Meta-Principled Family of Hyperparameter Scaling Strategies

Sho Yaida

Meta AI

Meta Platforms, Inc.

Menlo Park, California 94025, USA

shoyaida@meta.com

In this note, we first derive a one-parameter family of hyperparameter scaling strategies that interpolates between the neural-tangent scaling and mean-field/maximal-update scaling. We then calculate the scalings of dynamical observables – network outputs, neural tangent kernels, and differentials of neural tangent kernels – for wide and deep neural networks. These calculations in turn reveal a proper way to scale depth with width such that resultant large-scale models maintain their representation-learning ability. Finally, we observe that various infinite-width limits examined in the literature correspond to the distinct corners of the interconnected web spanned by effective theories for finite-width neural networks, with their training dynamics ranging from being weakly-coupled to being strongly-coupled.

I. INTRODUCTION

As we push the limits of AI-powered technologies – e.g., to draw like (Dalí+WALL-E)/2 [1, 2], to speak like an omnilingual translator [3], and to beat the imitation game like a human being [4, 5] – state-of-the-art models are inflating in size at a staggering pace. Anticipating this scaling trend to remain in vogue for the foreseeable future [6–8], it is imperative that we understand how to scale up models intelligently. Specifically, as deep-learning models grow both in width and depth, we'd like a principled way to adjust hyperparameters accordingly so that we won't have to restart the hyperparameter search from scratch each time models redouble.

Among a variety of hyperparameters used in practice, the most ubiquitous ones are those that control initialization distributions and learning rates. For these omnipresent hyperparameters, the following two scaling strategies have been suggested by theorists for wide and deep neural networks: one is the neural-tangent (NT) scaling strategy [9], which naturally arose from the study of infinitely-wide neural networks as kernel machines [10–12]; the other is the maximal-update (MU) scaling strategy [13], which generalizes the mean-field limit of one-hidden-layer neural networks [14–17] to deeper ones. In the wild today, unfortunately, neither strategies are followed systematically by practitioners: there are yet to be fair, methodical, and community-wide empirical tests as to which hyperparameter scaling strategies perform better for which dataset-task pairs with which network architectures and optimizers.[1]

In order to complement these putative empirical efforts, here we place various scaling strategies into a coherent theoretical framework.

In §II we first review minimum ingredients for our study while introducing the general class of hyperparameter scaling strategies that we call $p_\ell q_\ell r$ scaling strategies, which are equivalent to the abc-parametrizations introduced in Ref. [13].[2] In §III, we then derive a one-parameter family of hyperparameter scaling strategies from the principle of criticality for preactivations and the principle of learning-rate equivalence for the neural tangent kernel, that is, from the desiderata to avoid exploding/vanishing signals and to ensure that every layer contributes equally to learning [19]. The resulting family, (3.16)–(3.20), is parametrized by a metaparameter $s \in [0,1]$ and interpolates between the NT scaling at $s = 0$ and MU scaling at $s = 1$. In §IV, we further evaluate the degree of representation learning by analyzing the first-order and second-order differentials of the neural tangent kernel and find that their leading-order representation-learning effect is proportional to

$$\gamma \equiv \frac{L}{n^{1-s}}\,, \tag{1.1}$$

where n is the width and L is the depth of a neural network. Note that, had we fixed the depth in scaling up the model, the MU hyperparameter scaling strategy with $s = 1$ would be the only strategy that retains representation-learning capability for a large-scale neural network. However, it is critical to stress that, for any value of s, we can scale up the depth along with the width as $L \sim n^{1-s}$ to keep the emergent scale γ fixed. As we detail in §V, this leads to a richer taxonomy of representation-learning models in the infinite-model-size limit, and an even richer set of effective theories at finite size; some are governed by weakly-coupled training dynamics that can be analyzed by perturbation theory and the others are governed by strongly-coupled dynamics that call for nonperturbative approaches. Appendix A contains some further calculating, which presents the hierarchical structure of neural tangent kernel differentials for one-hidden-layer neural networks.

[1] See, e.g., Ref. [18] for a test of the MU scaling strategy.

[2] For the purpose of our presentation, we find it tidier to adopt a convention slightly different from Ref. [13]. For those familiar with Ref. [13], our conventions are related by

$$\begin{aligned} 2a_1 + 2b_1 =& p_1\,; \quad 2a_\ell + 2b_\ell = 1 + p_\ell \quad \text{for} \quad \ell = 2, \ldots, L\,, \\ 2a_1 =& q_1\,; \quad 2a_\ell = 1 + q_\ell \quad \text{for} \quad \ell = 2, \ldots, L\,, \\ c =& -r\,. \end{aligned}$$

We also identify the L-th layer as an output layer rather than attaching the extra $(L+1)$-th layer. More broadly, we follow the notations and conventions in Ref. [19] and extend the results therein to study general hyperparameter scaling strategies.

II. REVIEW AND $p_\ell q_\ell r$ SCALING STRATEGIES

Deep neural networks are composed of many structurally similar operations stacked on top of each other. A prototypical architecture – on which we focus all our concrete calculational effort throughout this note – is a multilayer perceptron (a.k.a. a fully-connected feedforward network), recursively defined by

$$z^{(1)}_{i;\delta} = b^{(1)}_i + \sum_{j=1}^{n_0} W^{(1)}_{ij} x_{j;\delta} \quad \text{for} \quad i = 1, \ldots, n_1 \,, \tag{2.1}$$

$$z^{(\ell+1)}_{i;\delta} = b^{(\ell+1)}_i + \sum_{j=1}^{n_\ell} W^{(\ell+1)}_{ij} \sigma\left(z^{(\ell)}_{j;\delta}\right) \quad \text{for} \quad i = 1, \ldots, n_{\ell+1} \,;\; \ell = 1, \ldots, L-1 \,, \tag{2.2}$$

where $x_{j;\delta}$ are inputs, indexed by a vectorial index j and a sample index $\delta \in \mathcal{D}$ in a dataset $\mathcal{D}$; biases $b^{(\ell)}_i$ and weights $W^{(\ell)}_{ij}$ are learnable model parameters; the single-variable function $\sigma(z)$ is called an activation function; and $z^{(\ell)}_{i;\delta}$'s are called ℓ-th-layer preactivations, with $f_{i;\delta} \equiv z^{(L)}_{i;\delta}$ being the network output for a sample δ. In addition, here, the input dimension n_0 and the output dimension n_L are fixed by the dataset and by the task while the hidden-layer widths $n_1, n_2, \ldots, n_{L-1}$ and the depth L are *architecture* hyperparameters one freely chooses. For the purpose of our calculations, we assume that the numbers of the hidden-layer neurons are all of similar order,

$$n_1, n_2, \ldots, n_{L-1} \sim n \,, \tag{2.3}$$

i.e., that they are all being increased simultaneously and uniformly in scaling up the model.

Before training starts, learnable model parameters need to be initialized in some way. Typically, the biases and weights are initialized identically and independently with mean-zero distributions – such as normal distributions and uniform distributions – whose covariances will be denoted as

$$\mathbb{E}\left[b^{(\ell)}_{i_1} b^{(\ell)}_{i_2}\right] = \frac{1}{n^{p_\ell}}\left(C^{(\ell)}_b \delta_{i_1 i_2}\right) \,, \quad \mathbb{E}\left[W^{(\ell)}_{i_1 j_1} W^{(\ell)}_{i_2 j_2}\right] = \frac{1}{n^{p_\ell}}\left(\frac{C^{(\ell)}_W}{n_{\ell-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\right) \,, \tag{2.4}$$

where $C^{(\ell)}_b$ and $C^{(\ell)}_W$ are order-one numbers – *initialization* hyperparameters – assigned to each layer, and expectation $\mathbb{E}\left[\cdot\right]$ is over the initialization distributions.[3] In particular, the NT scaling strategy corresponds to setting $p_\ell = 0$ which, as we shall see, results in order-one preactivations for all the layers. More generally, nonzero p_ℓ's would let us consider other possible scaling strategies with different powers of the typical hidden-layer width n.[4]

[3] An obligatory apology is in order for using the symbol δ both for the Kronecker delta and for the sample index; it should be clear from the context which are which.

[4] Even more generally, one can pick different p_ℓ's for biases and for weights; we will here make them equal such that biases and weights in a given layer contribute at the same order.

After initialization, model parameters $\theta_\mu \equiv \left\{b_i^{(\ell)}, W_{ij}^{(\ell)}\right\}_{\ell=1,\ldots,L}$ are typically trained by some gradient-based optimization algorithm. A prototypical optimizer – again to which we pay all our concrete calculational attention – is gradient descent, given by

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \sum_\nu \lambda_{\mu\nu} \frac{d\mathcal{L}_{\mathcal{A}_t}}{d\theta_\nu}\Big|_{\theta=\theta(t)}, \tag{2.5}$$

iterated from $t = 0$ at which point the network is initialized. Here, $\mathcal{L}_{\mathcal{A}_t}$ is a loss function on the batch of samples $\mathcal{A}_t \subset \mathcal{D}$ at the t-th iteration; η is a global learning rate; and $\lambda_{\mu\nu}$ is a learning-rate tensor that lets us control in more detail how the gradient on the ν-th model parameter affects the update of the μ-th model parameter. Typically, the learning-rate tensor is chosen to be diagonal,

$$\lambda_{b_{i_1}^{(\ell)} b_{i_2}^{(\ell)}} = \frac{1}{n^{q_\ell}} \left(\lambda_b^{(\ell)} \delta_{i_1 i_2}\right), \quad \lambda_{W_{i_1 j_1}^{(\ell)} W_{i_2 j_2}^{(\ell)}} = \frac{1}{n^{q_\ell}} \left(\frac{\lambda_W^{(\ell)}}{n_{\ell-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\right), \tag{2.6}$$

where $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$ are order-one numbers – *training* hyperparameters – assigned to each layer: in words, $\eta\lambda_b^{(\ell)}/n^{q_\ell}$ is the learning rate for the ℓ-th-layer biases and $\eta\lambda_W^{(\ell)}/(n^{q_\ell} n_{\ell-1})$ is the learning rate for the ℓ-th-layer weights.[5] Here again, the NT scaling corresponds to setting $q_\ell = 0$ and nonzero q_ℓ's allow more general scaling strategies. In the same spirit, we can think of further scaling the global learning rate with n,

$$\eta = n^r \eta_0, \tag{2.7}$$

with an order-one number η_0.

In passing, we note that there is trivial gauge redundancy among the exponents p_ℓ, q_ℓ, and r. That is, for any real number g, we readily see that the transformation

$$p_\ell \to p_\ell, \quad q_\ell \to q_\ell + g, \quad r \to r + g \tag{2.8}$$

leaves the initialization (2.4) and the update equation (2.5) invariant. I.e., out of the $2L+1$ exponents introduced in this section, one linear combination has no physical consequence.

Finally, to introduce various dynamical observables of our interest, let us Taylor-expand the update for the network output, $f_{i;\delta} = z_{i;\delta}^{(L)}$, in the update for the model parameters,

$$\theta_\mu(t+1) - \theta_\mu(t) = -\eta \sum_\nu \lambda_{\mu\nu} \frac{d\mathcal{L}_{\mathcal{A}_t}}{d\theta_\nu}(t) = -\eta \sum_\nu \lambda_{\mu\nu} \sum_{j=1}^{n_L} \sum_{\widetilde{\alpha}\in\mathcal{A}_t} \frac{df_{j;\widetilde{\alpha}}}{d\theta_\nu}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j;\widetilde{\alpha}}}(t). \tag{2.9}$$

[5] Even more typically, the learning-rate tensor is set to be a Kronecker delta, $\lambda_{\mu\nu} = \delta_{\mu\nu}$, giving the same learning rate for all the model parameters. That is the so-called standard parametrization, which is used most standardly in practice as of today but maybe not if you come from the theoretical future.

After doing just that and rearranging summations and dummy indices, we get

$$
\begin{aligned}
f_{i;\delta}(t+1) =& f_{i;\delta}(t) - \sum_{j=1}^{n_L} \sum_{\widetilde{\alpha}\in\mathcal{A}_t} \eta H_{ij;\delta\widetilde{\alpha}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j;\widetilde{\alpha}}}(t) \\
&+ \frac{1}{2} \sum_{j_1,j_2=1}^{n_L} \sum_{\widetilde{\alpha}_1,\widetilde{\alpha}_2\in\mathcal{A}_t} \eta^2 \mathrm{d}H_{ij_1j_2;\delta\widetilde{\alpha}_1\widetilde{\alpha}_2}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_1;\widetilde{\alpha}_1}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_2;\widetilde{\alpha}_2}}(t) \\
&- \frac{1}{6} \sum_{j_1,j_2,j_3=1}^{n_L} \sum_{\widetilde{\alpha}_1,\widetilde{\alpha}_2,\widetilde{\alpha}_3\in\mathcal{A}_t} \eta^3 \mathrm{dd_I}H_{ij_1j_2j_3;\delta\widetilde{\alpha}_1\widetilde{\alpha}_2\widetilde{\alpha}_3}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_1;\widetilde{\alpha}_1}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_2;\widetilde{\alpha}_2}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_3;\widetilde{\alpha}_3}}(t) + \dots ,
\end{aligned}
\tag{2.10}
$$

where we introduced

$$
H_{i_0i_1;\delta_0\delta_1} \equiv \sum_{\mu,\nu} \lambda_{\mu\nu} \frac{df_{i_0;\delta_0}}{d\theta_\mu} \frac{df_{i_1;\delta_1}}{d\theta_\nu} , \tag{2.11}
$$

$$
\mathrm{d}H_{i_0i_1i_2;\delta_0\delta_1\delta_2} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2} \frac{d^2 f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}} \frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}} \frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}} , \tag{2.12}
$$

$$
\mathrm{dd_I}H_{i_0i_1i_2i_3;\delta_0\delta_1\delta_2\delta_3} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2\\ \mu_3,\nu_3}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\lambda_{\mu_3\nu_3} \frac{d^3 f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}d\theta_{\mu_3}} \frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}} \frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}} \frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}} . \tag{2.13}
$$

They are the neural tangent kernel (NTK), the first-order differential of the NTK (dNTK), and the second-order differential of the NTK (ddNTK) *of type I*, respectively.[6]

For the system of equations to be closed, we need to further track the evolution of these dynamical observables: e.g., by Taylor-expanding the update for the NTK (2.11) in the update for the model parameters (2.9), we get

$$
\begin{aligned}
&H_{i_0i_1;\delta_0\delta_1}(t+1) \\
=& H_{i_0i_1;\delta_0\delta_1}(t) - \eta \sum_{j=1}^{n_L} \sum_{\widetilde{\alpha}\in\mathcal{A}_t} \left[\mathrm{d}H_{i_0i_1j;\delta_0\delta_1\widetilde{\alpha}}(t) + \mathrm{d}H_{i_1i_0j;\delta_1\delta_0\widetilde{\alpha}}(t) \right] \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j;\widetilde{\alpha}}}(t) \\
&+ \frac{1}{2}\eta^2 \sum_{j_2,j_3=1}^{n_L} \sum_{\widetilde{\alpha}_2,\widetilde{\alpha}_3\in\mathcal{A}_t} \left[\mathrm{dd_I}H_{i_0i_1j_2j_3;\delta_0\delta_1\widetilde{\alpha}_2\widetilde{\alpha}_3}(t) + \mathrm{dd_I}H_{i_1i_0j_2j_3;\delta_1\delta_0\widetilde{\alpha}_2\widetilde{\alpha}_3}(t) \right] \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_2;\widetilde{\alpha}_2}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_3;\widetilde{\alpha}_3}}(t) \\
&+ \eta^2 \sum_{j_2,j_3=1}^{n_L} \sum_{\widetilde{\alpha}_2,\widetilde{\alpha}_3\in\mathcal{A}_t} \mathrm{dd_{II}}H_{i_0i_1j_2j_3;\delta_0\delta_1\widetilde{\alpha}_2\widetilde{\alpha}_3}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_2;\widetilde{\alpha}_2}}(t) \frac{d\mathcal{L}_{\mathcal{A}_t}}{df_{j_3;\widetilde{\alpha}_3}}(t) + \dots ,
\end{aligned}
\tag{2.14}
$$

where we additionally introduced the *type-II* ddNTK,

$$
\mathrm{dd_{II}}H_{i_0i_1i_2i_3;\delta_0\delta_1\delta_2\delta_3} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2\\ \mu_3,\nu_3}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\lambda_{\mu_3\nu_3} \frac{d^2 f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}} \frac{d^2 f_{i_1;\delta_1}}{d\theta_{\nu_1}d\theta_{\mu_3}} \frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}} \frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}} . \tag{2.15}
$$

[6] Note that the transformation (2.8) leaves ηH, $\eta^2 \mathrm{d}H$, and $\eta^3 \mathrm{dd_I}H$ invariant.

Here, we see that the dNTK governs the leading-order change in the NTK after a gradient-descent step – hence its naming – and, carrying out the same exercise for the dNTK, we would find that a linear combination of the ddNTKs governs the leading-order change in the dNTK.

In general, these dynamical equations proliferate *ad infinitum*, both because there are an infinite number of Taylor terms for each dynamical equation and because there are an infinite number of dynamical objects with their own update equations.

Thus, at the first sight, these coupled equations look infinitely formidable and intractable. Fortunately, in some regimes – specifically, when the scale $\gamma = L/n^{1-s}$ (1.1) is small – we can systematically truncate the Taylor expansions and the resulting system of coupled dynamical equations can be solved through perturbation theory [19, 20]. What's more, for a given optimizer, the solution at the end of training can be expressed solely in terms of the dynamical objects *at initialization*. Therefore, we can mostly focus our study on the statistics and scalings of these dynamical objects at initialization, from which we can read off their statistics and scalings after training.[7]

With those remarks in mind, in the next section, we'll study the network output and NTK at initialization while, in the section after that, we'll study the dNTK and ddNTKs at initialization. Meanwhile, higher-order differentials of the NTK are consigned to Appendix A.

III. DERIVING A META-PRINCIPLED FAMILY

In this section, we'll use the principle of criticality for hidden-layer preactivations (§III A) and the principle of learning-rate equivalence for neural tangent kernels (§III B) to narrow the general class of $p_\ell q_\ell r$ hyperparameter scaling strategies down to a one-parameter family (§III C).

A. The Principle of Criticality at Meta

As a given input signal propagates forward through many layers of a deep neural network, the signal gets hit by many similar mathematical operations one after another. In turn, we generically expect that the forward signal either exponentially explodes or exponentially vanishes in magnitude, both of which would inhibit learning. The solution to this problem was proposed in a series of papers [22–24] and later extended in Ref. [19]. In particular, for the NT scaling with

[7] For this Taylor approach to work, the activation function needs to be smooth. This excludes `ReLU` activation function from our consideration due to its kink at origin: see Ref. [19] for more details on this subtlety. That said, empirically there is no observed superiority of `ReLU` over its smooth variants like `GELU` and `SWISH` – see, e.g., Ref. [21] – and hence this theoretical limitation is not practically limiting.

$p_\ell = 0$, the resulting *principle of criticality* gives a general prescription for tuning initialization hyperparameters $C_b^{(\ell)}$ and $C_W^{(\ell)}$ so as to ensure that signals stay of the same order for all layers.

Here, we'll apply this principle of criticality at a coarser – or more *meta* – level, requiring that forward signals stay *parametrically* of order one in all the hidden layers at initialization. Without such requirement, we would have a severe numerical problem for deep neural networks, with the values of preactivations differing by orders of magnitude from layer to layer.[8]

Specifically, at the point of initialization, the first-layer preactivations (2.1) have the covariance

$$\mathbb{E}\left[z^{(1)}_{i_1;\delta_1} z^{(1)}_{i_2;\delta_2}\right] = \frac{1}{n^{p_1}}\left[C_b^{(1)} + C_W^{(1)}\left(\frac{1}{n_0}\sum_{j=1}^{n_0} x_{j;\delta_1} x_{j;\delta_2}\right)\right]\delta_{i_1 i_2}\,, \tag{3.1}$$

where we used our general initialization hyperparameter scaling strategies (2.4). Similarly, the $(\ell+1)$-th-layer preactivations (2.2) have the covariance

$$\mathbb{E}\left[z^{(\ell+1)}_{i_1;\delta_1} z^{(\ell+1)}_{i_2;\delta_2}\right] = \frac{1}{n^{p_{\ell+1}}}\left\{C_b^{(\ell+1)} + C_W^{(\ell+1)}\frac{1}{n_\ell}\sum_{j=1}^{n_\ell}\mathbb{E}\left[\sigma\left(z^{(\ell)}_{j;\delta_1}\right)\sigma\left(z^{(\ell)}_{j;\delta_2}\right)\right]\right\}\delta_{i_1 i_2}\,. \tag{3.2}$$

Looking at these covariances, the *meta*-principle of criticality demands that we recursively set

$$p_1 = p_2 = \cdots = p_{L-1} = 0\,, \tag{3.3}$$

so that all the hidden-layer preactivations – and activations – stay of order one.[9]

Extremely importantly, differently from the prior work [19, 22–24], we intentionally left out the output layer at $\ell = L$ from this consideration. From our perspective, the ingenuity of Ref. [13] lies in realizing that we don't have to apply the meta-principle of criticality to the output layer, which leaves one free parameter,

$$p_L \equiv s\,. \tag{3.4}$$

Specifically, while a positive value of s results in parametrically small network outputs at initialization, it still allows nontrivial learning as long as the product of the global learning rate and the NTK, ηH, is of order one so that the network outputs acquire order-one values upon evolving with the update equation (2.10). We'll come back to this point in §III C.

[8] Yet another problem would be that, as we scale up models in width, the hidden-layer preactivations would explore different ranges of the activation function from scale to scale and hence perceive it as having different characteristics (with a possible exception of scale-invariant activation functions such as `linear` and `ReLU` activation functions).

[9] Surely, it appears intuitively obvious that the expectation value $\mathbb{E}\left[\sigma\left(z^{(\ell)}_{j;\delta_1}\right)\sigma\left(z^{(\ell)}_{j;\delta_2}\right)\right]$ is of order one for a generic activation function $\sigma(z)$ when the ℓ-th-layer preactivations are of order one. To be surer, for sufficiently large n, this expectation value can be turned into a series of Gaussian integrals through the $1/n$ expansion and evaluated order by order [19].

B. The Principle of Learning-Rate Equivalence at Meta

Similarly to forward signals, we also need to ensure that backward signals don't exponentially explode or exponentially vanish. More precisely, the gradients of the network outputs with respect to the model parameters must stay of the same order from layer to layer, so that we can stably train networks with gradient-based optimization algorithms.

A convenient way to diagnose this backward signal propagation is to look at the NTK (2.11),

$$\widehat{H}_{i_0 i_1;\delta_0\delta_1} \equiv \sum_{\mu,\nu} \lambda_{\mu\nu} \frac{df_{i_0;\delta_0}}{d\theta_\mu} \frac{df_{i_1;\delta_1}}{d\theta_\nu} \,, \tag{3.5}$$

which essentially is a square of the gradients. Here, we've decided to put a hat on the NTK to emphasize that we'll be studying its scaling at the point of initialization.[10]

To assess its scaling and statistics, it is further convenient to introduce the ℓ-th-layer NTK [9, 19],

$$\widehat{H}^{(\ell)}_{i_0 i_1;\delta_0\delta_1} \equiv \sum_{\mu,\nu} \lambda_{\mu\nu} \frac{dz^{(\ell)}_{i_0;\delta_0}}{d\theta_\mu} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta_\nu} \,, \tag{3.6}$$

which reproduces the output NTK (3.5) when $\ell = L$. More importantly, it obeys the forward equation

$$\begin{aligned}\widehat{H}^{(\ell+1)}_{i_1 i_2;\delta_1\delta_2} =& \frac{1}{n^{q_{\ell+1}}} \left\{ \lambda_b^{(\ell+1)} + \lambda_W^{(\ell+1)} \left[\frac{1}{n_\ell} \sum_{j=1}^{n_\ell} \sigma\left(z^{(\ell)}_{j;\delta_1}\right) \sigma\left(z^{(\ell)}_{j;\delta_2}\right) \right] \right\} \delta_{i_1 i_2} \\ &+ \sum_{j_1,j_2=1}^{n_\ell} W^{(\ell+1)}_{i_1 j_1} W^{(\ell+1)}_{i_2 j_2} \sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right) \sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right) \widehat{H}^{(\ell)}_{j_1 j_2;\delta_1\delta_2} \,,\end{aligned} \tag{3.7}$$

which can be derived by plugging in our general training hyperparameter scaling strategies (2.6) and using the chain rule with the familiar factor

$$\frac{dz^{(\ell+1)}_{i;\delta}}{dz^{(\ell)}_{j;\delta}} = W^{(\ell+1)}_{ij} \sigma'\left(z^{(\ell)}_{j;\delta}\right) \,, \tag{3.8}$$

that also appears in the backpropagation algorithm. The first term on the right-hand side of the NTK forward equation (3.7) captures the additive contributions from the gradients with respect to the $(\ell+1)$-th-layer model parameters; the second term captures the cumulative contributions from the previous layers, with the ℓ-th-layer NTK multiplied by two backpropagation factors – though we are emphatically going forward here.

[10] We note that in the so-called NTK limit [9] – i.e., in the infinite-width limit with the NT scaling and with fixed depth – the NTK is frozen at its value at initialization, so this hatting is redundant. But at any finite width – and also in some proper infinite-model-size limits – the NTK evolves due to nonzero dNTK and ddNTKs.

Now, the principle of learning-rate equivalence [19] gives a prescription for fine-tuning the training hyperparameters $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$ so as to ensure that gradients from all layers contribute equally to the output NTK – preconditioned on the principle of criticality that ensures hidden-layer preactivations neither exponentially explode nor exponentially vanish. Here, we apply the same principle – again at a more *meta* level – requiring that the two terms on the right-hand side of the NTK forward equation (3.7) are *parametrically* of the same order.

Putting that in action, on the one hand, the first term is of order $O\left(1/n^{q_{\ell+1}}\right)$.[11] On the other hand, the second term is of order $O\left(1/n^{p_{\ell+1}}\right) \times O\big(\widehat{H}^{(\ell)}\big)$.[12] To finish balancing these two terms, noting that the first-layer NTK

$$\widehat{H}^{(1)}_{i_1 i_2;\delta_1\delta_2} = \frac{1}{n^{q_1}}\left[\lambda_b^{(1)} + \lambda_W^{(1)}\left(\frac{1}{n_0}\sum_{j=1}^{n_0} x_{j;\delta_1}x_{j;\delta_2}\right)\right]\delta_{i_1 i_2} \tag{3.10}$$

scales as $O\left(1/n^{q_1}\right)$, hence consistently and inductively assuming that the ℓ-th-layer NTK scales as $O\left(1/n^{q_\ell}\right)$, and recalling the scalings of the two terms described in the two previous sentences, we get by induction

$$q_{\ell+1} = p_{\ell+1} + q_\ell \quad \text{for} \quad \ell = 1, \ldots, L-1\,. \tag{3.11}$$

Therefore, recalling that $p_1 = \cdots = p_{L-1} = 0$ (3.3) and $p_L = s$ (3.4), the meta-principle of learning-rate equivalence further demands that we recursively set

$$q_1 = q_2 = \cdots q_{L-1} \equiv t \tag{3.12}$$

and

$$q_L = t + s\,. \tag{3.13}$$

[11] Here and right after, we are preconditioning our scaling analysis on the meta-principle of criticality, i.e., that hidden-layer preactivations – and in turn hidden-layer activations and their derivatives – are parametrically of order one.

[12] To be a little more verbose, the factor of $1/n_\ell$ coming from the base NT scaling for the initial weights cancels out with the summation over $j = 1, \ldots, n_\ell$. To get the flavor of it, we can take the expectation value of the NTK forward equation (3.7), which, together with the initialization distributions (2.4), yields

$$\begin{aligned}\mathbb{E}\left[\widehat{H}^{(\ell+1)}_{i_1 i_2;\delta_1\delta_2}\right] =& \frac{1}{n^{q_{\ell+1}}}\left\{\lambda_b^{(\ell+1)} + \lambda_W^{(\ell+1)}\mathbb{E}\left[\frac{1}{n_\ell}\sum_{j=1}^{n_\ell}\sigma\left(z^{(\ell)}_{j;\delta_1}\right)\sigma\left(z^{(\ell)}_{j;\delta_2}\right)\right]\right\}\delta_{i_1 i_2} \\ &+ \frac{1}{n^{p_{\ell+1}}}\left\{C_W^{(\ell+1)}\mathbb{E}\left[\frac{1}{n_\ell}\sum_{j=1}^{n_\ell}\sigma'\left(z^{(\ell)}_{j;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j;\delta_2}\right)\widehat{H}^{(\ell)}_{jj;\delta_1\delta_2}\right]\right\}\delta_{i_1 i_2}\,,\end{aligned} \tag{3.9}$$

wherein we see the nice pairings of $1/n_\ell$ and $\sum_{j=1}^{n_\ell}$. For an even more verbose analysis, including the effects of the NTK fluctuations, please cf. Ref. [19]. In particular, we can show that the contributions from the NTK fluctuations are suppressed by a factor of $O\left(L/n\right)$.

C. Meta-Principled Family

At this point, unlike we promised, it appears as if we still got three parameters at our disposal: r from the global learning rate η (2.7), s from the last-layer preactivagtion scalings (3.4), and t from the bulk of the learning-rate tensor (3.12). Let's quickly cut these three down to one.

First, referring back to the network evolution (2.10), to have some learning in finite steps, we need the product of learning rate $\eta = O\left(n^r\right)$ and the output NTK $\widehat{H} = O\left(1/n^{q_L}\right)$ to be of order one. This requires that

$$r = q_L = t + s\,, \tag{3.14}$$

reducing the three down to two.[13]

Next, to cut the remaining two down to one, we recall that there is trivial gauge redundancy (2.8) without any physical consequence, which can be used to gauge away one redundant exponent. Specifically, choosing the transformation (2.8) with $g = -t$, we finally get

$$p_1 = \cdots = p_{L-1} = 0\,, \quad p_L = s\,, \quad q_1 = \cdots = q_{L-1} = 0\,, \quad q_L = s\,, \quad r = s\,. \tag{3.15}$$

In summary, we are left with only one *meta*parameter, s, that parametrizes the meta-principled family of hyperparameter scaling strategies:

$$\mathbb{E}\left[b^{(\ell)}_{i_1} b^{(\ell)}_{i_2}\right] = C^{(\ell)}_b \delta_{i_1 i_2}\,, \quad \mathbb{E}\left[W^{(\ell)}_{i_1 j_1} W^{(\ell)}_{i_2 j_2}\right] = \frac{C^{(\ell)}_W}{n_{\ell-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\,, \quad \text{for} \quad \ell = 1, \ldots, L-1\,, \tag{3.16}$$

$$\mathbb{E}\left[b^{(L)}_{i_1} b^{(L)}_{i_2}\right] = \frac{C^{(L)}_b}{n^s_{L-1}} \delta_{i_1 i_2}\,, \quad \mathbb{E}\left[W^{(L)}_{i_1 j_1} W^{(L)}_{i_2 j_2}\right] = \frac{C^{(L)}_W}{n^{1+s}_{L-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\,, \tag{3.17}$$

$$\lambda_{b^{(\ell)}_{i_1} b^{(\ell)}_{i_2}} = \lambda^{(\ell)}_b \delta_{i_1 i_2}\,, \quad \lambda_{W^{(\ell)}_{i_1 j_1} W^{(\ell)}_{i_2 j_2}} = \frac{\lambda^{(\ell)}_W}{n_{\ell-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\,, \quad \text{for} \quad \ell = 1, \ldots, L-1\,, \tag{3.18}$$

$$\lambda_{b^{(L)}_{i_1} b^{(L)}_{i_2}} = \frac{1}{n^s_{L-1}} \left(\lambda^{(L)}_b \delta_{i_1 i_2}\right)\,, \quad \lambda_{W^{(L)}_{i_1 j_1} W^{(L)}_{i_2 j_2}} = \frac{\lambda^{(L)}_W}{n^{1+s}_{L-1}} \delta_{i_1 i_2} \delta_{j_1 j_2}\,, \tag{3.19}$$

$$\eta = n^s_{L-1} \eta_0\,, \tag{3.20}$$

where, for concreteness, we set $n \equiv n_{L-1}$. In words, with respect to the NT scaling, we suppress the initialization hyperparameters and the learning-rate tensor components by a factor of $1/n^s$ for the last-layer biases and weights, while encouraging the global learning rate by a factor of n^s.[14]

[13] In general, there is also a nontrivial depth scaling coming from $\widehat{H}^{(L)}$. Here we account for that through $\eta_0 \lambda^{(\ell)}_{b,W}$.

[14] To apply the same meta-principles to more general architectures, the pragmatic rule of thumb is to just replace $n_{\ell-1}$ by the number of incoming signal components, often denoted as `fan_in` in PyTorch documentations. For hidden layers, our scaling strategies for initialization hyperparameters are then generally in line with the Xavier [25] and Kaiming [26] initialization schemes.

This family interpolates the NT scaling [9] at $s = 0$ and the MU scaling [13] at $s = 1$. In fact, these two strategies lie at the two ends of the sensible interval for the metaparameter $s \in [0, 1]$: $s \geq 0$ because, otherwise, the network outputs would be parametrically large; $s \leq 1$ because, otherwise, the differentials of the NTK would be parametrically larger, as we'll see next.[15]

IV. COMPUTING THE DEGREE OF REPRESENTATION LEARNING

In this section, we'll analyze the scalings of the dNTK (2.12) in §IV A and ddNTKs (2.13) (2.15) in §IV B. To motivate these calculations, let's look back at the NTK update equation (2.14). As noted there, these differentials control changes in the NTK during training. Viewing the kernel as a sum of features that comprise representation, the magnitudes of these differentials in turn quantify the degree of leading-order representation learning [19], which we'll compute here for wide and deep neural networks. This computation results in the emergent scale γ (4.10).

A. dNTK

Following our recursive analysis of the NTK, let's define the ℓ-th-layer dNTK as

$$\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2} \frac{d^2 z^{(\ell)}_{i_0;\delta_0}}{d\theta_{\mu_1} d\theta_{\mu_2}} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta_{\nu_1}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta_{\nu_2}} \,, \tag{4.1}$$

where we've again hatted it to emphasize that we are analyzing its scaling at initialization. This object reproduces the output dNTK (2.12) when $\ell = L$ while it vanishes in the first layer $\ell = 1$ because the first-layer preactivations (2.1) are linear in the model parameters. In the middle –

[15] At this point, for those familiar with Ref. [13], it may be helpful to pause and carry out an in-depth comparison between this part of our results against theirs.
Here are similarities. In Ref. [13], they invoked the "stability condition" to put hard constraints, $p_1 = \cdots p_{L-1} = 0$ (along with various soft constraints, i.e., inequalities), much like our meta-principle of criticality did. They then invoked the "nontriviality condition" to put a hard constraint $r = q_L$ – or another more complicated condition that is also satisfied by our family – much like our condition of finite learning (3.14), $\eta\widehat{H} = O(1)$, did. They further noticed the same symmetry under the transformation (2.8). These together cut the $2L + 1$ exponents down to L, so far in the same way as ours.
Here are differences. They invoked the "feature learning condition," $0 = 1 + \min_{\ell=1}^{L-1}(q_\ell - r) + \min\left(q_L - r, \frac{p_L+1}{2}\right)$. This condition was put forth to preserve feature learning in the infinite-width limit *with fixed depth*. To single out the MU scaling, they actually provoked stronger conditions, the "maximal-update conditions," $0 = 1 + (q_\ell - r) + \min\left(q_L - r, \frac{p_L+1}{2}\right)$ for $\ell = 1, \ldots, L-1$ (which automatically satisfy the "feature learning condition") and the "maximal-initialization condition," $0 = \frac{p_L+1}{2} + \min_{\ell=1}^{L-1}(q_\ell - r) + \min\left(q_L - r, \frac{p_L+1}{2}\right)$. Our family – which instead arose from the meta-principle of learning-rate equivalence – connects this MU scaling at $s = 1$ all the way down to the NT scaling at $s = 0$ while preserving representation learning for $s < 1$ by *scaling up the depth* with the width.

nearly reprinting an equation from Ref. [19] – it obeys the following forward equation:

$$
\begin{aligned}
\widehat{\mathrm{d}H}^{(\ell+1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} = & \frac{1}{n^{q_{\ell+1}}}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\delta_{i_0 i_1}\sum_{j_0,j_2=1}^{n_\ell} W^{(\ell+1)}_{i_2 j_2}\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma\left(z^{(\ell)}_{j_0;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2} \\
& + \frac{1}{n^{q_{\ell+1}}}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\delta_{i_0 i_2}\sum_{j_0,j_1=1}^{n_\ell} W^{(\ell+1)}_{i_1 j_1}\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma\left(z^{(\ell)}_{j_0;\delta_2}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1} \\
& + \sum_{j_0,j_1,j_2=1}^{n_\ell} W^{(\ell+1)}_{i_0 j_0}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2} \\
& + \sum_{j_0,j_1,j_2=1}^{n_\ell} W^{(\ell+1)}_{i_0 j_0}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2} .
\end{aligned}
\tag{4.2}
$$

This can again be derived – mindlessly object-wise but mindfully index-wise – by simply plugging in our general training hyperparameter scaling strategies (2.6) and repeatedly using the chain rule.

For the meta-principled family (3.15), because $p_1 = \cdots = p_{L-1} = q_1 = \cdots = q_{L-1} = 0$, the bulk of the dNTK forward equation (4.2) stays the same with the NT scaling strategy – for which the width and depth scalings of the dNTK have already been analyzed in detail [19]. We thus only need to pay special attention to the equation when $\ell + 1 = L$. There, for the first two terms on the right-hand side of the forward equation (4.2), with respect to the NT scaling, there is an additional factor of $1/n^s$ coming from $1/n^{q_L}$ in front and another additional factor of $1/n^{\frac{s}{2}}$ coming from the last-layer weight; for the last two terms, similarly, there is an additional factor of $1/n^{\frac{3s}{2}}$ coming from the three last-layer weights. Thus, the gauge-invariant combination, $\eta^2\widehat{\mathrm{d}H}^{(L)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}$, naively comes with an additional factor of $n^{2s}/n^{\frac{3s}{2}} = n^{\frac{s}{2}}$ with respect to the NT scaling. One twist is that it has an odd number of the last-layer weights, so its expectation vanishes; the first nontrivial expectation value is of the form $\mathbb{E}\left[z_{i;\delta}\eta^2\widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}\right]$, for which the s dependence cancels out.

All in all, downloading the scaling of this expectation value from Ref. [19], we get

$$
\mathbb{E}\left[z^{(L)}_{i;\delta}\eta^2\widehat{\mathrm{d}H}^{(L)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}\right] = O\left(\frac{L}{n}\right) . \tag{4.3}
$$

Thus, for a general value of s, the leading effect of the dNTK at initialization scales like the depth-to-width aspect ratio of the network.[16]

[16] To be precise, the equation (4.3) is imprecise. What actually happens in Ref. [19] on this topic is that *(i)* we study behaviors of signals in deep multilayer perceptrons and find various *universality classes* of activation functions; *(ii)* for each universality class, we solve the forward equations for the preactivations and the NTK, and find and choose the critical initialization hyperparameters $C^{(\ell)}_{b,W}$ and training hyperparameters $\lambda^{(\ell)}_{b,W}$ according to the principles of criticality and learning-rate equivalence so as to maximally avoid the exploding and vanishing problems; *(iii)* we then further solve the forward equations for the dNTK and ddNTKs to read off their width and depth scaling; and *(iv)* we look at dimensionless quantities such as $\mathbb{E}\left[z\eta^2\widehat{\mathrm{d}H}\right] / \left(\mathbb{E}\left[\eta\widehat{H}\right]\right)^2$ and $\mathbb{E}\left[z^2\right]\mathbb{E}\left[\eta^3\widehat{\mathrm{dd}_{\mathrm{I}}H}\right] / \left(\mathbb{E}\left[\eta\widehat{H}\right]\right)^2$, and – for the NT scaling strategy – find L/n scalings for all the leading corrections to the infinite-width limit.

B. ddNTKs

At this point, all the calculational procedures are familiar, so we'll briefly describe the calculations for the type-I ddNTK in the main text and the ones for the type-II ddNTK in a footnote.

We define the ℓ-th-layer type-I ddNTK as

$$\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2\\ \mu_3,\nu_3}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\lambda_{\mu_3\nu_3}\frac{d^3 z^{(\ell)}_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}d\theta_{\mu_3}}\frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta_{\nu_1}}\frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta_{\nu_2}}\frac{dz^{(\ell)}_{i_3;\delta_3}}{d\theta_{\nu_3}}\,, \tag{4.4}$$

which obeys the following forward equation that can be derived by plugging in our diagonal learning-rate tensor (2.6) and using the chain rule:

$$\begin{aligned}
&\widehat{\mathrm{dd_I}H}^{(\ell+1)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \qquad\qquad (4.5)\\
&= \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_0 i_1}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_1}W^{(\ell+1)}_{i_2 j_2}W^{(\ell+1)}_{i_3 j_3}\sigma\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right)\\
&\qquad\qquad \times\left[\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}+\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_2 j_3;\delta_0\delta_2\delta_3}\right]\\
&+ \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_0 i_2}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_2}W^{(\ell+1)}_{i_3 j_3}W^{(\ell+1)}_{i_1 j_1}\sigma\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\\
&\qquad\qquad \times\left[\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}+\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_3 j_1;\delta_0\delta_3\delta_1}\right]\\
&+ \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_0 i_3}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_3}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}\sigma\left(z^{(\ell)}_{j_3;\delta_3}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\\
&\qquad\qquad \times\left[\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}+\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2}\right]\\
&+ \sum_{j_0,j_1,j_2,j_3=1}^{n_\ell} W^{(\ell+1)}_{i_0 j_0}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}W^{(\ell+1)}_{i_3 j_3}\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right)\\
&\quad\times\Big[\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{dd_I}H}^{(\ell)}_{j_0 j_1 j_2 j_3;\delta_0\delta_1\delta_2\delta_3}+\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\\
&\qquad+\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_2 j_3;\delta_0\delta_2\delta_3}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}+\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_3 j_1;\delta_0\delta_3\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\\
&\qquad\qquad\qquad\qquad+\sigma'''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\Big]\,.
\end{aligned}$$

With respect to the NT scaling, all the terms get the multiplicative factor of $1/n^{2s}$ at $\ell+1=L$,

For those who could actually parse the last paragraph, there is one additional comment: in the last step *(iv)*, for the dimensionless combination for the ddNTK contribution, one might worry about the cancellation of the s dependence due to the factor of $\mathbb{E}\left[z^2\right]$, which scales like $1/n^s$; however, there is another dimensionless combination with $\mathbb{E}\left[z^2\right]$ replaced by y^2 with the training labels $y_{i;\tilde{\alpha}}$, and this combination does indeed appear in the end-of-training-prediction formula [19] and contributes dominantly without the s cancellation.

so, once again using the result from Ref. [19] for $s = 0$, the gauge-invariant combination scales as

$$\mathbb{E}\left[\eta^3 \widehat{\mathrm{dd_I}H}^{(L)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3}\right] = O\left(\frac{L}{n^{1-s}}\right) \tag{4.6}$$

for wide and deep neural networks.[17]

Through these calculations, there emerges the scale,

$$\gamma \equiv \frac{L}{n^{1-s}}\,, \tag{4.10}$$

that controls the effect of the ddNTKs. For $s = 0$, this is of the same order as the effect of the dNTK (4.3), while for any positive $s > 0$ this contribution is more dominant. In general, the emergent scale γ characterizes the amount of the leading-order change in the NTK and hence the

[17] As for the type-II ddNTK, we define the ℓ-th-layer version as

$$\widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \equiv \sum_{\substack{\mu_1,\nu_1\\ \mu_2,\nu_2\\ \mu_3,\nu_3}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\lambda_{\mu_3\nu_3} \frac{d^2 z^{(\ell)}_{i_0;\delta_0}}{d\theta_{\mu_1} d\theta_{\mu_2}} \frac{d^2 z^{(\ell)}_{i_1;\delta_1}}{d\theta_{\nu_1} d\theta_{\mu_3}} \frac{d z^{(\ell)}_{i_2;\delta_2}}{d\theta_{\nu_2}} \frac{d z^{(\ell)}_{i_3;\delta_3}}{d\theta_{\nu_3}}\,, \tag{4.7}$$

which obeys the forward equation

$$\begin{aligned}
&\widehat{\mathrm{dd_{II}}H}^{(\ell+1)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \\
&= \frac{1}{n^{2q_{\ell+1}}}\left(\frac{\lambda^{(\ell+1)}_W}{n_\ell}\right)^2 \delta_{i_0 i_2}\delta_{i_1 i_3} \sum_{j_0,j_1=1}^{n_\ell} \sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma\left(z^{(\ell)}_{j_0;\delta_2}\right)\sigma\left(z^{(\ell)}_{j_1;\delta_3}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1} \\
&+ \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_0 i_1} \sum_{j_0,j_1,j_2,j_3=1}^{n_\ell} \delta_{j_0 j_1} W^{(\ell+1)}_{i_2 j_2} W^{(\ell+1)}_{i_3 j_3} \\
&\qquad \times \sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right)\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3} \\
&+ \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_0 i_2} \sum_{j_0,j_1,j_2,j_3=1}^{n_\ell} \delta_{j_0 j_2} W^{(\ell+1)}_{i_1 j_1} W^{(\ell+1)}_{i_3 j_3}\sigma\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right)\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right) \\
&\qquad \times \left[\sigma''\left(z^{(\ell)}_{j_1;\delta_1}\right)\widehat{H}^{(\ell)}_{j_1 j_0;\delta_1\delta_0}\widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3} + \sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_1 j_0 j_3;\delta_1\delta_0\delta_3}\right] \\
&+ \frac{1}{n^{q_{\ell+1}}}\frac{\lambda^{(\ell+1)}_W}{n_\ell}\delta_{i_1 i_3} \sum_{j_0,j_1,j_2,j_3=1}^{n_\ell} \delta_{j_1 j_3} W^{(\ell+1)}_{i_0 j_0} W^{(\ell+1)}_{i_2 j_2}\sigma\left(z^{(\ell)}_{j_3;\delta_3}\right)\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right) \\
&\qquad \times \left[\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2} + \sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2}\right] \\
&+ \sum_{j_0,j_1,j_2,j_3=1}^{n_\ell} W^{(\ell+1)}_{i_0 j_0} W^{(\ell+1)}_{i_1 j_1} W^{(\ell+1)}_{i_2 j_2} W^{(\ell+1)}_{i_3 j_3}\sigma'\left(z^{(\ell)}_{j_2;\delta_2}\right)\sigma'\left(z^{(\ell)}_{j_3;\delta_3}\right) \\
&\quad \times \left[\sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\widehat{\mathrm{dd_{II}}H}^{(\ell)}_{j_0 j_1 j_2 j_3;\delta_0\delta_1\delta_2\delta_3} + \sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma''\left(z^{(\ell)}_{j_1;\delta_1}\right)\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3}\right. \\
&\qquad \left. + \sigma'\left(z^{(\ell)}_{j_0;\delta_0}\right)\sigma''\left(z^{(\ell)}_{j_1;\delta_1}\right)\widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2} + \sigma'\left(z^{(\ell)}_{j_1;\delta_1}\right)\sigma''\left(z^{(\ell)}_{j_0;\delta_0}\right)\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{\mathrm{d}H}^{(\ell)}_{j_1 j_0 j_3;\delta_1\delta_0\delta_3}\right],
\end{aligned} \tag{4.8}$$

which gets the multiplicative factor of $1/n^{2s}$ at $\ell + 1 = L$ with respect to the NT scaling, and thus

$$\mathbb{E}\left[\eta^3 \widehat{\mathrm{dd_{II}}H}^{(L)}\right] = O\left(\frac{L}{n^{1-s}}\right)\,. \tag{4.9}$$

degree of representation learning for wide and deep neural networks.[18] When small, it also serves as a perturbative parameter that quantifies the leading-order change to the frozen NTK limit and, at least for one-hidden-layer networks, we explicitly show in Appendix A that the higher-order differentials of the NTK are suppressed by higher-order powers of γ.

V. CASTING THE WEB OF EFFECTIVE THEORIES

Thus far, we've derived a family of hyperparameter scaling strategies, meta-parametrized by $s \in [0,1]$, from the meta-principle of criticality and the meta-principle of learning-rate equivalence.[19] We've then assessed the scalings of representation learning as a function of the hidden-layer width n and the depth L, which gave rise to the emergent scale, $\gamma = L/n^{1-s}$, that also controls the hierarchical structure of dynamical equations for wide and deep neural networks.

Theoretically, for each combination of the width n, depth L, and metaparameter s, we have an effective theory that describes the distribution of dynamical observables – at the point of initialization, at the end of training, and anywhen in between – for finite-width neural networks [19]. Taking a bird's-eye view on all of them, we now see the web of effective theories, with each thread – indexed by (n, L) and parametrized by s – smoothly connecting the NT scaling at $s = 0$ and the MU scaling at $s = 1$.[20]

Formally, to know what each scaling strategy would be doing for excessively large-scale models – and to make contact with the literature – it is instructive to take various infinite-model-size limits.

- The frozen NTK limit [9]: $0 \le s < 1$; $n \to \infty$; $L =$ fixed.

  For the NT scaling strategy with $s = 0$ – and for any $s \in (0, 1)$ – taking the infinite-width limit $n \to \infty$ with fixed depth results in the vanishing $\gamma \to 0$. In fact, the training dynamics of a neural network reduce to those of a linear model with an infinite number of random frozen features, resulting in kernel-machine predictions at the end of training [29]. In particular, while its dynamics are free and solvable, there is no representation learning [19, 30, 31].

[18] To be a bit more precise, the depth scaling is derived for deep neural networks, and for shallow neural networks like the one-hidden-layer ones with $L = 2$, the "L" in the numerator of the scale $\gamma = L/n^{1-s}$ should just be thought of as an order-one number.

[19] In a sense, our derivation gives a no-go theorem for other hyperparameter scaling strategies. To hedge, we point out that no-go theorems sometimes contain subtle loopholes in their assumptions which, upon realization and proper relaxation, lead to interesting phenomena: cf. the relation between the Coleman-Mandula theorem [27] and supersymmetry [28].

[20] Metaphorically, different architectures give rise to different webs. Also, to keep forcing this metaphor, each point on the web has internal dimensions, e.g., parametrized by order-one hyperparameters such as $C_{b,W}^{(\ell)}$ and $\lambda_{b,W}^{(\ell)}$.

- The infinite-width-and-depth NT limit [32–36]: $s = 0$; $n, L \to \infty$; $L/n = \text{fixed} > 0$.

  The scale γ stays positive and inflating neural networks can keep learning representation from data.[21] In addition, when the depth-to-width aspect ratio $\gamma = L/n$ of a network is small, its dynamics become weakly-coupled and can be solved perturbatively [19, 20].

- The mean-field [14–17] and MU [13] limit: $s = 1$; $n \to \infty$; $L = \text{fixed} > 1$.

  The scale γ is of order one and the dynamics in the resulting (non)effective theory description are strongly-coupled: Taylor expansions cannot be perturbatively truncated and the infinitudes of dynamical observables get coupled. This calls for new theoretical approaches.[22]

- A missing limit: $0 < s < 1$; $n, L \to \infty$; $L/n^{1-s} = \text{fixed} > 0$.

  This limit sits between the previous two limits. In the strict limit, this family degenerates, in that the value of s is irrelevant and the limit is completely parametrized by γ.[23] This limit is different from the infinite-width-and-depth NT limit with $s = 0$ because observables proportional to $O\left(L/n\right)$ vanish and those proportional to $O\left(L/n^{1-s}\right)$ survive, while both survived for $s = 0$. It is also different from the mean-field/MU limit with $s = 1$ because we have a continuous knob γ, while we had only a discrete knob L for $s = 1$. In particular, the resulting dynamics again become weakly-coupled and perturbatively analyzable for small γ.

Practically, given those limiting behaviors, it appears prudent to scale up the model with fixed $\gamma = L/n^{1-s}$ as long as tasks at hand benefit from representation learning. Beyond that, our theoretical understanding is yet incomplete to predict how each scaling strategy fairs against others – when and why – and for the time being the verdict is still out, both theoretically and empirically. But – if we may muse – just as the nature is described by weakly-coupled QED at one scale and strongly-coupled QCD at another scale, it is hard to imagine that one singular scaling strategy would be superior to all the others for all the problems.

[21] More specifically, the contributions from the dNTK, ddNTKs, and NTK fluctuations all scale with $L/n = \gamma$. With increasing γ, there is actually a tradeoff between the benefit of representation learning and the cost of instantiation-to-instantiation fluctuations, and this tradeoff leads to the notion of the optimal aspect ratio, $\gamma^\star$ [19].

[22] See, e.g., Ref. [37] for an approach inspired by dynamical mean field theory. There could also be some strong-coupling expansion or, better yet, some duality: modern field theorists don't even blink at such highly-speculative statements.

[23] With that said, at finite width, we should stress that the metaparameter s *does* affect the degree of separation between two scales, $\frac{L}{n^{1-s}}$ and $\frac{L}{n}$, which might have practical consequences.

Acknowledgements

The author is variously grateful to Mario Geiger, Andrey Gromov, Boris Hanin, Dan Roberts, Kushal Tirumala, and a duo of Bruno De Luca and Eva Silverstein for discussions with varying degrees of enlightenment, and to Yasaman Bahri, Surya Ganguli, Jaehoon Lee, Mitchell Wortsman, and Susan Zhang for varying degrees of positive reinforcement to write and put out this note.

[1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-Shot Text-to-Image Generation," in *International Conference on Machine Learning*, pp. 8821–8831. 2021.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," `arXiv:2204.06125 [cs.CV]`.

[3] M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, *et al.*, "No Language Left Behind: Scaling Human-Centered Machine Translation," `arXiv:2207.04672 [cs.CL]`.

[4] A. M. Turing, "Computing Machinery and Intelligence," *Mind* **LIX** no. 236, (1950) 433–460.

[5] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, *et al.*, "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models," `arXiv:2206.04615 [cs.CL]`.

[6] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep Learning Scaling is Predictable, Empirically," `arXiv:1712.00409 [cs.LG]`.

[7] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," `arXiv:2001.08361 [cs.LG]`.

[8] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, *et al.*, "Training Compute-Optimal Large Language Models," `arXiv:2203.15556 [cs.CL]`.

[9] A. Jacot, F. Gabriel, and C. Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 8571–8580. 2018.

[10] R. M. Neal, "Priors for Infinite Networks," in *Bayesian Learning for Neural Networks*, pp. 29–53. Springer, 1996.

[11] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep Neural Networks as Gaussian Processes," in *International Conference on Learning Representations*. 2018.

[12] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, "Gaussian Process Behaviour in Wide Deep Neural Networks," in *International Conference on Learning Representations*. 2018.

[13] G. Yang and E. J. Hu, "Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks," in *International Conference on Machine Learning*, pp. 11727–11737. 2021.

[14] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks," *Proceedings of the National Academy of Sciences* **115** no. 33, (2018) E7665–E7671.

[15] G. M. Rotskoff and E. Vanden-Eijnden, "Trainability and Accuracy of Neural Networks: An Interacting Particle System Approach," `arXiv:1805.00915 [stat.ML]`.

[16] J. Sirignano and K. Spiliopoulos, "Mean Field Analysis of Neural Networks: A Law of Large Numbers," *SIAM Journal on Applied Mathematics* **80** no. 2, (2020) 725–752.

[17] L. Chizat and F. Bach, "On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 3036–3046. 2018.

[18] G. Yang, E. J. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao, "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer," `arXiv:2203.03466 [cs.LG]`.

[19] D. A. Roberts, S. Yaida, and B. Hanin, *The Principles of Deep Learning Theory*. Cambridge University Press, 2022. `https://deeplearningtheory.com`.

[20] E. Dyer and G. Gur-Ari, "Asymptotics of Wide Networks from Feynman Diagrams," in *International Conference on Learning Representations*. 2020.

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986. 2022.

[22] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos," in *Advances in Neural Information Processing Systems*, vol. 29, pp. 3360–3368. 2016.

[23] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the Expressive Power of Deep Neural Networks," in *International Conference on Machine Learning*, pp. 2847–2854. 2017.

[24] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, "Deep Information Propagation," in *International Conference on Learning Representations*. 2017.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 249–256. 2010.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision*, pp. 1026–1034. 2015.

[27] S. Coleman and J. Mandula, "All Possible Symmetries of the S Matrix," *Physical Review* **159** no. 5, (1967) 1251.

[28] S. Weinberg, *The Quantum Theory of Fields, Volume 3: Supersymmetry*. Cambridge University Press, 2005.

[29] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide

Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 8572–8583. 2019.

[30] L. Chizat, E. Oyallon, and F. Bach, "On Lazy Training in Differentiable Programming," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 2937–2947. 2019.

[31] M. Geiger, S. Spigler, A. Jacot, and M. Wyart, "Disentangling feature and lazy training in deep neural networks," *Journal of Statistical Mechanics: Theory and Experiment* **2020** no. 11, (2020) 113301.

[32] B. Hanin and M. Nica, "Products of Many Large Random Matrices and Gradients in Deep Neural Networks," *Communications in Mathematical Physics* **376** no. 1, (2020) 287–322.

[33] B. Hanin and M. Nica, "Finite Depth and Width Corrections to the Neural Tangent Kernel," in *International Conference on Learning Representations.* 2019.

[34] Z. Hu and H. Huang, "On the Random Conjugate Kernel and Neural Tangent Kernel," in *International Conference on Machine Learning*, pp. 4359–4368. 2021.

[35] M. Li, M. Nica, and D. Roy, "The future is log-Gaussian: ResNets and their infinite-depth-and-width limit at initialization," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 7852–7864. 2021.

[36] M. B. Li, M. Nica, and D. M. Roy, "The Neural Covariance SDE: Shaped Infinite Depth-and-Width Networks at Initialization," `arXiv:2206.02768 [stat.ML]`.

[37] B. Bordelon and C. Pehlevan, "Self-Consistent Dynamical Field Theory of Kernel Evolution in Wide Neural Networks," `arXiv:2205.09653 [stat.ML]`.

[38] K. Aitken and G. Gur-Ari, "On the asymptotics of wide networks with polynomial activations," `arXiv:2006.06687 [cs.LG]`.

[39] J. Huang and H.-T. Yau, "Dynamics of Deep Neural Networks and Neural Tangent Hierarchy," in *International Conference on Machine Learning*, pp. 4542–4551. 2020.

Appendix A: Peeking into the Hierarchical Structure

In this Appendix, we'll compute the higher-order differentials of the NTK for one-hidden-layer neural networks. We'll do this in two steps: in §A 1 we'll explicitly calculate *type-I* differentials to see the hierarchical pattern; then, knowing the pattern, in §A 2 we'll implicitly show by induction that the same hierarchical structure holds for general types of the higher-order differentials.

Beyond seeing the hierarchical structure with the general metaparameter s, this Appendix serves two additional purposes: for one, it complements the existing work that estimates the width scalings of these higher-order differentials for deeper neural networks with a `linear` [20] or `polynomial` [38] activation function (see also Ref. [39] for various bounds on some combinations of higher-order differentials); for two, it might help someone in the future trying to show the same hierarchical structure for deeper neural networks with general activation functions, using the result and derivation herein as stepping stones for the recursive analysis.

Before diving into the actual calculations, let us make three preparatory general remarks.

- Thinking about the way in which higher-order differentials of the NTK are being generated, we can recognize that the k-th-order differentials can be generated by acting on a set of the $(k-1)$-th-order differentials with the differential operator

$$\sum_{\mu\nu}\left(\frac{df_{i;\delta}}{d\theta_{\nu}}\right)\lambda_{\mu\nu}\frac{d}{d\theta_{\mu}}\,. \tag{A1}$$

For instance, acting on the NTK with the operator generates the dNTK as

$$\begin{aligned}
&\sum_{\mu_2\nu_2}\left(\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}\right)\lambda_{\mu_2\nu_2}\frac{d}{d\theta_{\mu_2}}\left[\sum_{\mu_1,\nu_1}\lambda_{\mu_1\nu_1}\frac{df_{i_0;\delta_0}}{d\theta_{\mu_1}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\right]\\
=&\sum\lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\left(\frac{d^2f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}+\frac{df_{i_0;\delta_0}}{d\theta_{\mu_1}}\frac{d^2f_{i_1;\delta_1}}{d\theta_{\nu_1}d\theta_{\mu_2}}\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}\right)\\
=&\mathrm{d}H_{i_0i_1i_2;\delta_0\delta_1\delta_2}+\mathrm{d}H_{i_1i_0i_2;\delta_1\delta_0\delta_2}\,,
\end{aligned}\tag{A2}$$

and acting on the dNTK with the operator generates the two types of the ddNTKs as

$$\begin{aligned}
&\sum_{\mu_3\nu_3}\left(\frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}}\right)\lambda_{\mu_3\nu_3}\frac{d}{d\theta_{\mu_3}}\left[\sum_{\mu_1,\nu_1,\mu_2,\nu_2}\lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\frac{d^2f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}\right]\\
=&\sum\lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2}\lambda_{\mu_3\nu_3}\Bigg(\frac{d^3f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}d\theta_{\mu_3}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}\frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}}\\
&\qquad+\frac{d^2f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}}\frac{d^2f_{i_1;\delta_1}}{d\theta_{\nu_1}d\theta_{\mu_3}}\frac{df_{i_2;\delta_2}}{d\theta_{\nu_2}}\frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}}+\frac{d^2f_{i_0;\delta_0}}{d\theta_{\mu_1}d\theta_{\mu_2}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\frac{d^2f_{i_2;\delta_2}}{d\theta_{\nu_2}d\theta_{\mu_3}}\frac{df_{i_3;\delta_3}}{d\theta_{\nu_3}}\Bigg)\\
=&\mathrm{dd}_{\mathrm{I}}H_{i_0i_1i_2i_3;\delta_0\delta_1\delta_2\delta_3}+\mathrm{dd}_{\mathrm{II}}H_{i_0i_1i_2i_3;\delta_0\delta_1\delta_2\delta_3}+\mathrm{dd}_{\mathrm{II}}H_{i_0i_2i_1i_3;\delta_0\delta_2\delta_1\delta_3}\,.
\end{aligned}\tag{A3}$$

- Among various types of the higher-order differentials, we single out those with the form

$$\mathrm{d}_{\mathrm{I}}^{k-1}H_{i_0\cdots i_k;\delta_0\cdots\delta_k} \equiv \sum_{\substack{\mu_1,\nu_1\\ \cdots\\ \mu_k,\nu_k}} \lambda_{\mu_1\nu_1}\cdots\lambda_{\mu_k\nu_k}\frac{d^k f_{i_0;\delta_0}}{d\theta_{\mu_1}\cdots d\theta_{\mu_k}}\frac{df_{i_1;\delta_1}}{d\theta_{\nu_1}}\cdots\frac{df_{i_k;\delta_k}}{d\theta_{\nu_k}} \tag{A4}$$

  as the *type-I* $(k-1)$-th-order differentials of the NTK. These are the ones that appear in the dynamical update equation for the network outputs (2.10).

- We decompose the learning-rate tensor into those that act within each layer as

$$\lambda_{\mu\nu} = \sum_{\ell=1}^{L}\lambda_{\mu\nu}^{(\ell)}\,, \tag{A5}$$

  and, with a slight abuse of notation, denote the model parameters in the ℓ-th layer as $\theta_\mu^{(\ell)}$.

Now, let's dive in.

1. Type-I Higher-Order Differentials

First, let's write out the type-I $(k-1)$-th-order differential (A4) for the one-hidden-layer neural networks (i.e., those with $L = 2$), with the decomposition (A5) in mind:

$$\widehat{\mathrm{d}_{\mathrm{I}}^{k-1}H}_{i_0\cdots i_k;\delta_0\cdots\delta_k} = \sum_{\ell_1,\ldots,\ell_k=1}^{2}\sum_{\substack{\mu_1,\nu_1\\ \cdots\\ \mu_k,\nu_k}} \lambda_{\mu_1\nu_1}^{(\ell_1)}\cdots\lambda_{\mu_k\nu_k}^{(\ell_k)}\frac{d^k z_{i_0;\delta_0}^{(2)}}{d\theta_{\mu_1}^{(\ell_1)}\cdots d\theta_{\mu_k}^{(\ell_k)}}\frac{dz_{i_1;\delta_1}^{(2)}}{d\theta_{\nu_1}^{(\ell_1)}}\cdots\frac{dz_{i_k;\delta_k}^{(2)}}{d\theta_{\nu_k}^{(\ell_k)}}\,. \tag{A6}$$

Here, for one last time, we've put a hat to remind us that we are computing this object at initialization, which can be used to read off the scalings and statistics after training as well if wished. There are two kinds of nonzero terms in the sum over layers, the first kind with $\ell_1 = \cdots = \ell_k = 1$ and the second kind with $(\ell_1, \ell_2, \ldots, \ell_k) = (2, 1, \ldots, 1)$ along with its permutations thereof. All the other terms vanish as the output $z_{i_0;\delta_0}^{(2)}$ is linear in the second-layer model parameters $\theta_\mu^{(2)}$.

To evaluate the first kind, note that

$$\frac{dz_{i;\delta}^{(2)}}{d\theta_\nu^{(1)}} = \sum_{j=1}^{n_1}\frac{dz_{i;\delta}^{(2)}}{dz_{j;\delta}^{(1)}}\frac{dz_{j;\delta}^{(1)}}{d\theta_\nu^{(1)}} = \sum_{j=1}^{n_1} W_{ij}^{(2)}\sigma'\left(z_{j;\delta}^{(1)}\right)\frac{dz_{j;\delta}^{(1)}}{d\theta_\nu^{(1)}} \tag{A7}$$

by the chain rule, and similarly – using the fact that the first-layer preactivations $z_{i;\delta}^{(1)}$ are linear in the first-layer model parameters $\theta_\mu^{(1)}$ – note that

$$\frac{d^k z_{i_0;\delta_0}^{(2)}}{d\theta_{\mu_1}^{(1)}\cdots d\theta_{\mu_k}^{(1)}} = \sum_{j_0=1}^{n_1} W_{i_0j_0}^{(2)}\sigma^{[k]}\left(z_{j_0;\delta_0}^{(1)}\right)\frac{dz_{j_0;\delta_0}^{(1)}}{d\theta_{\mu_1}^{(1)}}\cdots\frac{dz_{j_0;\delta_0}^{(1)}}{d\theta_{\mu_k}^{(1)}}\,, \tag{A8}$$

where we denoted the k-th derivative of the activation function by $\sigma^{[k]}$. Putting them together, we get the contribution

$$
\begin{aligned}
&\sum_{\substack{\mu_1,\nu_1\\ \cdots\\ \mu_k,\nu_k}} \lambda^{(1)}_{\mu_1\nu_1}\cdots\lambda^{(1)}_{\mu_k\nu_k} \sum_{j_0,j_1,\ldots,j_k=1}^{n_1} W^{(2)}_{i_0j_0}W^{(2)}_{i_1j_1}\cdots W^{(2)}_{i_kj_k}\sigma^{[k]}\left(z^{(1)}_{j_0;\delta_0}\right)\sigma'\left(z^{(1)}_{j_1;\delta_1}\right)\cdots\sigma'\left(z^{(1)}_{j_k;\delta_k}\right)\\
&\qquad\qquad\qquad\qquad\times\frac{dz^{(1)}_{j_0;\delta_0}}{d\theta^{(1)}_{\mu_1}}\cdots\frac{dz^{(1)}_{j_0;\delta_0}}{d\theta^{(1)}_{\mu_k}}\frac{dz^{(1)}_{j_1;\delta_1}}{d\theta^{(1)}_{\nu_1}}\cdots\frac{dz^{(1)}_{j_k;\delta_k}}{d\theta^{(1)}_{\nu_k}}\\
&=\sum_{j_0,j_1,\ldots,j_k=1}^{n_1} W^{(2)}_{i_0j_0}W^{(2)}_{i_1j_1}\cdots W^{(2)}_{i_kj_k}\sigma^{[k]}\left(z^{(1)}_{j_0;\delta_0}\right)\sigma'\left(z^{(1)}_{j_1;\delta_1}\right)\cdots\sigma'\left(z^{(1)}_{j_k;\delta_k}\right)\widehat{H}^{(1)}_{j_0j_1;\delta_0\delta_1}\cdots\widehat{H}^{(1)}_{j_0j_k;\delta_0\delta_k}\\
&=\sum_{j=1}^{n_1} W^{(2)}_{i_0j}W^{(2)}_{i_1j}\cdots W^{(2)}_{i_kj}\sigma^{[k]}\left(z^{(1)}_{j;\delta_0}\right)\sigma'\left(z^{(1)}_{j;\delta_1}\right)\cdots\sigma'\left(z^{(1)}_{j;\delta_k}\right)H^{(1)}_{\delta_0\delta_1}\cdots H^{(1)}_{\delta_0\delta_k}\,,
\end{aligned}
\tag{A9}
$$

where in the last line we used the fact that the first-layer NTK (3.10),

$$
\widehat{H}^{(1)}_{j_0j_1;\delta_0\delta_1}=\frac{1}{n^{q_1}}\left[\lambda^{(1)}_b+\lambda^{(1)}_W\left(\frac{1}{n_0}\sum_{j=1}^{n_0}x_{j;\delta_0}x_{j;\delta_1}\right)\right]\delta_{j_0j_1}\equiv\delta_{j_0j_1}H^{(1)}_{\delta_0\delta_1},
\tag{A10}
$$

is diagonal in neural indices i.

To evaluate the second kind, note that

$$
\frac{dz^{(2)}_{i_1;\delta_1}}{dW^{(2)}_{ij}}=\delta_{i_1i}\sigma\left(z^{(1)}_{j;\delta_1}\right)\,,
\tag{A11}
$$

and

$$
\frac{d^kz^{(2)}_{i_0;\delta_0}}{dW^{(2)}_{ij}d\theta^{(1)}_{\mu_2}\cdots d\theta^{(1)}_{\mu_k}}=\delta_{i_0i}\sigma^{[k-1]}\left(z^{(1)}_{j;\delta_0}\right)\frac{dz^{(1)}_{j;\delta_0}}{d\theta^{(1)}_{\mu_2}}\cdots\frac{dz^{(1)}_{j;\delta_0}}{d\theta^{(1)}_{\mu_k}}\,.
\tag{A12}
$$

Putting them together – summing over all the components of the second-layer weights and noting that there is no second-layer bias contribution for $(k-1)\geq 1$ – for the $(\ell_1,\ell_2,\ldots,\ell_k)=(2,1,\ldots,1)$, we get the contribution

$$
\begin{aligned}
&\frac{1}{n^{q_2}}\frac{\lambda^{(2)}_W}{n_1}\delta_{i_0i_1}\sum_{\substack{\mu_2,\nu_2\\ \cdots\\ \mu_k,\nu_k}}\lambda^{(1)}_{\mu_2\nu_2}\cdots\lambda^{(1)}_{\mu_k\nu_k}\sum_{j,j_2,\ldots,j_k=1}^{n_1}W^{(2)}_{i_2j_2}\cdots W^{(2)}_{i_kj_k}\\
&\quad\times\sigma^{[k-1]}\left(z^{(1)}_{j;\delta_0}\right)\sigma\left(z^{(1)}_{j;\delta_1}\right)\sigma'\left(z^{(1)}_{j_2;\delta_2}\right)\cdots\sigma'\left(z^{(1)}_{j_k;\delta_k}\right)\frac{dz^{(1)}_{j;\delta_0}}{d\theta^{(1)}_{\mu_2}}\cdots\frac{dz^{(1)}_{j;\delta_0}}{d\theta^{(1)}_{\mu_k}}\frac{dz^{(1)}_{j_2;\delta_2}}{d\theta^{(1)}_{\nu_2}}\cdots\frac{dz^{(1)}_{j_k;\delta_k}}{d\theta^{(1)}_{\nu_k}}\\
&=\frac{1}{n^{q_2}}\frac{\lambda^{(2)}_W}{n_1}\delta_{i_0i_1}\sum_{j,j_2,\ldots,j_k=1}^{n_1}W^{(2)}_{i_2j_2}\cdots W^{(2)}_{i_kj_k}\sigma^{[k-1]}\left(z^{(1)}_{j;\delta_0}\right)\sigma\left(z^{(1)}_{j;\delta_1}\right)\sigma'\left(z^{(1)}_{j_2;\delta_2}\right)\cdots\sigma'\left(z^{(1)}_{j_k;\delta_k}\right)\widehat{H}^{(1)}_{jj_2;\delta_0\delta_2}\cdots\widehat{H}^{(1)}_{jj_k;\delta_0\delta_k}\\
&=\frac{1}{n^{q_2}}\delta_{i_0i_1}\frac{\lambda^{(2)}_W}{n_1}\sum_{j=1}^{n_1}W^{(2)}_{i_2j}\cdots W^{(2)}_{i_kj}\sigma^{[k-1]}\left(z^{(1)}_{j;\delta_0}\right)\sigma\left(z^{(1)}_{j;\delta_1}\right)\sigma'\left(z^{(1)}_{j;\delta_2}\right)\cdots\sigma'\left(z^{(1)}_{j;\delta_k}\right)H^{(1)}_{\delta_0\delta_2}\cdots H^{(1)}_{\delta_0\delta_k}\,,
\end{aligned}
\tag{A13}
$$

and we get similar contributions from the appropriate permutations.

Let's compute the scalings of these contributions with our meta-principled family of hyperparameter scaling strategies (3.15), $p_1 = q_1 = 0$ and $p_2 = q_2 = r = s$. With $p_1 = q_1 = 0$, we note that the first-layer preactivations $z^{(1)}$ and the first-layer NTK $H^{(1)}$ are both of order one, so we can essentially forget about them for a moment. With that noted, there are two cases to consider.

- For odd $k = 2m + 1$, that is, for even $k - 1 = 2m$, let's evaluate the following expectation value of the gauge-invariant product at initialization

  $$\mathbb{E}\left[\eta^{2m+1}\widehat{\mathrm{d}_{\mathrm{I}}^{2m}H}_{i_0\cdots i_{2m+1};\delta_0\cdots\delta_{2m+1}}\right] . \tag{A14}$$

  For the first kind of contributions (A9), we get a factor of $1/n^{(m+1)(p_2+1)} = 1/n^{(m+1)(s+1)}$ from $(2m + 2)$ weights while it gets a factor of n from the summation over j, so when multiplied by $\eta^{2m+1} \sim n^{2ms+s}$, we get $O\left(1/n^{m(1-s)}\right)$. For the second kind of contributions (A13), we get a factor of $1/n^{q_2} = 1/n^{s}$, a cancelation between the denominator and summation $(1/n_1)\sum_{j=1}^{n_1}$, a factor of $1/n^{m(p_2+1)} = 1/n^{m(s+1)}$ from $2m$ weights, and a factor of $\eta^{2m+1} \sim n^{2ms+s}$, so overall we once again get $O\left(1/n^{m(1-s)}\right)$. In summary,

  $$\mathbb{E}\left[\eta^{2m+1}\widehat{\mathrm{d}_{\mathrm{I}}^{2m}H}_{i_0\cdots i_{2m+1};\delta_0\cdots\delta_{2m+1}}\right] = O\left(\frac{1}{n^{m(1-s)}}\right) = O\left(\gamma^{m}\right) \tag{A15}$$

  with $\gamma \equiv 1/n^{1-s}$ – or $\gamma = 2/n^{1-s}$ if two means anything here (it doesn't). Thus, we see the hierarchical structure of higher-order differentials advertised in the main text.

- For even $k = 2m$, that is, for odd $k - 1 = 2m - 1$, let's evaluate the following expectation value of the gauge-invariant product at initialization

  $$\mathbb{E}\left[\eta^{2m}z^{(2)}_{i;\delta}\widehat{\mathrm{d}_{\mathrm{I}}^{2m-1}H}_{i_0\cdots i_{2m};\delta_0\cdots\delta_{2m}}\right] . \tag{A16}$$

  Note that we inserted one factor of the second-layer preactivations because – just as we saw for the dNTK – the expectation would otherwise vanish due to a dangling factor of the second-layer weights. With that factor added, for the first kind of contributions (A9), we get a factor of $1/n^{(m+1)(p_2+1)} = 1/n^{(m+1)(s+1)}$ from $(2m + 1 + 1)$ weights while it gets a factor of n from the summation over j, so when multiplied by $\eta^{2m} \sim n^{2ms}$, we get $O\left(1/n^{m(1-s)+s}\right)$. Running the similar analysis, we get the same order for the second kind of contributions (A13). In summary

  $$\mathbb{E}\left[\eta^{2m}z^{(2)}_{i;\delta}\widehat{\mathrm{d}_{\mathrm{I}}^{2m-1}H}_{i_0\cdots i_{2m};\delta_0\cdots\delta_{2m}}\right] = O\left(\frac{1}{n^{m(1-s)+s}}\right) = O\left(\gamma^{m}\frac{1}{n^{s}}\right) = O\left(\gamma^{m-1}\frac{1}{n}\right) . \tag{A17}$$

  We see that, for $s > 0$, the contributions from the odd-order differentials are more suppressed than the ones from the even-order differentials.

2. General Types of Higher-Order Differentials

We'll now inductively show that the $(k-1)$-th-order differentials have the form of

$$\left(\frac{1}{n^{q_2}}\frac{\lambda_W^{(2)}}{n_1}\right)^u \sum_{j=1}^{n_1} W_{i_1 j}^{(2)} \cdots W_{i_v j}^{(2)} F\left(z_j^{(1)}, H^{(1)}\right), \tag{A18}$$

with $2u+v=k+1$ where the last factor $F\left(z_j^{(1)}, H^{(1)}\right)$ is an order-one function of the first-layer mean NTK $H_{\delta_1\delta_2}^{(1)}$ and the *j-th component* of the first-layer preactivation $z_{j;\delta}^{(1)}$. Note that the two contributions (A9) (A13) that appeared above for the type-I differentials both respect this form.

The key to induction is to decompose the generative differential operator (A1) as

$$\begin{aligned}
&\sum_{\mu\nu}\left(\frac{df_{i;\delta}}{d\theta_\nu}\right)\lambda_{\mu\nu}\frac{d}{d\theta_\mu} &\text{(A19)}\\
=&\sum_{\mu\nu}\left(\frac{dz_{i;\delta}^{(2)}}{d\theta_\nu^{(1)}}\right)\lambda_{\mu\nu}^{(1)}\frac{d}{d\theta_\mu^{(1)}}+\left(\frac{1}{n^{q_2}}\frac{\lambda_W^{(2)}}{n_1}\right)\sum_{i'=1}^{n_2}\sum_{j'=1}^{n_1}\left(\frac{dz_{i;\delta}^{(2)}}{dW_{i'j'}^{(2)}}\right)\frac{d}{dW_{i'j'}^{(2)}}+\left(\frac{1}{n^{q_2}}\lambda_b^{(2)}\right)\sum_{i'=1}^{n_2}\left(\frac{dz_{i;\delta}^{(2)}}{db_{i'}^{(2)}}\right)\frac{d}{db_{i'}^{(2)}}\\
=&\sum_{j'=1}^{n_1}W_{ij'}^{(2)}\sigma'\left(z_{j';\delta}^{(1)}\right)\left[\sum_{\mu\nu}\lambda_{\mu\nu}^{(1)}\frac{dz_{j';\delta}^{(1)}}{d\theta_\nu^{(1)}}\frac{d}{d\theta_\mu^{(1)}}\right]+\left(\frac{1}{n^{q_2}}\frac{\lambda_W^{(2)}}{n_1}\right)\sum_{j'=1}^{n_1}\sigma\left(z_{j';\delta}^{(1)}\right)\frac{d}{dW_{ij'}^{(2)}}+\left(\frac{1}{n^{q_2}}\lambda_b^{(2)}\right)\frac{d}{db_i^{(2)}}\,.
\end{aligned}$$

Going from the last to the first: when the last operator hits the expression (A18), it returns zero because there is no second-layer bias; when the middle operator hits the expression (A18), it increases u by one while it decreases v by one – note that the sum over j' collapses onto j – so in total increases $2u+v$ by one; and when the first operator hits the expression (A18), it increases v – and hence $2u+v$ – by one while again the sum over j' collapses onto j because the first-layer NTK – which gets generated when the operator in the square parentheses hits the first-layer preactivation in the function F – is diagonal in the neural indices. So it keeps the form (A18) with $2u+v=k+1$ increased by one, completing the induction.

From the expression (A18), we can run the same scaling analysis as before and find the γ^m scaling for $2m$-th-order differentials and the γ^m/n^s scaling for $(2m-1)$-th-order differentials.