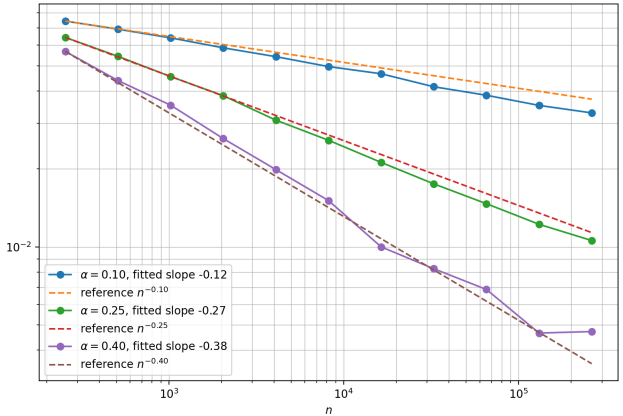
Feature Learning in the Proportional Depth-Width Limit

Mufan Li

University of Waterloo and Vector Institute

Based on joint work with Boris Hanin (Princeton) and Lorenzo Noci (ETH Zürich/OpenAI)

Scaling Laws and Scaling Limits



Central Limit Theorem:

$X_i \overset{\text{iid}}{\sim} P$ with $\mathbb{E}X_i = 0, \mathbb{E}X_i^2 = 1$. Then

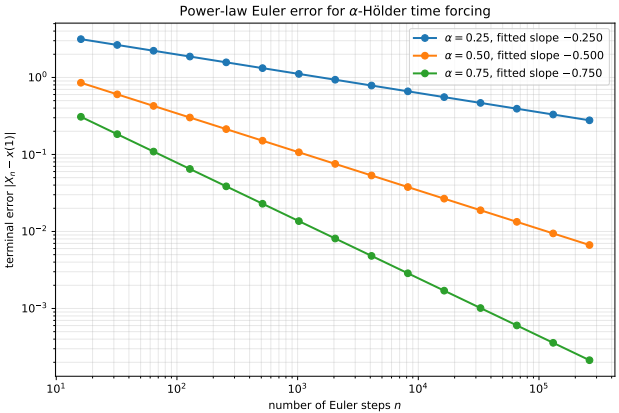
$$S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i \xrightarrow{d} \mathcal{N}(0,1)\,.$$

Question: What is the convergence rate?

Answer: Depends on largest $\alpha \in (0, 1/2]$ such that $\mathbb{E}|X_i|^{2(1+\alpha)}$ is bounded, in which case the Kolmogorov distance behaves like a **power law**

$$d_K(\mathcal{L}(S_n), \mathcal{N}(0,1)) \propto n^{-\alpha}\,.$$

Scaling Laws and Scaling Limits



Euler discretization of an ODE:

Let $\dot{x}(t) = b(t, x(t)), x(0) = x_0$, where b is Lipschitz in x but only Hölder in $t \in [0, T]$

$$|b(t,x) - b(s,x)| \leq C|t-s|^{\alpha}\,, \quad \alpha \in (0,1)\,.$$

Then for an explicit Euler discretization with stepsize $h = T/n$

$$X_{k+1} = X_k + hb(kh, X_k)\,,$$

we have the **power law** error bound

$$\max_{k \leq n} |X_k - x(kh)| \propto n^{-\alpha}\,.$$

Scaling Laws and Scaling Limits

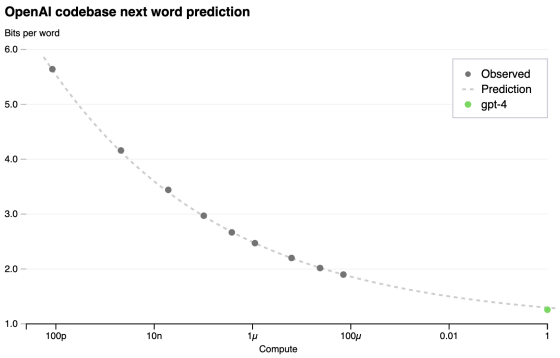


Figure 1: GPT-4 scaling law [Ope+24].

What if we view:

- The x-axis (Compute) as a discrete **index** that is taken to infinity,
- Each network as an indexed **element** in a "space of neural networks",
- The performance metric as a **test function** on this space,

Question: Would the scaling law be a consequence of the convergence rate of the **scaling limit**?

Scaling Laws and Scaling Limits

Roughly speaking

$$\text{Compute} \propto \text{Width}^2 \times \text{Depth} \times \text{Batch Size} \times \text{Number of Training Steps}\,.$$

This would correspond to studying the **joint limit** of width, depth, data, and training iterations.

Due to the technical challenges of studying the full joint limit, most existing work primarily focuses on the **depth** and **width** limits, where data and training remain **finite**.

However, even within this limited range, we already understand several **distinct limiting regimes**, with vastly different behaviours.

Kernel and Feature Learning Regimes

The most tractable and well studied infinite-width limit is the **kernel regime** first introduced by Neal [Nea95]. However, it was quickly understood that the **features were not learned** (or updated) throughout training in the limit.

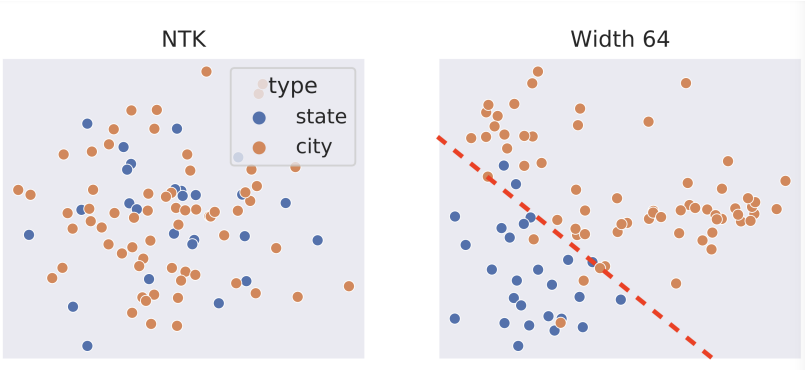


Figure 2: Figure from [YH21].

Kernel and Feature Learning Regimes

Yang and Hu [YH21] then proposed the μP-scaling, which strengthens the updates of the hidden layers while maintaining stability, to remedy this problem.

More precisely, for a linear layer with width n (to be taken to infinity)

$$h_{\ell+1} = \frac{1}{\sqrt{n}} W_\ell h_\ell \,,$$

the authors considered the change $\Delta W_\ell = W_\ell(k+1) - W_\ell(k)$ and defined the **maximal update** condition as

$$\Delta_{W_\ell} h_{\ell+1} = \frac{1}{\sqrt{n}} \Delta W_\ell h_\ell = \Theta(1) \,,$$

where $\Theta(1)$ refers to entries of the vector being asymptotically tight and not converging to zero as width $n \to \infty$. Later, for infinite-depth ResNets with

$$h_{\ell+1} = h_\ell + \frac{1}{d^\alpha} \frac{1}{\sqrt{n}} W_\ell h_\ell \,, \quad \alpha \in [1/2, 1] \,,$$

we would require $\Delta_{W_\ell} h_{\ell+1} \in \Theta(\frac{1}{d})$ instead for stability [Bor+23; Yan+24; Dey+25].

Kernel and Feature Learning Regimes

One of the most important empirical consequences of feature learning is **hyperparameter transfer**

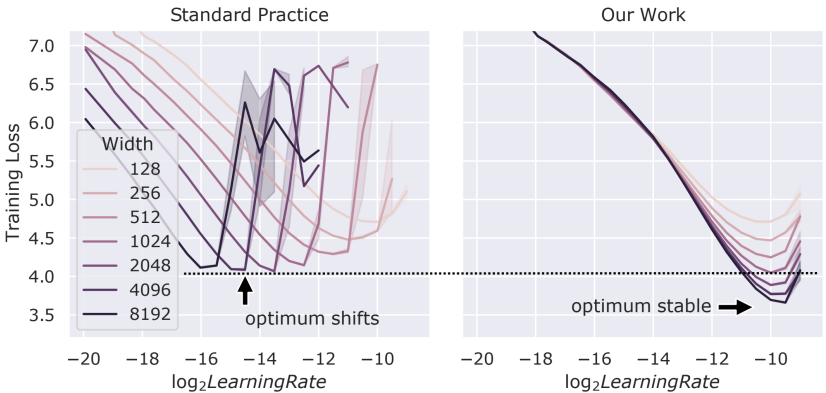


Figure 3: Figure from [Yan+22].

The Proportional Scaling Limit

However, μP is not the only scaling regime that yields feature learning.

Hanin and Nica [HN19; HN20] also showed that increasing depth **proportionally** to width will also yield the neural tangent kernel updating, i.e. feature learning, albeit the result is only a scale computation for one data point and one step of training.

In an earlier work [LNR22], we showed that it is necessary to weaken the non-linearities as depth increases, a method first introduced by Martens et al. [Mar+21] and Zhang et al. [ZBM22].

Furthermore, we provide a precise characterization of the forward pass via a limiting SDE for the feature covariance matrix.

We further extended this approach to Transformers [Noc+23], which resolved the rank collapse and vanishing gradient problems in deep Transformers.

In this work, we will finally extend this SDE approach to **one step of training**, and prescribe the exact learning rate scaling required for feature learning.

The Proportional Scaling Limit

Let us briefly review the SDE approach of Li et al. [LNR22].

Starting with a linear network with only one input data point

$$h_{\ell+1} = \frac{1}{\sqrt{n}} W_\ell h_\ell \,, \quad h_1 = \frac{1}{\sqrt{n_0}} W_0 x \,,$$

where $h_\ell \in \mathbb{R}^n, x \in \mathbb{R}^{n_0}$ and $W_{\ell,ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$.

Observe that $h_{\ell+1}|\sigma(h_\ell) \sim \mathcal{N}(0, \frac{1}{2}\|h_\ell\|^2 I_n)$. Letting $\Phi_\ell = \frac{1}{n}\|h_\ell\|^2$ be the feature kernel, we have

$$\Phi_{\ell+1} = \frac{1}{n}\left\|\frac{1}{\sqrt{n}} W_\ell h_\ell\right\|^2 = \frac{1}{n}\left\|\frac{1}{\sqrt{n}} W_\ell \frac{h_\ell}{\|h_\ell\|}\right\|^2 \|h_\ell\|^2 \overset{d}{=} \frac{1}{n}\chi_n^2 \Phi_\ell = \Phi_\ell + \Phi_\ell\left(\frac{1}{n}\chi_n^2 - 1\right) ,$$

where we note that $\frac{1}{n}\chi_n^2 - 1$ has zero mean and variance $\frac{2}{n}$.

Essentially $\Phi_{\ell+1} \overset{d}{=} \Phi_\ell + \sqrt{\frac{2}{n}}\Phi_\ell \xi_\ell$ is a **discretization** of the SDE $d\Phi_\tau = \sqrt{2}\Phi_\tau \, dB_\tau$ with step size $\frac{1}{n}$.

Therefore the total **depth time** is number of layers multiplied by the step size, which is $\bar{\tau} = \frac{d}{n}$.

The Proportional Scaling Limit

We can further extend this to multiple data points, which requires us to consider

$$h_\ell = [h_\ell^1, \ldots, h_\ell^m] \in \mathbb{R}^{n \times m}\,, \quad \Phi_\ell = \frac{1}{n} h_\ell^\top h_\ell \in \mathbb{R}^{m \times m}\,.$$

If we view Φ_ℓ as the upper triangular entries flattened into a vector in $\mathbb{R}^{m(m+1)/2}$, we can show that Φ_ℓ is a discretization of an SDE again with step size $\frac{1}{n}$

$$d\Phi_\tau = \Sigma(\Phi_\tau)^{1/2}\, dB_\tau\,, \quad \Sigma(\Phi)^{\alpha\beta,\gamma\delta} = \Phi^{\alpha\gamma}\Phi^{\beta\delta} + \Phi^{\alpha\delta}\Phi^{\beta\gamma}\,.$$

Similarly, activation functions must therefore also contribute a factor of step size $\frac{1}{n}$ as well in order to remain stable, e.g. a leaky ReLU with slopes $1 + \frac{1}{\sqrt{n}}$, which turns out to yield a drift term in the SDE

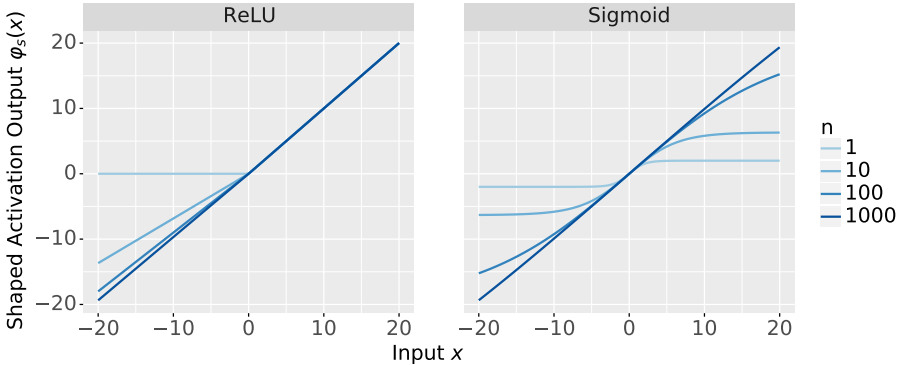
$$\Phi_{\ell+1} = \Phi_\ell + \frac{b(\Phi_\ell)}{n} + \frac{1}{\sqrt{n}}\Sigma(\Phi_\ell)^{1/2}\xi_\ell \quad \to \quad d\Phi_\tau = b(\Phi_\tau)\, d\tau + \Sigma(\Phi_\tau)^{1/2}\, dB_\tau\,.$$

$b(\Phi)$ can be interpreted as an **instantaneous** or downscaled version of the Cho and Saul kernel [CS09].

Shaped Activation

Martens et al. [Mar+21] and Zhang et al. [ZBM22] proposed numerically optimizing the **shape** of activation functions to achieve a target output correlation.

We observed the resulting activations are closer to **linear** as the network size increases.

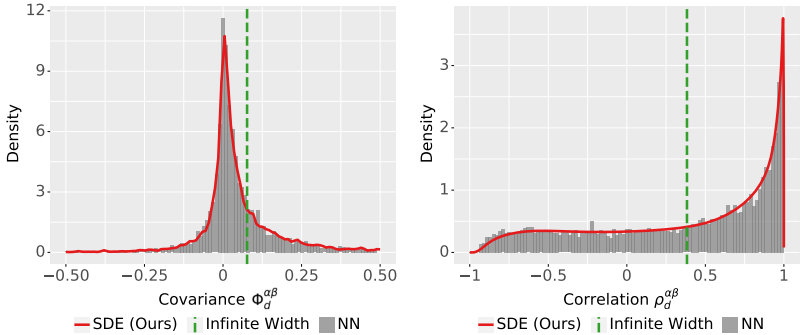


Neural Covariance SDE - Remarks

To recover the output distribution, observe that we can condition on the penultimate layer to get

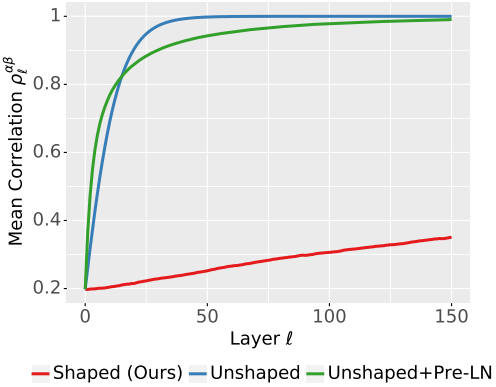
$$[f^{\alpha}]_{\alpha=1}^{m} | \mathcal{F}_d \sim N\left(0\,, \Phi_d \otimes I_{n_{\text{out}}}\right) .$$

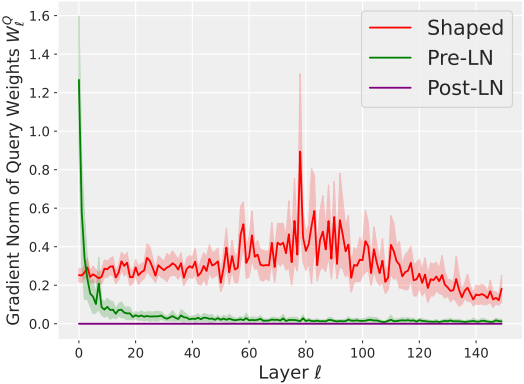
Therefore we can sample Φ_τ from the covariance SDE at time $\frac{d}{n} \to \overline{\tau}$, then sample $[f^{\alpha}]_{\alpha=1}^{m}$ conditioned on $\Phi_{\overline{\tau}}$.



Application - Shaped Transformer

We extended the **shaping** approach to Transformer layers, remedying vanishing gradients [NL**L**+23].





Noci, **L**., Li, He, Hofmann, Maddison, and Roy. "The Shaped Transformer..." NeurIPS 2023.

Spectrum of the Covariance SDE

Let us consider a **linear** neural network, which leads to $d\Phi_\tau = \Sigma(\Phi_\tau)^{1/2}\, dB_\tau$.

It turns out that Σ is the **affine-invariant cometric**, which arose from the following property. Let $P_\tau : \Phi_0 \mapsto \Phi_\tau$ be the random flow map of the SDE, then

$$AP_\tau(\Phi)A^\top \overset{d}{=} P_\tau(A\Phi A^\top)\,, \quad \forall A \in \mathrm{GL}(m)\,.$$

Without loss of generality, we can start the diffusion at $\Phi_0 = I_m$ since $\Phi_\tau \overset{d}{=} \Phi_0^{1/2} P_\tau(I_m)\Phi_0^{1/2}$. This invariant property also allows us to derive the following.

Theorem (LDN+26)

Let λ_i be the i-th smallest eigenvalue of Φ_τ, then

$$d\lambda_i = \sqrt{2}\lambda_i\, dB_i + \sum_{j\neq i} \frac{\lambda_i\lambda_j}{\lambda_i - \lambda_j}\, d\tau\,.$$

L., de Dios Pont, Nica, and Roy. "Geometric Dyson Brownian Motion..." In Prep.

Spectrum of the Covariance SDE

In order for the $m \to \infty$ limit to exist, we need to consider the time change $\tau \mapsto \frac{\tau}{m}$, which corresponds to maintaining the ratio $\frac{dm}{n} = \tau$, also considered in Bayesian neural network settings [HZ23; HZ24].

Let $\rho_\tau^{(m)}(dx) = \frac{1}{m}\sum_i \delta_{\lambda_i}$ be the empirical spectral distribution, and let $G_\tau^{(m)}(z) = \int \frac{x}{x-z}\, d\rho_\tau^{(m)}(x)$ be the T-transform. Then we have the following result.

Theorem (LDN+26)

As $m \to \infty$, we have that $G_\tau^{(m)}(z) \to G_\tau(z)$ which is the unique (viscosity) solution to

$$\partial_\tau G_\tau(z) = -z G_\tau(z) \partial_z G_\tau(z)\,.$$

Furthermore, if $\rho_0(dx) = \delta_1(dx)$, then we have that $G_\tau(z)$ solves the following fixed point equation

$$G_\tau(z) = \frac{1}{z e^{-\tau G_\tau(z)} - 1}\,.$$

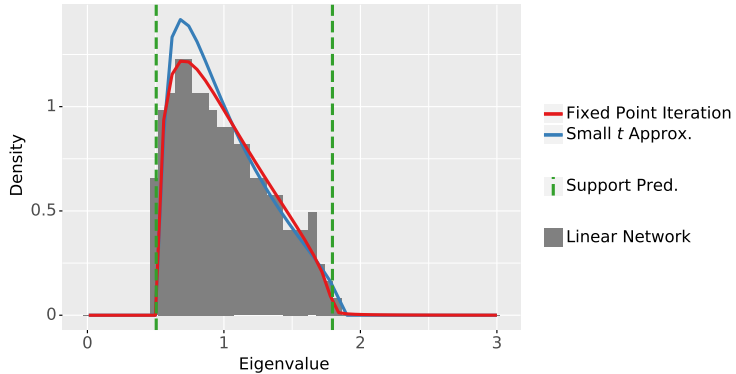
Let ν_τ be the distribution with T-transform $G_\tau(z)$, then $\rho_\tau = \nu_\tau \boxtimes \rho_0$.

Spectrum of the Covariance SDE

The fixed point equation allows us to recover a small τ approximation

$$\rho_\tau(x) = \frac{\sqrt{(x-\lambda_-)(\lambda_+ - x)}}{2\pi x^2 \tau} + O(\tau^2)\,, \qquad \text{where } \sigma^2 = 1+\tau, \lambda = \frac{\tau}{1+\tau}, \lambda_\pm = \sigma^2(1 \pm \sqrt{\lambda})^2\,,$$

and observe it is almost the **Marchenko–Pastur** law.



Gaussian Conditioning and the Backward Pass

So far we have relied on the weights being iid. Gaussian, so how can we extend this to **training**?

Lemma (Gaussian Conditioning)

Let $W \in \mathbb{R}^{n\times n}$ have iid $\mathcal{N}(0,1)$ entries, and let $\varphi, g \in \mathbb{R}^{n\times m}$ be deterministic. Denote by P_φ, P_g the orthogonal projections onto their column spans. Then

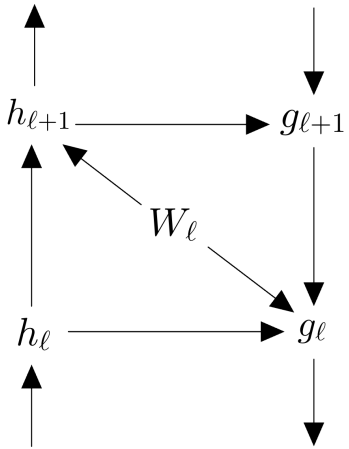
$$W \,\big|\, \sigma(W\varphi, W^\top g) \overset{d}{=} WP_\varphi + P_g W - P_g W P_\varphi + P_g^\perp \widetilde{W} P_\varphi^\perp,$$

where the first three terms are measurable with respect to $\sigma(W\varphi, W^\top g)$, and $\widetilde{W}$ is an independent iid Gaussian copy of W.

This lemma allows us to **isolate** exactly which part of the weight matrix has been used in a previous forward or backward pass, and which part remains (almost) iid Gaussian.

In particular, Gerbelot et al. [Ger+24] used this conditioning approach to derive a rigorous version of DMFT equations.

Gaussian Conditioning and the Backward Pass



Let us consider the first **backward pass** of a linear network with one data point, where we have for output $f = \frac{1}{\sqrt{n}} W_d h_d$ with $W_d \in \mathbb{R}^{1\times n}$, and the backward pass is given by

$$h_{\ell+1} = \frac{1}{\sqrt{n}} W_\ell h_\ell \,, \quad g_\ell = \sqrt{n}\frac{\partial f}{\partial h_\ell} \,, \quad g_\ell = \frac{1}{\sqrt{n}} W_\ell^\top g_{\ell+1} \,.$$

Conditioning on the forward passes $\{h_k\}_{k=1}^{d+1}$, we have

$$W_\ell | \sigma(W_\ell h_\ell)) \overset{d}{=} W_\ell P_{h_\ell} + \widetilde{W}_\ell P_{h_\ell}^\perp \,, \quad P_{h_\ell} = \frac{h_\ell h_\ell^\top}{\|h_\ell\|^2} \,.$$

Note this conditioning essentially makes the backward pass look like a "Markov chain" update, where all the randomness comes from $\widetilde{W}_\ell$ which is independent of the forward pass.

Gaussian Conditioning and the Backward Pass

In this linear one sample case, it so happens that

$$Q_\ell = \frac{1}{\sqrt{n}}\langle h_\ell, g_\ell \rangle = f\,,$$

which means Q_ℓ is $\Theta(1)$ and constant in ℓ.

This further gives us the backward recursion

$$g_\ell = \frac{\langle h_{\ell+1}, g_{\ell+1}\rangle}{\|h_\ell\|^2} h_\ell + \frac{1}{\sqrt{n}} P_{h_\ell}^\perp \widetilde{W}_\ell^\top g_{\ell+1} = \frac{Q_{\ell+1}}{\sqrt{n}\Phi_\ell} h_\ell + \sqrt{G_{\ell+1}} P_{h_\ell}^\perp \zeta_\ell\,,$$

where $G_\ell = \frac{1}{n}\|g_\ell\|^2$, and $\zeta_\ell \sim \mathcal{N}(0, I_n)$ independent of the forward pass.

A similar calculation to the forward pass leads us to

$$G_\ell = G_{\ell+1}\frac{\chi^2_{n-1}}{n} + \frac{Q_\ell^2}{n\Phi_\ell} = G_{\ell+1} + \frac{1}{n}\left(\frac{Q_\ell^2}{\Phi_\ell} - G_{\ell+1}\right) + \sqrt{\frac{2}{n}} G_{\ell+1}\xi_\ell + o_p(n^{-1/2})\,,$$

where ξ_ℓ are iid with zero mean and unit variance.

Gaussian Conditioning and the Backward Pass

Consequently, we can view G_ℓ as a discretization of the SDE

$$dG_\tau = \left(\frac{Q_\tau^2}{\Phi_\tau} - G_\tau \right) d\tau + \sqrt{2} G_\tau \bullet dB_\tau \,,$$

where $\bullet$ denotes the right hand Itô integral, since we are going **backwards** in depth.

Remark: We can derive a backward version of the covariance SDE for G **without** requiring the weights W_ℓ to be **iid.** Gaussian.

We further note the neural tangent kernel has an interesting form if we normalize it by depth

$$\frac{1}{d} \|\nabla_\theta f\|^2 = \frac{1}{d} \sum_{\ell=1}^{d} \Phi_\ell G_{\ell+1} \to \int_0^{\bar{\tau}} \Phi_\tau G_\tau \, d\tau \,.$$

In general Q_τ is not constant in τ, but instead Q_τ, G_τ together satisfy a **system of backward (matrix) SDEs** (in the right hand Itô sense) with drift terms from the activation.

One Step of Training

To study the scaling limit of training dynamics, we need to choose a learning rate scale to ensure the output update is stable

$$\Delta f(x) \approx -\eta \underbrace{\sum_{\ell=1}^{d}}_{\Theta(d)} \underbrace{\Phi_\ell G_{\ell+1}(f(x)-y)}_{\Theta(1)},$$

then we **necessarily** need to choose $\eta = \Theta(\frac{1}{d})$, or more precisely let $\eta = \frac{\eta_{\text{base}}}{d}$ for some $\eta_{\text{base}} = \Theta(1)$.

We note this is **distinct** from the μP prescription of $\eta = \eta_{\text{base}} n$, which is increasing with scale.

One can analyze the update of the feature kernel by splitting it into three terms

$$\Delta\Phi_\ell = \underbrace{\frac{1}{n}\langle \Delta h_\ell, h_\ell\rangle}_{=:R_\ell^{\varphi}} + \underbrace{\frac{1}{n}\langle h_\ell, \Delta h_\ell\rangle}_{=:R_\ell^{\varphi}} + \underbrace{\frac{1}{n}\langle \Delta h_\ell, \Delta h_\ell\rangle}_{\dot{\Phi}_\ell}.$$

We will also need to analyze an additional kernel $R^g = \frac{1}{\sqrt{n}}\langle \Delta h_\ell, g_\ell\rangle$.

One Step of Training

For example, let us consider the recursion for δh_ℓ

$$\Delta h_{\ell+1} = -\frac{\eta_{\text{base}}}{d}\frac{1}{\sqrt{n}}W_\ell h_\ell (f(x)-y) + \frac{1}{\sqrt{n}}W_\ell \Delta h_\ell = \frac{\eta_{\text{base}}}{n^{3/2}} r g_\ell(\Phi_\ell + R_\ell^\varphi) + \frac{1}{\sqrt{n}}W_\ell \Delta h_\ell \,.$$

Turns out the first term is lower order so we have the recursion

$$\dot{\Phi}_{\ell+1} = \frac{1}{n}\left\langle \frac{1}{\sqrt{n}}W_\ell \Delta h_\ell, \frac{1}{\sqrt{n}}W_\ell \Delta h_\ell \right\rangle + \text{ lower order terms.}$$

Using Gaussian conditioning we can derive

$$\frac{1}{\sqrt{n}}W_\ell \Delta h_\ell = \frac{R_\ell^\varphi}{\Phi_\ell}h_{\ell+1} + \frac{R_\ell^z - \frac{R_\ell^\varphi}{\Phi_\ell}Q_\ell}{G_{\ell+1}}\frac{g_{\ell+1}}{\sqrt{n}} + \frac{1}{\sqrt{n}}P^\perp_{g_{\ell+1}}\widetilde{W}_\ell P^\perp_{h_\ell}\Delta h_\ell \,.$$

Finally expanding the inner product gives us the desired recursion for $\dot{\Phi}_\ell$.

One Step of Training

The final formulae look a bit complicated...

$$
\begin{aligned}
dR_\tau^g &= \frac{\eta_{\text{base}}}{\bar{\tau}}\, r\, \Phi_\tau G_\tau\, d\tau, \\
dR_\tau^\varphi &= \left(\frac{Q_\tau}{G_\tau} R_\tau^g - \frac{Q_\tau^2}{\Phi_\tau G_\tau} R_\tau^\varphi\right) d\tau + \frac{R_\tau^\varphi}{\Phi_\tau}\, d\Phi_\tau + \sqrt{\Phi_\tau \dot{\Phi}_\tau - (R_\tau^\varphi)^2}\, dB_\tau^{R^\varphi}, \\
d\dot{\Phi}_\tau &= \left(\frac{(R_\tau^g)^2}{G_\tau} - \frac{Q_\tau^2 (R_\tau^\varphi)^2}{\Phi_\tau^2 G_\tau} - \left(\dot{\Phi}_\tau - \frac{(R_\tau^\varphi)^2}{\Phi_\tau}\right)\right) d\tau + \frac{(R_\tau^\varphi)^2}{\Phi_\tau^2}\, d\Phi_\tau \\
&\quad + 2R_\tau^\varphi \sqrt{\frac{1}{\Phi_\tau}\left(\dot{\Phi}_\tau - \frac{(R_\tau^\varphi)^2}{\Phi_\tau}\right)}\, dB_\tau^{R^\varphi} + \sqrt{2}\left(\dot{\Phi}_\tau - \frac{(R_\tau^\varphi)^2}{\Phi_\tau}\right) dB_\tau^{\dot{\Phi}},
\end{aligned}
$$

where $r = f(x) - y$ is the residual, and $B^{R^\varphi}, B^{\dot{\Phi}}$ are independent Brownian motions.

Remark: The convergence to SDEs implies the current scaling of learning rate is **stable**.

We have also derived the full **non-linear** and multiple data point version of this SDE system. Furthermore, we believe this process can be further iterated inductively to yield a characterization of **finitely many steps** of training, in a similar spirit to the DMFT equations.

Hyperparameter Transfer

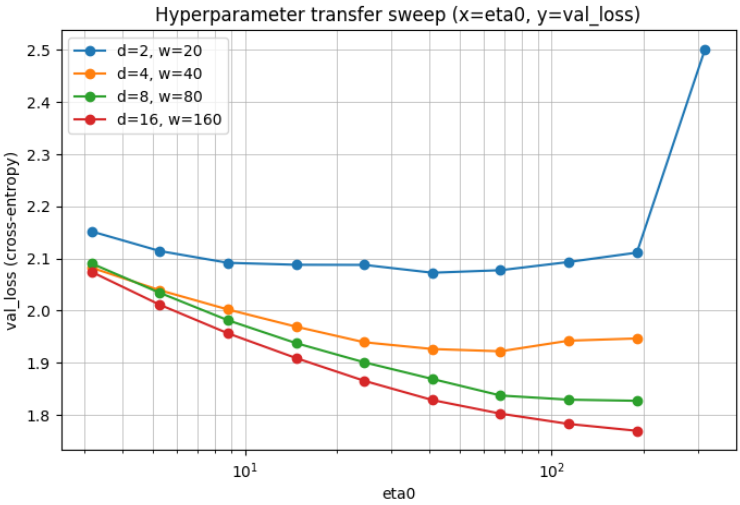


Figure 6: Preliminary Hyperparameter Transfer Results in the proportional limit.

On the Notion of Maximal Update

However, one can show that this scaling **does not satisfy** maximal update

$$\Delta_{W_\ell} h_{\ell+1} = \frac{1}{\sqrt{n}} \Delta W_\ell h_\ell = \Theta\left(\frac{1}{d\sqrt{n}}\right) < \Theta\left(\frac{1}{d}\right) .$$

The reason is a bit subtle: the forward and backward neuron inner product is on the CLT scale

$$g_\ell := \sqrt{n} \frac{\partial f}{\partial h_\ell} \implies Q_\ell = \frac{1}{\sqrt{n}} \langle h_\ell, g_\ell \rangle = \Theta(1) \text{ and } R^g_\ell = \frac{1}{\sqrt{n}} \langle \Delta h_\ell, g_\ell \rangle = \Theta(1) .$$

In fact, we can show that this is the **maximum** scale of update to the **kernels** Φ_ℓ, such that the update it retains the **most number of terms** in the limit, while also does not lead to any diverging terms.

Furthermore, we note our notion of maximal learning is **not** a measure of the update scale on **neurons** h_ℓ directly as in μP.

Given that there is hyperparameter transfer and that feature kernel updates in this regime, we should **rethink the desiderata** of feature learning in terms of **kernels**.

Summary and Future Directions

Analyzing scaling limits allows us to prescribe **how to** scale up the architecture and algorithms, providing **cheap estimates** for optimal hyperparameters of large networks.

In particular, we derive a system of forward-backward SDEs that characterize the **one step of training** in the proportional limit, and show that this scaling exhibits **feature learning** and **hyperparameter transfer**.

There are several interesting directions to explore given this set of results:

- Extend the scaling to Shaped Transformers [Noc+23], and test the **scaling performance** against μP and CompleteP [Dey+25].
- Using our new results on spectrum of linear network covariance kernel [Li+26], we can analyze the effect of one step of update in this regime where the **number of data points** also diverges.
- Extend the framework to finitely many steps of training, forming a DMFT like system of forward-backward SDEs.

Appendix - Neural Covariance SDE

Theorem (LNR22)

Consider the shaped ReLU-like activation

$$\varphi_s(x) = s_+ \max(x, 0) + s_- \min(x, 0)\,, \qquad \text{with } s_\pm = 1 + \frac{c_\pm}{\sqrt{n}}\,.$$

As $d, n \to \infty$ and $\frac{d}{n} \to \overline{\tau} > 0$, the upper triangular entries of the covariance matrix $\Phi_\ell = \frac{c}{n}[\langle \varphi_\ell^\alpha, \varphi_\ell^\beta \rangle]_{\alpha \le \beta}$ (flattened to a vector) converge to the solution of the following SDE

$$d\Phi_\tau = b(\Phi_\tau)\, d\tau + \Sigma(\Phi_\tau)^{1/2}\, dB_\tau\,, \quad \Phi_0 = \frac{1}{n_{\text{in}}}[\langle x^\alpha, x^\beta \rangle]_{\alpha \le \beta}\,,$$

where $\Sigma(\Phi)_{\alpha\beta,\gamma\delta} = \Phi^{\alpha\gamma}\Phi^{\beta\delta} + \Phi^{\alpha\delta}\Phi^{\beta\gamma}$, $b(\Phi)_{\alpha\beta} = \nu(\rho^{\alpha\beta})\sqrt{\Phi^{\alpha\alpha}\Phi^{\beta\beta}}$ with $\rho^{\alpha\beta} = \frac{\Phi^{\alpha\beta}}{\sqrt{\Phi^{\alpha\alpha}\Phi^{\beta\beta}}}$ and

$$\nu(\rho) = \frac{(c_+ - c_-)^2}{2\pi}\left[\sqrt{1-\rho^2} - \arccos(\rho)\rho\right]\,.$$

L., Nica, and Roy. "The Neural Covariance SDE..." NeurIPS 2022 (Oral).

References

- [Bor+23] B. Bordelon, L. Noci, M. B. Li, B. Hanin, and C. Pehlevan. "Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit". *arXiv preprint arXiv:2309.16620* (2023).
- [CS09] Y. Cho and L. K. Saul. "Kernel methods for deep learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2009, pp. 342–350.
- [Dey+25] N. Dey, B. C. Zhang, L. Noci, M. Li, B. Bordelon, S. Bergsma, C. Pehlevan, B. Hanin, and J. Hestness. "Don't be lazy: CompleteP enables compute-efficient deep transformers". *arXiv preprint arXiv:2505.01618* (2025).
- [Ger+24] C. Gerbelot, E. Troiani, F. Mignacco, F. Krzakala, and L. Zdeborova. "Rigorous dynamical mean-field theory for stochastic gradient descent methods". *SIAM Journal on Mathematics of Data Science* 6.2 (2024), pp. 400–427.
- [HN19] B. Hanin and M. Nica. "Products of many large random matrices and gradients in deep neural networks". *Communications in Mathematical Physics* (2019), pp. 1–36.

References (cont.)

- [HN20] B. Hanin and M. Nica. "Finite Depth and Width Corrections to the Neural Tangent Kernel". In: *Int. Conf. Learning Representations (ICLR)*. 2020.
- [HZ23] B. Hanin and A. Zlokapa. "Bayesian interpolation with deep linear networks". *Proceedings of the National Academy of Sciences* 120.23 (2023), e2301345120.
- [HZ24] B. Hanin and A. Zlokapa. "Bayesian inference with deep weakly nonlinear networks". *arXiv preprint arXiv:2405.16630* (2024).
- [Li+26] M. Li, J. de Dios Pont, M. Nica, and D. M. Roy. "Geometric Dyson Brownian Motion and the Free Log-Normal for Minor of Products of Random Matrices". In Preparation. 2026.
- [LNR22] M. Li, M. Nica, and D. Roy. "The neural covariance SDE: Shaped infinite depth-and-width networks at initialization". *Advances in Neural Information Processing Systems* 35 (2022), pp. 10795–10808.

References (cont.)

- [Mar+21] J. Martens, A. Ballard, G. Desjardins, G. Swirszcz, V. Dalibard, J. Sohl-Dickstein, and S. S. Schoenholz. "Rapid training of deep neural networks without skip connections or normalization layers using Deep Kernel Shaping". *arXiv preprint arXiv:2110.01765* (2021).
- [Nea95] R. M. Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 1995.
- [Noc+23] L. Noci, C. Li, M. B. Li, B. He, T. Hofmann, C. Maddison, and D. M. Roy. "The shaped transformer: Attention models in the infinite depth-and-width limit". *arXiv preprint arXiv:2306.17759* (2023).
- [Ope+24] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].
- [PB20] M. Potters and J.-P. Bouchaud. *A First Course in Random Matrix Theory: For Physicists, Engineers and Data Scientists*. Cambridge University Press, 2020.

References (cont.)

- [Yan+22] 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 preprint arXiv:2203.03466* (2022).
- [Yan+24] G. Yang, D. Yu, C. Zhu, and S. Hayou. "Tensor Programs VI: Feature Learning in Infinite Depth Neural Networks". In: *The Twelfth International Conference on Learning Representations*. 2024.
- [YH21] G. Yang and E. J. Hu. "Feature Learning in Infinite-Width Neural Networks". In: *Int. Conf. Machine Learning (ICML)*. 2021. arXiv: 2011.14522.
- [ZBM22] G. Zhang, A. Botev, and J. Martens. "Deep Learning without Shortcuts: Shaping the Kernel with Tailored Rectifiers". *arXiv preprint arXiv:2203.08120* (2022).