UNDERSTANDING NETWORK SIZE SCALING THROUGH LINEAR REGION GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural network test losses are observed to evolve as power laws with the network size, but there is no theory that can fully predict their parameters after training. We present an empirical study of the learned properties that describes the scaling by exploiting the linear region partition of ReLU networks. We show that trained neural networks do not constitute ideal piecewise-linear approximators and identify that the regression loss is dominated by deviations of regional affine offsets from their optimal values. This offset error is described locally as the first-order variation of the target over a characteristic regional scale, and its mean depends approximately on the square of that same scale averaged over high-gradient regions. These results connect neural scaling to measurable properties of the piecewise-affine structure of networks and target geometry.

1 INTRODUCTION

Scaling has become an essential aspect of modern neural network development. Measurements across a wide variety of fields show that the test loss of any neural network scales as a power law on the model size, dataset size and compute used in training over several orders of magnitude (Hestness et al., 2017; Rosenfeld et al., 2020; Kaplan et al., 2020; Henighan et al., 2020; Zhai et al., 2022; Hoffmann et al., 2022; Caballero et al., 2023). This nearly universal observation is often leveraged to extrapolate neural network performance from small experiments and has been used to inform the optimal compute allocation for large-scale trainings. Despite the simplicity of the power law form, no theory yet has been able to fully characterize the neural scaling or fully predict the values of its parameters from first principles.

Several approaches have been explored to explain neural scaling laws. These include descriptions based on the effective dimension of the data manifold (Sharma and Kaplan, 2020) and spectral properties of the data distribution in solvable models (Bahri et al., 2024; Maloney et al., 2022). While these works provide important mechanisms for understanding the emergence of scaling laws, their results rely on idealized assumptions or effective descriptions and do not generally yield a direct prediction for the scaling of a neural network after training.

In this paper, we study the model-size scaling of simple multi-layer neural networks with piecewise-linear activation functions trained on regression tasks. Our approach is based on the analysis of the linear region partition, which is defined as the set of connected sections of the input space on which the network function is represented by a single affine map (Montúfar et al., 2014; Raghu et al., 2017; Balestriero and Baraniuk, 2018; Serra et al., 2018). This partition naturally decomposes the loss into a sum of regional contributions, whose scaling can be quantified in terms of their local features by using tools from approximation theory. This characterization allows us to probe which aspects of region geometry and the target function control the scaling. We examine this connection on trained networks that approximate low-dimensional analytical target functions with a well-defined notion of mathematical complexity.

For any piecewise-linear network, its ideal limit will be realized when an optimal affine approximation is represented in each region. We use this description as our starting point and quantify how much the learned offsets and slopes in each region deviate from their optimal realization. Across our experiments, the offset error dominates the loss and decreases more slowly than the other contributions, motivating a closer examination of its scaling.

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.

Our study of the offset error reveals that its scaling cannot be described through a simple model depending on the number of network parameters or a single global descriptor of the region partition. Instead, it is described by a non-trivial interplay between the region distribution and the derivative structure of the target function. By incorporating these effects into a single variable, we are able to express the offset error scaling as a constant approximation error rate.

Our main contributions are:

- We demonstrate that neural networks with piecewise-linear activation functions trained on regression tasks generally do not represent ideal piecewise linear approximators because of the deviation of the learned affine maps from their optimal realization within each region.
- We measure that the affine offset error is the dominant and slowest-decaying contribution to the global approximation error. By contrast, the affine slopes are more precise in approximating the target gradients and their error decreases at a faster rate.
- We empirically show that the offset error can be described as the average of the joint distribution of local target gradients and a characteristic regional scale that does not need to match the region size.
- We identify a quadratic power law that describes the scaling of the offset error in terms of a variable that aggregates all relevant aspects from the linear region partition and the target functional structure.

Our results show that the ideal piecewise-affine picture assumed by established explanations of the scaling does not describe trained networks, and that a description of the scaling requires an account of both network and target properties.

2 Background and related work

Neural scaling laws The dependence of the test loss on the number of network parameters N can be described by a power law with an asymptotic floor (Hestness et al., 2017; Kaplan et al., 2020; Rosenfeld et al., 2020):

$$\mathcal{E} = CN^{-\alpha} + \mathcal{E}_\infty, \tag{1}$$

There is currently no general theory that can predict $\mathcal{E}_\infty$, C, and α from the properties of a trained neural network. They can only be determined through empirical fits on sequences of trained models. Several theoretical approaches have sought to explain the exponent α through the intrinsic dimension of the data manifold (Sharma and Kaplan, 2020; Bahri et al., 2024), the spectral structure and task alignment of infinite-width models (Bordelon et al., 2020; Canatar et al., 2021), and on solvable random-feature regression models (Maloney et al., 2022; Atanasov et al., 2026). More recent work has extended these analyses to feature-learning regimes (Bordelon et al., 2025). However, these proposals rely on idealized assumptions or simplified settings that do not match the conditions of neural networks after training and thus cannot determine the scaling on finite trained networks.

Linear regions A network with piecewise-linear activations splits its input space Ω into linear regions R on which the network acts as

$$\hat{f}_R(x) = A_R^\mathsf{T} x + b_R, \qquad x \in R, \tag{2}$$

This partition is realized through the composition of linear mappings and activation functions across layers. First-layer neurons introduce region boundaries directly in the input space, whereas neurons from the deeper layers subdivide regions formed by the preceding layers. This structure implies that both the set of all affine region parameters $\{A_R, b_R\}$ and region boundaries are controlled by the network weights through a complex functional dependence (Balestriero and Baraniuk, 2018; Zhang et al., 2018). Each region can be delimited by examining its activation pattern, which will remain fixed within its domain[1].

[1]We remark that the characterization of the regions through activation patterns is more aligned with activation regions (Hanin and Rolnick, 2019b; Stargalla et al., 2025) and does not account for cases where neighboring activation patterns represent the same affine map. However, such degeneracies are rare in practice.

Early work on linear regions focused on characterizing the maximum number of linear regions as measures of network expressivity (Montúfar et al., 2014; Raghu et al., 2017; Serra et al., 2018). Subsequent analyses illustrated that these counts underestimate the number of regions realized in practice by several orders of magnitude and showed that their number grows polynomially with depth, in contrast with the exponential rate predicted by the maximal estimate (Hanin and Rolnick, 2019a;b).

More recent studies have shown that region distributions are spatially heterogeneous, and that their local density depends on input distribution and task details (Novak et al., 2018; Zhang and Wu, 2020; Humayun et al., 2023; Patel and Montúfar, 2025). These studies characterize important aspects of the region partition, but none of them identifies its effects on loss scaling.

Piecewise-linear approximators for neural networks Let f be a smooth function and a region R with diameter ℓ_R and choose $c_R \in R$. The optimal linear approximator achievable by the network function $\hat{f}_R(x)$ will be given by $\hat{f}_R(x) = f(c_R) + \nabla f(c_R) \cdot (x - c_R)$, whose error is bound by

$$|f(x) - \hat{f}_R(x)| \leq \tfrac{1}{2} M_R \ell_R^2, \qquad M_R = \sup_{x \in R} \|H_f(x)\|_{\mathrm{op}}, \tag{3}$$

where $\|H_f(x)\|_{\mathrm{op}}$ is operator norm of the Hessian matrix of f. This is the classical linear interpolation rate (Ciarlet, 1978; Babuška and Aziz, 1976), which implies that the mean square error (MSE) over R will scale as ℓ_R^4. A similar argument applies if we consider a constant approximator $\hat{f}_R(x) = f(c_R)$, where the MSE will instead scale as ℓ_R^2.

This approach is applied by some studies to characterize bounds for neural network losses (Yarotsky, 2017; DeVore et al., 2021) and their scaling (Sharma and Kaplan, 2020; Bahri et al., 2024) by considering that the optimal affine setting is realized on every region and that the partition itself is uniform.

While this picture yields a simple scaling law prediction in terms of a fixed power law on the region scale, trained networks are not constrained to represent optimal affine approximators over every region or to produce a uniform partition. Indeed, Adcock and Dexter (2021) documents a substantial gap between this ideal benchmark and practical network performance. In this work, we adapt this characterization to a realistic setting where we do not assume any ideal conditions.

3 Experimental setup

3.1 Datasets

We consider Pfaffian target functions (Khovanskii, 1991; Gabrielov and Vorobjov, 2004), which are defined as polynomials of degree β whose arguments are r functions $f_1, \ldots, f_r$ that act on inputs of dimension n. The n, β and r parameters give a measure of topological complexity for functions, so working with them allow us to study the dependence of the scaling on a set of targets that cover a wide array of meaningful features. We consider two types of functions for the f_j sets:

- `gauss`: $f_j(x) = \exp\big(w_j(x)\big)$,
- `tanh`: $f_j(x) = \tanh\big(w_j(x)\big)$,

where

$$w_j(x) \;=\; x^{\mathsf{T}} Q_j \, x \;+\; v_j^{\mathsf{T}} \cdot x \;+\; w_j^0. \tag{4}$$

The Q_j, v_j and w_j^0 parameters are drawn randomly for each dataset. We set Q_j to be a symmetric matrix, and keep it negative definite for the `gauss` datasets. The choice of these two function types is motivated by their differing derivative structures, which will allow us to probe how the scaling changes with different gradient and curvature patterns. We consider $n \in \{3, 4, 5\}$ for the input dimension and $r \in \{3, 4, 5\}$ for the chain length while keeping $\beta = 3$ fixed. Evaluating every combination for both types amounts to 18 datasets.

Inputs for all datasets are sampled uniformly over a $[-15, 15]^n$ range that is then rescaled to a unit box, and the outputs are standardized to zero mean and unit variance. Every dataset consists of 10^6 data points, split $70/10/20$ into training, validation and test sets.

3.2 NETWORKS

We train fully connected neural networks with 3 hidden layers and ReLU activation functions, trained with Adam and a cosine learning rate scheduler. Widths are chosen to cover 18 network sizes spanning approximately 5×10^3 to 2×10^5 parameters. We use maximal-update parametrization (μP) (Yang and Hu, 2021; Yang et al., 2021), which ensures that the optimal learning rate transfers across widths. The learning rate and batch size are tuned separately for each dataset, and no weight decay is applied. We train every network on an MSE loss using 4 independent initialization seeds over 100 epochs for the `gauss` datasets and 200 epochs for the `tanh` datasets. We evaluate the network on the validation set after every epoch and keep the weights that produce the best validation loss. Training $10\times$ longer or on $10\times$ more data does not improve the test loss, so we consider the networks converged.

3.3 METHODS

Loss decomposition We study the network scaling by decomposing the loss into terms that quantify the gaps of the region affine parameters with respect to their optimal values and the error that remains under an ideal linear approximation. We accomplish this by fitting a linear function $\Delta_0 + \Delta A \cdot (x - c)$ to the network error within each region, resulting in

$$\mathcal{E}_i = \Delta_0 + \Delta A \cdot (x_i - c) + r_i. \tag{5}$$

The resulting Δ_0 will be the offset of the affine map constant $A_R c + b_R$ with respect to the average of the target function over the region, whereas the fitted ΔA will be the average gap between A_R and the slope of the optimal affine approximation to the target g. If we subtract this affine map at a point x_i from $\mathcal{E}_i$ we will get a remainder r_i, which represents the part of the error that cannot be explained by an affine approximation. The full expressions for Δ_0 and ΔA after the fit are:

$$\Delta_0 \;=\; \langle f \rangle_R - \hat{f}_R(c), \tag{6}$$

$$\Delta A \;=\; C_R^{+} \big\langle (x - c)\, f \big\rangle_R - A_R \;\equiv\; g - A_R, \qquad C_R \;=\; \frac{1}{m} \sum_{i=1}^{m} (x_i - c)(x_i - c)^{\mathsf{T}}, \tag{7}$$

where C_R is the covariance of the sampled points, C_R^{+} is its pseudoinverse, and m is the number of data points within the region. The key advantage of this decomposition is that it allows us to keep the same three-term structure with no cross-terms after computing the MSE over a region if we set $c = \langle x \rangle_R$, since $\langle x - \langle x \rangle_R \rangle_R = 0$, $\langle r_R \rangle_R = 0$ and $\langle (x - \langle x \rangle_R) r_R \rangle_R = 0$. Then, if we square and average, we obtain

$$\langle e^2 \rangle_R \;=\; \underbrace{\Delta_0^2}_{\text{offset error}} \;+\; \underbrace{\Delta A^{\mathsf{T}} C_R \Delta A}_{\text{slope error}} \;+\; \underbrace{\langle r^2 \rangle_R}_{\text{non-affine remainder}}, \tag{8}$$

Since the regions partition the input space, the sum of these averages over all regions reproduces the total MSE once each region is weighted by its volume in input space. We compute region averages and centroids by sampling 5000 points over the region containing each test point.

Working with this decomposition, an ideal piecewise linear approximator is achieved only if the network matches the optimal affine approximation over all regions and thus the first two terms in Eq. (8) vanish.

Region size computation Since the ideal characterization in Section 2 is expressed in terms of the region diameter ℓ_R, we need a comparable measure for trained networks. We use the average radius s_R from the region centroid:

$$s_R = \frac{1}{K_s} \sum_{j=1}^{K_s} d_R(\bar{x}_R, u_j), \tag{9}$$

where the directions u_j are sampled isotropically, the distances d_R are measured from the region centroid until the first point where its activation pattern changes, and $K_s = 48$. The variable s_R, which we call region size, provides a measure of the typical radial extent of the region that is robust against anisotropies. At the network level, we cannot use the mean of s_R since empirical observations show that the region distribution contains abnormally large regions which exceed the typical region

size by orders of magnitude while carrying a very small fraction of the error. In order to minimize this effect, we use the median $\mathrm{med}\, s_R$ instead as the scaling variable for the loss.

It is intractable to account for the whole region distribution when computing the median, since even moderately sized networks can contain millions of regions (Balestriero and Baraniuk, 2018). However, we observe that the standard deviation of the median s_R estimates for all our trained networks is below 5% with only 1000 samples. We use 5000 regions for all our partition computations.

Scaling law fits We fit the loss to the scaling law ansatz $y = Cx^{\alpha}$, which reproduces the functional form of Eq. 1 with $\mathcal{E}_\infty \to 0$. We remove the asymptotic floor parameter by working in the infinite data limit, where the loss is expected to evolve as a pure power law with the network size (Bahri et al., 2024). This is done by fitting a separate model with an added floor parameter and iteratively removing network sizes from the upper end of the range until the exponent estimates from both models match. This approach always retains at least 70% of the networks from the original range. Working in a regime where the asymptotic floor is removed provides more stable estimates for the scaling exponent α despite the reduced number of points available for the fit since we do not need to identify whether the fit has a floor. We execute the fit on a combined data pool with all four seeds, and we check that their correlation does not affect the α estimates. Uncertainties on α are computed as the half-width of the 95% confidence interval of the least-squares slope in log-log space.

4 Results

4.1 Deviation from the ideal regime

We start our analysis by measuring the magnitude of the offset error Δ_0^2, the slope error $\Delta A^\mathsf{T} C_R \Delta A$ and the non-affine remainder $\langle r^2 \rangle_R$ in all `gauss` and `tanh` datasets. Table 1 shows the average percentage of the total loss each term in Eq. (8) holds for the smallest and largest networks in the adjusted range described in Section 3.3. The offset error Δ_0^2 holds the largest share of the loss and improves at a slower rate than the other two terms. This trend is confirmed in Table 2, where we report the fitted exponents of each term and the global loss as a function of $\mathrm{med}\, s_R$. Furthermore, the non-affine remainder, which should scale as ℓ_R^4, produces $\alpha \approx 3.2 - 3.5$. Given its dominance over the loss, we focus our analysis on describing Δ_0^2 and leave the study of the slope error and the non-affine remainder for future work.

			Δ_0^2		$\Delta A^\mathsf{T} C_R \Delta A$		$\langle r^2 \rangle_R$	
Type	n	r	small	large	small	large	small	large
`gauss`	3	3	0.594	0.795	0.265	0.158	0.141	0.047
`gauss`	3	5	0.588	0.800	0.265	0.156	0.147	0.044
`gauss`	5	3	0.664	0.821	0.207	0.126	0.129	0.053
`gauss`	5	5	0.746	0.824	0.184	0.144	0.070	0.032
`tanh`	3	3	0.535	0.732	0.316	0.214	0.149	0.054
`tanh`	3	5	0.600	0.755	0.302	0.206	0.098	0.039
`tanh`	5	3	0.630	0.818	0.258	0.152	0.111	0.030
`tanh`	5	5	0.698	0.924	0.231	0.070	0.070	0.007

Table 1: Fraction of the loss carried by the volume-weighted sum of each term of Eq. (8), evaluated on the smallest and the largest network of each range and averaged over the four seeds, for both target types. The full results are shown in Table 4 in Appendix A.

4.2 Offset error scaling

The offset error exponents in Table 2 are close to 2, the value expected for a constant approximation error. However, for most datasets the confidence interval excludes 2 itself. To understand this discrepancy, we examine in detail the quantities that determine the offset error.

First-order crossing description In order to parametrize Δ_0 as a constant approximation error, we need to define the point ξ_R where the target function is evaluated and the point $x_R^\times$ where

Type	n	r	Δ_0^2	$\Delta A^\mathsf{T} C_R \Delta A$	$\langle r^2 \rangle_R$	MSE
gauss	3	3	1.81 ± 0.13	2.65 ± 0.14	3.27 ± 0.15	2.11 ± 0.12
gauss	3	5	1.50 ± 0.13	2.28 ± 0.15	2.89 ± 0.18	1.77 ± 0.14
gauss	5	3	1.75 ± 0.15	2.53 ± 0.17	3.16 ± 0.13	2.00 ± 0.14
gauss	5	5	1.7 ± 0.4	2.5 ± 0.3	3.3 ± 0.3	1.9 ± 0.3
tanh	3	3	1.61 ± 0.11	2.26 ± 0.12	2.86 ± 0.17	1.90 ± 0.10
tanh	3	5	2.74 ± 0.18	3.33 ± 0.16	3.76 ± 0.15	2.95 ± 0.17
tanh	5	3	1.59 ± 0.11	2.49 ± 0.11	3.24 ± 0.13	1.88 ± 0.10
tanh	5	5	1.52 ± 0.09	2.51 ± 0.08	3.30 ± 0.09	1.71 ± 0.09

Table 2: Measured scaling exponents of the average offset error Δ_0^2, the slope error $\Delta A^\mathsf{T} C_R \Delta A$ and the non-affine remainder $\langle r^2 \rangle_R$ terms of Eq. (8) and of the total loss against the median region size, for both target types. The full results are shown in Table 5 in Appendix A.

$f(x) = \hat{f}_R(\bar{x}_R)$, which we call crossing point. For each region, we define

$$\xi_R = \arg \min_{x \in R:\, f(x) = \langle f \rangle_R} \|x - \langle x \rangle_R\|, \tag{10}$$

as the point on the target level set $f(x) = \langle f \rangle_R$ closest to the regional centroid, which we estimate through a projected descent along the level set. Using ξ_R as a starting point, we estimate $x_R^\times$ by searching over 1024 random directions and 96 radial lengths. Once a sign change is found, the exact location of the crossing is determined through a linear interpolation between the bracketing radial points. $x_R^\times$ is chosen as the closest crossing point over all directions.

By construction, $\Delta_{0,R} = f(\xi_R) - f(x_R^\times)$. We can then expand the target function about ξ_R to obtain

$$\Delta_{0,R} = \nabla f(\xi_R) \cdot \delta_{0,R} + \mathcal{O}(|\delta_{0,R}|^2) \tag{11}$$

where $\delta_{0,R} = \xi_R - x_R^\times$, whose norm we call the crossing distance.

We test whether this first-order expansion is enough to describe the average offset error by computing $\nabla f(\xi_R) \cdot \delta_0$ directly and comparing it to $\Delta_{0,R}$. We perform this test on samples of 5000 regions, on which we cut the ones where the higher-order corrections are large relative to the first-order term. We apply this cut using the ratio

$$R_0 \;=\; \frac{\left|\delta_{0,R}^\top \nabla^2 f(\xi_R)\, \delta_{0,R}\right|}{2\left|\nabla f(\xi_R) \cdot \delta_{0,R}\right|}. \tag{12}$$

Demanding $R_0 < 0.5$ removes under 5% of the total error magnitude for every tested network. With this cut in place, the left panel of Fig. 1 shows a comparison of the offset error per region against the first order approximation on a network with 50000 parameters trained on a representative dataset. Along with this measurement, we also measure the $|\nabla f(\xi_R)||\delta_{0,R}|$ factorization and the simplified expression $|\nabla f(\xi_R)| s_R$.

These results show that the offset error is accurately reproduced by the first-order prediction and by the proposed factorization. This indicates that $\delta_{0,R}$ is approximately aligned with the direction of the gradient. In contrast, there is a clear disconnect between $\delta_{0,R}$ and s_R, which implies that the offset error does not behave as a constant approximation error on s_R.

From region size to crossing distance In order to understand why $\delta_{0,R}$ does not track s_R, we study three separate effects: the $|\langle x \rangle_R - \xi_R|$ displacement, the $|\langle x \rangle_R - x_R^\times|$ displacement, and the region radius along the crossing direction $|\langle x \rangle_R - \tau_R|$, where τ_R is the point where a ray from $\langle x \rangle_R$ in the direction of $\delta_{0,R}$ leaves the region.

We fit the medians of each of these distances against $\operatorname{med} s$, and show the measured exponents in Table 3. We observe a significant overlap between the $\operatorname{med} |\langle x \rangle_R - x_R^\times|$ and $\operatorname{med} |\langle x \rangle_R - \tau_R|$ exponents, suggesting that much of the crossing-point scaling is due to the region crossing axes evolving at a lower rate than s_R. By contrast, ξ_R falls much faster, which is explained by considering that $\xi_R = \langle x \rangle_R$ for linear target functions. The displacement is then set by the target function curvature of f and its scaling is of order $\mathcal{O}(s_R^2)$.

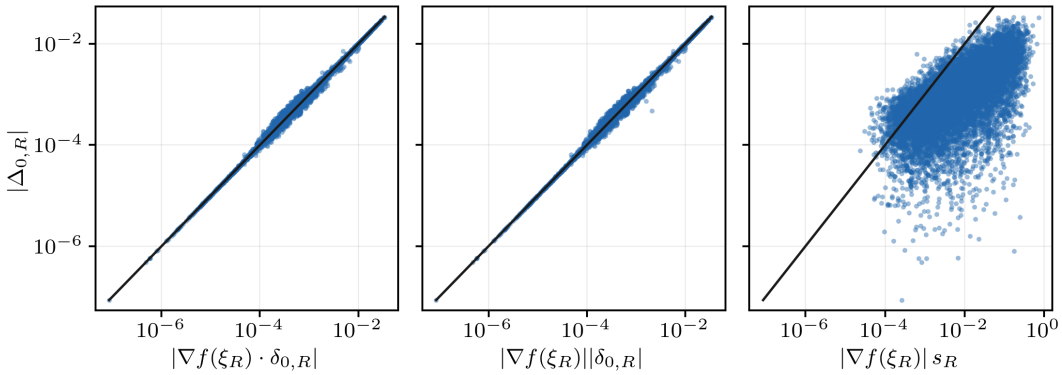


Figure 1: Measured offset $|\Delta_{0,R}|$ against $|\nabla f(\xi_R) \cdot \delta_{0,R}|$ $|\nabla f(\xi_R)||\delta_{0,R}|$, and $|\nabla f(\xi_R)| s_R$, for a network with 50000 parameters trained on the `gauss` dataset with $n = 3$, $r = 3$ after applying the $R_0 < 0.5$ cut.

Type	n	r	med $\lvert\langle x\rangle_R - \xi_R\rvert$	med $\lvert\langle x\rangle_R - x_R^\times\rvert$	med $\lvert\langle x\rangle_R - \tau_R\rvert$	med $\lvert\delta_{0,R}\rvert$
`gauss`	3	3	1.74 ± 0.11	1.04 ± 0.07	0.96 ± 0.03	0.89 ± 0.08
`gauss`	3	5	1.76 ± 0.08	0.93 ± 0.08	0.97 ± 0.02	0.82 ± 0.08
`gauss`	5	3	1.47 ± 0.14	1.01 ± 0.08	0.89 ± 0.06	0.87 ± 0.09
`gauss`	5	5	1.59 ± 0.09	0.74 ± 0.13	0.92 ± 0.05	0.67 ± 0.15
`tanh`	3	3	1.85 ± 0.07	1.07 ± 0.07	0.97 ± 0.04	0.69 ± 0.07
`tanh`	3	5	1.80 ± 0.03	1.28 ± 0.05	0.96 ± 0.02	1.00 ± 0.05
`tanh`	5	3	1.76 ± 0.04	1.00 ± 0.05	0.94 ± 0.04	0.51 ± 0.05
`tanh`	5	5	1.81 ± 0.03	0.77 ± 0.03	0.910 ± 0.010	0.49 ± 0.03

Table 3: Scaling exponents against med s of the median distances from the centroid $\langle x\rangle_R$ to ξ_R, $x_R^\times$ and τ_R, and of the median $|\delta_{0,R}|$. The full results are shown in Table 6 in Appendix A.

Taken together, these measurements explain why $|\delta_{0,R}|$ does not scale proportionally to s_R. We can see this by considering that ξ_R is often located between $\langle x\rangle_R$ and $x_R^\times$ along a line, which allows us to approximate $|\delta_{0,R}| \approx |\langle x\rangle_R - x_R^\times| - |\langle x\rangle_R - \xi_R|$. This approximation holds within 10% for every dataset once we substitute all measured distances. This reconstruction is incomplete, but it shows that the crossing-distance scaling is affected by both target curvature and anisotropic region reduction.

Scaling in terms of the crossing distance Describing the scaling in terms of med $|\delta_{0,R}|$ still does not describe $\langle\Delta_0^2\rangle$ as a constant approximation error, as we show in Table 7. The reason is that the $|\delta_{0,R}|$ and $\nabla f(\xi_R)$ distributions are strongly anticorrelated: the median Pearson correlation between their logarithms is on average -0.677 for `gauss` targets and -0.567 for `tanh` targets. The full correlations for all datasets are shown in Table 8 in Appendix A.

In order to understand the effect of this correlation on the scaling, we separate the regions into 5 gradient bins defined by the 60th, 80th, 90th, and 95th percentiles of $|\nabla f(\xi_R)|$ for every dataset. This split will allow us to measure how the offset error is distributed across each gradient bin and how the scaling changes on an environment with a reduced $|\nabla f(\xi_R)|$ range. For these tests we increase the region count to 10000 before the R_0 cut to ensure that there is enough statistics on each gradient bin.

We show the shares of the $\Delta_{0,R}^2$, $|\delta_{0,R}|^2$ and $|\nabla f(\xi_R)|^2$ distributions over each bin in Tables 9-11 in Appendix A. Together, they show that the `gauss` error is concentrated in the highest gradient bins, while the `tanh` error mostly occupies the largest three bins. This distribution broadly matches the pattern for the gradient shares and strongly opposes the crossing distance, whose largest values are concentrated in the lowest gradient bins.

Within each gradient bin k, we expect that $\langle\Delta_0^2\rangle_k$ will scale quadratically with $\mathrm{med}_k|\delta_{0,R}|$. We show that is the case in Fig. 4 and Table 12 in Appendix A, where all exponents range from 1.85 to 2.14. This result heavily deviates on the lowest gradient bins, which hold a negligible part of the total error.

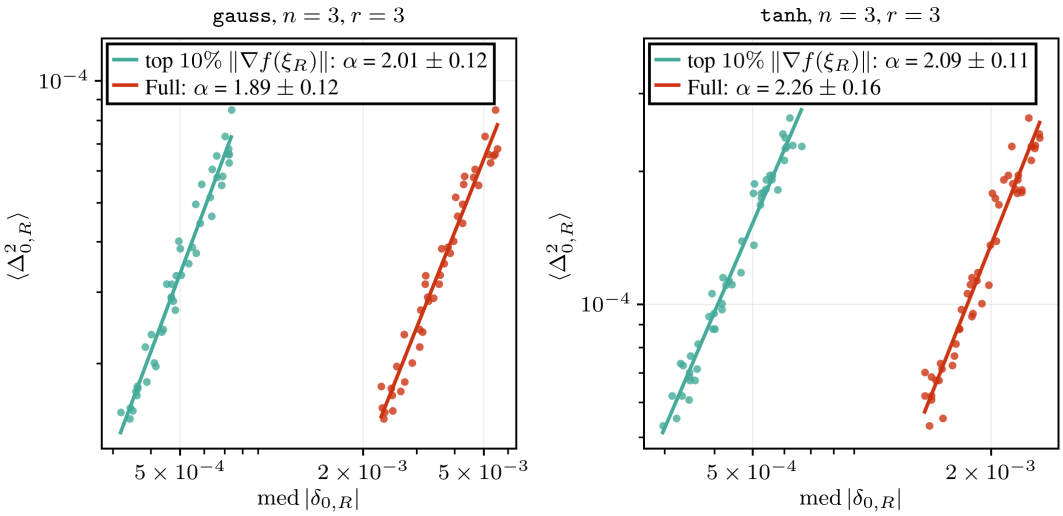


Figure 2: Offset error $\langle \Delta^2_{0,R} \rangle$ against the median crossing distance $\mathrm{med}\,|\delta_{0,R}|$ taken over the highest decile of $\|\nabla f(\xi_R)\|$ and over all regions, for one `gauss` and one `tanh` dataset.

These results suggest that the full $\langle \Delta^2_0 \rangle$ can be described as a constant approximation error with the crossing distance averaged over the high gradient domain of the target function as its natural scale instead of the median. Indeed, we show in Table 13 of Appendix A and in Fig. 2 that the scaling of $\langle \Delta^2_0 \rangle$ is approximately quadratic with respect to the median $|\delta_{0,R}|$ over the 10% highest $|\nabla f(\xi_R)|$, with the average `gauss` exponent being 1.97 ± 0.13 and the average `tanh` exponent being 2.20 ± 0.12.

These observations also imply that the crossing distance distribution must generally scale non-uniformly. We check this for both $|\delta_{0,R}|$ and s_R in Tables 14 and 15 in Appendix A, where we measure the scaling of $\mathrm{med}|\delta_{0,R}|$ against each $\mathrm{med}_k|\delta_{0,R}|$. The results show that all s_R and the `gauss` $|\delta_{0,R}|$ scale at approximately the same rate, but the `tanh` $|\delta_{0,R}|$ exponents range from -0.2 to 1.5. This demonstrates that all s_R shrink at the same rate, but that uniformity does not generally propagate to $|\delta_{0,R}|$.

The heterogeneous crossing distance scaling and the gradient bin decomposition provides a full account for the scaling of $\langle \Delta^2_0 \rangle$ against $\mathrm{med}\,|\delta_{0,R}|$. We demonstrate this by replacing each term in the decomposition across gradient bins by the two power laws we have measured for it:

$$\langle \Delta^2_0 \rangle = \sum_k \pi_k \langle \Delta^2_0 \rangle_k, \qquad \langle \Delta^2_0 \rangle_k = C_k (\mathrm{med}_k\,|\delta_{0,R}|)^2 \approx C_k A_k^2 (\mathrm{med}\,|\delta_{0,R}|)^{2\gamma_k}, \tag{13}$$

where π_k is the fraction of regions in bin k, γ_k is measured in Table 14, and we refit C_k after fixing the $\mathrm{med}_k\,|\delta_{0,R}|$ exponent to 2. Holding π_k at its mean over the network range, this prediction differs from the measured $\langle \Delta^2_0 \rangle$ by at most 12% on any one of our datasets.

4.3 Relation between the number of parameters and the region size

We have identified a structure that relates network size scaling of neural networks to linear region geometry. However, this structure cannot provide a complete prediction for the network size scaling unless there exists a relation between $\mathrm{med}\,s_R$ and the number of parameters in the neural network. There are only two approaches in the literature that attempt to model this dependence. Sharma and Kaplan (2020) and Bahri et al. (2024) propose $s \propto N^{-1/n}$, whereas Hanin and Rolnick (2019a) and Goujon et al. (2024) instead predict $s \propto N^{-1/2}$ for fixed depth networks.

We show in Fig. 3 a comparison of the measured dependence between these two variables and both of these predictions for two of our datasets with different n. We observe that none of the proposed models predict the correct dependence across dimensions. If we apply our fitting method to all datasets, we find an average exponent of -0.37 ± 0.02 for the `gauss` datasets and of -0.39 ± 0.03

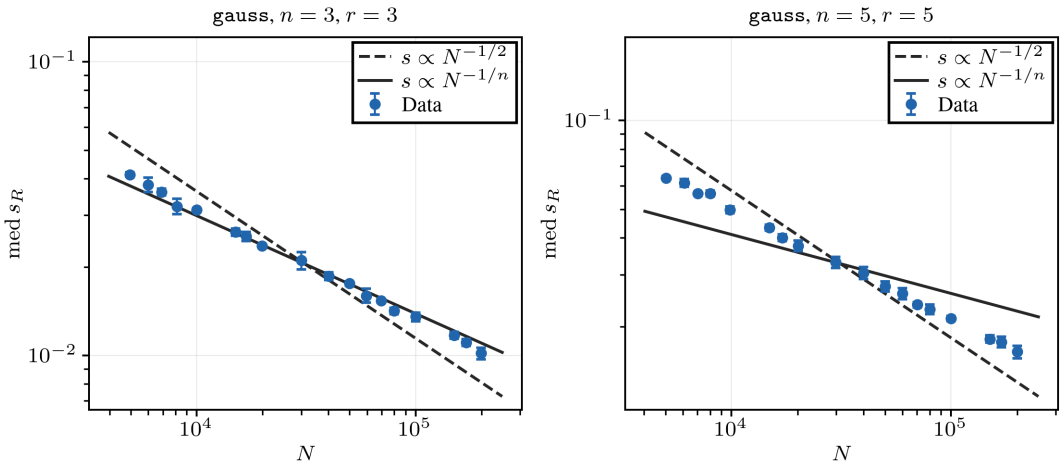


Figure 3: Median region size $\mathrm{med}\, s_R$ against the number of parameters N for two `gauss` datasets, averaged over the four seeds, with error bars given by the seed standard deviation. We compare the observed dependence with two models with predicted exponents $-1/2$ and $-1/n$.

for the `tanh` datasets, which does not depend on n or r. Therefore, further studies need to be made to understand the relation between the number of parameters and the region size.

5 Conclusion and Limitations

We have studied the model-size scaling of trained ReLU networks by analyzing their linear region partition. By decomposing the regional MSE into the sum of an offset error, a slope error and a non-affine remainder, we find that trained networks do not realize the ideal piecewise-linear approximation regime commonly assumed in descriptions of neural scaling. Instead, the loss is dominated by deviations of the regional affine maps from their optimal values, with the offset error being the dominant contribution in both its magnitude and scaling.

We have characterized this offset error as a first-order variation of the target over a characteristic scale called crossing distance. We have shown that the crossing distance does not scale proportionally with the region size, and that different parts of its distribution can evolve at different rates despite the region sizes scaling uniformly. We have also identified that the offset error concentrates on regions located at the higher end of the target gradient distribution, and that it represents a constant approximation error with the crossing distance if it is evaluated within a limited gradient range. As a consequence, we can produce a quadratic parametrization of the scaling in terms of the median of the crossing distance over a set of high-gradient regions. These results provide a new approach to quantify the neural scaling in terms of measurable properties of the learned piecewise-affine structure of the network and simple features of the target function.

However, our analysis leaves several important aspects of the scaling unresolved. Our experiments are restricted to ReLU networks trained on low-dimensional analytic regression problems, and it remains to be established whether the same mechanisms apply to larger architectures, higher-dimensional data, and other learning tasks. Even within our setting, we do not identify a mechanism for the different scaling rates of the crossing distance distribution or the relation between the crossing distance and the region size. Additionally, we have not been able to identify any feature from our datasets that affects the scaling despite leveraging a strict notion of functional complexity when building them. Finally, while our results describe the structure of the scaling in relation to the region partition, a predictive theory relating the region scale to the number of network parameters is still missing. Establishing this relation and understanding how training determines the learned network function are necessary steps towards building a fully predictive theory of neural scaling.

AI USE STATEMENT

In this work, we used large language model (Claude Opus 5 and ChatGPT 5.6) to obtain critical feedback on the manuscript in the form of simulated peer reviews, to suggest rewordings of individual sentences in the main text, to check the existence and bibliographic details of related-work references, and to verify the page count against the ICLR formatting requirements. We also used them to implement methods, write and edit analysis code, and to design the plot formats. We did not use generative AI tools to generate synthetic data, formulate hypotheses or mathematical claims, design the experiments, or interpret results. All suggestions were reviewed and, where adopted, rewritten by the authors, and all numerical statements were checked against our own results tables. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The target function families, sampling ranges, normalization and train/validation/test split are specified in Section 3. Network architectures, optimizer, training budget, validation routines selection and convergence checks are described in Section 3.2. The loss decomposition, region sampling, region size estimator, crossing-point construction, the R_0 cut and the scaling fit procedure are given in Section 3.3 and Section 4. Complete per-dataset results for every table in the main text are reported in Appendix A. Code will be released upon publication.

REFERENCES

Ben Adcock and Nick Dexter. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):624–655, 2021. doi:10.1137/20M131309X.

Alexander Atanasov, Jacob A. Zavatone-Veth, and Cengiz Pehlevan. Scaling and renormalization in high-dimensional regression. *Journal of Statistical Mechanics: Theory and Experiment*, 2026(4):043404, 2026. doi:10.1088/1742-5468/ae4bba.

Ivo Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976. doi:10.1137/0713021.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024. doi:10.1073/pnas.2311878121.

Randall Balestriero and Richard G. Baraniuk. A spline theory of deep networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 374–383. PMLR, 2018.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. How feature learning can improve neural scaling laws. In *International Conference on Learning Representations (ICLR)*, 2025.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1024–1034. PMLR, 2020.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. In *International Conference on Learning Representations (ICLR)*, 2023.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12:2914, 2021. doi:10.1038/s41467-021-23103-1.

Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and Its Applications*. North-Holland, Amsterdam, 1978.

Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021. doi:10.1017/S0962492921000052.

Andrei Gabrielov and Nicolai Vorobjov. Complexity of computations with Pfaffian and Noetherian functions. In *Normal Forms, Bifurcations and Finiteness Problems in Differential Equations*, volume 137 of *NATO Science Series II*. Kluwer, Dordrecht, 2004.

Alexis Goujon, Arian Etemadi, and Michael Unser. On the number of regions of piecewise linear neural networks. *Journal of Computational and Applied Mathematics*, 441:115667, 2024. doi:10.1016/j.cam.2023.115667.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604. PMLR, 2019a.

Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, volume 32, 2019b.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard G. Baraniuk. Training dynamics of deep network linear regions. *arXiv preprint arXiv:2310.12977*, 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.

A. G. Khovanskii. *Fewnomials*, volume 88 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 1991.

Alexander Maloney, Daniel A. Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

Guido F. Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 27, pages 2924–2932, 2014.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: An empirical study. In *International Conference on Learning Representations (ICLR)*, 2018.

Niket Patel and Guido Montúfar. On the local complexity of linear regions in deep ReLU networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. PMLR, 2017.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations (ICLR)*, 2020.

Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4558–4566. PMLR, 2018.

Utkarsh Sharma and Jared Kaplan. A neural scaling law from the dimension of the data manifold. *arXiv preprint arXiv:2004.10802*, 2020.

Moritz Stargalla, Christoph Hertrich, and Daniel Reichman. The computational complexity of counting linear regions in ReLU neural networks. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Greg Yang and Edward J. Hu. Tensor Programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11727–11737. PMLR, 2021.

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.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017. doi:10.1016/j.neunet.2017.07.002.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, 2022.

Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5824–5832. PMLR, 2018.

Xiao Zhang and Dongrui Wu. Empirical studies on the properties of linear regions in deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

A COMPLEMENTARY RESULTS

This appendix reports the complete results for all tests and measurements described in Section 4.

			Δ_0^2		$\Delta A^\mathsf{T} C_R \Delta A$		$\langle r^2 \rangle_R$	
Type	n	r	small	large	small	large	small	large
gauss	3	3	0.594	0.795	0.265	0.158	0.141	0.047
gauss	3	4	0.629	0.824	0.262	0.148	0.109	0.028
gauss	3	5	0.588	0.800	0.265	0.156	0.147	0.044
gauss	4	3	0.681	0.787	0.206	0.152	0.113	0.061
gauss	4	4	0.666	0.796	0.223	0.158	0.111	0.046
gauss	4	5	0.680	0.844	0.228	0.128	0.092	0.028
gauss	5	3	0.664	0.821	0.207	0.126	0.129	0.053
gauss	5	4	0.715	0.857	0.192	0.111	0.092	0.032
gauss	5	5	0.746	0.824	0.184	0.144	0.070	0.032
tanh	3	3	0.535	0.732	0.316	0.214	0.149	0.054
tanh	3	4	0.560	0.667	0.305	0.266	0.135	0.068
tanh	3	5	0.600	0.755	0.302	0.206	0.098	0.039
tanh	4	3	0.606	0.822	0.286	0.150	0.108	0.028
tanh	4	4	0.646	0.869	0.267	0.116	0.086	0.016
tanh	4	5	0.650	0.848	0.257	0.132	0.092	0.020
tanh	5	3	0.630	0.818	0.258	0.152	0.111	0.030
tanh	5	4	0.677	0.894	0.246	0.096	0.077	0.010
tanh	5	5	0.698	0.924	0.231	0.070	0.070	0.007

Table 4: Fraction of the loss carried by the volume-weighted sum of each term of Eq. (8), evaluated on the smallest and the largest network of each range and averaged over the four seeds, for both target types.

Type	n	r	Δ_0^2	$\Delta A^\mathsf{T} C_R \Delta A$	$\langle r^2 \rangle_R$	MSE
gauss	3	3	1.81 ± 0.13	2.65 ± 0.14	3.27 ± 0.15	2.11 ± 0.12
gauss	3	4	1.57 ± 0.09	2.40 ± 0.10	3.05 ± 0.14	1.83 ± 0.08
gauss	3	5	1.50 ± 0.13	2.28 ± 0.15	2.89 ± 0.18	1.77 ± 0.14
gauss	4	3	1.84 ± 0.20	2.62 ± 0.17	3.25 ± 0.21	2.10 ± 0.18
gauss	4	4	1.83 ± 0.16	2.63 ± 0.18	3.33 ± 0.23	2.09 ± 0.16
gauss	4	5	1.73 ± 0.15	2.65 ± 0.13	3.35 ± 0.14	1.98 ± 0.13
gauss	5	3	1.75 ± 0.15	2.53 ± 0.17	3.16 ± 0.13	2.00 ± 0.14
gauss	5	4	1.72 ± 0.15	2.58 ± 0.17	3.22 ± 0.16	1.93 ± 0.15
gauss	5	5	1.7 ± 0.4	2.5 ± 0.3	3.3 ± 0.3	1.9 ± 0.3
gauss	Mean		$\mathbf{1.71 \pm 0.12}$	$\mathbf{2.53 \pm 0.13}$	$\mathbf{3.20 \pm 0.15}$	$\mathbf{1.97 \pm 0.12}$
tanh	3	3	1.61 ± 0.11	2.26 ± 0.12	2.86 ± 0.17	1.90 ± 0.10
tanh	3	4	3.0 ± 0.4	3.5 ± 0.4	4.2 ± 0.5	3.3 ± 0.4
tanh	3	5	2.74 ± 0.18	3.33 ± 0.16	3.76 ± 0.15	2.95 ± 0.17
tanh	4	3	1.63 ± 0.09	2.51 ± 0.07	3.17 ± 0.10	1.90 ± 0.08
tanh	4	4	1.71 ± 0.09	2.66 ± 0.09	3.33 ± 0.10	1.96 ± 0.09
tanh	4	5	2.22 ± 0.14	3.09 ± 0.13	3.86 ± 0.15	2.46 ± 0.13
tanh	5	3	1.59 ± 0.11	2.49 ± 0.11	3.24 ± 0.13	1.88 ± 0.10
tanh	5	4	1.57 ± 0.08	2.55 ± 0.08	3.44 ± 0.09	1.79 ± 0.08
tanh	5	5	1.52 ± 0.09	2.51 ± 0.08	3.30 ± 0.09	1.71 ± 0.09
tanh	Mean		$\mathbf{2.0 \pm 0.6}$	$\mathbf{2.8 \pm 0.4}$	$\mathbf{3.5 \pm 0.4}$	$\mathbf{2.2 \pm 0.6}$

Table 5: Measured scaling exponents of the average offset error Δ_0^2, the slope error $\Delta A^\mathsf{T} C_R \Delta A$ and the non-affine remainder $\langle r^2 \rangle_R$ terms of Eq. (8) and of the total loss against the median region size, for both target types.

Type	n	r	$\mathrm{med}\,\lvert\langle x\rangle_R - \xi_R\rvert$	$\mathrm{med}\,\lvert\langle x\rangle_R - x_R^{\times}\rvert$	$\mathrm{med}\,\lvert\langle x\rangle_R - \tau_R\rvert$	$\mathrm{med}\,\delta_{0,R}$
gauss	3	3	1.74 ± 0.11	1.04 ± 0.07	0.96 ± 0.03	0.89 ± 0.08
gauss	3	4	1.70 ± 0.04	1.02 ± 0.04	0.94 ± 0.02	0.91 ± 0.04
gauss	3	5	1.76 ± 0.08	0.93 ± 0.08	0.97 ± 0.02	0.82 ± 0.08
gauss	4	3	1.64 ± 0.08	0.91 ± 0.09	0.95 ± 0.06	0.81 ± 0.10
gauss	4	4	1.75 ± 0.06	0.98 ± 0.07	0.98 ± 0.04	0.88 ± 0.08
gauss	4	5	1.70 ± 0.06	0.98 ± 0.06	0.97 ± 0.04	0.90 ± 0.06
gauss	5	3	1.47 ± 0.14	1.01 ± 0.08	0.89 ± 0.06	0.87 ± 0.09
gauss	5	4	1.60 ± 0.11	0.98 ± 0.07	0.91 ± 0.05	0.87 ± 0.07
gauss	5	5	1.59 ± 0.09	0.74 ± 0.13	0.92 ± 0.05	0.67 ± 0.15
gauss	Mean		$\mathbf{1.66 \pm 0.10}$	$\mathbf{0.95 \pm 0.09}$	$\mathbf{0.94 \pm 0.03}$	$\mathbf{0.85 \pm 0.07}$
tanh	3	3	1.85 ± 0.07	1.07 ± 0.07	0.97 ± 0.04	0.69 ± 0.07
tanh	3	4	1.81 ± 0.10	1.42 ± 0.14	0.98 ± 0.05	1.05 ± 0.15
tanh	3	5	1.80 ± 0.03	1.28 ± 0.05	0.96 ± 0.02	1.00 ± 0.05
tanh	4	3	1.75 ± 0.05	0.94 ± 0.04	0.91 ± 0.03	0.53 ± 0.04
tanh	4	4	1.82 ± 0.04	0.94 ± 0.04	0.94 ± 0.02	0.50 ± 0.04
tanh	4	5	1.90 ± 0.03	0.99 ± 0.04	0.95 ± 0.02	0.63 ± 0.05
tanh	5	3	1.76 ± 0.04	1.00 ± 0.05	0.94 ± 0.04	0.51 ± 0.05
tanh	5	4	1.87 ± 0.03	0.82 ± 0.03	0.94 ± 0.02	0.46 ± 0.03
tanh	5	5	1.81 ± 0.03	0.77 ± 0.03	0.910 ± 0.010	0.49 ± 0.03
tanh	Mean		$\mathbf{1.82 \pm 0.05}$	$\mathbf{1.03 \pm 0.21}$	$\mathbf{0.94 \pm 0.02}$	$\mathbf{0.65 \pm 0.22}$

Table 6: Scaling exponents against $\mathrm{med}\, s$ of the median distances from the centroid $\langle x\rangle_R$ to ξ_R, $x_R^{\times}$ and τ_R, and of the median $|\delta_{0,R}|$. Medians are over regions with $R_0 < 0.5$, a quoted with the standard deviation across datasets.

Type	n	r	$\langle \Delta_{0,R}^2 \rangle$ vs $\mathrm{med}\,\delta_{0,R}$
gauss	3	3	1.95 ± 0.15
gauss	3	4	1.72 ± 0.09
gauss	3	5	1.81 ± 0.14
gauss	4	3	2.01 ± 0.20
gauss	4	4	1.90 ± 0.18
gauss	4	5	2.02 ± 0.15
gauss	5	3	1.80 ± 0.18
gauss	5	4	1.94 ± 0.17
gauss	5	5	1.8 ± 0.5
gauss	Mean		$\mathbf{1.88 \pm 0.11}$
tanh	3	3	2.25 ± 0.20
tanh	3	4	2.85 ± 0.20
tanh	3	5	2.72 ± 0.09
tanh	4	3	2.88 ± 0.18
tanh	4	4	3.18 ± 0.16
tanh	4	5	3.18 ± 0.14
tanh	5	3	2.80 ± 0.23
tanh	5	4	3.11 ± 0.13
tanh	5	5	2.90 ± 0.11
tanh	Mean		$\mathbf{2.9 \pm 0.3}$

Table 7: Measured scaling exponent of the mean squared offset $\langle \Delta_{0,R}^2 \rangle$ against the median crossing distance $\mathrm{med}\,\delta_{0,R}$ for both target types.

Type	n	r	$\rho(\log \lvert\delta_{0,R}\rvert, \log \lvert\nabla f(\xi_R)\rvert)$
gauss	3	3	-0.759
gauss	3	4	-0.697
gauss	3	5	-0.736
gauss	4	3	-0.674
gauss	4	4	-0.683
gauss	4	5	-0.639
gauss	5	3	-0.649
gauss	5	4	-0.627
gauss	5	5	-0.617
gauss	Mean		$\mathbf{-0.68 \pm 0.05}$
tanh	3	3	-0.673
tanh	3	4	-0.636
tanh	3	5	-0.635
tanh	4	3	-0.568
tanh	4	4	-0.605
tanh	4	5	-0.526
tanh	5	3	-0.502
tanh	5	4	-0.494
tanh	5	5	-0.483
tanh	Mean		$\mathbf{-0.57 \pm 0.07}$

Table 8: Median Pearson correlation between the logarithms of the crossing and the target-gradient norm $|\nabla f(\xi_R)|$ for each dataset. Function type values show the mean and standard deviation across the nine datasets.

			$k=1$		$k=2$		$k=3$		$k=4$		$k=5$	
Type	n	r	small	large	small	large	small	large	small	large	small	large
gauss	3	3	0.056	0.044	0.148	0.129	0.182	0.180	0.201	0.179	0.413	0.468
gauss	3	4	0.073	0.048	0.126	0.085	0.146	0.102	0.145	0.124	0.510	0.642
gauss	3	5	0.052	0.038	0.090	0.055	0.141	0.103	0.189	0.165	0.529	0.639
gauss	4	3	0.088	0.096	0.110	0.118	0.138	0.138	0.145	0.150	0.519	0.498
gauss	4	4	0.105	0.089	0.129	0.114	0.141	0.127	0.133	0.127	0.492	0.543
gauss	4	5	0.099	0.083	0.133	0.110	0.128	0.107	0.117	0.111	0.523	0.590
gauss	5	3	0.056	0.053	0.084	0.074	0.117	0.095	0.121	0.109	0.622	0.668
gauss	5	4	0.083	0.065	0.136	0.102	0.152	0.126	0.133	0.120	0.495	0.587
gauss	5	5	0.045	0.049	0.081	0.076	0.091	0.084	0.120	0.105	0.664	0.686
tanh	3	3	0.000	0.001	0.120	0.129	0.311	0.262	0.205	0.215	0.363	0.393
tanh	3	4	0.008	0.016	0.202	0.220	0.334	0.308	0.194	0.192	0.262	0.265
tanh	3	5	0.010	0.027	0.250	0.250	0.290	0.271	0.192	0.176	0.258	0.276
tanh	4	3	0.000	0.001	0.056	0.072	0.249	0.259	0.255	0.251	0.440	0.418
tanh	4	4	0.001	0.001	0.050	0.074	0.211	0.207	0.237	0.219	0.502	0.499
tanh	4	5	0.004	0.006	0.104	0.151	0.259	0.260	0.252	0.202	0.381	0.383
tanh	5	3	0.001	0.001	0.060	0.076	0.191	0.205	0.231	0.230	0.518	0.487
tanh	5	4	0.004	0.009	0.097	0.133	0.250	0.246	0.226	0.206	0.423	0.406
tanh	5	5	0.013	0.024	0.155	0.207	0.256	0.243	0.220	0.182	0.357	0.343

Table 9: Fraction of the offset term Δ_0^2 carried by each $|\nabla f(\xi_R)|$ bin, evaluated on the smallest and largest network of each range averaged over the four seeds for both target types. Bins are fixed by the 60th, 80th, 90th, and 95th percentiles of $|\nabla f(\xi_R)$, with $k = 1$ being the flattest and $k = 5$ the steepest.

			$k=1$		$k=2$		$k=3$		$k=4$		$k=5$	
Type	n	r	small	large	small	large	small	large	small	large	small	large
gauss	3	3	0.000	0.000	0.015	0.015	0.073	0.077	0.163	0.158	0.750	0.749
gauss	3	4	0.001	0.001	0.006	0.006	0.021	0.021	0.072	0.073	0.900	0.899
gauss	3	5	0.000	0.000	0.001	0.001	0.010	0.010	0.060	0.060	0.929	0.929
gauss	4	3	0.002	0.002	0.015	0.015	0.044	0.045	0.091	0.096	0.848	0.842
gauss	4	4	0.003	0.003	0.020	0.019	0.065	0.066	0.107	0.106	0.804	0.805
gauss	4	5	0.006	0.005	0.027	0.027	0.048	0.047	0.068	0.069	0.851	0.852
gauss	5	3	0.002	0.002	0.017	0.017	0.048	0.049	0.092	0.090	0.841	0.841
gauss	5	4	0.006	0.006	0.044	0.044	0.102	0.102	0.139	0.137	0.710	0.711
gauss	5	5	0.001	0.001	0.007	0.007	0.022	0.023	0.072	0.070	0.899	0.900
tanh	3	3	0.000	0.000	0.003	0.003	0.037	0.038	0.107	0.110	0.853	0.849
tanh	3	4	0.000	0.000	0.019	0.022	0.081	0.096	0.141	0.139	0.758	0.743
tanh	3	5	0.000	0.000	0.024	0.027	0.107	0.114	0.148	0.147	0.720	0.712
tanh	4	3	0.000	0.000	0.005	0.007	0.060	0.079	0.148	0.152	0.787	0.762
tanh	4	4	0.000	0.000	0.004	0.005	0.047	0.056	0.128	0.137	0.821	0.802
tanh	4	5	0.000	0.000	0.012	0.019	0.083	0.095	0.161	0.176	0.744	0.710
tanh	5	3	0.000	0.000	0.009	0.013	0.081	0.089	0.160	0.173	0.750	0.725
tanh	5	4	0.000	0.000	0.018	0.025	0.102	0.118	0.184	0.191	0.696	0.666
tanh	5	5	0.001	0.002	0.035	0.047	0.123	0.143	0.170	0.163	0.671	0.646

Table 10: Fraction of the total squared target gradient $\sum_R |\nabla f|^2$ carried by each $|\nabla f(\xi_R)|$ bin, evaluated on the smallest and largest network of each range averaged over the four seeds for both target types. Bins are fixed by the 60th, 80th, 90th and 95th percentiles of $|\nabla f(\xi_R)|$, with $k = 1$ being the flattest and $k = 5$ the steepest, and only regions with $R_0 < 0.5$ enter.

			$k=1$		$k=2$		$k=3$		$k=4$		$k=5$	
Type	n	r	small	large	small	large	small	large	small	large	small	large
gauss	3	3	0.982	0.997	0.016	0.003	0.001	0.000	0.000	0.000	0.000	0.000
gauss	3	4	0.982	0.998	0.015	0.002	0.002	0.000	0.000	0.000	0.000	0.000
gauss	3	5	0.987	0.997	0.011	0.003	0.001	0.000	0.000	0.000	0.000	0.000
gauss	4	3	0.987	0.997	0.011	0.002	0.002	0.000	0.001	0.000	0.000	0.000
gauss	4	4	0.990	0.997	0.008	0.002	0.001	0.000	0.000	0.000	0.000	0.000
gauss	4	5	0.991	0.998	0.006	0.001	0.002	0.000	0.001	0.000	0.000	0.000
gauss	5	3	0.992	0.998	0.006	0.001	0.001	0.000	0.000	0.000	0.000	0.000
gauss	5	4	0.979	0.993	0.016	0.005	0.004	0.001	0.001	0.000	0.001	0.000
gauss	5	5	0.960	0.993	0.033	0.006	0.006	0.001	0.001	0.000	0.001	0.000
tanh	3	3	0.987	0.997	0.012	0.003	0.001	0.000	0.000	0.000	0.000	0.000
tanh	3	4	0.960	0.996	0.036	0.004	0.003	0.000	0.000	0.000	0.001	0.000
tanh	3	5	0.921	0.996	0.073	0.004	0.004	0.000	0.001	0.000	0.000	0.000
tanh	4	3	0.883	0.966	0.109	0.033	0.006	0.001	0.001	0.000	0.001	0.000
tanh	4	4	0.824	0.989	0.168	0.011	0.006	0.001	0.001	0.000	0.000	0.000
tanh	4	5	0.739	0.980	0.224	0.017	0.016	0.003	0.016	0.000	0.005	0.000
tanh	5	3	0.934	0.981	0.054	0.017	0.009	0.002	0.002	0.000	0.001	0.000
tanh	5	4	0.819	0.972	0.162	0.026	0.014	0.002	0.003	0.000	0.001	0.000
tanh	5	5	0.883	0.986	0.093	0.011	0.017	0.002	0.004	0.000	0.003	0.000

Table 11: Fraction of the total squared crossing distance $\sum_R \delta_0^2$ carried by each $|\nabla f(\xi_R)|$ bin, evaluated on the smallest and largest network of each range averaged over the four seeds for both target types. Bins are fixed by the 60th, 80th, 90th and 95th percentiles of $|\nabla f(\xi_R)|$, with $k = 1$ being the flattest and $k = 5$ the steepest, and only regions with $R_0 < 0.5$ enter.

Type	n	r	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
gauss	3	3	2.20 ± 0.13	2.07 ± 0.08	2.05 ± 0.08	1.99 ± 0.14	1.97 ± 0.11
gauss	3	4	2.17 ± 0.06	1.99 ± 0.05	2.03 ± 0.06	2.01 ± 0.08	1.82 ± 0.09
gauss	3	5	2.30 ± 0.10	2.09 ± 0.06	2.01 ± 0.06	2.06 ± 0.10	1.92 ± 0.12
gauss	4	3	2.13 ± 0.19	1.96 ± 0.10	1.92 ± 0.13	1.87 ± 0.15	2.00 ± 0.19
gauss	4	4	2.20 ± 0.08	1.96 ± 0.07	1.91 ± 0.09	1.93 ± 0.17	1.73 ± 0.15
gauss	4	5	2.19 ± 0.10	2.01 ± 0.07	2.01 ± 0.07	1.99 ± 0.10	1.98 ± 0.15
gauss	5	3	2.12 ± 0.17	1.99 ± 0.08	2.04 ± 0.08	1.90 ± 0.11	1.71 ± 0.23
gauss	5	4	2.18 ± 0.10	2.04 ± 0.06	1.98 ± 0.08	2.01 ± 0.10	1.93 ± 0.17
gauss	5	5	2.39 ± 0.24	1.96 ± 0.13	1.97 ± 0.14	1.85 ± 0.14	1.60 ± 0.21
gauss	Mean		$\mathbf{2.21 \pm 0.09}$	$\mathbf{2.01 \pm 0.05}$	$\mathbf{1.99 \pm 0.05}$	$\mathbf{1.96 \pm 0.07}$	$\mathbf{1.85 \pm 0.14}$
tanh	3	3	1.1 ± 0.7	2.53 ± 0.24	2.13 ± 0.11	2.14 ± 0.12	2.02 ± 0.12
tanh	3	4	3.0 ± 0.9	2.20 ± 0.13	1.91 ± 0.10	2.03 ± 0.15	2.10 ± 0.16
tanh	3	5	4.7 ± 0.7	2.28 ± 0.05	2.08 ± 0.04	2.12 ± 0.09	2.08 ± 0.08
tanh	4	3	0.2 ± 0.9	3.6 ± 0.4	2.04 ± 0.07	1.98 ± 0.09	2.01 ± 0.09
tanh	4	4	0.1 ± 0.9	3.7 ± 0.3	2.07 ± 0.07	1.94 ± 0.07	2.02 ± 0.08
tanh	4	5	3.0 ± 1.2	2.87 ± 0.11	2.10 ± 0.06	1.94 ± 0.07	2.04 ± 0.07
tanh	5	3	0.8 ± 1.3	2.80 ± 0.22	2.08 ± 0.12	1.98 ± 0.13	1.98 ± 0.16
tanh	5	4	3.5 ± 2.1	2.52 ± 0.10	2.07 ± 0.07	1.99 ± 0.09	1.97 ± 0.09
tanh	5	5	6.1 ± 0.8	2.33 ± 0.06	2.00 ± 0.04	1.98 ± 0.06	1.97 ± 0.07
tanh	Mean		$\mathbf{2.5 \pm 2.1}$	$\mathbf{2.8 \pm 0.6}$	$\mathbf{2.05 \pm 0.06}$	$\mathbf{2.01 \pm 0.07}$	$\mathbf{2.02 \pm 0.05}$

Table 12: Measured scaling exponents of the mean offset error per gradient bin $\langle \Delta_0^2 \rangle_k$ against each median crossing distance $\mathrm{med}_k\,|\delta_{0,R}|$ for both target types. Bins are fixed by the 60th, 80th, 90th, and 95th percentiles of $\|\nabla f(\xi_R)\|$, with $k=1$ being the flattest and $k=5$ being the steepest.

Type	n	r	Top 5%	Top 10%	Top 20%	Top 40%	Full
gauss	3	3	2.03 ± 0.11	2.01 ± 0.12	2.06 ± 0.09	1.97 ± 0.08	1.89 ± 0.12
gauss	3	4	2.01 ± 0.11	2.00 ± 0.08	1.92 ± 0.08	1.76 ± 0.08	1.75 ± 0.08
gauss	3	5	2.11 ± 0.12	2.03 ± 0.12	1.89 ± 0.12	1.73 ± 0.12	1.86 ± 0.15
gauss	4	3	1.97 ± 0.19	1.98 ± 0.13	1.99 ± 0.13	2.07 ± 0.12	2.25 ± 0.17
gauss	4	4	1.86 ± 0.14	1.88 ± 0.16	1.89 ± 0.14	1.89 ± 0.15	1.98 ± 0.18
gauss	4	5	2.10 ± 0.12	2.06 ± 0.07	2.02 ± 0.08	1.96 ± 0.09	2.06 ± 0.14
gauss	5	3	1.86 ± 0.18	1.82 ± 0.14	1.81 ± 0.12	1.81 ± 0.13	1.82 ± 0.17
gauss	5	4	2.10 ± 0.16	2.01 ± 0.11	1.88 ± 0.10	1.80 ± 0.09	1.79 ± 0.12
gauss	5	5	1.73 ± 0.19	1.77 ± 0.18	1.80 ± 0.17	1.8 ± 0.2	2.0 ± 0.4
gauss	Mean		$\mathbf{1.97 \pm 0.13}$	$\mathbf{1.95 \pm 0.10}$	$\mathbf{1.92 \pm 0.09}$	$\mathbf{1.87 \pm 0.11}$	$\mathbf{1.93 \pm 0.16}$
tanh	3	3	1.98 ± 0.13	2.09 ± 0.11	2.03 ± 0.09	2.07 ± 0.12	2.26 ± 0.16
tanh	3	4	2.4 ± 0.2	2.32 ± 0.13	2.17 ± 0.09	2.27 ± 0.09	2.9 ± 0.2
tanh	3	5	2.25 ± 0.10	2.27 ± 0.07	2.26 ± 0.04	2.26 ± 0.05	2.76 ± 0.08
tanh	4	3	2.09 ± 0.10	2.07 ± 0.07	2.09 ± 0.06	2.60 ± 0.09	2.80 ± 0.10
tanh	4	4	2.21 ± 0.10	2.11 ± 0.06	2.15 ± 0.06	2.70 ± 0.11	3.09 ± 0.13
tanh	4	5	2.27 ± 0.08	2.16 ± 0.06	2.21 ± 0.05	2.69 ± 0.08	3.13 ± 0.11
tanh	5	3	2.18 ± 0.18	2.12 ± 0.13	2.14 ± 0.09	2.55 ± 0.12	2.91 ± 0.17
tanh	5	4	2.16 ± 0.10	2.13 ± 0.07	2.17 ± 0.07	2.49 ± 0.08	3.11 ± 0.11
tanh	5	5	2.29 ± 0.09	2.19 ± 0.06	2.17 ± 0.05	2.38 ± 0.06	2.95 ± 0.09
tanh	Mean		$\mathbf{2.20 \pm 0.12}$	$\mathbf{2.16 \pm 0.08}$	$\mathbf{2.15 \pm 0.07}$	$\mathbf{2.5 \pm 0.2}$	$\mathbf{2.9 \pm 0.3}$

Table 13: Measured scaling exponents of the offset term $\langle \Delta_0^2 \rangle$ against the median crossing distance $\mathrm{med}\,|\delta_{0,R}|$ taken over a widening pool of regions, for both target types. Each pool holds the regions whose target-gradient norm $\|\nabla f(\xi_R)\|$ falls in the stated upper fraction of the frozen gradient distribution, so the pool grows from the steepest 5% of regions to the whole population. The fractions correspond to the 60th, 80th, 90th and 95th percentile bin edges used throughout.

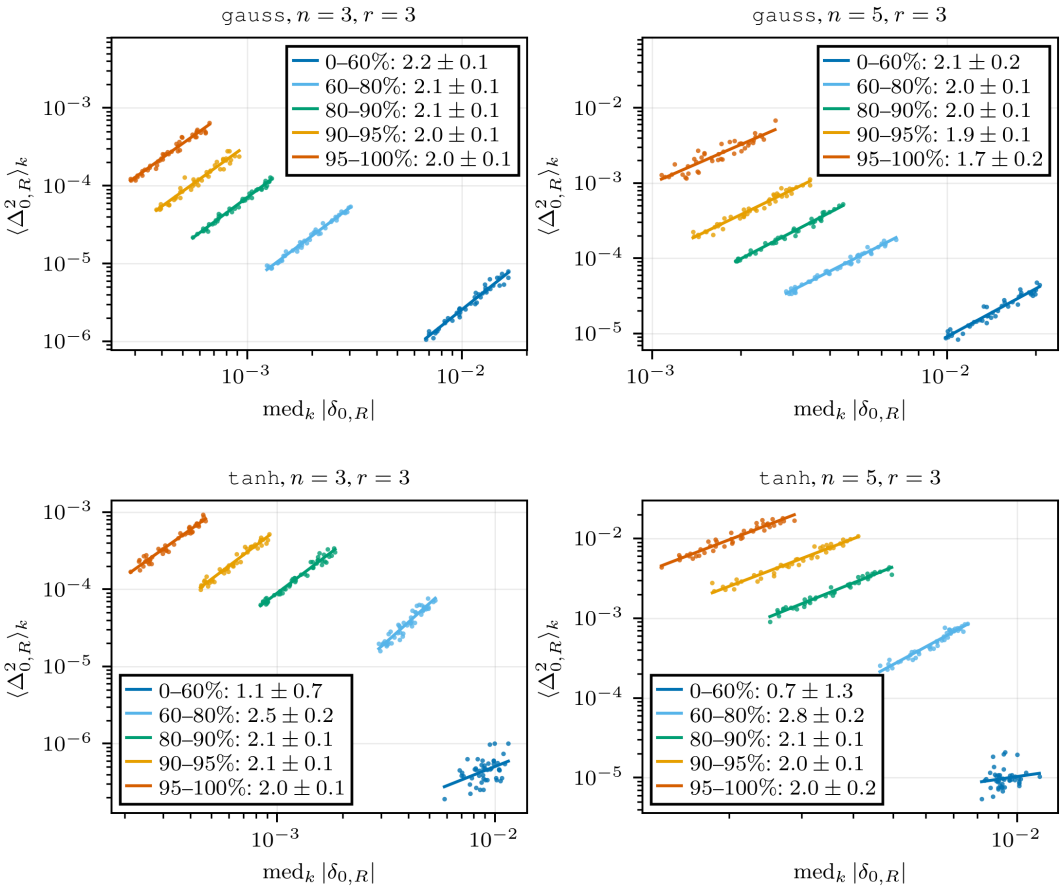


Figure 4: Scaling of $\langle \Delta_0^2 \rangle_k$ against $\mathrm{med}_k\, |\delta_{0,R}|$ for four selected `gauss` and `tanh` datasets. Regions are split into five $|\nabla f(\xi_R)|$ bins using the 60th, 80th, 90th, and 95th percentiles, with $k = 1$ being the flattest and $k = 5$ the steepest.

Type	n	r	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
gauss	3	3	0.93 ± 0.06	0.97 ± 0.05	0.92 ± 0.06	0.91 ± 0.09	0.91 ± 0.06
gauss	3	4	0.99 ± 0.03	1.03 ± 0.03	0.98 ± 0.04	0.90 ± 0.05	0.83 ± 0.06
gauss	3	5	0.94 ± 0.06	1.09 ± 0.05	1.05 ± 0.06	0.93 ± 0.08	0.85 ± 0.08
gauss	4	3	0.97 ± 0.05	1.08 ± 0.07	1.14 ± 0.08	1.12 ± 0.14	1.09 ± 0.11
gauss	4	4	1.00 ± 0.06	1.06 ± 0.06	1.06 ± 0.09	1.02 ± 0.12	1.02 ± 0.12
gauss	4	5	0.98 ± 0.03	1.08 ± 0.05	1.06 ± 0.06	1.03 ± 0.07	0.95 ± 0.08
gauss	5	3	0.92 ± 0.06	1.02 ± 0.05	1.04 ± 0.07	1.04 ± 0.10	0.93 ± 0.09
gauss	5	4	0.96 ± 0.04	1.03 ± 0.05	1.00 ± 0.06	0.95 ± 0.06	0.81 ± 0.08
gauss	5	5	0.87 ± 0.13	1.15 ± 0.12	1.14 ± 0.14	1.11 ± 0.19	1.13 ± 0.22
gauss	Mean		$\mathbf{0.95 \pm 0.04}$	$\mathbf{1.06 \pm 0.05}$	$\mathbf{1.04 \pm 0.07}$	$\mathbf{1.00 \pm 0.08}$	$\mathbf{0.95 \pm 0.11}$
tanh	3	3	-0.20 ± 0.19	0.76 ± 0.08	1.12 ± 0.08	1.01 ± 0.11	1.09 ± 0.11
tanh	3	4	0.48 ± 0.18	1.20 ± 0.08	1.48 ± 0.15	1.31 ± 0.13	1.15 ± 0.12
tanh	3	5	0.36 ± 0.06	1.19 ± 0.03	1.31 ± 0.04	1.23 ± 0.06	1.20 ± 0.06
tanh	4	3	-0.57 ± 0.20	0.55 ± 0.05	1.37 ± 0.06	1.40 ± 0.07	1.31 ± 0.07
tanh	4	4	-0.37 ± 0.23	0.70 ± 0.05	1.49 ± 0.05	1.56 ± 0.09	1.35 ± 0.09
tanh	4	5	0.33 ± 0.14	1.00 ± 0.03	1.40 ± 0.05	1.46 ± 0.07	1.35 ± 0.07
tanh	5	3	-0.24 ± 0.12	0.90 ± 0.08	1.32 ± 0.09	1.52 ± 0.13	1.26 ± 0.12
tanh	5	4	0.06 ± 0.07	1.10 ± 0.04	1.44 ± 0.05	1.49 ± 0.09	1.40 ± 0.08
tanh	5	5	0.31 ± 0.05	1.17 ± 0.03	1.45 ± 0.04	1.44 ± 0.06	1.26 ± 0.06
tanh	Mean		$\mathbf{0.0 \pm 0.4}$	$\mathbf{0.95 \pm 0.24}$	$\mathbf{1.38 \pm 0.12}$	$\mathbf{1.38 \pm 0.17}$	$\mathbf{1.26 \pm 0.10}$

Table 14: Measured scaling exponents of the median crossing scale $\mathrm{med}_k\, \delta_0$ in each gradient bin against the global median crossing scale $\mathrm{med}\, |\delta_{0,R}$, for both target types. Gradient bins are fixed by the 60th, 80th, 90th, and 95th percentiles of $|\nabla f(\xi_R)|$, with $k = 1$ being the flattest and $k = 5$ the steepest.

Type	n	r	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
gauss	3	3	1.05 ± 0.03	0.97 ± 0.03	1.00 ± 0.05	1.03 ± 0.06	0.94 ± 0.06
gauss	3	4	1.05 ± 0.03	0.95 ± 0.03	0.93 ± 0.04	0.97 ± 0.06	0.96 ± 0.07
gauss	3	5	1.05 ± 0.02	0.99 ± 0.03	0.91 ± 0.05	0.91 ± 0.07	0.91 ± 0.08
gauss	4	3	1.01 ± 0.03	1.04 ± 0.04	1.01 ± 0.06	1.03 ± 0.07	0.96 ± 0.08
gauss	4	4	1.01 ± 0.02	1.02 ± 0.03	1.01 ± 0.07	0.98 ± 0.10	0.99 ± 0.11
gauss	4	5	1.00 ± 0.02	1.05 ± 0.03	1.04 ± 0.03	1.02 ± 0.05	0.97 ± 0.05
gauss	5	3	1.02 ± 0.02	1.01 ± 0.03	0.93 ± 0.05	0.94 ± 0.05	0.97 ± 0.06
gauss	5	4	1.02 ± 0.02	0.99 ± 0.03	0.95 ± 0.04	0.98 ± 0.06	0.92 ± 0.08
gauss	5	5	1.01 ± 0.03	1.02 ± 0.06	1.01 ± 0.06	0.99 ± 0.07	0.97 ± 0.08
gauss	Mean		$\mathbf{1.02 \pm 0.02}$	$\mathbf{1.01 \pm 0.03}$	$\mathbf{0.98 \pm 0.05}$	$\mathbf{0.98 \pm 0.04}$	$\mathbf{0.95 \pm 0.03}$
tanh	3	3	1.13 ± 0.09	1.09 ± 0.04	0.92 ± 0.03	0.94 ± 0.05	0.88 ± 0.09
tanh	3	4	1.13 ± 0.07	1.18 ± 0.08	0.99 ± 0.06	1.03 ± 0.08	1.08 ± 0.12
tanh	3	5	1.19 ± 0.04	1.09 ± 0.02	1.01 ± 0.03	1.05 ± 0.04	1.10 ± 0.05
tanh	4	3	1.13 ± 0.05	1.12 ± 0.03	0.92 ± 0.02	0.96 ± 0.04	1.04 ± 0.04
tanh	4	4	1.08 ± 0.05	1.12 ± 0.02	0.99 ± 0.02	0.93 ± 0.02	1.02 ± 0.04
tanh	4	5	1.08 ± 0.04	1.020 ± 0.010	1.11 ± 0.02	1.10 ± 0.03	1.02 ± 0.03
tanh	5	3	1.13 ± 0.05	0.99 ± 0.03	0.97 ± 0.03	0.90 ± 0.05	1.04 ± 0.05
tanh	5	4	1.11 ± 0.03	1.03 ± 0.02	1.01 ± 0.02	1.00 ± 0.03	1.00 ± 0.03
tanh	5	5	1.110 ± 0.010	0.990 ± 0.010	0.990 ± 0.010	1.04 ± 0.02	1.07 ± 0.03
tanh	Mean		$\mathbf{1.12 \pm 0.03}$	$\mathbf{1.07 \pm 0.07}$	$\mathbf{0.99 \pm 0.05}$	$\mathbf{1.00 \pm 0.06}$	$\mathbf{1.03 \pm 0.06}$

Table 15: Measured scaling exponents of the median region size $\mathrm{med}_k\, s_R$ in each gradient bin against the global median region size $\mathrm{med}\, s_R$. The region size s_R is the mean exact ray-exit distance from the regional centroid over 48 isotropic directions. Gradient bins are fixed by the 60th, 80th, 90th, and 95th percentiles of $|\nabla f(\xi_R)|$, with $k = 1$ being the flattest and $k = 5$ the steepest.