.1]$ in the non-zero bias setting. We use pre-activation LayerNorm with $\epsilon = 10^{-5}$, and residual blocks of the form $y = x + W_2\phi(W_1x)$. While this study focuses on classification, Appendix D discusses potential extensions to regression settings.

Reconstructing the Goldilocks Zone. We vary two initialization parameters: weight scale $\alpha \in [10^{-3}, 10^3]$ or $\alpha \in [10^{-3L}, 10^{3L}]$, and softmax temperature $T \in [10^{-3L}, 10^{3L}]$, with L the network depth. Excess of positive curvature is measured at initialization, typically using a batch size of 128.

Connecting to Optimization. We train all models with full-batch GD for 1000 epochs. To account for the interaction between scaling and gradient magnitudes, we adjust the learning rates accordingly (derivations in Appendix I). Final train accuracy serves as our main metric to assess whether initial curvature correlates with trainability under realistic optimization. It is worth noting that test accuracy exhibits the same behavior and is presented in Appendix K.

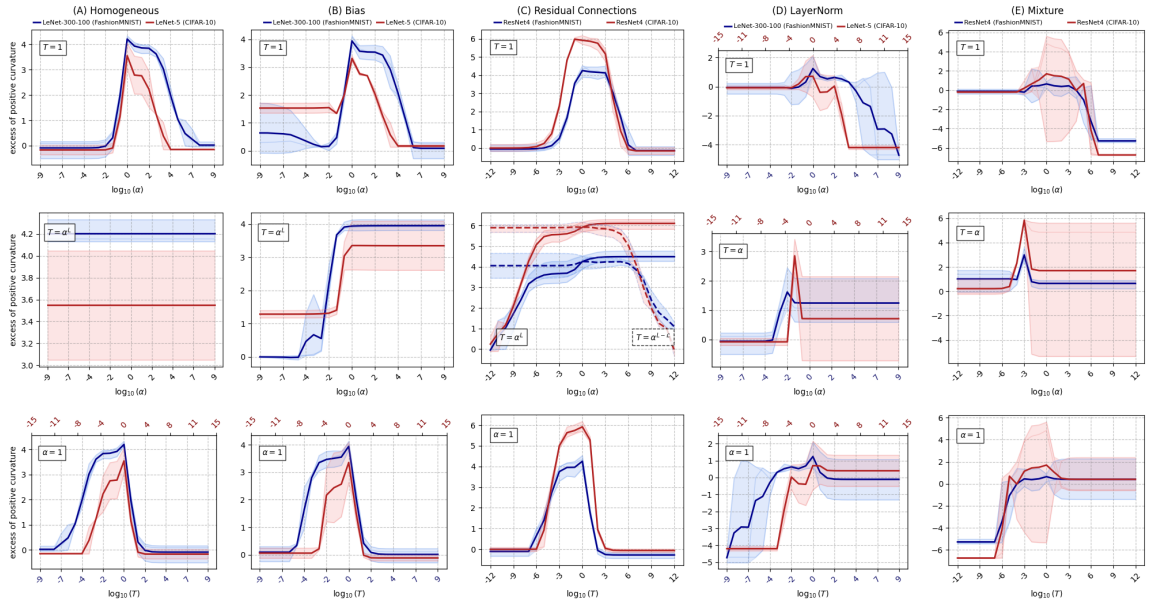

Figure 1: **Excess Positive Curvature for Different Architectural Variants and Scaling Strategies.** Columns correspond to a distinct architectural settings: (A) homogeneous baseline, (B) non-zero biases, (C) residual connections, (D) LayerNorm, and (E) a combined model with all three inhomogeneities. Rows display results under different scaling strategies: (top) weight scaling α, (middle) joint weight and temperature scaling, and (bottom) temperature T scaling.

3. Results

3.1. Recreating the Goldilocks zone

This study aims to examine how the Goldilocks zone manifests in inhomogeneous networks. A common way to probe this zone is through weight scaling (α), which moves the model along rays in parameter space and modulates output confidence. Small α values push the network toward low-confidence, while large values lead to overconfidence. Softmax temperature scaling offers an alternative that adjusts confidence directly, without altering the weight norm, and is therefore especially useful for comparing architectures where α-scaling behaves inconsistently.

Results. Figure 1 shows the excess of positive curvature across scaling regimes for various architectures. In addition to weight scaling (α) and softmax temperature scaling (T), we also analyze their joint scaling (middle row) to understand where the two strategies are equivalent. Specifically, we scale the softmax temperature to counteract the effect of α-scaling on the network output. If the strategies are truly equivalent, the curvature should remain constant under joint scaling.

We observe that across all architectures, a Goldilocks zone, marked by a peak of positive curvature, emerges under both scaling strategies. This includes networks with biases, residuals, and even LayerNorm, although the curvature in LayerNorm models is much lower and can become negative outside the 0-scaling zone. This general presence of a peak suggests that the Goldilocks zone is a robust feature of neural networks, even beyond homogeneous cases.

The baseline homogeneous model serves as a reference: it shows perfect symmetry between α- and T-scaling, confirming that both can produce the same output effect [13]. In contrast, inhomogeneous networks break this symmetry. Although a Goldilocks zone still emerges with both strategies, their behavior diverges. This reflects the fact that in inhomogeneous architectures, small and large α lead to markedly different dynamics, while softmax temperature, by acting only on output confidence, is less sensitive to such effects.

The middle row of Figure 1, where both α and T are scaled jointly, is especially informative. In homogeneous networks, the curvature stays flat across all α in this plot, confirming that $T = \alpha^L$ cancels the effect of α-scaling and matches the theoretical prediction.

In inhomogeneous networks, we also observe flat curvature in the large-α regime, indicating that in this regime, weight scaling can still be compensated by an appropriate temperature scaling. Specifically, we can choose T to recover the output confidence and match the curvature, at least approximately or asymptotically. This includes: (i) biases and residuals, which behave similarly to homogeneous networks when α is large, and (ii) LayerNorm, where the output scales linearly.

However, for small α, this cancellation breaks down. The output behavior no longer follows simple scaling rules, and softmax temperature alone cannot recover the distortions caused by weight scaling. Notably, in networks with biases, the last-layer bias dominates the output at small α, and temperature has no effect on additive terms (Appendix E). In LayerNorm, the output becomes nearly constant due to the dominance of the ε term (Appendix G), leading to curvature collapse or reversal.

Residual connections are a notable exception. In the small-α regime, the identity path dominates and the output scales as $\alpha^{L-\mathcal{L}}$, where $\mathcal{L}$ is the total number of layers in residual blocks. In this case, the effect of weight scaling can be approximately canceled by scaling the temperature as $T = \alpha^{L-\mathcal{L}}$ (Appendix F). This is the only architecture where small-α behavior remains recoverable through a modified temperature scaling.

3.2. Connection to Optimization

The second goal of this study is to investigate whether there exists a consistent link between the Goldilocks zone and trainability in inhomogeneous networks. While Vysogorets et al. [13] has shown that this connection does not always hold for homogeneous settings, we aim to revisit this question in the presence of architectural inhomogeneities. In particular, we focus on networks with normalization, for which we observed notably different curvature behaviors (Figure 1).

To compare trainability across architectures, we focus on softmax temperature scaling rather than weight scaling. As discussed, in inhomogeneous networks, α-scaling yields distinct behaviors in small and large α regimes due to architecture-specific output scaling. For instance, at small α, biases can dominate in biased networks, LayerNorm is governed by the ε term, and residuals are dominated by the identity path. These effects alter output scaling and, in turn, the gradients, making learning rate selection non-trivial and architecture-dependent. Appendix I details these dynamics, including gradient behavior and learning rate adjustment rules.

In contrast, softmax temperature scaling modifies the output confidence directly and admits a simple learning rate adjustment ($\eta' = T \cdot \eta_0$) that remains valid across architectures (Appendix I). For this reason, we use T-scaling to evaluate the relationship between curvature and trainability in inhomogeneous settings. For completeness, results obtained under α-scaling are also reported in Appendix J. While informative, they are harder to interpret due to architecture-specific dynamics and inconsistent scaling behavior.

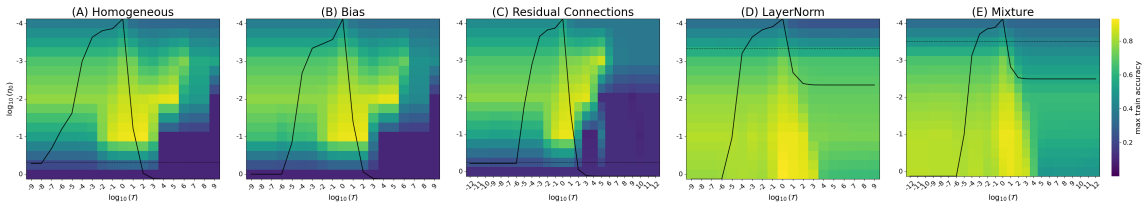

Figure 2: **Connection to Optimization - Train Accuracy for LeNet-300-100 under Varying Learning Rates and Architectural Inhomogeneities.** Each column corresponds to a different architecture: (A) baseline (homogeneous), (B) non-zero biases, (C) residual connections, (D) LayerNorm, and (E) a mixture of all three inhomogeneities. Overlaid on each plot is the Goldilocks zone boundary (as defined in Figure 1), allowing comparison between regions of favorable curvature and trainability. Test accuracy exhibits the same general patterns and is presented in Appendix K.

Results. Figure 2 presents the train accuracy across temperature scales for each variant. The corresponding test accuracy plots, which exhibit similar trends, are available in Appendix K. In networks with biases and residual connections, we observe a strong alignment between the Goldilocks zone and trainability: both architectures achieve their best performance near the temperature-defined curvature peak.

For residual connections, however, we note a sharper decline in trainability as temperature increases beyond this zone. This may suggest that residual architectures are more sensitive to initialization. A similar observation was made by Zhang et al. [15], who showed that in deep residual networks, failing to account properly for the scaling effects of residual paths, particularly in the absence of normalization, can lead to unstable gradients and hinder optimization.

In contrast, LayerNorm (and the combined model, which is largely driven by LayerNorm behavior) displays a more surprising pattern. Despite weak or even negative excess positive curvature, these models maintain stable and high training performance across a wide temperature range. This supports prior observations on the stabilizing role of normalization [2, 12].

4. Conclusion

Our study extends the analysis of the Goldilocks zone to inhomogeneous neural networks. We show that both weight and temperature scaling can reveal zones of high positive curvature in architectures with biases and residual connections, and that these zones generally align with improved trainability. However, the behavior becomes more complex in networks with normalization: while a weak signature of the Goldilocks zone remains, its predictive power for optimization becomes more illusive. These findings highlight the need for new tools to understand how architectural choices, especially normalization, alter the geometry of the loss landscape and its connection to learning dynamics.

Acknowledgments

We would like to thank the Mila community for insightful discussions and support throughout this project. Sarath Chandar is supported by the Canada CIFAR AI Chairs program, the Canada Research Chair in Lifelong Machine Learning, and the NSERC Discovery Grant. Ekaterina Lobacheva is supported by IVADO and the Canada First Research Excellence Fund.

References

[1] Maksym Andriushchenko, Francesco Croce, Maximilian Mueller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2302.07011.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[3] Stanislav Fort and Adam Scherlis. The Goldilocks zone: Towards better understanding of neural network loss landscapes. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://arxiv.org/abs/1807.02581.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

[5] Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace, 2018. URL https://arxiv.org/abs/1812.04754.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision*, 2015. URL https://arxiv.org/abs/1502.01852.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. URL https://arxiv.org/abs/1512.03385.

[8] Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, 2023. URL https://proceedings.mlr.press/v187/kaur23a.html.

[9] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[10] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. URL http://www.jstor.org/stable/2236334.

[11] Yann LeCun. Lenet-5, convolutional neural networks. http://yann.lecun.com/exdb/lenet, 2015. Accessed on 30 January 2023.

[12] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1805.11604.

[13] Artem M Vysogorets, Anna Dawid, and Julia Kempe. Deconstructing the Goldilocks zone of neural network initialization. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/vysogorets24a.html.

[14] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

[15] Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Residual learning without normalization via better initialization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gsz30cKX.

Appendix A. Related Work

Loss Landscape Geometry and Initialization. The geometry of neural network loss landscapes is central to understanding optimization and initialization. Seminal work by Glorot and Bengio [4] and He et al. [6] established initialization norms that empirically promote trainability. Fort and Scherlis [3] later formalized this intuition by identifying the Goldilocks zone, a region of parameter space where the loss exhibits excess positive curvature, measured via the Hessian matrix. They showed that common initializers (e.g., [4, 6]) lie within this zone, explaining their effectiveness. Crucially, they observed that training dynamics naturally pull networks into this region, suggesting it contains a high density of optimal solutions. However, the zone's outer edges exhibit unstable curvature variance, leading to erratic optimization.

Refining the Goldilocks Zone. Recent work by Vysogorets et al. [13] challenges the assumption that initialization norm alone governs positive curvature. They decompose the Hessian into its positive semi-definite component (G-term), linking curvature to softmax saturation and logit gradient vanishing. Their analysis reveals that excess curvature correlates with low initial loss and gradient norms but does not always predict final performance. Notably, they demonstrate that softmax temperature scaling can mimic the effects of weight norm adjustments, offering an alternative control mechanism. However, their study focuses on homogeneous networks, leaving open questions about inhomogeneous architectures (e.g., with residuals, biases or LayerNorm).

Optimization Dynamics and Limitations. The connection between curvature and trainability remains nuanced. While the Goldilocks zone facilitates gradient descent [5], Vysogorets et al. [13] show that networks within the zone can still exhibit degenerate behaviors (e.g., zero logits). Adaptive optimizers (e.g., Adam) may further complicate this relationship by escaping flat regions more effectively than SGD. These results suggest that initialization-induced curvature is neither necessary nor sufficient to guarantee success, motivating further investigation of architecture-dependent effects.

Our Contribution. We extend this line of work by investigating the Goldilocks zone in inhomogeneous networks (e.g., with biases, residuals, and LayerNorm), addressing gaps left by prior studies. We empirically evaluate how weight scaling and softmax temperature interact with optimization hyperparameters (e.g., learning rate), bridging theoretical curvature analysis with practical trainability.

Appendix B. Relation to Curvature-Generalization Debates

Recent work has questioned the usefulness of curvature-based metrics, particularly the Hessian's largest eigenvalue ($\lambda_{\max}$), as predictors of generalization [1, 8]. These studies highlight that sharpness measures may be sensitive to factors such as learning rate, batch size, reparameterization, or the use of normalization, and that in some cases, flatter minima do not reliably correspond to better generalization.

While we acknowledge this growing body of evidence, we emphasize that our study is focused not on generalization, but on trainability. Specifically, we investigate how curvature, quantified via the excess of positive eigenvalues of the Hessian, relates to the network's ability to optimize the training loss from a random initialization, and how this relationship is affected by architectural inhomogeneities.

We believe that decoupling these two aspects, trainability and generalization, is essential, and hope that our results contribute to a more precise understanding of how initialization geometry affects early optimization dynamics, particularly in inhomogeneous networks.

Appendix C. Preliminaries

We consider a K-way classification task with dataset $D = \{(X_\mu, y_\mu)\}_{\mu=1}^N$, where $X_\mu \in \mathbb{R}^d$ and $y_\mu \in [K]$. A neural network f_θ, parameterized by $\theta \in \mathbb{R}^P$, maps each input to logits $\{z_k^{(\mu)}\}_{k=1}^K$, which are transformed into class probabilities via the softmax function with temperature T:

$$p_k^{(\mu)} = \sigma_T(z^{(\mu)})_k = \frac{\exp(z_k^{(\mu)}/T)}{\sum_{c=1}^K \exp(z_c^{(\mu)}/T)}.$$

The total loss is the average cross-entropy over the dataset:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{\mu=1}^N \ell(p^{(\mu)}, y_\mu) = -\frac{1}{N}\sum_{\mu=1}^N \log p_{y_\mu}^{(\mu)}.$$

We study the curvature of this loss landscape through the Hessian matrix:

$$H_{ij} = \frac{\partial^2 \mathcal{L}}{\partial\theta_i \partial\theta_j},$$

which reflects second-order sensitivity of the loss over the entire dataset (or batch, in the minibatch setting).

Appendix D. Extension to Regression Tasks

While our study focuses on classification, the broader question of how initialization curvature relates to trainability also applies to regression tasks. Even though neural network regressors do not have a softmax layer, the temperature scaling methodology can be extended to this setting as well.

Softmax temperature scaling, in practice, corresponds to dividing the logits (i.e., the pre-softmax outputs) by a temperature parameter T, effectively modulating the model's output confidence. For regression, a direct analogue does not exist, since the output is not a probability distribution. Still, a similar operation can be introduced by scaling the output of the final layer by a scalar $1/T$, yielding a controllable "output temperature" that influences the magnitude of predictions. This leads to a natural analogue of temperature scaling for regression tasks.

Such a temperature-like parameter could be introduced post hoc or implemented by scaling only the final layer's weights. In contrast, weight scaling uniformly rescales all learnable parameters, including earlier layers. This distinction offers a parallel between full-network scaling and output-only scaling.

Appendix E. Scaling Properties with non-zero biases

Consider a fully connected neural network with L layers, ReLU activation functions, and non-zero biases. Let $f_\theta(x)$ denote the output before the softmax for an input x, where θ represents the

network's parameters (weights and biases). In homogeneous models (without biases), scaling the weights by $\alpha > 0$ scales the output by α^L. However, with biases, this property breaks due to their additive nature. We analyze how biases affect the scaling behavior and discuss the implications for the Goldilocks zone and softmax temperature.

Scaling with Non-Zero Biases. For a network with non-zero biases, the output of each layer h_l is given by:

$$h_1 = \alpha W_1 x + \alpha b_1,$$

$$h_2 = \alpha W_2 h_1 + \alpha b_2 = \alpha W_2(\alpha W_1 x + \alpha b_1) + \alpha b_2,$$

$$h_2 = \alpha^2 W_2 W_1 x + \alpha^2 W_2 b_1 + \alpha b_2.$$

Extending this to L layers, the output h_L becomes:

$$h_L = \alpha^L (W_L \dots W_1 x) + \alpha^L (W_L \dots W_2 b_1) + \alpha^{L-1} (W_L \dots W_3 b_2) + \dots + \alpha b_L.$$

Large and Small Scaling Regimes.

- Large α ($\alpha \gg 1$): The term $\alpha^L \left[(W_L \dots W_1 x) + (W_L \dots W_2 b_1)\right]$ dominates the output, while the remaining bias terms $(\dots + \alpha^2 W_L b_{L-1} + \alpha b_L)$ become negligible. In this regime, the output scales approximately as $\alpha^L f_\theta(x)$, as long as the depth of the network is reasonable. Indeed for homogeneous, the theoretically expected behavior, including all biases, is:

  $$\alpha^L f_\theta(x) = \alpha^L \left[(W_L \dots W_1 x) + (W_L \dots W_2 b_1) + \dots + b_L\right],$$

  This could correspond to the case where each bias b_l is scaled by α^l. However, for networks with α-scaled biases, for large α, the output effectively reduces to a truncated form that depends only on the input and the first-layer bias $\alpha^L \left[(W_L \dots W_1 x) + (W_L \dots W_2 b_1)\right]$. As the depth L increases, this divergence from the full homogeneous behavior grows: deeper networks amplify only the early components, leading to an increasingly distorted representation. Thus, paradoxically, large α does not lead to homogeneous behavior, but rather to a biased approximation dominated by early layers. In our experiments, which use networks of depth 3 and 5, the behavior remains close to the homogeneous case (Figure 1.B). However, we expect that for significantly deeper architectures, the divergence from the full expression would become much more pronounced.

- Small α ($\alpha \ll 1$): The last bias terms dominate, and the output scales as αb_L. In this regime, the scaling behavior deviates significantly from the homogeneous case, as the biases break the homogeneity of the network.

These theoretical insights are empirically validated in Figure 1.B of the main text, which illustrates how the excess positive curvature varies across different scaling regimes in networks with non-zero biases.

Appendix F. Scaling Properties in Networks with Residual Connections

In this section, we analyze the scaling properties of neural networks with residual connections [7]. Residual blocks introduce a skip connection that adds the input x directly to the output of a sequence of transformations. We assume no biases and no layer normalization in the residual block. We show that the output of the scaled model depends on α, with α^L-behavior (homogeneous) for large α and $\alpha^{L-\mathcal{L}}$-behavior for small α, where $\mathcal{L}$ is the total number of layers in residual blocks.

Effect of Residual Connections. Consider a residual block defined as follows:

$$\text{ResidualBlock}(x) = F(x) + x,$$

where $F(x)$ represents a sequence of transformations (e.g., linear layers and activations). In our case, $F(x)$ consists of:

$$F(x) = W_2 \cdot \text{ReLU}(W_1 x),$$

where W_1 and W_2 are weight matrices. When scaling the weights by $\alpha > 0$, the residual block becomes:

$$\text{ResidualBlock}'(x) = \alpha W_2 \cdot \text{ReLU}(\alpha W_1 x) + x.$$

Asymptotic Scaling Regimes. The residual connection introduces two distinct scaling regimes:

- **Large α ($\alpha \gg 1$):** The term $\alpha W_2 \cdot \text{ReLU}(\alpha W_1 x)$ dominates, and the residual block behaves like a homogeneous transformation: $\text{ResidualBlock}'(x) \approx \alpha^2 W_2 \cdot \text{ReLU}(W_1 x)$. In this regime, the output scales with α^L, and the network behaves similarly to a homogeneous model.
- **Small α ($\alpha \ll 1$):** The term $\alpha W_2 \cdot \text{ReLU}(\alpha W_1 x)$ becomes negligible, and the residual block reduces to the identity function: $\text{ResidualBlock}'(x) \approx x$. In this regime, the output scales with $\alpha^{L-\mathcal{L}}$.

These theoretical insights are validated by the empirical results shown in Figure 1.C of the main text. The middle line of 1.C further confirms them under both scaling of T (α^L and $\alpha^{L-\mathcal{L}}$), aligning with the network transformation passing through the identity path.

Appendix G. Scaling Properties in Networks with Layer Normalization

In this section, we analyze the scaling properties of neural networks with layer normalization (LayerNorm, Ba et al. [2]) applied after each linear or convolutional layer. LayerNorm introduces a different form of inhomogeneity by normalizing the activations at each layer. We show that LayerNorm cancels out the effect of weight scaling for some layers, making the output independent of the network's depth. The behavior of those networks can align with homogeneous models but requires more extreme scaling of weights and softmax temperature to observe similar effects.

Consider a fully connected neural network with L layers, ReLU activation functions, and LayerNorm applied after each linear transformation. Let $f_\theta(x)$ denote the output before the softmax for an input x, where θ represents the network's parameters. We assume no biases are present in the linear layers.

Effect of Layer Normalization. Layer normalization normalizes the activations of each layer to have zero mean and unit variance. For a given layer l with input h_{l-1}, the output h_l is computed as:

$$h_l = \text{LayerNorm}(W_l h_{l-1}),$$

where LayerNorm(z) is defined as:

$$\text{LayerNorm}(z) = \gamma \odot \frac{z - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta,$$

with μ and σ being the mean and standard deviation of z, and γ and β being learnable parameters. For simplicity, we assume $\gamma = 1$, $\beta = 0$, and ε negligible, reducing LayerNorm to:

$$\text{LayerNorm}(z) = \frac{z - \mu}{\sigma}.$$

Scaling Invariance of LayerNorm. Suppose we scale the weights of each layer by $\alpha > 0$, i.e., $W_l' = \alpha W_l$. The output of the first layer becomes:

$$h_1' = \text{LayerNorm}(\alpha W_1 x) = \frac{\alpha W_1 x - \mu_1'}{\sigma_1'},$$

where μ_1' and σ_1' are the mean and standard deviation of $\alpha W_1 x$. Since scaling W_1 by α scales both the mean and standard deviation by α, we have:

$$h_1' = \frac{\alpha W_1 x - \alpha \mu_1}{\alpha \sigma_1} = \frac{W_1 x - \mu_1}{\sigma_1} = h_1.$$

Thus, LayerNorm cancels out the effect of weight scaling at each layer, making the output invariant to α:

$$h_l' = h_l \quad \text{for all layers } l \text{ containing LayerNorm.}$$

Output Scaling. The final output of the network before the softmax is given by the last linear layer, which does not have LayerNorm applied:

$$f_{\theta'}(x) = W_L' h_{L-1}' = \alpha W_L h_{L-1}.$$

Since $h_{L-1}' = h_{L-1}$, the output scales linearly with α:

$$f_{\theta'}(x) = \alpha f_\theta(x).$$

Implications for Scaling. Scaling the weights by α scales the output by α, but the intermediate activations remain invariant due to LayerNorm. This behavior is similar to homogeneous models but does not depend on the network's depth. However, when α becomes very small, the pre-normalization activations $z_\ell = \alpha W_\ell h_{\ell-1}$ approach zero. In this regime, the normalization operation becomes unstable: the standard deviation of the activation vector tends to zero, and the ε term (typically added for numerical stability) begins to dominate. Consequently, the output becomes insensitive to both the input and the parameters, leading to vanishing gradients and degraded trainability. Figure 3 shows the Kullback-Leibler Divergence [10], as a similarity metric, between

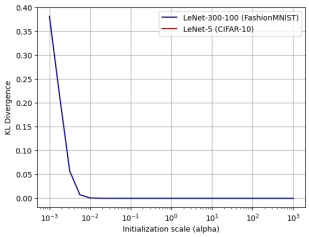

Figure 3: $D_{\mathbf{KL}}$ divergence between expected logits αf_θ and the actual logits $f_{\theta'}$.

the expected scaled output $\alpha f_\theta(x)$ and the actual output $f_{\theta'}(x)$, confirming that the linear scaling relationship holds except when α becomes very small.

These theoretical insights are empirically validated in Figure 1.D of the main text, which presents three subplots highlighting the impact of LayerNorm on the emergence of the Goldilocks zone. Interestingly, networks initialized with LayerNorm struggle to reach a well-defined Goldilocks region.

Appendix H. Scaling Properties in Networks with Layer Normalization, Residual Connections, and Biases

Consider a neural network with L layers. Let $f_\theta(x)$ denote the output before the softmax for an input x, where θ represents all parameters (weights, biases). In this section we analyze the effect of scaling the weights and biases by $\alpha > 0$ for a residual model with biases and LayerNorm.

Scaling invariance of LayerNorm. When we scale the weights and biases by α, the pre-norm activations become:

$$z_l = \alpha W_l h_{l-1} + \alpha b_l = \alpha(W_l h_{l-1} + b_l).$$

The mean and standard deviation scale linearly with α:

$$\mu_l' = \alpha \mu_l, \quad \sigma_l' = \alpha \sigma_l,$$

where μ_l and σ_l are the mean and standard deviation of $W_l h_{l-1} + b_l$. Thus, LayerNorm cancels the scaling:

$$\text{LayerNorm}(z_l) = \frac{\alpha(W_l h_{l-1} + b_l) - \alpha\mu_l}{\alpha\sigma_l} = \frac{W_l h_{l-1} + b_l - \mu_l}{\sigma_l} = \text{LayerNorm}(W_l h_{l-1} + b_l).$$

This shows that the output of LayerNorm is invariant to α, regardless of the presence of biases.

Residual connection preserves scaling invariance. The layer output with residual connection is:

$$h_l = \text{LayerNorm}(W_l h_{l-1} + b_l) + h_{l-1},$$

which is independent of α. Thus, all intermediate activations h_l remain unchanged when scaling weights and biases by α.

Final layer scaling. The network's final output (before softmax) is:

$$f_{\theta'}(x) = \alpha W_L h_{L-1} + \alpha b_L,$$

since the last layer typically does not have LayerNorm or residual connections. This scales linearly with α:

$$f_{\theta'}(x) = \alpha(W_L h_{L-1} + b_L) = \alpha f_\theta(x).$$

In this combined architecture, LayerNorm dominates the scaling behavior, resulting in a linear scaling relationship with α that is independent of network depth. This behavior is empirically validated in Figure 1.E of the main text.

Appendix I. Learning Rate Adjustments for Scaled Models

To ensure that training dynamics remain comparable across different scaling regimes, we adjust the learning rate depending on (1) the type of scaling applied (weight or softmax temperature) and (2) the architectural components present in the network. We consider three types of architectural inhomogeneities (biases, residual connections, and layer normalization) and analyze their behavior under both small and large scaling factors. When possible, we derive learning rate rules that preserve the relative update magnitude with respect to the model's output and parameter norm.

Softmax Temperature Scaling

Scaling the softmax by a temperature T modifies the gradient of the cross-entropy loss as follows:

$$\nabla_\theta \ell = \frac{1}{T} \left(\frac{\partial z}{\partial \theta} \right)^\top (\sigma_T(z) - y),$$

which leads to the update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{T} \left(\frac{\partial z}{\partial \theta} \right)^\top (\sigma_T(z) - y).$$

To maintain a consistent effective step size, we set the scaled learning rate as:

$$\eta' = T \cdot \eta_0.$$

This rule is simple, general, and largely independent of architectural details, making softmax temperature scaling a robust baseline for connecting initialization and trainability.

Weight Scaling and Architectural Inhomogeneities

Let $\theta' = \alpha\theta$ be the weight-scaled parameters and $f_{\theta'}(x)$ the corresponding network output. The required learning rate adjustment depends strongly on how the architecture responds to the scaling factor α. We now analyze each case.

Biases. **Large α:** In relatively shallow networks, the output scales as $f_{\theta'}(x) \approx \alpha^L f_\theta(x)$, resembling homogeneous behavior (Appendix E). The logit gradients scale as $\nabla_{\theta'} z'_k = \alpha^{L-1} \nabla_\theta z_k$, and to preserve update size relative to parameter norm, we adopt the scaling as previously used by Vysogorets et al. [13]:

$$\eta' = \alpha^{2-L} \cdot \eta_0.$$

Small α: The network collapses toward a constant function, dominated by the output bias term, which is only scaled linearly. In this case, the homogeneity assumption fails, and no clear gradient scaling rule applies.

Residual Connections. **Large α:** The residual branch dominates and the output scales as α^L, as in homogeneous networks (Appendix F). The same learning rate scaling rule applies:

$$\eta' = \alpha^{2-L} \cdot \eta_0.$$

Small α: The skip (identity) path dominates, and the output scales as $\alpha^{L-\mathcal{L}}$, where $\mathcal{L}$ is the number of layers in residual branches. The gradient then scales as $\nabla_{\theta'} z'_k = \alpha^{L-\mathcal{L}-1} \nabla_\theta z_k$, and the appropriate learning rate is:

$$\eta' = \alpha^{2-L+\mathcal{L}} \cdot \eta_0.$$

Layer Normalization. **Large α:** The normalization cancels out intermediate scaling effects, and the output scales linearly with α: $f_{\theta'}(x) = \alpha f_\theta(x)$ (Appendix G). The gradient remains invariant: $\nabla_{\theta'} z'_k = \nabla_\theta z_k$. Thus, the learning rate should scale linearly:

$$\eta' = \alpha \cdot \eta_0.$$

Small α: The output becomes nearly constant due to the dominance of the ε term in the normalization denominator. In this regime, scaling effects vanish and gradient magnitudes become unpredictable. Consequently, no principled learning rate adjustment can be applied, and performance becomes sensitive to initialization noise.

Appendix J. Connection to Optimization with weight scaling

While our main analysis focuses on softmax temperature scaling, we also report results under weight scaling α for completeness (Figure 4). As discussed in the main text and in Appendix I, optimization under α-scaling introduces architecture-dependent effects on gradient magnitude and learning rate adjustment, especially for small α. To minimize these complications, we focus on the large-α regime where the scaling behavior is more predictable.

For networks with biases or residual connections, we apply the learning rate adjustment $\eta' = \alpha^{2-L} \cdot \eta_0$, which holds under the assumption of homogeneous-like scaling at large α. For LayerNorm and the combined model, where the output scales linearly with α, we use $\eta' = \alpha \cdot \eta_0$, again assuming the scaling behavior holds in the large-α limit.

Results. For homogeneous networks, as well as those with biases and residual connections, we recover similar trends as previous work [13] and as our experiment with softmax temperature scaling. In these cases, the divergence observed at low α (analogous to high temperatures) is consistent across both methods. Since the divergence regime correspond to the most sensitive region to learning rate and gradient dynamics, the differences between the two scaling strategies remain relatively small.

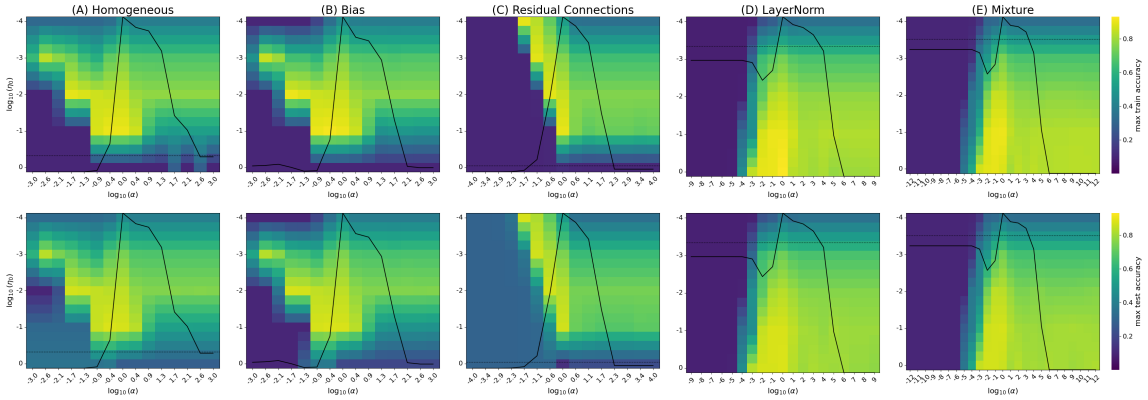

Figure 4: **Connection to Optimization - Train (top) and Test (bottom) Accuracy across Weight Scaling for LeNet-300-100 under Varying Learning Rates and Architectural Inhomogeneities.** Each column corresponds to a different architectural configuration: (A) baseline (homogeneous), (B) non-zero biases, (C) residual connections, (D) LayerNorm, and (E) a mixture of all three inhomogeneities. Overlaid on each plot is the Goldilocks zone boundary (as defined in Figure 1), allowing comparison between regions of favorable curvature and actual generalization.

However, for LayerNorm and the combined model, the behavior diverges more clearly. Under α-scaling, these models exhibit early training instability at small α, while they remain stable under large softmax temperatures. This instability is explained by the dominant role of the ε term in LayerNorm, which affects the backward gradients and disrupts the relationship between output scaling and update magnitude.

These results offer additional empirical support for our decision to focus on softmax temperature scaling in the main analysis. Temperature scaling provides more reliable and architecture-agnostic behavior, aligning better with theoretical expectations and enabling consistent learning rate adjustments.

Appendix K. Connection to Optimization: Test Accuracy and Temperature Scaling

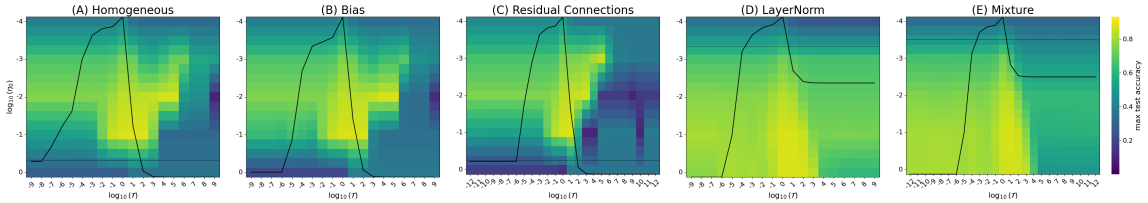

Figure 5: **Connection to Optimization - Test Accuracy across Softmax Temperature for LeNet-300-100 under Varying Learning Rates and Architectural Inhomogeneities.** Each column corresponds to a different architectural configuration: (A) baseline (homogeneous), (B) non-zero biases, (C) residual connections, (D) LayerNorm, and (E) a mixture of all three inhomogeneities. Overlaid on each plot is the Goldilocks zone boundary (as defined in Figure 1), allowing comparison between regions of favorable curvature and actual generalization.

The test accuracy results shown in Figure 5 mirror the trends observed in training accuracy (Figure 2). Across all architectures, the alignment between the Goldilocks zone and trainability remains consistent, further supporting the relevance of our analysis beyond the training set.

Appendix L. Limitations and Future Work

Our work investigates the Goldilocks zone in a range of inhomogeneous networks, focusing on curvature at initialization and its link to trainability. We highlight differences between weight and temperature scaling, and show how elements like biases, residuals, and LayerNorm reshape the loss landscape, but several questions remain.

First, our analysis is limited to initialization and early training. We do not track how curvature evolves during training or how it interacts with optimization steps. In particular, we do not separate Gauss-Newton and full Hessian contributions, which play distinct roles in homogeneous networks [13]. Extending this analysis to inhomogeneous settings would provide a fuller picture of how curvature relates to optimization dynamics.

Second, our experiments use controlled setups with fixed architectures and datasets. Whether these findings hold across tasks, depths, or training regimes remains open. Studying how learning rate schedules, optimizers (e.g., Adam), or regularization interact with initialization-induced curvature could clarify the Goldilocks zone's practical impact.

Although our study focuses on classification, we provide an initial discussion in Appendix D on how similar ideas may extend to regression.

Finally, while we use softmax temperature as a reliable probe of model confidence and curvature, it likely has a broader impact on training dynamics, which should be examined in future work.