A Experimental details

Datasets: We considered the MNIST [10], Fashion-MNIST [64], and CIFAR-10 [42] datasets. We standardized the images and used one-hot encoding for the labels.

Models: We considered fully connected networks (FCNs), Myrtle family CNNs [58] and ResNets (version 1) [28] trained using the JAX [7], and Flax libraries [29]. We use d and w to denote the depth and width of the network. Below, we provide additional details of the models and clarify what width corresponds to for CNNs and ResNets.

1. **FCNs:** We considered ReLU FCNs with constant width w in Neural Tangent Parameterization (NTP) / Standard Parameterization (SP), initialized at criticality [52]. The models do not include bias or normalization. The forward pass of the pre-activations from layer l to $l+1$ is given by

$$h_i^{l+1} = \gamma^l \sum_j^w W_{ij}^l \phi(h_j^l), \tag{2}$$

   where $\phi(.)$ is the ReLU activation and γ^l is a constant. For NTP, $\gamma^l = 2/\sqrt{w}$ and the weights W^l are initialized using normal distribution, i.e., $W_{ij}^l \sim \mathcal{N}(0,1)$. For SP, $\gamma^l = 1$ and the weights W^l are initialized as $W_{ij}^l \sim \mathcal{N}(0, 2/w)$. For the last layer, we have $\gamma^L = 1/\sqrt{w}$ for NTP and $W_{ij}^L \sim \mathcal{N}(0, 1/w)$ for SP.

   We considered $d \in \{4, 8, 16\}$ and $w \in \{256, 512, 1024, 2048\}$. For $d/w \gtrsim 1/16$, the dynamics is noisier, and it becomes challenging to separate the underlying deterministic dynamics from random fluctuations (see Appendix E).

2. **CNNs:** We considered Myrtle family ReLU CNNs [58] without any bias or normalization in Standard Parameterization (SP), initialized using He initialization [28]. The above model uses a fixed number of channels in each layer, which we refer to as the width of the network. In this case, the forward pass equations for the pre-activations from layer l to layer $l+1$ are given by

$$h_i^{l+1}(\alpha) = \sum_j^w \sum_{\beta \in ker} W_{ij}^{l+1}(\beta)\phi(h_i^l(\alpha+\beta)), \tag{3}$$

   where α, β label the spacial location. The weights are initialized as $W_{ij}^l(\beta) \sim \mathcal{N}(0, 2/k^2w)$, where k is the filter size. We considered $d \in \{5, 7, 10\}$ and $w \in \{64, 128, 256\}$.

3. **ResNets:** We considered version 1 ResNet [28] implementations from Flax examples without Batch Norm or regularization. For ResNets, width corresponds to the number of channels in the first block. For example, the standard ResNet-18 has four blocks with widths $[w, 2w, 4w, 8w]$, with $w = 64$. We refer to w as the width or the widening factor. We considered ResNet-18 and ResNet-34 with $w \in \{32, 64, 128\}$.

All the models are trained with the average loss over the batch $\mathcal{D}_B = \{(x_\mu, y_\mu)\}_{\mu=1}^B$, i.e., $L(x, y_{\mathcal{D}_B}) = 1/B \sum_{\mu=1}^B \ell(x_\mu, y_\mu)$, where $\ell(.)$ is the loss function. This normalization, along with initialization, ensures that the loss is $\mathcal{O}(1)$ at initialization.

Bias: Throughout this work, we have primarily focused on models without any bias for simplicity. In Appendix K, we demonstrate that bias does not have an appreciable impact on the results.

Batch size: We use a batch size of 512 and scale the learning rate as $\eta = c/\lambda_0^H$ in all our experiments, unless specified. Appendix J shows results for a smaller batch size $B = 32$.

Learning rate: We scale the learning rate constant as $c = 2^x$, with $x \in \{-1.0, \ldots x_{max}\}$ in steps of 0.1. Here, x_{max} is related to the maximum learning rate constant as $c_{max} = 2^{x_{max}}$.

Sharpness measurement: We measure sharpness using the power iteration method with 20 iterations. We found that 20 iterations suffice both for MSE and cross-entropy loss. For MSE loss, we use $m = 2048$ randomly selected training examples for evaluating sharpness at each step. In comparison, we found that cross-entropy requires a large number of training examples to obtain a good approximation of sharpness. Given the computational constraints, we use 4096 training examples to approximate sharpness for cross-entropy loss.

Averages over initialization and SGD runs: All the critical constants depend on both the random initializations and the SGD runs. In our experiments, we found that the fluctuations from initialization at large d/w outweigh the randomness coming from different SGD runs. Thus, we focus on initialization averages in all our experiments.

A.1 Compute usage

We utilized different computational resources depending on the task complexity. For less demanding tasks, we performed computation for a total of 2800 hours, utilizing a seventh of an NVIDIA A100 GPU. For more computationally intensive tasks, we utilized a full NVIDIA A100 GPU for a total 300 hours.

A.2 Reproducibility

We provide a notebook in the supplementary material that reproduces the main results of the paper.

A.3 Details of Figures in the main text:

Figure 1: A shallow CNN ($d = 5$, $w = 128$) in SP trained on the CIFAR-10 dataset with MSE loss for 1000 epochs using SGD with learning rates $\eta = c/\lambda_0^H$ and batch size $B = 512$. We measure sharpness at every step for the first epoch, every epoch between 10 and 100 epochs, and every 10 epochs beyond 100.

Figure 2: (top panel) A wide ($d = 5$, $w = 512$) and (bottom panel) a deep CNN ($d = 10, w = 128$) in SP trained on the CIFAR-10 dataset with MSE loss for $t = 10$ steps using vanilla SGD with learning rates $\eta = c/\lambda_0^H$ and batch size $B = 512$.

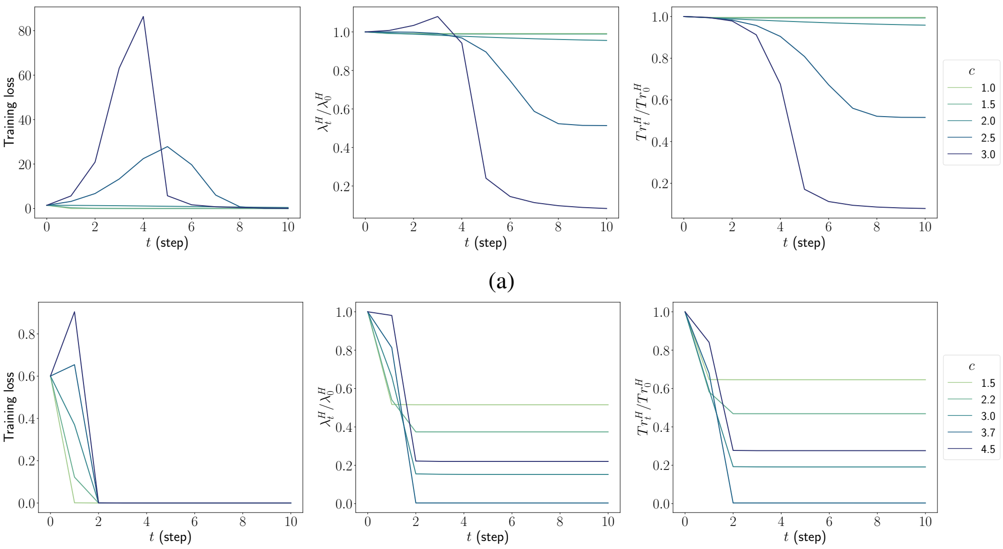
Figure 3: (a) FCNs in NTP with $d = 8$ and $w \in \{256, 512, 1024, 2048\}$trained on the MNIST dataset, (b) CNNs in SP with $d = 7$ and $w \in \{64, 128, 256, 512\}$ trained on the Fashion-MNIST dataset, (c) ResNet in SP with $d = 18$ and $w \in \{32, 64, 128\}$ trained on the CIFAR-10 dataset (without batch normalization). Each data point in the figure represents an average of ten distinct initialization, and the solid lines represent a two-degree polynomial $y = a + bx + cx^2$ fitted to the raw data points. Here, where $x = 1/w$, and y can take on one of three values: c_{loss}, c_{sharp} and c_{max}. We show the error bars in Appendix C only as they may give an impression that the inequality $c_{loss} \leq c_{sharp} \leq c_{max}$ can be violated.

Figure 4: (a) FCNs in NTP with $d \in \{4, 8, 16\}$ and $w \in \{256, 512, 1024, 2048\}$trained on the MNIST dataset, (b) CNNs in SP with $d \in \{5, 7, 10\}$ and $w \in \{64, 128, 256, 512\}$ trained on the Fashion-MNIST dataset, (c) ResNet in SP with $d \in \{18, 34\}$ and $w \in \{32, 64, 128\}$ trained on the CIFAR-10 dataset (without batch normalization).

Figure 5: Normalized sharpness measured at $c\tau = 200$ against the learning rate constant for 7-layer CNNs in SP trained on the CIFAR-10 dataset, with $w \in \{128, 256, 512\}$. Each data point is an average over five random initialization. Smoothening details are provided in Appendix I.2.

Figure 6: The phase diagram of the uv model trained with MSE loss using gradient descent with (a) the top eigenvalue of Hessian λ_t^H, (b) the trace of Hessian $\mathrm{tr}(H_t)$ and (c) the square of the Frobenius norm $\mathrm{tr}(H_t^T H_t)$ used as a measure of sharpness. In (a), the learning rate is scaled as $\eta = c/\lambda_0^H$, while in (b) and (c), the learning rate is scaled as $\eta = k/\mathrm{tr}(H_0)$. The vertical dashed line shows $c = 2$ ($k = 2$) for reference. Each data point is an average over 500 random initializations.

Figure 7: Training trajectories of the uv model with (a, b) large ($w = 512$) and (c, d) small ($w = 2$) width, trained for $t = 10$ training steps on a single example $(x, y) = (1, 0)$ with MSE loss using vanilla gradient descent with learning rates (a, c) $c = 0.5$ and (b, d) $c = 2.50$.



(a)

(b)

Figure 8: Training trajectories of the uv model with (a) large ($w = 512$) and (v) a small ($w = 2$) widths trained for $t = 10$ training steps on a single example $(x, y) = (1, 0)$. For the wide network, $c_{loss} = 2.1$, $c_{sharp} = 2.6$, $c_{max} = 4.0$, and for the narrow network, $c_{loss} = 3.74$, $c_{sharp} = 4.63$, $c_{max} = 4.93$.

B Additional results for the uv model

B.1 Details of the model

Consider a two-layer linear network in (NTP) with unit input-output dimensions

$$f(x) = \frac{1}{\sqrt{w}} v^T u x, \tag{4}$$

where $x, f(x) \in \mathbb{R}$. Here, $u, v \in \mathbb{R}^w$ are trainable parameters, with each element initialized using the normal distribution, $u_i, v_i \sim \mathcal{N}(0, 1)$ for $i \in \{1, \ldots, w\}$. The model is trained using MSE loss on a single training example $(x, y) = (1, 0)$, which simplifies the loss to

$$\mathcal{L}(u, v) = \frac{1}{2} f^2. \tag{5}$$

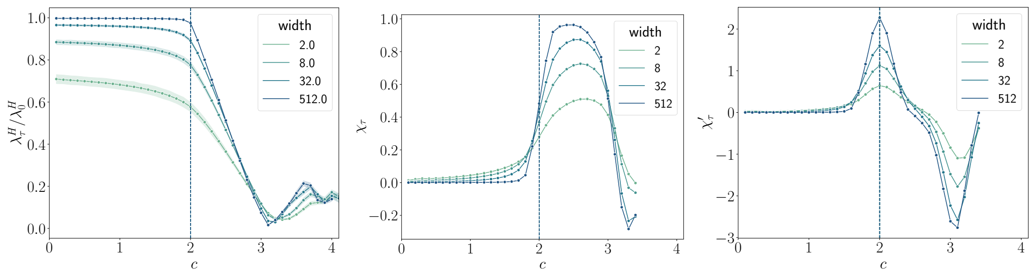
The trace of the Hessian $\mathrm{tr}(H)$ has a simple expression in terms of the norms of the weight vectors

$$\mathrm{tr}(H) = \frac{x^2}{w} \left(\|u\|^2 + \|v\|^2 \right), \tag{6}$$

which is equivalent to the NTK for this model. The Frobenius norm of the Hessian $\|H\|_F$ can be written in terms of the loss $\mathcal{L}$ and $\mathrm{tr}(H)$

$$\|H\|_F^2 = \mathrm{tr}(H)^2 + 2f^2 \left(1 + \frac{2}{w} \right) = \mathrm{tr}(H)^2 + 4\mathcal{L} \left(1 + \frac{2}{w} \right) \tag{7}$$

The gradient descent updates of the model trained using MSE loss on a single training example $(x, y) = (1, 0)$ are given by



(a)

(b)

(c)

Figure 9: (a) Normalized sharpness measured at $\tau = 100$ steps against the learning rate constant for the uv model trained on $(x, y) = (1, 0)$, with varying widths. Each data point is an average of over 500 initializations, where the shaded region depicts the standard deviation around the mean trend. (b, c) Smooth estimations of the first two derivatives, χ_τ and χ'_τ, of the, averaged normalized sharpness wrt the learning rate constant. The vertical dashed lines denote c_{crit} estimated for each width, using the maximum of χ'_τ. Here, we have removed the points beyond $c = 3.5$ for the calculation of derivatives to avoid large fluctuations near the divergent phase. Smoothening details are described in Appendix I.2.

$$v_{t+1} = v_t - \eta f_t \frac{1}{\sqrt{w}} u_t x \tag{8}$$

$$u_{t+1} = u_t - \eta f_t \frac{1}{\sqrt{w}} v_t x \tag{9}$$

The update equations in function space can be written in terms of the trace of the Hessian $\mathrm{tr}(H)$.

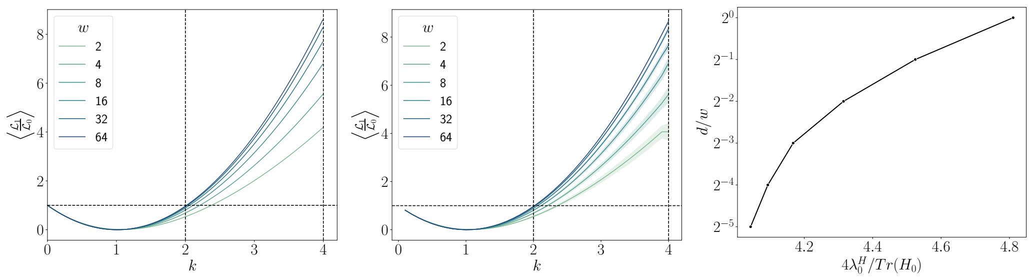
$$\begin{aligned} f_{t+1} &= f_t \left(1 - \eta\, \mathrm{tr}(H_t) + \frac{\eta^2 f_t^2}{w}\right) \\ \mathrm{tr}(H_{t+1}) &= \mathrm{tr}(H_t) + \frac{\eta f_t^2}{w} \left(\eta\, \mathrm{tr}(H_t) - 4\right). \end{aligned} \tag{10}$$

Figure 8 shows the training trajectories of the uv model trained on $(x, y) = (1, 0)$ using MSE loss for 10 training steps. The model shows similar dynamics to those presented in Section 2. It is worth mentioning that the above equations have been analyzed in [47] at large width. In the following subsections, we extend their analysis by incorporating the higher-order terms to analyze the effect of finite width.

B.2 The intermediate saturation regime

The uv model trained on $(x, y) = (1, 0)$ does not show the progressive sharpening and late-time regimes (iii) and (iv) described in Section 1. Hence, we can measure sharpness at the end of training to analyze how it is reduced upon increasing the learning rate and to compare it with the intermediate saturation regime results in Section 3.

Figure 9(a) shows the normalized sharpness measured at $\tau = 100$ steps for various widths. This behavior reproduces the results observed in the intermediate saturation regime in Section 3. In particular, we can see stages (1) and (2), where $\lambda^H_\tau/\lambda^H_0$ starts off fairly independent of learning rate constant c, and then dramatically reduces when $c > 2$; stage (3), where $\lambda^H_\tau/\lambda^H_0$ plateaus at a small value as a function of c is too close to the divergent phase in this model to be clearly observed. The corresponding derivatives of the averaged normalized sharpness, χ_τ, and χ'_τ, are shown in Figure 9(b, c). The vertical dashed lines denote c_{crit} estimated for each width, using the maximum of χ'_τ. We observe that $c_{crit} = 2$ for all widths.



(a) (b) (c)

Figure 10: (a, b) The averaged loss at the first step $\langle {}^{\mathcal{L}_1}/_{\mathcal{L}_0} \rangle$ against the learning rate constant k for varying widths obtained from (a) inequality 16 and (b) numerical experiments. The intersection of $\langle {}^{\mathcal{L}_1}/_{\mathcal{L}_0} \rangle$ with the horizontal line $y = 1$ depicts k_{loss}. The two vertical lines $k = 2$ and $k = 4$ mark the endpoints of k_{loss} at small and large widths. The shaded region in (b) shows the standard deviation around the mean trend. (c) The scaling of λ_0^H and $\mathrm{tr}(H_0)$ with width.

B.3 Opening of the sharpness reduction phase in the uv model

This section shows that $\mathcal{O}({}^{1}/_{w})$ terms in Equation (10) effectively lead to the opening of the sharpness reduction phase with ${}^{1}/_{w}$ in the uv model. In Appendix B.2, we demonstrated that for the uv model, $c_{crit} = 2$ for all values of widths. Hence, it suffices to show that c_{loss} increases from the value 2 as ${}^{1}/_{w}$ increases. We do so by finding the smallest k such that the averaged loss over initializations increases during early training.

It follows from Equation 10 that the averaged loss increases in the first training step if the following holds

$$\left\langle \frac{\mathcal{L}_1}{\mathcal{L}_0} \right\rangle = \left\langle \left(1 - \eta \, \mathrm{tr}(H_0) + \frac{\eta^2 f_0^2}{w} \right)^2 \right\rangle > 1, \tag{11}$$

where $\langle . \rangle$ denotes the average over initializations. On scaling the learning rate with trace as $\eta = {}^{k}/_{\mathrm{tr}(H_0)}$, we have

$$\left\langle \frac{\mathcal{L}_1}{\mathcal{L}_0} \right\rangle = \left\langle \left(1 - k + \frac{k^2}{w} \frac{f_0^2}{\mathrm{tr}(H_0)^2} \right) \right\rangle > 1 \tag{12}$$

$$\left\langle \frac{\mathcal{L}_1}{\mathcal{L}_0} \right\rangle = \left((1-k)^2 + 2(1-k)\frac{k^2}{w} \left\langle \frac{f_0^2}{\mathrm{tr}(H_0)^2} \right\rangle + \frac{k^4}{w^2} \left\langle \frac{f_0^4}{\mathrm{tr}(H_0)^4} \right\rangle \right) > 1. \tag{13}$$

The required two averages have the following expressions as shown in Appendix B.8.

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = \frac{w}{4(w+1)} \tag{14}$$

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle = \frac{3(w+2)w^3}{16} \frac{\Gamma(w)}{\Gamma(w+4)}. \tag{15}$$

Inserting the above expressions in Equation 13, on average the loss increases in the very first step if the following inequality holds

$$\left\langle \frac{\mathcal{L}_1}{\mathcal{L}_0} \right\rangle = \left((1-k)^2 + \frac{k^2(1-k)}{2(w+1)} + \frac{3k^4}{16(w+3)(w+1)} \right) > 1 \tag{16}$$

The graphical representation of the above inequality shown in Figure 10(a) is in excellent agreement with the experimental results presented in Figure 10(b).

Let us denote k'_{loss} as the minimum learning rate constant such that the average loss increases in the first step. Similarly, let k_{loss} denote the learning rate constant if the loss increases in the first 10 steps. Then, k'_{loss} increases from the value 2 as $1/w$ increases as shown in Figure 10(a). By comparison, the trace reduces at any step if $\eta \operatorname{tr}(H_t) < 4$. At initialization, this condition becomes $k < 4$. Hence, for $k < k'_{loss}$, both the loss and trace monotonically decrease in the first training step. These arguments can be extended to later training steps, revealing that the loss and trace will continue to decrease for $k < k'_{loss}$.

Next, let η_{loss} denote the learning rate corresponding to c_{loss}. Then, we have $\eta_{loss} = \frac{c_{loss}}{\lambda_0^H} = \frac{k_{loss}}{\operatorname{tr}(H_0)}$, implying

$$c_{loss} = k_{loss} \frac{\lambda_0^H}{\operatorname{tr}(H_0)}. \tag{17}$$

Figure 10(c) shows that $\lambda_0^H \geq \operatorname{tr}(H_0)$ for all widths, implying $c_{loss} \geq k_{loss}$. Hence, c_{loss} increases with $1/w$ as observed in Figure 6(a). In Appendix B.2, we demonstrated that for the uv model, $c_{crit} = 2$ for all values of widths. Incorporating this with c_{loss} increases with $1/w$, we have sharpness reduction phase opening up as $1/w$ increases.

B.4 Opening of the loss catapult phase at finite width

In this section, we use the Frobenius norm of the Hessian $\|H\|_F$ as a proxy for the sharpness and demonstrate the emergence of the loss-sharpness catapult phase at finite width. In particular, We analyze the expectation value $\langle \operatorname{tr}(H^T H) \rangle$ after the first training step near $k = k_{loss}$ and show that $k_{loss} \leq k_{frob}$, with the difference increasing with $1/w$. First, we write $\operatorname{tr}(H_t^T H_t)$ in terms of $\mathcal{L}_t$ and $\operatorname{tr}(H_t)$

$$\operatorname{tr}(H_t^T H_t) = \operatorname{tr}(H_t)^2 + 4\left(1 + \frac{2}{w}\right)\mathcal{L}_t. \tag{18}$$

Next, using Equations 1, we write down the change in $\operatorname{tr}(H_t^T H_t)$ after the first training step in terms of $\operatorname{tr}(H_0)$ and $\mathcal{L}_0$

$$\begin{aligned}
\Delta \operatorname{tr}(H_1^T H_1) &= \operatorname{tr}(H_1^T H_1^T) - \operatorname{tr}(H_0^T H_0) = \operatorname{tr}(H_1)^2 - \operatorname{tr}(H_0)^2 + 4\left(1 + \frac{2}{w}\right)(\mathcal{L}_1 - \mathcal{L}_0) \\
\Delta \operatorname{tr}(H_1^T H_1) &= \frac{\eta f_0^2}{w}\left(\eta \operatorname{tr}(H_0) - 4\right)\left[\frac{\eta f_0^2}{w}\left(\eta \operatorname{tr}(H_0) - 4\right) + 2\operatorname{tr}(H_0)\right] + 4\left(1 + \frac{2}{w}\right)(\mathcal{L}_1 - \mathcal{L}_0)
\end{aligned} \tag{19}$$

Next, we substitute $\eta = k/\operatorname{tr}(H_0)$ to obtain the above equation as a function of k

$$\Delta \operatorname{tr}(H_1^T H_1) = \frac{k(k-4)}{w}\left[\frac{k(k-4)}{w}\frac{f_0^4}{\operatorname{tr}(H_0)^2} + 2f_0^2\right] + 4\left(1 + \frac{2}{w}\right)(\mathcal{L}_1 - \mathcal{L}_0) \tag{20}$$

Finally, we calculate the expectation value of $\langle \Delta \operatorname{tr}(H_1^T H_1) \rangle$

$$\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle = \frac{k(k-4)}{w}\left[\frac{k(k-4)}{w}\left\langle \frac{f_0^4}{\operatorname{tr}(H_0)^2} \right\rangle + 2\langle f_0^2 \rangle\right] + 4\left(1 + \frac{2}{w}\right)\langle \mathcal{L}_1 - \mathcal{L}_0 \rangle, \tag{21}$$

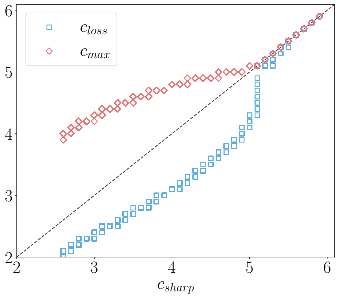


Figure 11: The relationship between the critical constants for the uv model trained on a single training examples $(x, y) = (1, 0)$ with MSE loss using gradient descent. Each data point corresponds to a random initialization

by estimating $\left\langle \frac{f_0^4}{\operatorname{tr}(H_0)^2} \right\rangle$ using the approach demonstrated in the previous section

$$\left\langle \frac{f_0^4}{\operatorname{tr}(H_0)^2} \right\rangle = \frac{3w}{4(w+3)}. \tag{22}$$

Inserting $\left\langle \frac{f_0^4}{\operatorname{tr}(H_0)^2} \right\rangle$ in Equation 21 along with $\langle f_0^2 \rangle = 1$, we have

$$\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle = \underbrace{\frac{k(k-4)}{w} \left[\frac{3k(k-4)}{4(w+3)} + 2 \right]}_{I(k,w)} + 4 \left(1 + \frac{2}{w} \right) \langle \mathcal{L}_1 - \mathcal{L}_0 \rangle \tag{23}$$

At infinite width, the above equation reduces to $\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle = 4 \langle \mathcal{L}_1 - \mathcal{L}_0 \rangle$, and hence, $k_{frob} = k_{loss}$. For any finite width, $I(k, w) < 0$ for $0 < k < 4$. At $k \leq k_{loss}$, $\mathcal{L}_1 - \mathcal{L}_0 \leq 0$, and therefore $\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle < 0$. In order for the sharpness to catapult, we require $\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle > 0$ and therefore $k_{frob} > k_{loss}$. As $1/w$ increases $|I(k, w)|$ also increases, which means a higher value of $\mathcal{L}_1 - \mathcal{L}_0$ is required to reach a point where $\left\langle \Delta \operatorname{tr}(H_1^T H_1) \right\rangle \geq 0$. Thus $k_{frob} - k_{loss}$ increases with $1/w$.

B.5 The early training trajectories

Figure 8 shows the early training trajectories of the uv model with large ($w = 512$) and small ($w = 2$) widths. The dynamics depicted show several similarities with early training dynamics of real-world models shown in Figure 2. At small widths, the loss catapults at relatively higher learning rates (specifically, at $c_{loss} = 3.74$, which is significantly higher than the critical value of $c_{crit} = 2$).

B.6 Relationship between critical constants

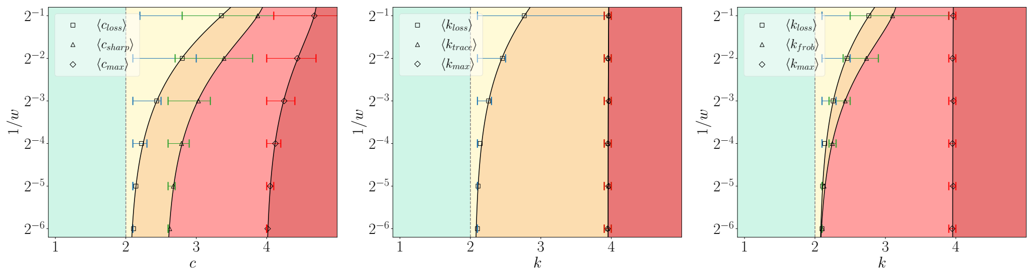
Figure 11 shows the relationship between various critical constants for the uv model. The data show that the inequality $c_{loss} \leq c_{sharp} \leq c_{max}$ holds for every random initialization of the uv model.

B.7 Phase diagrams with error bars

This section shows the variation in the phase diagram boundaries of the uv model shown in Figure 6(a, b). Figure 12 shows these phase diagrams. Each data point is an average of over 500 initializations. The horizontal bars around each data point indicate the region between 25% and 75% quantile.

B.8 Derivation of the expectation values

Here, we provide the detailed derivation of the averages $\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle$ and $\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle$. We begin by finding the average $\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle$



(a) (b) (c)

Figure 12: The phase diagram of the uv model trained with MSE loss using gradient descent with (a) sharpness λ_t^H (b) trace of Hessian tr_0^H and (c) the square of the Frobenius norm $\mathrm{tr}(H_t^T H_t)$ used as a measure of sharpness. In (a), the learning rate is scaled as $\eta = {}^{c}/_{\lambda_0^H}$, while in (b) and (c), the learning rate is scaled as $\eta = {}^{k}/_{\mathrm{tr}(H_0)}$. Each data point denotes an average of over 500 initialization, and the smooth curve represents a 2-degree polynomial fitted to the raw data. The horizontal bars around the average data point indicate the region between 25% and 75% quantile.

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = w \int_{-\infty}^{\infty} \prod_{i=1}^{w} \left(\frac{dv_i du_i}{2\pi} \right) \exp\left(-\frac{\|u\|^2 + \|v\|^2}{2} \right) \frac{\sum_{j,k=1}^{w} u_j v_j u_k v_k}{\left(\|u\|^2 + \|v\|^2 \right)^2}, \tag{24}$$

where $\|.\|$ denotes the norm of the vectors.

The above integral is non-zero only if $j = k$. Hence, it is a sum of w identical integrals. Without any loss of generality, we solve this integral for $j = 1$ and multiply by w to obtain the final result, i.e.,

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = w^2 \int_{-\infty}^{\infty} \prod_{i=1}^{w} \left(\frac{dv_i du_i}{2\pi} \right) \exp\left(-\frac{\|u\|^2 + \|v\|^2}{2} \right) \frac{u_1^2 v_1^2}{\left(\|u\|^2 + \|v\|^2 \right)^2} \tag{25}$$

Consider a transformation of $u, v \in \mathbb{R}^w$ into w dimensional spherical coordinates such that

$$u_1 = r_u \cos \varphi_{u_1}, \qquad v_1 = r_v \cos \varphi_{v_1}, \tag{26}$$

which yields,

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = \frac{w^2}{(2\pi)^w} \int dr_u dr_v d\Omega_{u,w} d\Omega_{v,w} r_u^{w-1} r_v^{w-1} \exp\left(-\frac{r_u^2 + r_v^2}{2} \right) \frac{r_u^2 \cos^2 \varphi_{u_1} r_v^2 \cos^2 \varphi_{u_1}}{\left(r_u^2 + r_v^2 \right)^2} \tag{27}$$

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = \frac{w^2}{(2\pi)^w} \int dr_u dr_v \exp\left(-\frac{r_u^2 + r_v^2}{2} \right) \frac{r_u^2 r_v^2}{\left(r_u^{w+1} + r_v^{w+1} \right)^2} \int d\Omega_{u,w} d\Omega_{v,w} \cos^2 \varphi_{u_1} \cos^2 \varphi_{v_1} \tag{28}$$

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = \frac{w^2}{(2\pi)^w} \underbrace{\int dr_u dr_v \exp\left(-\frac{r_u^2 + r_v^2}{2} \right) \frac{r_u^2 r_v^2}{\left(r_u^{w+1} + r_v^{w+1} \right)^2}}_{I_r} \Bigg(\underbrace{\int d\Omega_w \cos^2 \varphi_1}_{I_\varphi} \Bigg)^2, \tag{29}$$

where $d\Omega$ denotes the w dimensional solid angle element. Here, we denote the radial and angular integrals by I_r and I_φ respectively. The radial integral I_r is

$$I_r = \int_0^\infty dr_u dr_v \frac{r_u^{w+1} r_v^{w+1}}{(r_u^2 + r_v^2)^2} \exp\left(-\frac{r_u^2 + r_v^2}{2}\right). \tag{30}$$

Let $r_u = R\cos\theta$ and $r_v = R\sin\theta$ with $R \in [0, \infty)$ and $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, then we have

$$I_r = \int_0^\infty dR\, R^{2w-1} e^{-R^2/2} \int_0^{\pi/2} d\theta \cos^{w+1}\theta \sin^{w+1}\theta \tag{31}$$

$$I_r = \frac{\sqrt{\pi}}{2^3} \frac{\Gamma(w)\, \Gamma\left(\frac{w+2}{2}\right)}{\Gamma\left(\frac{w+3}{2}\right)}, \tag{32}$$

where $\Gamma(.)$ denotes the Gamma function. The angular integral I_φ is

$$I_\varphi = \int d\varphi_1 d\varphi_2 \dots d\varphi_{w-1} \sin^{w-2}\varphi_1 \cos^2\varphi_1 \sin^{w-3}\varphi_2 \dots \sin\varphi_{w-2} \tag{33}$$

$$I_\varphi = \int d\varphi_1 d\varphi_2 \dots d\varphi_{w-1} \sin^{w-2}\varphi_1 \sin^{w-3}\varphi_2 \dots \sin\varphi_{w-2} \frac{\int_0^\pi d\varphi_1 \sin^{w-2}\varphi_1 \cos^2\varphi_1}{\int_0^\pi d\varphi_1 \sin^{w-2}\varphi_1} \tag{34}$$

$$I_\varphi = \frac{\pi^{w/2}}{\Gamma(\frac{w+2}{2})}. \tag{35}$$

Plugging in Equations 32 and 35 into Equation 29, we obtain a very simple expression

$$\left\langle \frac{f_0^2}{Tr(H_0)^2} \right\rangle = \frac{w^2}{2^{w+3}} \frac{\sqrt{\pi}\Gamma(w)}{\Gamma(\frac{w+2}{2})\Gamma(\frac{w+3}{2})} = \frac{w}{4(w+1)}. \tag{36}$$

The other integral $\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle$ can be obtained by generalizing the above approach as described below

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle = w^2 \int_{-\infty}^{\infty} \prod_{i=1}^{w} \left(\frac{dv_i du_i}{2\pi}\right) \exp\left(-\frac{\|u\|^2 + \|v\|^2}{2}\right) \frac{\sum_{j,k,l,m=1}^{w} u_j v_j u_k v_k u_l v_l u_m v_m}{(\|u\|^2 + \|v\|^2)^4}. \tag{37}$$

The integral is zero if either $j = k$ and $l = m$ or $j = k = l = m$, which we consider separately. Without loss of generality, we find the following integrals

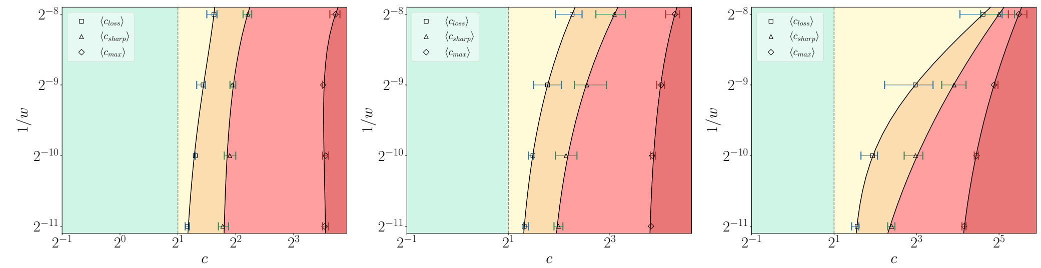
$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{22} = w^2 \int_{-\infty}^{\infty} \prod_{i=1}^{w} \left(\frac{dv_i du_i}{2\pi}\right) \exp\left(-\frac{\|u\|^2 + \|v\|^2}{2}\right) \frac{u_1^2 u_2^2 v_1^2 v_2^2}{(\|u\|^2 + \|v\|^2)^4} \tag{38}$$

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{4} = w^2 \int_{-\infty}^{\infty} \prod_{i=1}^{w} \left(\frac{dv_i du_i}{2\pi}\right) \exp\left(-\frac{\|u\|^2 + \|v\|^2}{2}\right) \frac{u_1^4 v_1^4}{(\|u\|^2 + \|v\|^2)^4}, \tag{39}$$

which have the following expressions

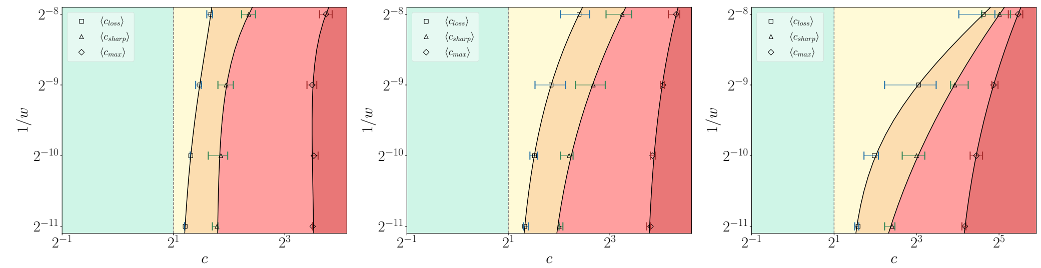
$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{22} = \frac{w^2}{16} \frac{\Gamma(w)}{\Gamma(w+4)} \tag{40}$$

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{4} = \frac{9w^2}{16} \frac{\Gamma(w)}{\Gamma(w+4)}, \tag{41}$$



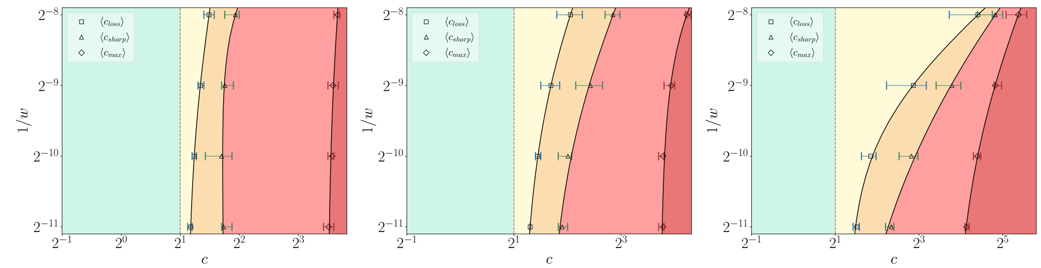
(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

Figure 13: Phase diagrams of FCNs in NTP with varying depths trained on the MNIST dataset using MSE loss.



(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

Figure 14: Phase diagrams of FCNs in NTP with varying depths trained on the Fashion-MNIST dataset.



(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

Figure 15: Phase diagrams of FCNs in NTP with varying depths trained on the CIFAR-10 dataset.

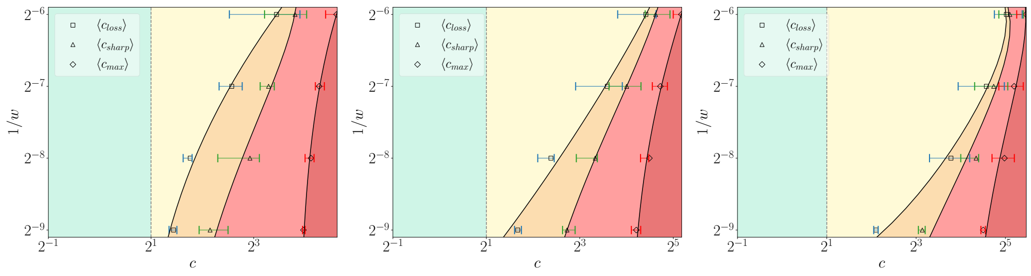
where $\Gamma(.)$ denotes the gamma function. On combining the expressions with their multiplicities, we obtain the final result

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle = 3w(w-1) \left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{22} + w \left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle_{4} \tag{42}$$

$$\left\langle \frac{f_0^4}{Tr(H_0)^4} \right\rangle = \frac{3(w+2)w^3}{16} \frac{\Gamma(w)}{\Gamma(w+4)} \tag{43}$$

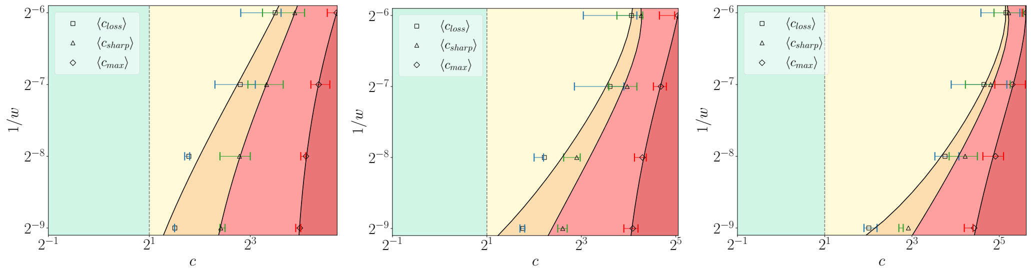
C Phase diagrams of early training

This section describes experimental details and shows additional phase diagrams of early training. The results include (1) FCNs in NTP trained on MNIST, Fashion-MNIST, and CIFAR-10 datasets, (2) CNNs in SP trained on Fashion-MNIST and CIFAR-10, and (3) ResNets in SP trained on CIFAR-



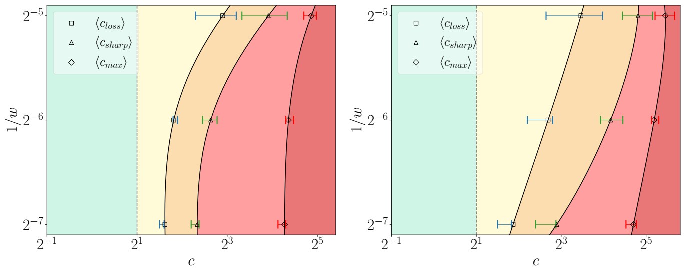
(a) $d = 5$ (b) $d = 7$ (c) $d = 10$

Figure 16: Phase diagrams of Convolutional Neural Networks (CNNs) in SP with varying depths trained on the Fashion-MNIST dataset.



(a) $d = 5$ (b) $d = 7$ (c) $d = 10$

Figure 17: Phase diagrams of Convolutional Neural Networks (CNNs) in SP with varying depths trained on the CIFAR-10 dataset.

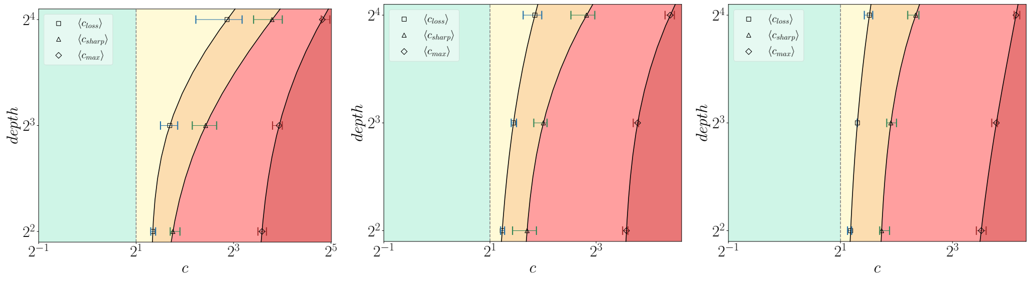


(a) $d = 18$ (b) $d = 34$

Figure 18: Phase diagrams of Resnets in SP with different depths trained on the CIFAR-10 dataset.

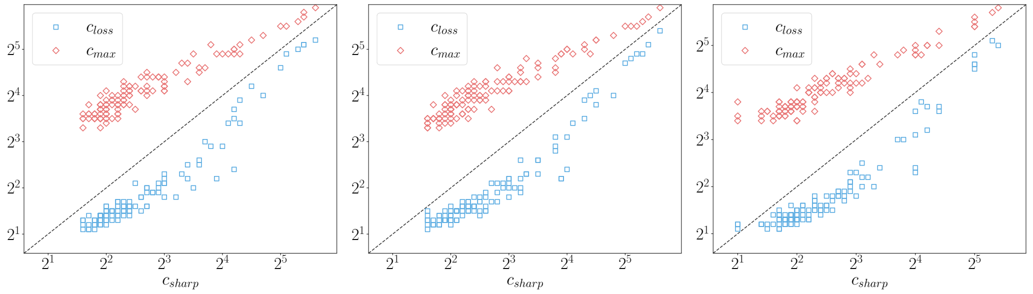
10 datasets using MSE loss. Figures 13 to 18 show these results. The depths and widths are the same as specified in Appendix A. Each data point is an average over 10 initializations. The horizontal bars around the average data point indicate the region between 25% and 75% quantile. Phase diagrams for cross-entropy results are shown in Appendix F.

Additional experimental details : We train each model for $t = 10$ steps using SGD with learning rates $\eta = c/\lambda_0^H$ and batch size of 512, where $c = 2^x$ with $x \in \{0.0, \dots x_{max}\}$ in steps of 0.1. Here, x_{max} is relatd to the maximum trainable learning rate constant as $c_{max} = 2^{x_{max}}$. We have considered 10 random initializations for each model. As mentioned in Appendix A, we do not consider averages over SGD runs as the randomness from initialization outweighs it. Hence, we obtain 10 values for each of the critical values in the following results. For each initialization, we compute the critical constants using Definitions 1 and 3. To avoid a random increase in loss and sharpness due to fluctuations, we round off the values of λ_t^H/λ_0^H and $\mathcal{L}_t/\mathcal{L}_0$ to their second decimal places before comparing with 1. We denote the average values using data points and variation using

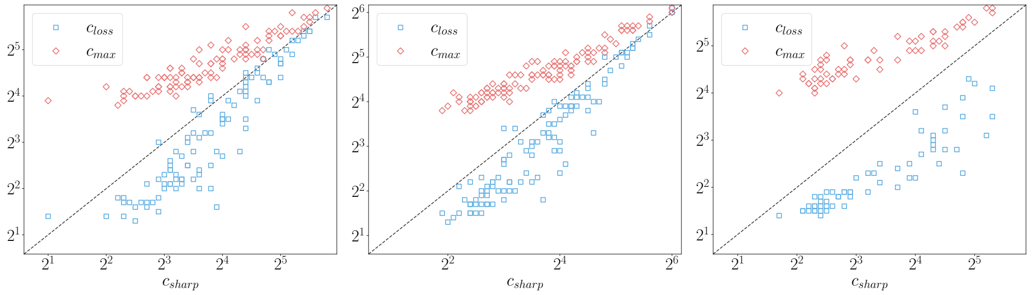


(a) $w = 512$ (b) $w = 1024$ (c) $d = 2048$

Figure 19: Phase diagrams of FCNs in NTP with varying widths trained on the CIFAR-10 dataset.



(a) FCNs on MNIST (b) FCNs on Fashion-MNIST (c) FCNs on CIFAR-10



(d) CNNs on Fashion-MNIST (e) CNNs on CIFAR-10 (f) ResNets on CIFAR-10

Figure 20: The relationship between various critical constants for various models and datasets. Each data point corresponds to a model with random initialization. The dashed line denotes the values where $x = y$.

horizontal bars around the average data points, which indicate the region between 25% and 75% quantile. The smooth curves are obtained by fitting a two-degree polynomial $y = a + bx + cx^2$ with $x = {}^{1}/_{w}$ and y can take on one of three values: c_{loss}, c_{sharp} and c_{max}.

Phase diagrams with depth Figure 19: shows the phase diagrams with depth for FCNs in NTP trained on the CIFAR-10 dataset. The phase diagrams look qualitatively similar compared to the ${}^{1}/_{w}$ phase diagrams.

D Relationship between various critical constants

Figure 20 illustrates the relationship between the early training critical constants for models and datasets. The experimental setup is the same as in Appendix C. Typically, we find that $c_{loss} \leq c_{sharp} \leq c_{sharp}$ holds true. However, there are some exceptions, which are observed at high values of ${}^{d}/_{w}$ (see 20 (d, e)), where the trends of the critical constants converge, and large fluctuations can cause deviations from the inequality.

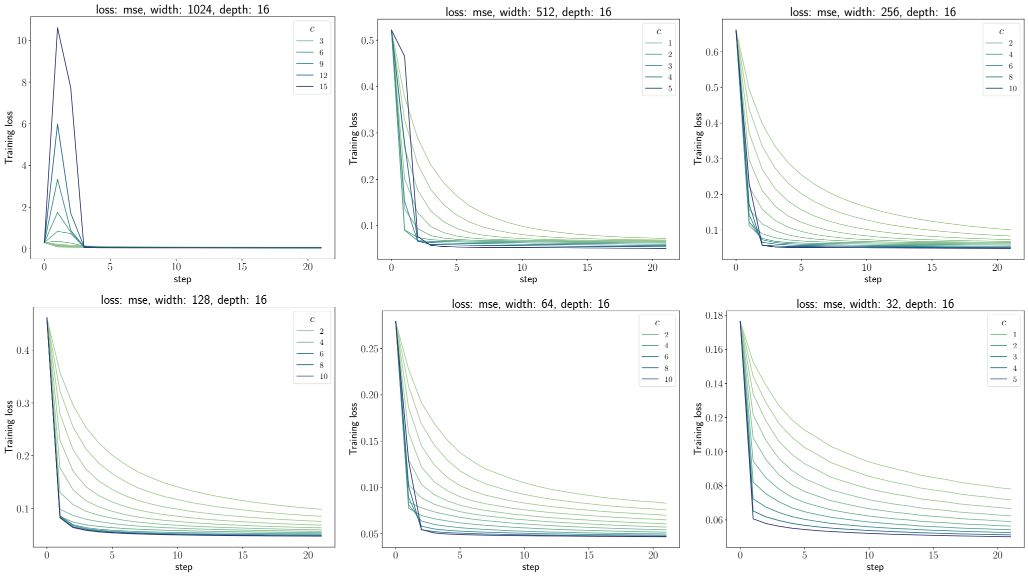


Figure 21: Training loss trajectories of ReLU FCNs with $d = 16$ trained on the CIFAR-10 dataset with MSE loss using SGD with learning rate $\eta = c/\lambda_0^H$ and batch size $B = 512$.

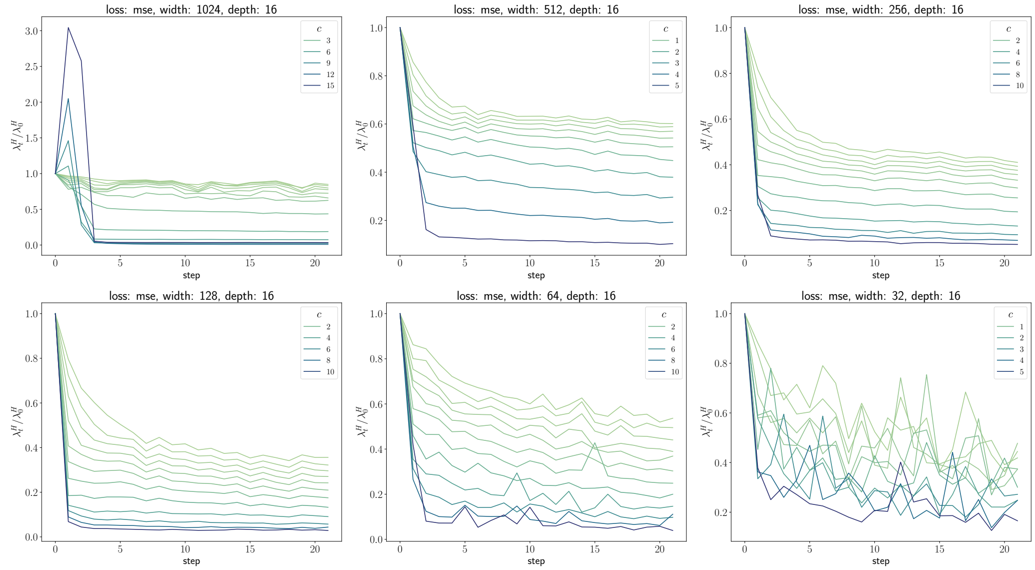
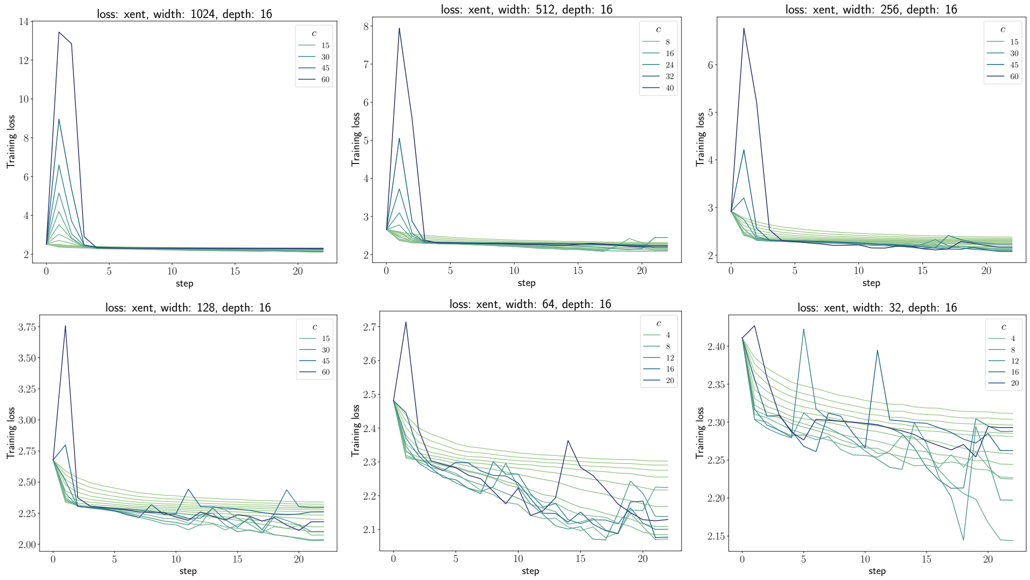


Figure 22: Sharpness trajectories of ReLU FCNs with $d = 16$ trained on the CIFAR-10 dataset with MSE loss using SGD with learning rate $\eta = c/\lambda_0^H$ and batch size $B = 512$.

E The effect of d/w on the noise in dynamics

In this section, we demonstrate that for FCNs with $d/w \gtrsim 1/16$, the dynamics becomes noise-dominated. This aspect makes it challenging to distringuish the underlying deterministic dynamics from random fluctuations. To demonstrate this, we consider FCNs trained on CIFAR-10 using MSE and cross-entropy loss and use 4096 training examples for estimating sharpness.

Figures 21 and 22 show the training loss and sharpness of FCNs with $d = 16$ and varying widths, trained on CIFAR-10 using MSE loss. We observe that the sharpness dynamics becomes noisier for $w \lesssim 64$.


loss: xent, width: 1024, depth: 16

loss: xent, width: 512, depth: 16

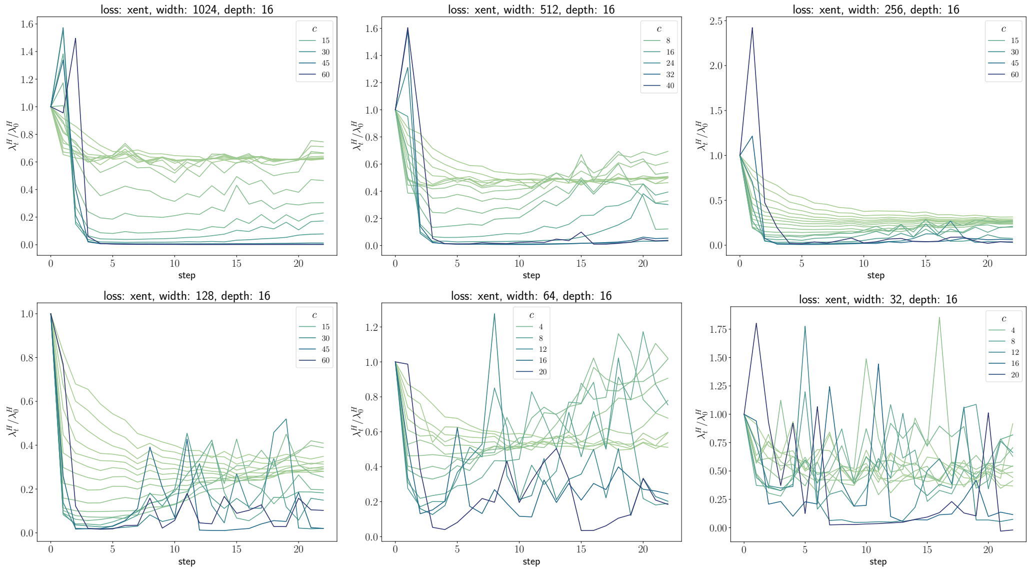
loss: xent, width: 256, depth: 16

loss: xent, width: 128, depth: 16

loss: xent, width: 64, depth: 16

loss: xent, width: 32, depth: 16

Figure 23: Training loss trajectories of ReLU FCNs with $d = 16$ trained on the CIFAR-10 dataset with cross-entropy loss using SGD with learning rate $\eta = c/\lambda_0^H$ and batch size $B = 512$.


loss: xent, width: 1024, depth: 16

loss: xent, width: 512, depth: 16

loss: xent, width: 256, depth: 16

loss: xent, width: 128, depth: 16

loss: xent, width: 64, depth: 16

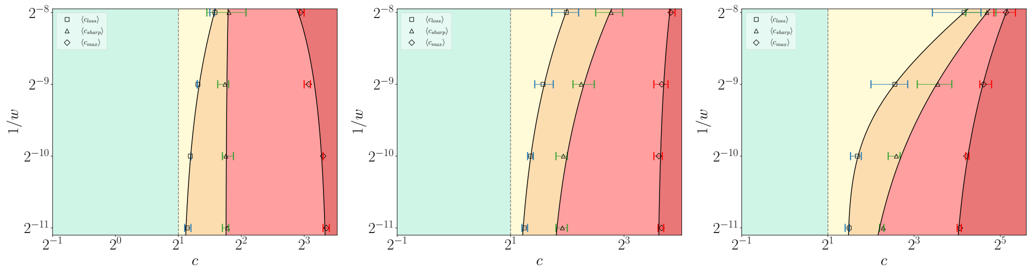
loss: xent, width: 32, depth: 16

Figure 24: Sharpness trajectories of ReLU FCNs with $d = 16$ trained on the CIFAR-10 dataset with cross-entropy loss using SGD with learning rate $\eta = c/\lambda_0^H$ and batch size $B = 512$.

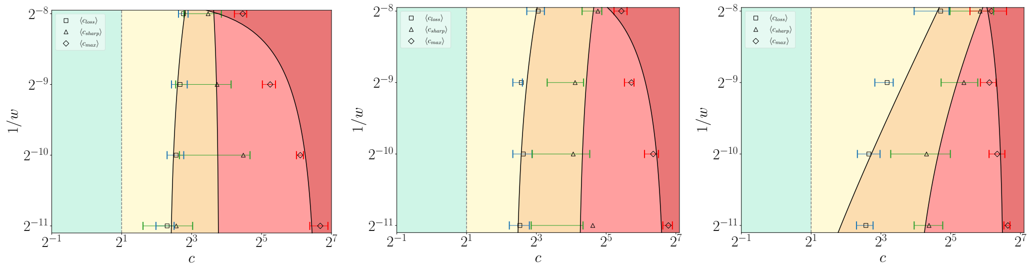
Figures 23 and 24 shows the training dynamics with loss switched to cross-entropy, while keeping the initialization and SGD batch sequence the same as in the MSE loss case. In comparison to MSE loss, the training loss and sharpness dynamics show a higher level of noise, especially for $w \lesssim 256$. As a result, it becomes difficult to characterize the training dynamics for $d/w \gtrsim 1/16$.

F Crossentropy

In this section, we provide additional results for models trained with cross-entropy (xent) loss and compare them with MSE results. Broadly speaking, models trained with cross-entropy loss show similar characterstics to those trained with MSE loss, such as, (i) sharpness reduction during early training, (ii) an increase in critical constants c_{loss}, c_{sharp} with d and $1/w$, (iii) $c_{loss} \leq c_{sharp} \leq c_{max}$.

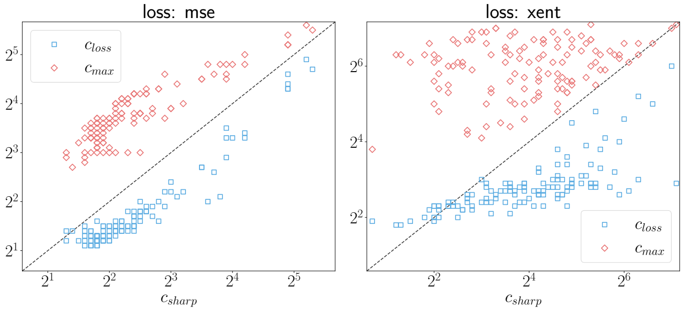


(a) MSE, $d = 4$ (b) MSE, $d = 8$ (c) MSE, $d = 16$



(d) xent, $d = 4$ (e) xent, $d = 8$ (f) xent, $d = 16$

Figure 25: The phase diagrams of early training of FCNs trained on the CIFAR-10 dataset using (a, b, c) MSE and (d, e, f) cross-entropy loss. Each data point is an average over 10 initializations, and solid lines represent a smooth curve fitted to raw data points. The horizontal bars around the averaged data point indicates the region between 25% and 75% quantile. For cross-entropy phase diagrams, the $c = 2$ line is shown for reference only and does not relate to c_{crit}.



(a) (b)

Figure 26: Comparison of the relationship between critical constants for FCNs in SP trained on CIFAR-10 using MSE and cross-entropy loss. Each data point corresponds to a randomly initialized model with depths and widths mentioned in Appendix A.

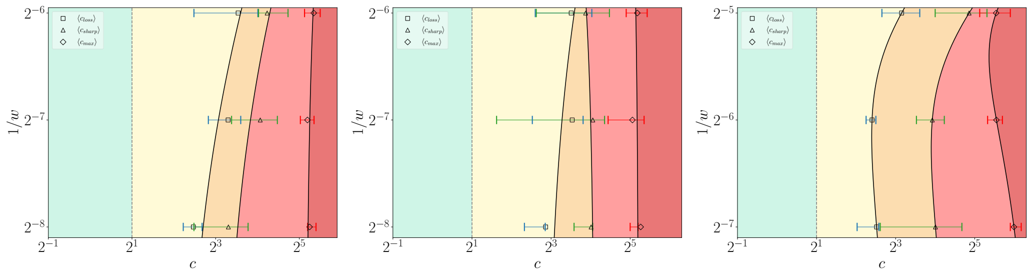
However, the dynamics of models trained with cross-entropy loss is noisier compared to MSE as shown in the previous section, and characterizing these dynamics can be more complex. In the following experiments, we consider models trained on the CIFAR-10 dataset and used 4096 training examples to estimate sharpness.

F.1 Phase diagrams

Figure 25 compares the phase diagrams of FCNs in SP trained on the CIFAR-10 dataset, using both MSE and cross-entropy loss. The estimated critical constants for cross-entropy loss are generally more noisy, as quantified by the confidence intervals. In comparison to phase diagrams of models trained with MSE loss, we observe a few notable differences. First, the loss starts to catapult at a value appreciably larger than $c = 2$ at large widths. Primarly, $4 \lesssim c_{loss} \lesssim 8$. Additionally, c_{max} generally decreases with $1/w$. This decreasing trend becomes less sharp at large depths.

Despite these differences, the phase diagrams for both loss functions share various similarities. First, we observe sharpness reduces during early training for $c < c_{sharp}$ (see the first row of Figure 24). Next, we observe that the inequality $c_{loss} \leq c_{sharp} \leq c_{max}$ generally holds for both loss functions as demonstrated in Figure 26, barring some exceptions.

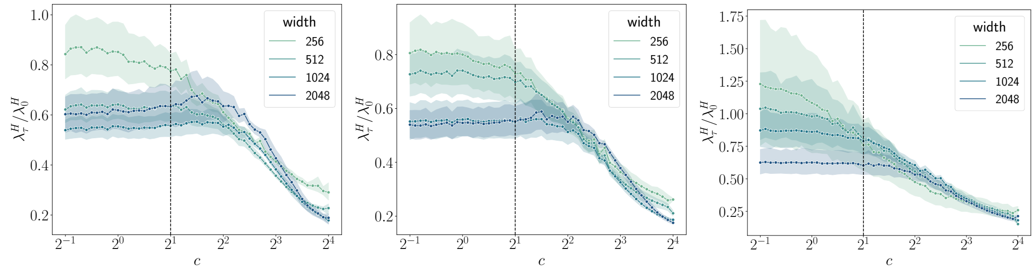
Figure 27 shows the phase diagrams for CNNs and ResNets trained on the CIFAR-10 dataset using cross-entropy loss. The observed critical constants are much noisier as quantified by the confidence intervals. Nevertheless, the phase diagram shows similar trends as mentioned above. For large $^1/_w$ models, we found that progressive sharpening begins after $5 - 10$ training steps. For these cases, we only use the first 5 steps to measure sharpness to avoid progressive sharpening. For CNNs, we observed that the dynamics becomes difficult to characterize for $w \lesssim 32$ and $d \gtrsim 10$, due to large fluctuations. Consequently, we've opted not to include these particular results.



(a) CNN, $d = 5$ (b) CNN, $d = 7$ (c) ResNet, $d = 18$

Figure 27: Phase diagrams of (a, b) CNNs and (c) ResNets trained on the CIFAR-10 dataset with cross-entropy loss using SGD with $\eta = {}^c/_{\lambda_0^H}$ and $B = 512$.

F.2 Intemediate saturation regime



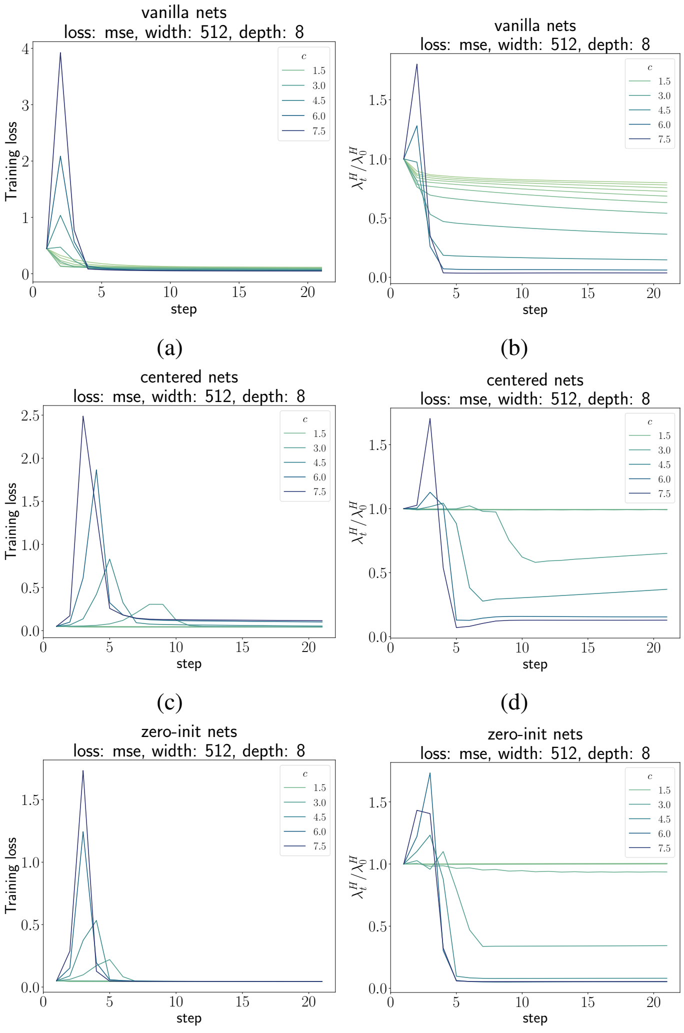
(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

Figure 28: Sharpness measured at $c\tau = 100$ against the learning rate constant for FCNs trained on the CIFAR-10 dataset using cross-entropy loss, with varying depths and widths. Each curve is an average over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical dashed line shows $c = 2$ for reference.

Figure 28 shows the normalized sharpness measured at $c\tau = 100$ for FCNs trained on CIFAR-10 using cross-entropy loss. [3] Similar to MSE loss, we observe an abrupt drop in sharpness at large learning rates. However, this abrupt drop occurs at $2 \lesssim c_{crit} \lesssim 4$. The estimated sharpness is noisier (compare with Figure 37), which hinders a reliable estimation of c_{crit}. We speculate that we require a large number of averages for a reliable estimation of c_{crit} for cross-entropy loss. We leave the precise characterization of c_{crit} for cross-entropy loss for future work.

---

[3] The time step $\tau = {}^{100}/_c$ is in the middle of the intermediate saturation regime for most of the models. For further details on estimating sharpness, see Appendix I.1.



(a) (b)

(c) (d)

(e) (f)

Figure 29: Comparison of the early training dynamics of (a, b) vanilla, (c, d) centered, and (e, f) zero-initialized FCNs (with depth $= 8$ and width $= 512$), trained on the CIFAR-10 dataset with MSE loss using gradient descent for 20 steps.

G The effect of setting model output to zero at initialization

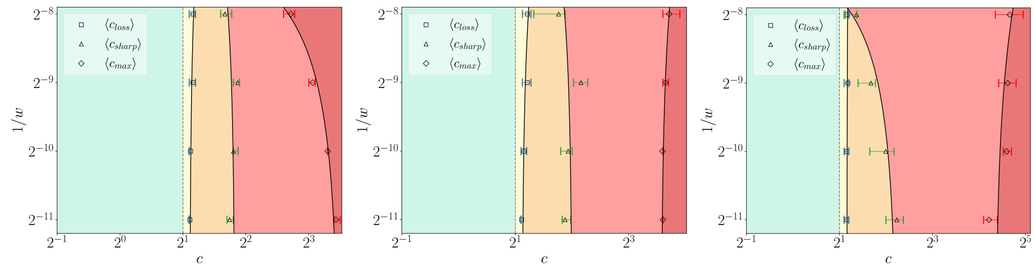
In this section, we demonstrate the effect of network output $f(x;\theta_t)$ at initialization on the early training dynamics. In particular, we set the network output to zero at initialization, $f(x;\theta_0) = 0$, by (1) 'centering' the network by its initial value $f_c(x;\theta_t) = f(x;\theta) - f(x;\theta_0)$ or (2) setting the last layer weights to zero at initialization. We show that both (1) and (2) remove the opening of the sharpness reduction phase with $1/w$. Resultantly, the average onset of loss catapult occurs at $c_{loss} \approx 2$, independent of depth and width.

Throughout this section, we use 'vanilla' networks to refer to networks initialized in the standard way. For simplicity, we train FCNs using full batch gradient descent with MSE loss using a subset consisting of 4096 examples of the CIFAR-10 dataset.

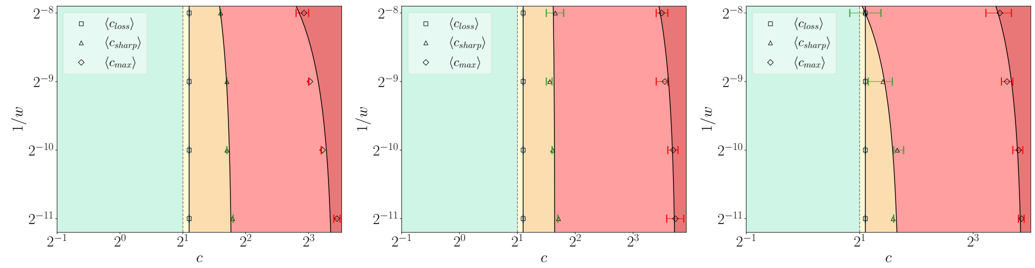
G.1 The effect of centering networks

Given a network function $f(x;\theta_t)$, we define the centered network $f_c(x;\theta_t)$ as

$$f_c(x;\theta_t) = f(x;\theta_t) - f(x;\theta_0), \tag{44}$$



(a) centered, $d = 4$ (b) centered, $d = 8$ (c) centered, $d = 16$



(d) zero-init, $d = 4$ (e) zero-init, $d = 8$ (f) zero-init, $d = 16$

Figure 30: The phase diagrams of early training dynamics of (a, b, c) centered and (d, e, f) zero-init networks trained on CIFAR-10 using MSE using gradient descent. Each data point is an average over 10 initializations. The horizontal bars around the average data point indicate the region between 25% and 75% quantile.

where $f(x; \theta_0)$ is the network output at intialization. By construction, the network output is zero at initialization. It is noteworthy that centering a network is an unusual way of training deep networks as it doubles the cost of training because of two forward passes.

Figure 29 compares the training loss and sharpness dynamics of vanilla networks and centered networks. Unlike vanilla networks, we do not observe a decrease in sharpness for $c < c_{loss}$ during early training. Rather, we observe a slight increase in sharpness. To distinguish this slight increase from sharpness catapult, we introduce a threshold ϵ, comparing normalized sharpness $\lambda^H_t / \lambda^H_0$ with $1 + \epsilon$, to define a sharpness catapult.[4] As demonstrated in Appendix G.3, the uv model trained on a single training example (x, y) with $y \neq 0$ sheds lights on this initial increase in sharpness.

Interestingly, irrespective of depth and width, we observe that loss catapults at $c_{loss} \approx 2$, as demonstrated in the phase diagrams in Figure 30(a, b, c). These findings suggest a strong correlation between a large network output at initialization $\|f(x; \theta_0)\|$ and the opening of the sharpness reduction phase discussed in Section 2.

G.2 The effect of setting the last layer to zero

An alternative way to train networks with $f(x; \theta_0) = 0$ is by setting the last layer to zero at initialization. The principle of criticality at initialization [52, 55, 65] does not put any constraints on the last layer weights. Hence, setting the last layer to zero does not affect signal/gradient propagation at initialization. Yet, setting the last layer to zero results in initialization in a flat curvature region at initialization, resulting in access to larger learning rates. We refer to these networks as 'zero-init' networks.

Figure 29 compares the training dynamics of zero-init networks with vanilla and centered networks. We observe that the dynamics is quite similar to the centered networks: (i) sharpness does not reduce for small learning rates and (ii) loss catapults $c_{loss} \approx 2$, irrespective of depth and width. Figure 30(d, e, f) show the phase diagrams of networks with zero-initialized networks. Like centered networks, the critical constants do not scale with depth and width. Again, suggesting that a large network output

---

[4] In experiments, we set $\epsilon = 0.05$. We use the same threshold for zero-init networks.

at initialization $\|f(x;\theta_0)\|$ is related to the opening of the sharpness reduction phase in the early training results shown in Section 2.

G.3 Insights from uv model trained on (x, y)

In this section, we gain insights into the effect of setting network output to zero at initialization using uv model trained on an example (x, y). In particular, we show that loss catapults at $k_{loss} = 2$ and sharpness increases during early training.

Consider the uv model trained on a single training example (x, y) with $y \neq 0$[5]

$$f(x) = \frac{1}{\sqrt{w}} \sum_{i}^{w} u_i v_i \, x.$$

This simplifies the loss function to

$$\mathcal{L} = \frac{1}{2} \left(f(x) - y \right)^2 = \frac{1}{2} \Delta f^2, \tag{45}$$

where Δf is the residual. The trace of the Hessian $\operatorname{tr}(H)$ is

$$\operatorname{tr}(H) = \frac{x^2}{w} \left(\|v\|^2 + \|u\|^2 \right). \tag{46}$$

The Frobenius norm can be written in terms of the trace and the network output

$$\|H\|_F^2 = \lambda^2 + 2x^2 \Delta f^2 \left(1 + \frac{2f}{w \Delta f} \right). \tag{47}$$

The function and residual updates are given by

$$f_{t+1} = f_t - \eta \operatorname{tr}(H_t) + \frac{\eta^2 x^2}{w} f_t \Delta f_t^2 \tag{48}$$

$$\Delta f_{t+1} = \Delta f_t \left(1 - \eta \operatorname{tr}(H_t) + \frac{\eta^2 x^2}{w} f_t \Delta f_t \right). \tag{49}$$

Similarly, we can obtain the trace update equations

$$\operatorname{tr}(H_{t+1}) = \operatorname{tr}(H_t) + \frac{\eta \Delta f_t^2 x^2}{w} \left(\eta \operatorname{tr}(H_t) - 4 \frac{f_t}{\Delta f_t} \right). \tag{50}$$

Let us analyze them for the networks with zero output at initialization. The loss at the first step increases if

$$\left\langle \frac{L_1}{L_0} \right\rangle = \left\langle \left(1 - \eta \operatorname{tr}(H_0) + \frac{\eta^2 x^2}{n} f_0 \Delta f_0 \right)^2 \right\rangle > 1 \tag{51}$$

$$\tag{52}$$

Setting $f_0 = 0$ and scaling the learning rate as $\eta = k / \operatorname{tr}(H_0)$, we see that the loss increases at the first step if $k > 2$.

---

[5] Note that for $y = 0$, the network is already at a minimum.

$$\left\langle \frac{L_1}{L_0} \right\rangle = \left\langle (1-k)^2 \right\rangle > 1 \tag{53}$$

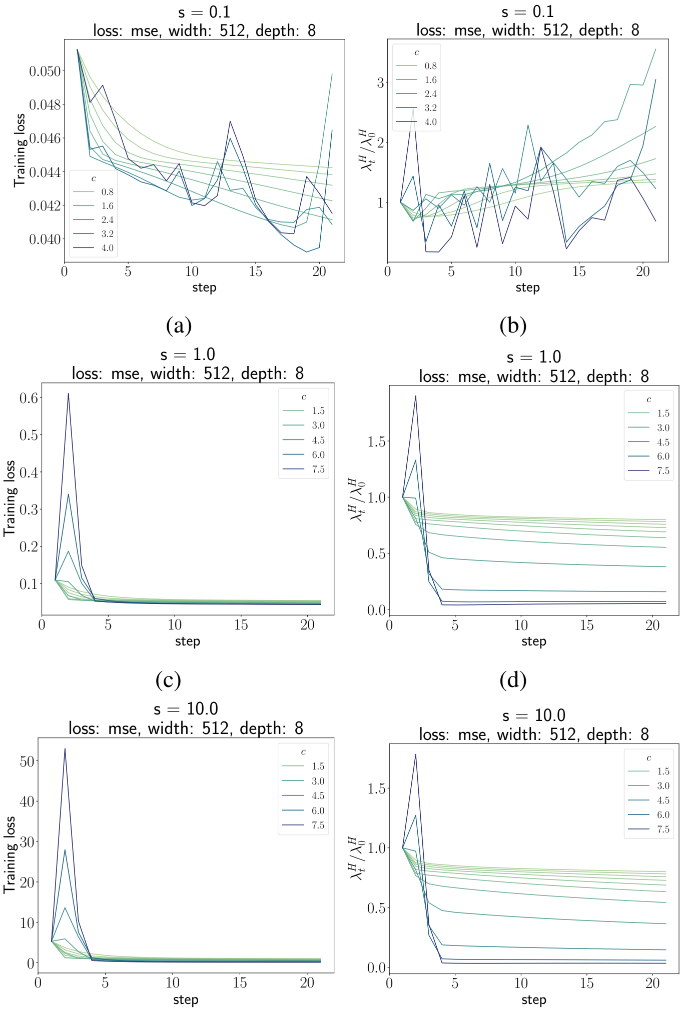
Next, we analyze the change in trace during the first training step. Setting $f_0 = 0$, we observe that the trace increases for all learning rates

$$\operatorname{tr}(H_1) = \operatorname{tr}(H_0) + \frac{\eta^2 x^2}{w} \Delta f_0^2 \operatorname{tr}(H_0), \tag{54}$$

modulated by the learning rate and width. Finally, we analyze the change in Frobenius norm in the first training step at $k = k_{loss}$, which implies $\Delta f_1^2 = \Delta f_0^2$,

$$\left\langle \Delta \|H_1\|^2 \right\rangle = \left\langle \operatorname{tr}(H_1)^2 - \operatorname{tr}(H_0)^2 + 2x^2 \left(\Delta f_1^2 - \Delta f_0^2 \right) \right\rangle . \tag{55}$$

As $\operatorname{tr}(H)$ increases in the first training step, $\|H\|_F$ also increases in the first training step.



(a) (b)

(c) (d)

(e) (f)

Figure 31: The early training dynamics of FCNs with a fixed output scale trained on the CIFAR-10 dataset with MSE loss using gradient descent.

H The effect of output scale on the training dynamics

Given a neural network function $f(x)$ with depth d and width w, we define the scaled network as $f_s(x) = \alpha f(x)$, where α is referred to as the output scale. In this section, we empirically study the impact of the output scale on the early training dynamics. In particular, we show that a large (resp. small) value of $\|f(x; \theta_0)\|$ relative to the one-hot encodings of the labels causes the sharpness to decrease (resp. increase) during early training. Interestingly, we still observe an increase in $\langle c_{loss} \rangle$ with d and $1/w$, unlike the case of initializing network output to zero, highlighting the unique impact of output scale on the dynamics. For simplicity, we train FCNs using gradient descent with MSE loss using a subset consisting of 4096 examples of the CIFAR-10 dataset, as in the previous section.

H.1 The effect of fixed output scale at initialization

In this section, we study the training dynamics of models trained with a fixed output scale at initialization. Given a network output function $f(\theta)$, we define the 'scaled network' as

$$f_s(\theta) = \frac{s f(\theta)}{\|f(\theta_0)\|}, \tag{56}$$

where s is a scalar, fixed throughout training. By construction, the network output norm $\|f_s(\theta_0)\|$ equals s. For standard initialization, $s = \|f(\theta_0)\| = \mathcal{O}(\sqrt{k})$, where k are the number of classes.

Figure 31 shows the training dynamics of FCNs for three different values of the output scale s. The training dynamics of networks with $s = 1.0$ and $s = 10.0$ share qualitative similarities. In contrast, networks initialized with a smaller output scale ($s = 0.1$) exhibit distinctly different dynamics. In particular, we observe that for large output scales ($s \gtrsim 0.5$) sharpness decreases during early training, while sharpness increases for small output scales [6]. Furthermore, the training dynamics tends to be noisier at small output scales, making it difficult to characterize catapult dynamics amidst these fluctuations. In summary, the training dynamics of networks with small output scale deviate from the training dynamics discussed in the main text, particularly as the sharpness quickly increases during early training.

Figure 32 shows the trends of various critical constants with width for FCNs for three different values of s. Similar to vanilla networks, we observe that c_{loss} increases with d and $1/w$. In comparison, sharpness decreases (increases) for large (small) values of s. These experiments suggest that the output scale primarily influences the increase/decrease in sharpness during early training and does not affect the scaling of c_{loss} with depth and width.

Note that we do not generate phase diagrams for these experiments as the training dynamics of networks with small output scales at initialization deviate from the training dynamics disucssed in the main text.

H.2 Scaling the output scale with width

In this section, we study the training dynamics of models with an output scale scaled with width as $\alpha = w^{-\sigma}$, which is commonly used in the literature [18, 6, 4]. We consider three distinct σ values $\{-0.5, 0.0, 0.5\}$, where $\sigma = -0.5$ represents the lazy regime, $\sigma = 0.5$ corresponds to feature learning (rich) regime and $\sigma = 0.0$ correponds to standard (vanilla) initialization.

Figure 33 shows the training loss and sharpness trajectories of FCNs trained on for different σ values. We observe that the training trajectories in the lazy regime look identical to standard initialization. In comparison, the training trajectories in the feature learning regime is distinctly different. We observe that in the standard and lazy regimes, sharpness decreases during early training, whereas sharpness tends to increase in the feature learning regime and eventually oscillates around the edge of stability regime. Moreover, we observe that sharpness can catapult before the training loss in the feature learning regime (compare catapult peaks in 33(e, f)). These results are in parallel to the fixed output scale networks studied in the pervious section.

---

[6] We empirically observed that sharpness reduces for output scales as small as $s \sim 0.5$, which is relatively small compared to $\sqrt{k}$.

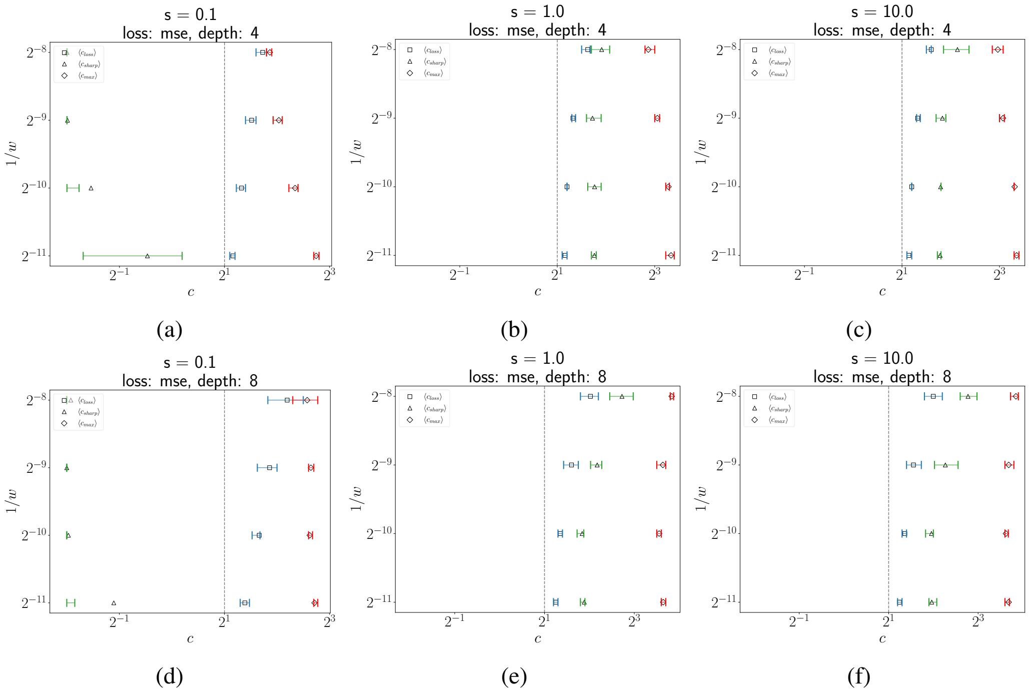


Figure 32: The phase diagrams of early training dynamics for ReLU FCNs with fixed output scale trained on a subset of the CIFAR-10 dataset using MSE loss using gradient descent. Each data point is an average over 10 initializations. The horizontal bars around the average data point indicate the region between 25% and 75% quantile.

Figure 34 summarizes the early training dynamics of FCNs with different σ values. We observe similar results as in the previous section. The output scale affects the initial increase/decrease of sharpness but does not affect the scaling trend of c_{loss} with depth and width. Moreover, we observe a systematic pattern of c_{max} scaling with width. In the lazy regime, we observe that c_{max} increases with $1/w$, while c_{max} decreases with $1/w$ in the feature learning regime.

I Sharpness curves in the intermediate saturation regime

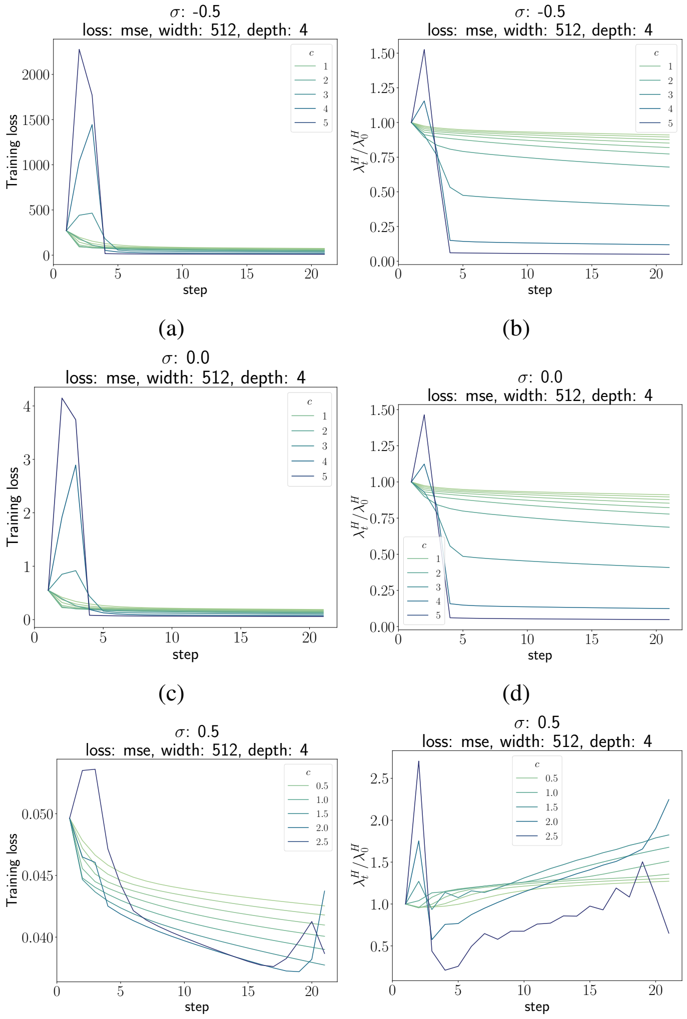
This section shows additional results for Section 3 for MSE loss. Cross-entropy results are shown in Appendix F. Figures 35 to 39 show the normalized sharpness curves for different depths and widths.

I.1 Estimating the sharpness

This paragraph describes the procedure for measuring the sharpness to study the effect of the learning rate, depth, and width in the intermediate saturation regime. We measure the sharpness λ_τ^H at a time τ in the middle of the intermediate saturation regime. We choose τ so that $c\tau \approx 200$, for learning rates $c = 2^x$, where $x \in [-1.0, 4.0]$ in steps of 0.1. The value 200 is chosen such that τ is in the middle of the intermediate saturation regime. Next, we measure sharpness over a range of steps $t \in [\tau - 5, \tau + 5]$ and average over t to reduce fluctuations. We repeat this process for various initializations and obtain the average sharpness.

I.2 Estimating the critical constant c_{crit}

This subsection explains how to estimate c_{crit} from sharpness measured at time τ. First, we normalize the sharpness with its initial value, and then average over random initializations. Next, we estimate the critical point c_{crit} using the second derivative of the order parameter curve. Even if the obtained averaged normalized sharpness curve is somewhat smooth, the second derivative may become extremely noisy as minor fluctuations amplify on taking derivatives. This can cause difficulties in



(a) (b) (c) (d) (e) (f)

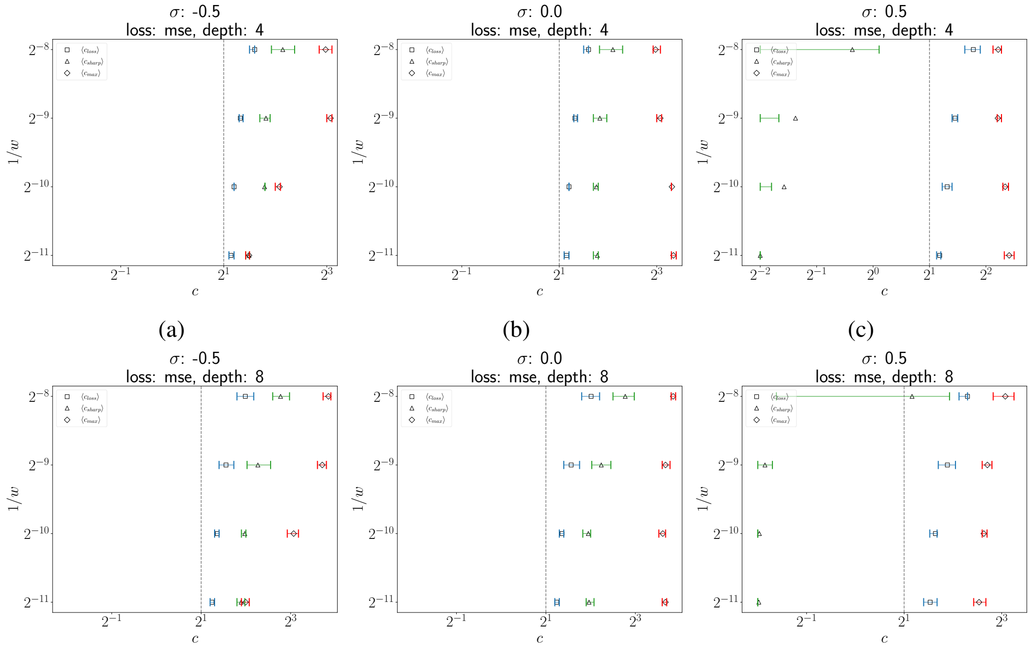
Figure 33: The early training dynamics of FCNs with output scale $\alpha = w^{-\sigma}$ trained on the CIFAR-10 dataset with MSE loss using gradient descent.

obtaining c_{crit}. We resolve this issue by estimating the smooth derivatives of the averaged order parameter with the Savitzky–Golay filter [56] using its scipy implementation [60]. The estimated c_{crit} is shown by vertical lines in the sharpness curves in Figures 35 to 39.

J The effect of batch size on the reported results

J.1 The early transient regime

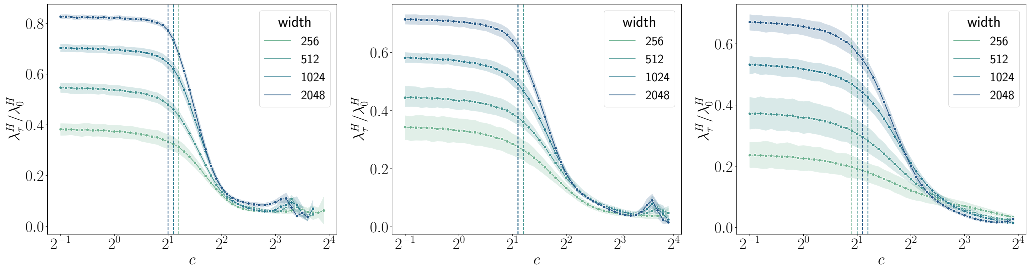
Figure 40 shows the phase diagrams of early training dynamics of FCNs with $d = 4$ trained on the CIFAR-10 dataset using two different batch sizes. The phase diagram obtained is consistent with the findings presented in Section 2, except for one key difference. Specifically, we observe that when d/w is small and small batch sizes are used for training, sharpness may increase from initialization at relatively smaller values of c. This is reflected in Fig. 40 by $\langle c_{sharp} \rangle$ moving to the left as B is reduced from 512 to 128. However, this initial increase in sharpness is small compared to the sharpness catapult observed at larger batch sizes. We found that this increase at small batch sizes is due to fluctuations in gradient estimation that can cause sharpness to increase above its initial value by chance.



(a) (b) (c)

(d) (e) (f)

Figure 34: The phase diagrams of early training dynamics for ReLU FCNs with varying depths and output scale.



(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

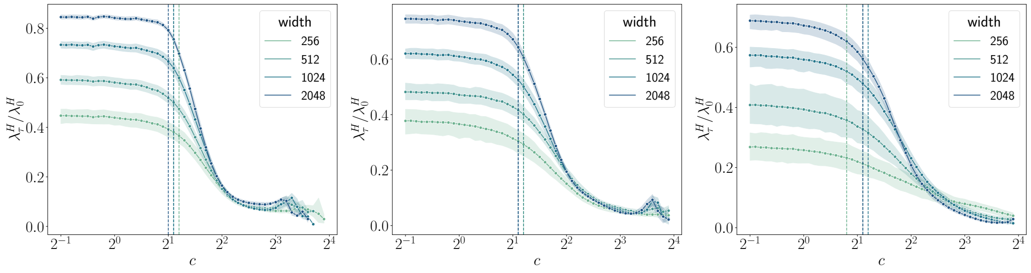
Figure 35: Sharpness measured at $c\tau = 200$ against the learning rate constant for FCNs trained on the MNIST dataset, with varying depths and widths. Each curve is an average over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ.

J.2 The intermediate saturation regime

Figure 41 shows the normalized sharpness, measured at $c\tau = 200$, and its derivatives for various widths and batch sizes. The results are consistent with those in Section 3, with a lowering in the peak heights of the derivatives χ and χ' at small batch sizes. The lowering of the peak heights means the full width at half maximum increases, which implies a broadening of the transition around c_{crit} at smaller batch sizes.

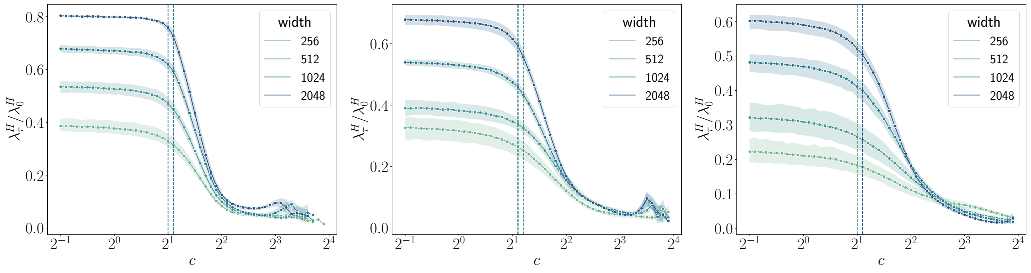
K The effect of bias on the reported results

In this section, we show that FCNs with bias show similar results as presented in the main text. We considered FCNs in SP initialized with He initialization [28].



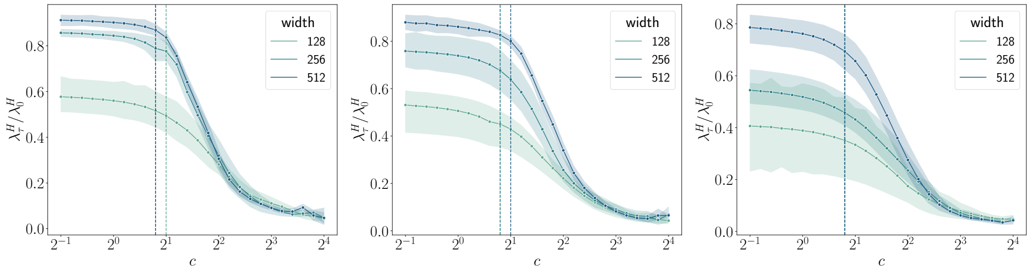
(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

Figure 36: Sharpness measured at $c\tau = 200$ against the learning rate constant for FCNs trained on the Fashion-MNIST dataset, with varying depths and widths. Each curve is an average over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ.



(a) $d = 4$ (b) $d = 8$ (c) $d = 16$

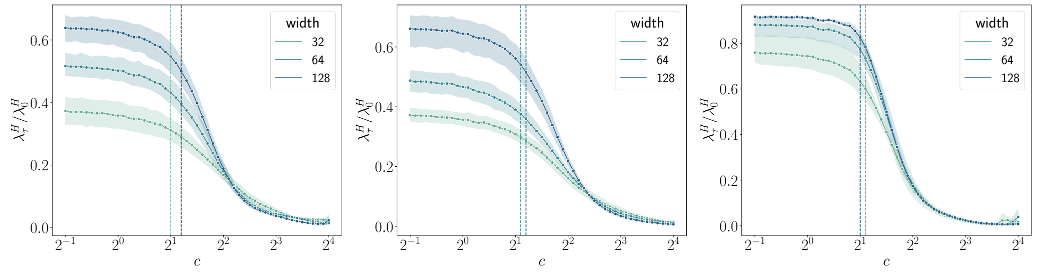
Figure 37: Sharpness measured at $c\tau = 200$ against the learning rate constant for FCNs trained on the CIFAR-10 dataset, with varying depths and widths. Each curve is an average over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ.



(a) $d = 5$ (b) $d = 7$ (c) $d = 10$

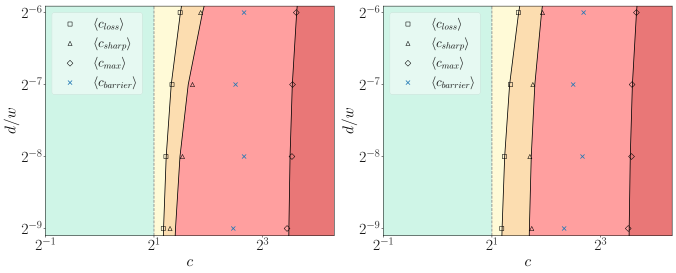
Figure 38: Sharpness measured at $c\tau = 200$ against the learning rate constant for Myrtle-CNNs trained on the CIFAR-10 dataset, with varying depths and widths. Each curve is an average of over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ.

Figure 42 shows the phase diagrams of early training for FCNs with bias trained on the CIFAR-10 dataset. We observe a similar phase diagram compared to the no-bias case (compare with Figure 25).



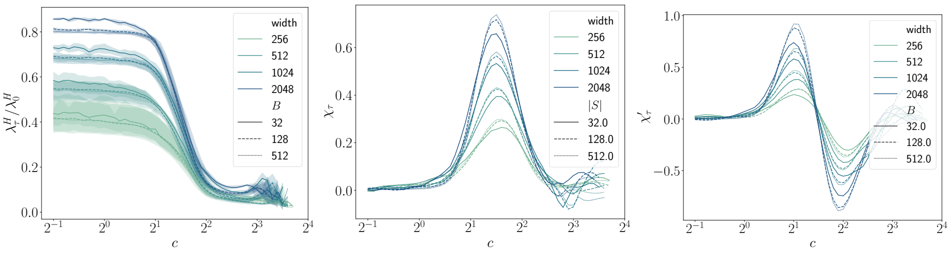
(a) $d = 18$ (b) $d = 34$ (c) $d = 50$

Figure 39: Sharpness measured at $c\tau = 200$ against the learning rate constant for ResNets trained on the CIFAR-10 dataset, with varying depths and widths. Each curve is an average of over ten initializations, where the shaded region depicts the standard deviation around the mean trend. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ.



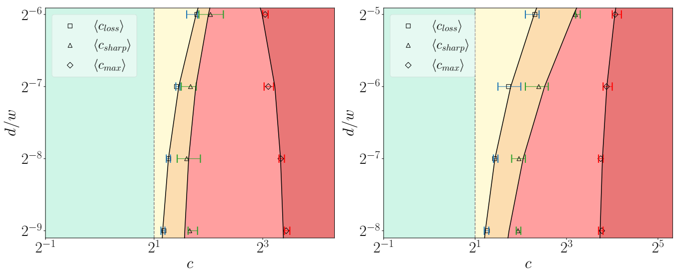
(a) $B = 128$ (b) $B = 512$

Figure 40: The phase diagram of early training for FCNs with $d = 4$ trained on the CIFAR-10 dataset with MSE loss using SGD with different batch sizes.



(a) (b) (c)

Figure 41: (a) Normalized sharpness measured at $c\tau = 200$ against the learning rate constant for FCNs with $d = 4$ trained on the CIFAR-10 dataset, with varying widths. Each data point is an average over 10 initializations, where the shaded region depicts the standard deviation around the mean trend. (b, c) Smooth estimations of the first two derivatives, χ_τ and χ'_τ, of the averaged normalized sharpness wrt the learning rate constant.



(a) $d = 4$

(b) $d = 8$

Figure 42: The phase diagram of early training for FCNs with bias trained on the CIFAR-10 dataset with MSE loss using SGD with different depths.