The present document contains revised figures from the main submission. Figure A extends Figure 1 from the main paper, presenting an upper bound on the true Lipschitz constant (Equation 2), together with the lower bound studied in the main paper (Equation 1, reported here for convenience).

1 Observations on the Lipschitz constant

Equation 1 in the present rebuttal (i.e. Equation 2 in the main paper) provides an empirical estimate of the Lipschitz constant of a piece-wise linear neural network function f. Particularly, Equation 1 may underestimate the true Lipschitz constant γ as it entails restricting the computation of the supremum to the activation regions containing training data.

$$(\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}\|_2^2)^{\frac{1}{2}} := \Big(\frac{1}{N}\sum_{n=1}^{N}\sup_{\mathbf{x}:\|\mathbf{x}\|\neq 0}\frac{\|\boldsymbol{\theta}_{\epsilon_n}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2}\Big)^{\frac{1}{2}} \tag{1}$$

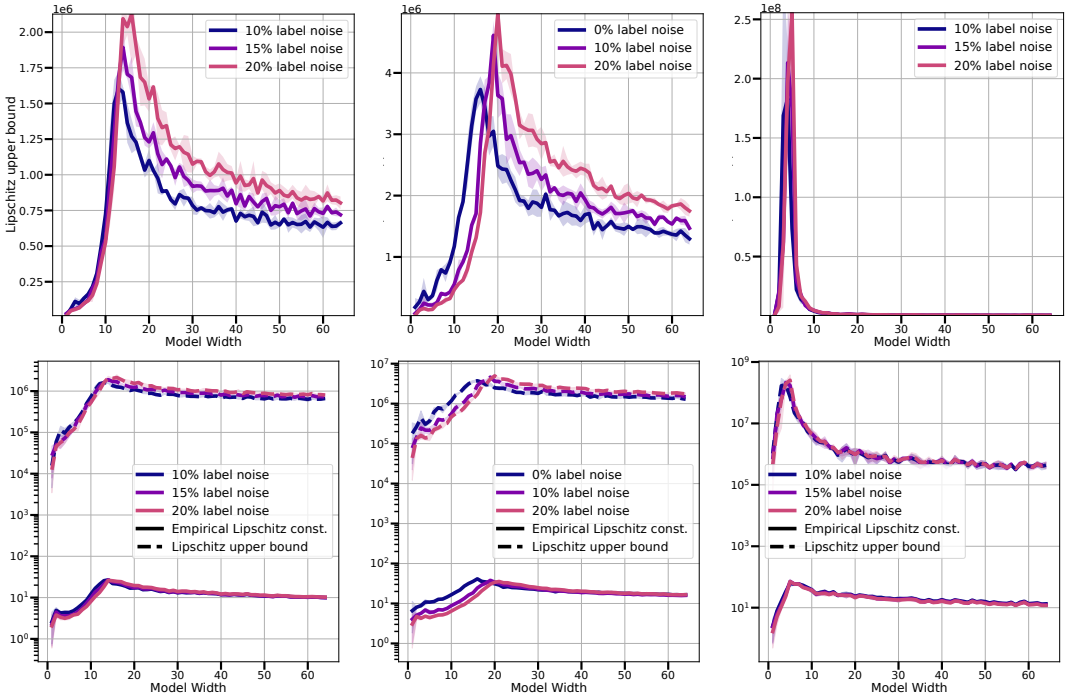

Figure A: (Top) Upper bound on the true Lipschitz constant, estimated according to Equation 2. (Bottom) The same upper bound on the true Lipschitz constant, visualized in y-logscale together with the empirical Lipschitz constant. From left to right: ConvNets trained on CIFAR-10 (left), CIFAR-100 (middle) and ResNets trained on CIFAR-10 (right).

In order to study whether the double descent behaviour observed in Figure 1 is reflected in the

true Lipschitz constant γ, we consider the following upper bound, where $\Omega = \text{dom}(\mathbf{f})$:

$$\gamma := \sup_{\mathbf{x}\in\Omega} \|\nabla_{\mathbf{x}}\mathbf{f}\| = \sup_{\mathbf{x}\in\Omega} \| \prod_{\ell=1}^{L} \phi^{\ell}(\mathbf{x}^{\ell})\theta^{\ell}\| \leq \sup_{\mathbf{x}\in\Omega} \prod_{\ell}^{L} \|\theta^{\ell}\| \leq \prod_{\ell=1}^{L} \lambda_{\max}(\theta^{\ell}) \tag{2}$$

where $\lambda_{\max}(\phi^{\ell}(\mathbf{x}^{\ell})) = 1$ for ReLU activation function ϕ.

Figure A presents a double descent trend for the Lipschitz contant upper bound, closely mirroring (albeit at a different scale) the behaviour of the empirical lower bound.

2 Loss Hessian Spectrum

Theorem 2 in the main paper presents an upper bound on input-space smoothness of the loss landscape, as measured by $\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)\|$, in terms of the mean loss curvature H.

To complement our analysis and validate Theorem 2, Figure B extends the results presented in Figure 2 of the main paper, visualizing the loss input-space gradient $\mathbb{E}_{\mathcal{D}}\|\nabla_{\mathbf{x}}\mathcal{L}(\theta, x, y)\|$, together with the mean loss curvature $\text{tr}(H)$, the maximum Hessian eigenvalue $\lambda_{\max}(H)$, and the smallest Hessian eigenvalue $\lambda_r(H)$, as model size increases. It can be observed how all quantities undergo the same trend, peaking near the interpolation threshold.

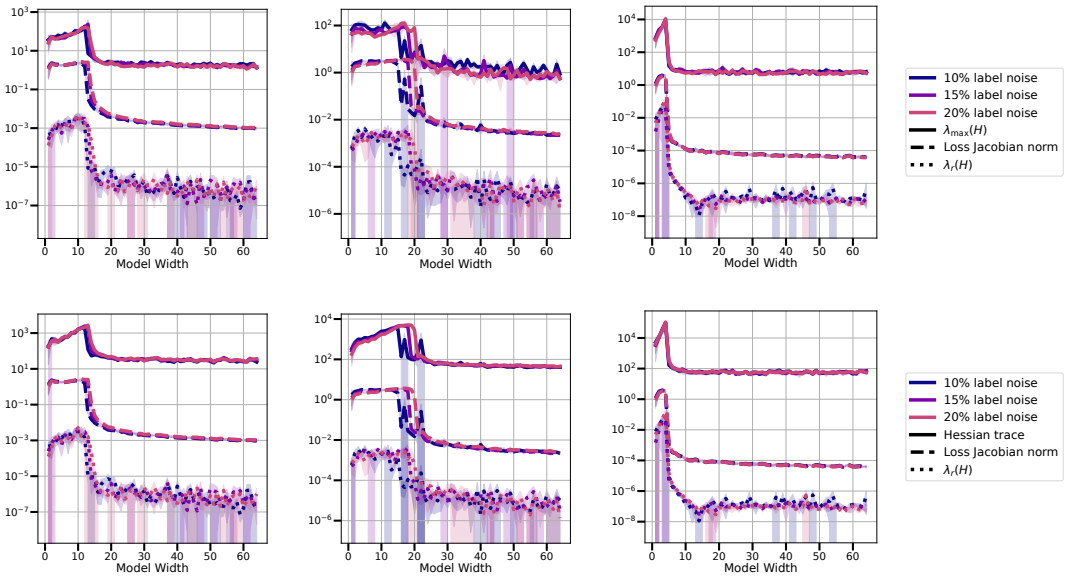

Figure B: (Top) **Maximum and minimum non-zero loss curvature** in parameter space, together with input-space loss Jacobian. (Bottom) **Mean curvature, and minimum non-zero curvature** for the loss in parameter space. From left to right: ConvNets trained on CIFAR-10 (left), CIFAR-100 (middle) and ResNets trained on CIFAR-10 (right). All values are reported in log-y scale to better separate models in the interpolating regime.

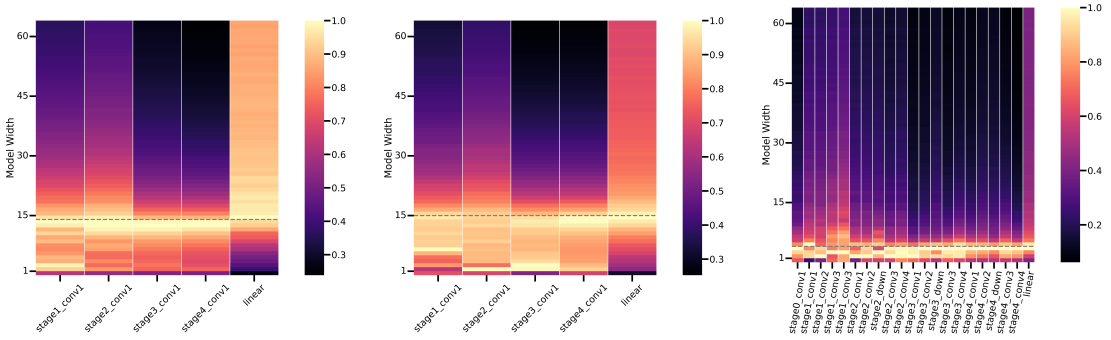

Figure C: **Scaled distance from initialization for each layer** of ConvNets trained on CIFAR-10 with 20% noisy labels (left), CIFAR-100 (middle), and ResNet18s trained on CIFAR-10 with 20% noisy labels (right). For each ConvNet and most ResNet layers, distance of the converged layer's parameters from initialization follows double descent, peaking at the interpolation threshold (dashed), suggesting global boundedness of the model function beyond training data for large models.

3 Distance from Initialization of Trained Parameters

In Figure C, we plot the saled distance of the parameters $\boldsymbol{\theta}_T^\ell$ of every layer from their value at initialization $\boldsymbol{\theta}_0^\ell$, namely $d_{0,T}^\ell = \frac{\|\theta_T^\ell - \theta_0^\ell\|_2}{\theta_0^\ell}$.