DOUBLE DESCENT IN LEARNING RATE IN THE FEATURE LEARNING REGIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Double descent is the non-monotonic behavior of test error as a function of a single hyperparameter. It is well documented for model width, dataset size and training time. We show that it also happens along a different axis: the learning rate. We give a mechanistic account of how this happens and under what conditions it appears. We also separate it from other large-learning-rate effects. The Maximal Update Parametrization (μP) is widely thought to make the optimal learning rate stay the same as width changes, so it can be tuned on small models and reused on large ones. We show that on some datasets this transfer fails due to the learning rate double descent. As width increases, the optimal learning rate jumps from the low-learning-rate regime to the high-learning-rate regime. As a result, a learning rate tuned on a small proxy model can lead to clearly worse choices at scale.

1 INTRODUCTION

At scale, learning rates are no longer tuned directly but transferred from small proxy models (Yang et al., 2021; Mlodozeniec et al., 2025). Under the Maximal Update Parametrization (μP) (Yang & Hu, 2021), the test-optimal learning rate stays nearly constant across width, empirically (Yang et al., 2021; Everett et al., 2024) and, in an idealized setting, provably (Hayou, 2026). Transfer quietly assumes a single learning rate optimum that moves predictably with scale.

Double descent shows that test error can be non-monotone in model size, sample size and training time (Belkin et al., 2019; Nakkiran et al., 2021). If gradient descent (GD) tracks gradient flow (GF), the learning rate only sets the time ηT reached within a budget T. We show that in the feature-learning regime this breaks. At a fixed budget, test error can have two minima in the learning rate, two to five orders of magnitude apart (fig. 2): an early-stopping minimum (*Dip 1*), an interpolation plateau, and a second minimum (*Dip 2*) where the network still interpolates but generalizes better than on the plateau.

What matters is less the path GD takes than where it may stop: on the squared loss, GD at rate η can only converge to interpolants whose tangent kernel ($\boldsymbol{K}$) satisfies $\lambda_{\max}(\boldsymbol{K}/n) \leq 2/\eta$ (Wu et al., 2018), and this ceiling regularizes the learned features. Because the two minima compete, the optimal learning rate can jump with scale, even under μP. Our **contributions** are threefold:

- **Double descent in the learning rate:** We find it in MLPs and ResNet-18 on CIFAR-10, and without interpolation in a transformer. Dip 2 is not epoch-wise double descent but a finite-step effect.
- **When and why it appears:** We derive necessary conditions (large steps, feature learning and overparameterization) that, with ablations, explain why standard training hides it. For SiLU networks, we prove under explicit assumptions how the higher learning rate can lead to improved generalization while interpolating the (noisy) training set (sections 3 and 4.2).
- **A failure mode of μTransfer:** With clean labels under μP, the optimal learning rate can jump between dips when increasing width, by more than two orders of magnitude (section 5).

1.1 RELATED WORK

Double descent was first observed as non-monotonicity in the test loss versus model capacity (Belkin et al., 2019). Subsequent studies found similar effects in training duration and other hyperparame-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

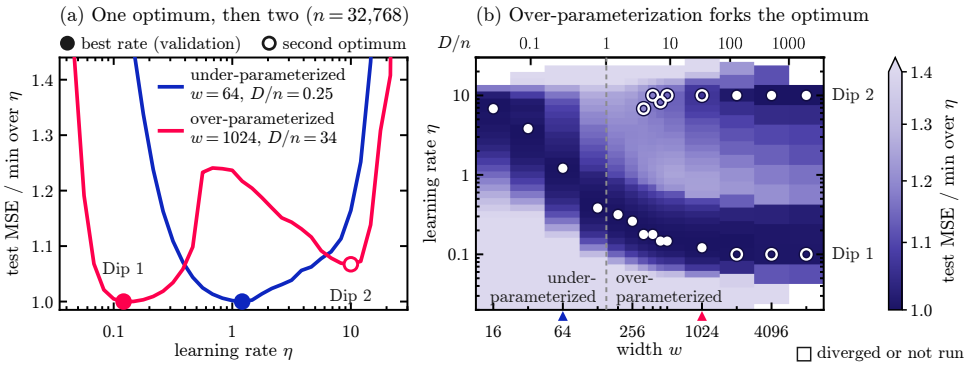

Figure 1: **Over-parameterization splits the learning-rate optimum, even under μP.** Two-layer ReLU networks in μP trained by gradient descent on $n = 32{,}768$ examples; D = parameter count. **(a)** An under-parameterized net has one optimal learning rate; an over-parameterized one has two (Dip 1, Dip 2). **(b)** Test loss over width and learning rate: the optimum splits once $D > n$. μP works as intended (see Section C.10 for a confirmation), yet the optimum's structure does not transfer across widths.

ters, like training set size and explicit kernel regularization (Nakkiran et al., 2021; d'Ascoli et al., 2020; Adlam & Pennington, 2020a; Yang et al., 2020; Hastie et al., 2019; Liang & Rakhlin, 2020; Holzmüller, 2020). These works vary capacity, sample size, training time, or explicit regularization. We instead hold these quantities fixed while sweeping the learning rate (LR) itself and observe a double descent. We contrast this to other forms of double descent in section 3. Recent years have shown a large interest in analyzing finite learning rate effects. Central findings are large LR induced implicit regularization (Barrett & Dherin, 2020; Beugnot et al., 2022), catapult mechanism where early instabilities improve model generalization (Lewkowycz et al., 2020; Li et al., 2019; Sadrtdinov et al., 2024; Zhu et al., 2023) as well as the edge of stability (EoS)(Cohen et al., 2021; Wu et al., 2018; Arora et al., 2022) and concrete models of the induced oscillatory trajectories like central flows (Cohen et al., 2024; Arora et al., 2022; Damian et al., 2022; Wang et al., 2021; Chen & Bruna, 2022; Kalra et al., 2023; Wang et al., 2023). We investigate these models in section 3.3 and show that the EoS is not the necessary driver of the improved generalization of the second LR dip.

Throughout the paper we will make use of the maximum update parameterization (μP) (Yang et al., 2021; Yang & Hu, 2021). This network parameterization is believed to allow for LR width hyperparameter transfer, which was recently proven in an idealized setting (Hayou, 2026). We show that double descent in LR can create scenarios where the optimal LR jumps orders of magnitude (this jump differs from the findings in Kalra & Barkeshli). See section E for a more comprehensive related work section.

2 Setup and the Phenomenon

2.1 Setup

Inputs $\boldsymbol{x} \sim p$ have a clean target $f_*(\boldsymbol{x})$ (the one-hot label in classification). A network $f_{\boldsymbol{\theta}}$ is trained on n examples whose labels may be noisy, e.g. $y_i = f_*(\boldsymbol{x}_i) + \sigma z_i$ with $z_i \sim \mathcal{N}(0, 1)$. We write

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\big), \qquad \mathcal{R}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim p}\, \ell_{\text{eval}}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), f_*(\boldsymbol{x})\big) \tag{1}$$

for the training loss and the population risk. The risk is always measured against clean targets, on an independent test set. Here ℓ is the half squared error or the cross-entropy, and ℓ_{eval} is the squared error unless stated otherwise, so that $\mathcal{R}(\boldsymbol{\theta}) = \|f_{\boldsymbol{\theta}} - f_*\|^2_{\text{pop}}$ with $\|g\|^2_{\text{pop}} = \mathbb{E}_{\boldsymbol{x}\sim p}\|g(\boldsymbol{x})\|^2$. We train with full-batch GD,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \nabla\mathcal{L}(\boldsymbol{\theta}_t), \qquad t = 0, \ldots, T-1, \tag{2}$$

or with mini-batch SGD, at a constant rate η for a fixed budget of T updates. We use no weight decay, clipping or schedule, so the step size is the only knob. All networks use the Maximal Update Parametrization (μP) in its (a, b, c) form (Yang & Hu, 2021; Yang et al., 2021; Everett et al., 2024). Architectures, scaling exponents and data subsampling are given in App. C.

2.2 Double descent in the learning rate

Fix the budget and sweep the learning rate: the test error does not have a single optimum (fig. 2). This holds for a student–teacher MLP with label noise, for ResNet-18 on CIFAR-10, and for a transformer on a WikiText subset small enough to be memorized. The curve has three regimes:

- **Dip 1** (small rates): early stopping. The model has not yet fit the training set, and the minimum moves to smaller rates as the budget grows, roughly as $\eta \propto 1/T$.
- **Plateau**: the model fits the training set, noise included, and the test error sits well above Dip 1.
- **Dip 2** (large rates): the test error falls again although the model still fits the training set.

The two minima lie several decades apart and arise differently. Dip 1 depends only on the integrated time ηT, whereas Dip 2 has no counterpart in gradient flow (section 3). In the transformer the high-rate endpoints do not yet interpolate at this budget, so our analysis concerns the interpolating Dip 2 of the MLP and the ResNet.

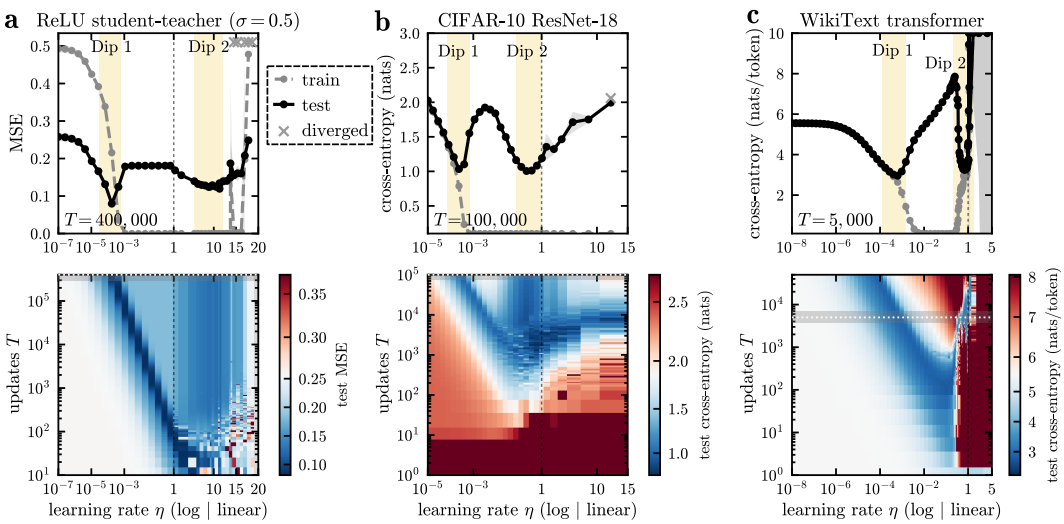

Figure 2: **Top:** training (dashed grey) and clean test error (black) against the learning rate η at a fixed update budget T for (a) a ReLU student–teacher MLP with label noise ($\sigma = 0.5$, $T = 400{,}000$), (b) ResNet-18 on CIFAR-10 ($T = 100{,}000$) and (c) a transformer on a WikiText subset ($T = 5{,}000$); (b) and (c) have no label noise. Test error has two minima separated by a plateau. In (a) and (b) Dip 2 interpolates the training set; in (c) the high-rate endpoints do not yet interpolate at this budget. $\times$: divergence. **Bottom:** test error over (η, T). Dip 1 moves along lines of constant ηT. Dip 2 opens above a threshold rate and does not move with T.

Question 1: What makes a second dip possible? (section 3).
Question 2: How can a higher learning rate lower generalization error while still memorizing the training data? (section 4).
Question 3: What does this mean for learning-rate transfer? (section 5).

3 What does and does not create the second dip?

A first-principles theory of the second dip for arbitrary architectures and data is beyond the scope of this paper. We give relevant necessary conditions. Throughout, $\boldsymbol{\theta}_F$ is the interpolant that GF reaches

from the same initialization, Λ_F its sharpness, and $\eta_c = 2/\Lambda_F$ its edge. $\mathcal{L}$ and $\mathcal{R}$ are the training loss (half squared error) and the population risk of equation 1, and GD is equation 2.

Necessary conditions for double descent in the learning rate (informal; Thm. B.2, App. B.2)

- **(i) Endpoint selection by stability:** For almost every initialization, GD at a constant rate $\eta > \eta_c$ cannot converge to the GF interpolant $\boldsymbol{\theta}_F$: a linearly stable interpolating endpoint has $\lambda_{\max}(\nabla^2 \mathcal{L}) \leq 2/\eta < \Lambda_F$. Below the edge, if every step keeps $\eta\lambda_{\max} \leq 2-\delta$, GD ends within $O(\eta/\delta)$ of $\boldsymbol{\theta}_F$ in parameter space.
- **(ii) Feature learning:** For a fixed kernel (a model that is linear in its parameters) and the squared loss, GD from a fixed initialization converges to the same interpolant at every stable constant rate $0 < \eta < \eta_c$ and diverges above η_c (Fig. 3c). The converged predictor therefore does not depend on the rate; only early-stopped kernel predictors can. If a feature-learning network instead converges at $\eta > \eta_c$ to a linearly stable interpolant, its NTK Gram matrix on the training inputs, $\boldsymbol{K}_\infty$, must have moved away from that of the GF interpolant, $\boldsymbol{K}_F$: $\|\boldsymbol{K}_\infty - \boldsymbol{K}_F\|_{\text{op}} \geq (1 - \eta_c/\eta)\, \lambda_{\max}(\boldsymbol{K}_F)$.
- **(iii) Overparameterization:** An interpolating second dip needs interpolants that agree on the training inputs but differ elsewhere. Locally this is guaranteed when the training Jacobian has full row rank n and $n < D$, the rank of the population tangent map: near such a point the interpolants form a manifold with $D - n$ directions that change the learned function. Conversely, interpolants that can be joined by a piecewise smooth path of constant training outputs, along which the training Jacobian has rank D, all compute the same function. For $n > D$, generic noisy labels cannot be interpolated at all.

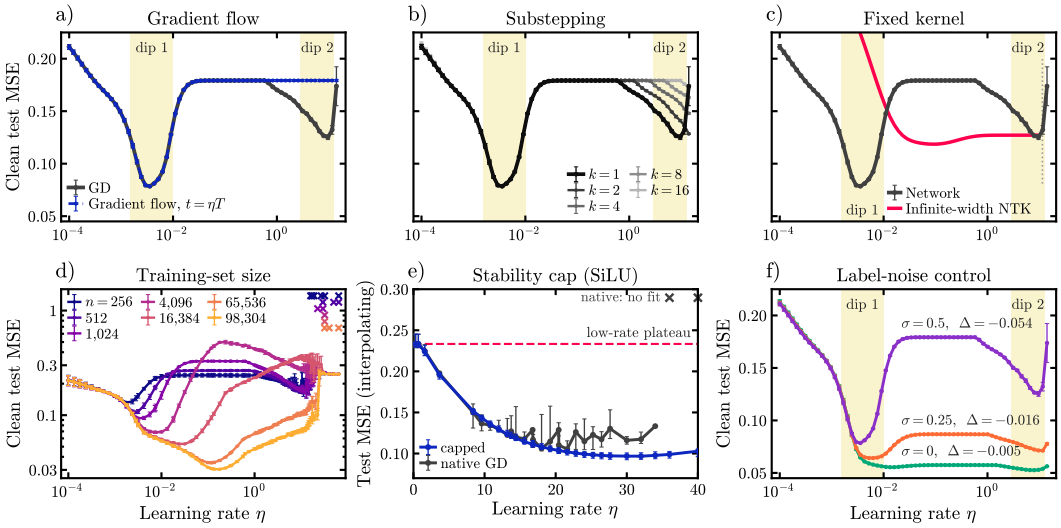

Figure 3: Unless stated, width-256 two-hidden-layer ReLU student and teacher, $d = 64$, $n = 1024$, $\sigma = 0.5$, full-batch GD for $T = 20{,}000$ steps. Yellow: dip 1 and dip 2 (where the $\sigma = 0.5$ GD curve is below half depth). **(a)** Gradient flow at $t = \eta T$ matches GD through dip 1, then stays on the plateau. **(b)** k substeps of size η/k remove dip 2 monotonically in k. **(c)** The infinite-width NTK has one minimum, on the network's plateau (dotted: kernel stability limit). **(d)** Width 64, varying n: dip 1 deepens and shifts right; no dip 2 for $n \geq 4096$. Crosses: a seed diverged. **(e)** SiLU. GD with step $\eta/2$ and top NTK eigenvalues capped at $2/\eta$ stays stable ($\max h\lambda_H = 1.81 < 2$), yet ends 59% below the plateau. Crosses: no native seed fits. **(f)** Δ: paired change from the plateau to the best $\eta \geq 1$; roughly $\propto \sigma^2$.

3.1 Empirical tests

We now show how each of the conditions influences the double dip.

Difference to epoch-wise double descent and GF (i). GD matches GF read off at $t = \eta T$ up to the plateau, including Dip 1, and then separates: at the high-rate minimum GF stays on the plateau (0.179) while GD reaches 0.125 (fig. 3a). Splitting each step into k steps of size η/k at fixed ηT loosens the ceiling to $2k/\eta$ and removes the gain monotonically (fig. 3b). At $k = 16$ the step (0.60) falls below $\eta_c \approx 0.63$, and the risk returns exactly to the plateau. Dip 1 depends on integrated time and Dip 2 on the step size, so this is not epoch-wise double descent.

The kernel must move (ii). With the infinite-width NTK, kernel gradient flow has a single minimum, at a rate where the network is on its plateau. It then converges to one rate-independent interpolant until it diverges (fig. 3c). At the canonical Dip 2 the network's NTK has moved by 94% of its top eigenvalue, on the bound of (ii). Hence feature learning is central to the second dip.

Interpolation and dataset size (iii). The high LR can only choose a different interpolant if the data leave some of the function undetermined, and for $n > D$ generic noisy labels cannot be interpolated at all. Increasing n from 256 to 98,304 (n/D up to 12.1) lowers the plateau and weakens the second dip until a single minimum remains (fig. 3d). Hence, the network needs to be sufficiently overparameterized for the second dip to occur.

Noise amplifies the second dip but is not required. The gain of Dip 2 over the plateau grows roughly quadratically in σ (fig. 3f). However, Dip 2 still exists for a noiseless teacher.

3.2 Where is the double descent in realistic training setups?

LR double descent requires overparameterization, where the model has enough capacity to interpolate its training set (iii). Modern LLMs are trained on far more tokens than parameters, so they never enter this regime. Consistent with this, we see no second dip once $n \gg D$, and the effect is absent in hyperparameter-transfer studies (Li et al., 2025; Mlodozeniec et al., 2025). Even in overparameterized settings, common practice hides it. Strong weight decay weakens the second descent (App. D.1), as does lazy training (NTK does not move sufficiently), and early stopping or short budgets remove the dips. When both minima do exist, they lie 2.3–3.7 decades apart at $T = 10^4$ and further apart at longer budgets, which is wider than a typical sweep (Fig. 18). Lastly, we note that GD is not commonly used in practice. We show in Section D.2 that Adam shows the double dip as well (Muon interestingly not, at least in our setup).

3.3 Is a catapult or EoS necessary for the dip?

At high learning rates, plain GD first goes through a catapult and then keeps oscillating at the edge of stability (App. D.3). Condition (i) only says that GD cannot end at the gradient-flow solution above η_c. It does not say how GD gets away from that solution, nor that the instability is what improves generalization. To separate the two, we impose the ceiling directly. We run GD with a small, stable step $h = \eta/2$, and whenever the largest eigenvalues of $\boldsymbol{K}/n$ approach $2/\eta$, we remove the part of the update that would push them higher (App. C.5). This is the constraint that central flows attribute to the edge of stability (Cohen et al., 2024). The resulting optimizer stays below the stability threshold at every recorded checkpoint, and its training loss never rises above its initial value. Yet it keeps Dip 2 and matches or improves on native GD: for SiLU across the whole range of rates, and for ReLU over a verified band of rates (fig. 3e). The same small step without the constraint is clearly worse, so the gain does not come from the smaller step. This does not contradict (i). An ordinary gradient step moves the gradient-flow endpoint only at second order in the step size, whereas the constraint moves it at first order (App. B.4). In native GD, unstable steps are how the ceiling is reached. The capped runs show that the instability itself is not needed for the gain. They are consistent with the ceiling being what helps, but they do not prove it on their own (section 4).

4 How can a higher LR lower generalization error?

Now, we discuss Question 2 from above. For that, we need to understand which component of the generalization error changes between the first dip, interpolation plateau, and second dip.

4.1 A DEEPER LOOK INTO THE BIAS VARIANCE TRADEOFF

Let $\boldsymbol{\theta}_\eta$ be the endpoint of training at rate η and $f_\eta := f_{\boldsymbol{\theta}_\eta}$. It is random through the initialization I, the training inputs D, the label noise N and the minibatch stream M. With $\bar{f}_\eta := \mathbb{E}[f_\eta]$,

$$\mathcal{R}_\eta := \mathbb{E}\big[\mathcal{R}(\boldsymbol{\theta}_\eta)\big] = \mathbb{E}\big[\|f_\eta - f_*\|_{\text{pop}}^2\big] = \underbrace{\|\bar{f}_\eta - f_*\|_{\text{pop}}^2}_{\text{pooled bias}^2} + \underbrace{\sum_{\varnothing \neq A \subseteq \{\mathsf{I},\mathsf{D},\mathsf{N},\mathsf{M}\}} V_A}_{\text{main effects and interactions}} , \tag{3}$$

where $V_A = \mathbb{E}\|g_A\|_{\text{pop}}^2$ with $g_A = \sum_{B \subseteq A} (-1)^{|A \setminus B|}\, \mathbb{E}[f_\eta \mid B]$ is the functional-ANOVA term of the sources in A. Sources absent in a setting are dropped. For classification, f_η is the softmax probability vector and $\|\cdot\|_{\text{pop}}^2$ is the multiclass Brier risk.

We estimated this decomposition by retraining matched ensembles at the first dip, interpolation plateau, and second dip (see Section C.6 for experimental details, Adlam & Pennington (2020b)). For the MLP and ResNet, the first dip is an underfit solution with high bias and low variance. Reaching the interpolation plateau substantially reduces bias but creates a large increase in variance. The second dip then reduces risk while retaining interpolation. In the teacher–student MLPs, the plateau-to-second-dip improvement is almost entirely a variance reduction. For clean ReLU and SiLU networks, variance falls by 0.00721 and 0.00256, respectively, while pooled bias slightly worsens. With noisy labels, variance reduction explains close to 100% of the ReLU gain and SiLU gain. CIFAR-10 exhibits a different balance. For ResNet-18, improvement of the pooled mean predictor accounts for 58.7% of the clean second-dip gain and 72.4% under 20% label noise, while variance reduction accounts for the remaining 41.3% and 27.6%. Thus the MLP second dip primarily selects a less variable interpolant, whereas the CIFAR-10 second dip also learns a substantially better mean predictor.

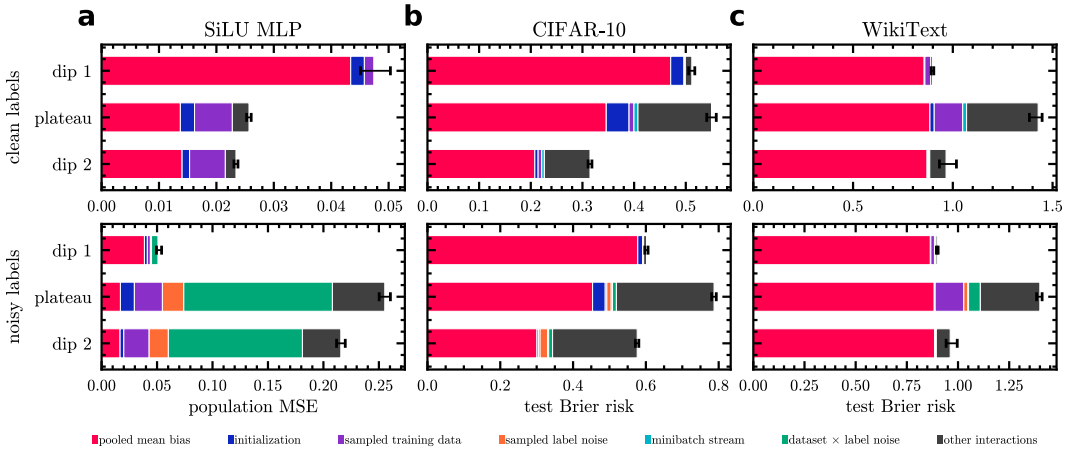

Figure 4: Rows show clean and noisy labels; columns show the **a)** SiLU teacher–student MLP, **b)** CIFAR-10 ResNet-18, and **c)** the WikiText transformer. Each bar decomposes population risk into pooled mean-function error and exact main or interaction effects of the sampled training factors. Black whiskers are 95% paired product-factor bootstrap intervals. Horizontal scales are panel-specific because the MLP uses population MSE whereas CIFAR-10 and WikiText use multiclass Brier risk. The MLP and CIFAR-10 dip-2 endpoints interpolate. At 5,000 updates the WikiText high-rate endpoints do not interpolate, so those panels describe the finite-time high-rate valley rather than a matched interpolating second dip.

4.2 A RIDGE ON THE LEARNED FEATURES

Now, we try to understand how a higher learning rate can lower the variance of the network. In the noisy SiLU MLP the pooled mean predictor does not move, so the Dip-2 gain is all variance (Fig. 4). We explain the terms containing the label-noise draw (53%), one network's variance over its label noise at a fixed training set; for the initialization terms (40%), only the direction.

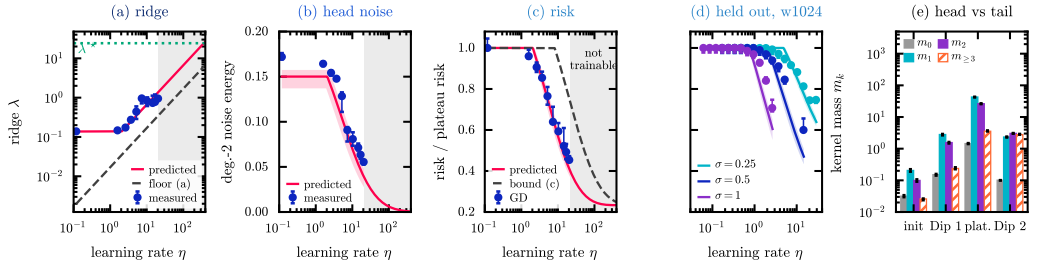

Figure 5: **Theorem 4.1 against trained SiLU networks.** Points are measured (median and range over seeds); lines are the theorem's closed form, computed from plateau measurements and two constants fixed on one network; bands show its uncertainty (details in App. B.7). **(a)–(c)** test items **(a)–(c)** on the canonical network; dashed lines are the floor in **(a)** and the risk bound it implies in **(c)**; grey rates do not train. **(d)** Held-out networks of width 1024, on the axis of **(c)**. **(e)** Kernel mass by Hermite degree: training inflates the head (degrees ≤ 2) more than the tail (hatched), and at Dip 2 the cap shrinks the head while the tail stays. Title colours match the formula each panel tests.

Head and tail. Let $\boldsymbol{j}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$ be the tangent features, so $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{j}(\boldsymbol{x}), \boldsymbol{j}(\boldsymbol{x}') \rangle$. For $p = \mathcal{N}(0, \boldsymbol{I}_d)$, expanding each feature in Hermite polynomials, $\boldsymbol{j} = \sum_{k \geq 0} \boldsymbol{j}_k$ with $\boldsymbol{j}_k$ of degree k, defines degree-k masses and splits the kernel into a *head* (degrees ≤ 2) and a *tail* (degrees ≥ 3),

$$m_k = \mathbb{E}_{\boldsymbol{x} \sim p} \|\boldsymbol{j}_k(\boldsymbol{x})\|^2, \qquad \boldsymbol{K}_{\mathrm{h}} = \langle \boldsymbol{j}_{\leq 2}, \boldsymbol{j}_{\leq 2} \rangle, \qquad \boldsymbol{K}_{\mathrm{t}} = \langle \boldsymbol{j}_{\geq 3}, \boldsymbol{j}_{\geq 3} \rangle, \tag{4}$$

with $\boldsymbol{K} = \boldsymbol{K}_{\mathrm{h}} + \boldsymbol{K}_{\mathrm{t}}$ up to cross terms of zero mean. Degree k spans $N_k = \binom{d+k-1}{k}$ directions. At $d = 64$, $n = 1024$, degrees ≤ 1 ($N_0 + N_1 = 65$) are resolved; degree 2 ($N_2 = m = 2080$) is learnable but underdetermined, $\psi = m/n_1 = 2.17$ with $n_1 = n - d - 1$, and carries the teacher's degree-2 energy r^2. Degrees ≥ 3 ($N_3 = 45760 \approx 45n$) are essentially unlearnable at this sample size: a kernel predictor's degree-3 part lies in an n-dimensional subspace, and for rotation-invariant kernels with $n \asymp d^2$ the degree-≥ 3 part of any target is not learned but acts as noise plus a ridge (Misiakiewicz, 2022; Hu & Lu, 2022). We therefore treat the teacher's energy there, $B_{\geq 3}$, as noise, $\sigma^2_{\mathrm{eff}} = \sigma^2 + B_{\geq 3}$.

For the same reason, tail features at distinct inputs are nearly orthogonal: the tail is nearly diagonal on the training inputs and nearly zero at a new input. Idealizing this as $\boldsymbol{K}_{\mathrm{t}}(\boldsymbol{X}, \boldsymbol{X}) = q\, m_{\geq 3} \boldsymbol{I}_n$ and $\boldsymbol{K}_{\mathrm{t}}(\boldsymbol{x}, \boldsymbol{X}) = \boldsymbol{0}$ (A2), an interpolant that extends its training values through $\boldsymbol{K}$ (A1) is ridge regression on the head:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{X})\, \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1} \boldsymbol{y} \approx \boldsymbol{K}_{\mathrm{h}}(\boldsymbol{x}, \boldsymbol{X}) \big(\boldsymbol{K}_{\mathrm{h}}(\boldsymbol{X}, \boldsymbol{X}) + q\, m_{\geq 3} \boldsymbol{I}_n \big)^{-1} \boldsymbol{y}. \tag{5}$$

Training inflates the head more (m_2 grows $265\times$ from initialization to the plateau, $m_{\geq 3}$ $138\times$; Section D.4, Figure 5e). The stability cap limits only the head, so a larger rate shifts noise into the tail:

❶ **Stability caps the coherent mode**	❷ **The cap reaches the head, not the tail**	❸ **The tail stores noise locally: a ridge**	❹ **Noise leaves the head, the risk falls**
$m_0 \leq (1+\delta)\, 2/\eta$	$m_2 \leq (1+\epsilon)(d+1)\, m_0$	$\lambda \propto m_{\geq 3}/m_2 \propto \eta$	head noise ↓ $\mathcal{R}$ ↓
the cap alone helps: Sec. 3.3	Lemma B.8; ϵ, T_0 measured	Thm. 4.1(a) · Fig. 5a	Thm. 4.1(b, c) · Fig. 5b–d

❶ **Stability caps the coherent mode.** GD at rate η settles only where $\lambda_{\max}(\boldsymbol{K}/n) \leq 2/\eta$; it gets there via EoS, but the cap alone gives the gain (Sec. 3.3). On the training inputs the degree-0 part of $\boldsymbol{K}/n$ is $m_0 \boldsymbol{1}\boldsymbol{1}^{\top}/n$: the coherent mode, in which all outputs move together, rank one with eigenvalue m_0. So the cap binds $m_0 \leq (1+\delta)\, 2/\eta$, where $1 + \delta = m_0/\lambda_{\max} \leq 1.05$ on the branch.

❷ **The cap reaches the head, not the tail.** In the first layer the tangent features are $g_u(\boldsymbol{x})\, \boldsymbol{x}$, g_u the back-propagated signal at unit u, and by Stein's lemma, $\mathbb{E}[g_u(\boldsymbol{x})\boldsymbol{x}] = \mathbb{E}\nabla g_u$, the linear part of g_u yields coherent and quadratic mass $\|\mathbb{E}\nabla g_u\|^2$ and $(d+1)\|\mathbb{E}\nabla g_u\|^2$. So $m_2 = (1+\epsilon)(d+1) S_1$ with $S_1 = \sum_u \|\mathbb{E}\nabla g_u\|^2 \leq m_0$ (Lemma B.8; $\epsilon = 0.04$–0.21). The tail has no such link: by (A2) it only shifts $\boldsymbol{K}/n$ by $q\, m_{\geq 3} \boldsymbol{I}_n/n$ and stays free ((A3): $T_0 = 0.77\, m^{\mathrm{pl}}_{\geq 3}$). Measured (Fig. 5e), from the plateau to Dip 2 the cap shrinks m_0 $14\times$ and m_2 $8.7\times$, while $m_{\geq 3}$ falls only $1.3\times$, so the tail-to-head ratio $m_{\geq 3}/m_2$ rises from 0.14 to 0.92.

❸ **The tail stores noise locally, so it acts as a ridge.** By (5), with $\boldsymbol{\alpha} = \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{y}$ the fit splits as $\boldsymbol{y} = \boldsymbol{K}_{\mathrm{h}}(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{\alpha} + q\, m_{\geq 3}\boldsymbol{\alpha}$: the head part reaches every test input, the tail part none. So the tail is a ridge on the head, $\lambda = q\, m_{\geq 3} m/(n_1 m_2)$ in units of the degree-2 features, as known for fixed kernels (Liang & Rakhlin, 2020); by ❶–❷ it grows at least linearly in η (Theorem 4.1(a)). Storage is cheap in the tail: a minimum-norm fit of noise energy E with $N > n$ features leaves $\approx E\, n/(N-n)$ at test inputs, $E/(\psi - 1) \approx 0.86E$ in the head but $\approx 0.02E$ in degree 3. Measured, the ridge grows linearly (Fig. 5a); (A2) is approximate: the tail extends 16–24% of the fitted noise.

❹ **Noise leaves the head, and the risk falls.** A larger ridge leaves the head a smaller share of the fitted noise and less noise energy at test inputs (Theorem 4.1(b)). The risk falls until $\lambda^\star$, beyond which the ridge removes more signal than noise (Theorem 4.1(c)). Measured (predicted), the head's share falls from 0.80 to 0.48 (0.84 → 0.51), the degree-2 energy of the fitted noise from 0.17 to 0.06 (0.15 → 0.04), and the risk to 0.45 of the plateau at Dip 2 (0.41; Fig. 5b, c) and to 0.845 at the rates of Fig. 4 (0.851), where the model books the whole gain to the noise terms (App. B.7).

Theorem 4.1 (The ceiling is a ridge on the learned features)**.** *Let $\boldsymbol{\theta}_\eta$ be a linearly stable interpolating GD limit at rate η. Assume that (A1) it extends its training values through its tangent kernel; (A2) degrees ≤ 1 are resolved, the degree-2 head is isotropic, and on the training inputs the tail acts as $q\, m_{\geq 3}\boldsymbol{I}$ and extends nothing; and (A3) $m_{\geq 3} \geq T_0$. Average risks over the direction of the teacher's degree-2 part and over the noise, treating the teacher's unlearned part as noise. Let $\nu = \big(1 + \psi + \lambda - \sqrt{(1+\psi+\lambda)^2 - 4\psi}\big)/(2\psi)$. Then:*

- *(a)* Ridge. *$f_{\boldsymbol{\theta}_\eta}$ is ridge regression on the degree-2 features, with $\lambda = q\, m_{\geq 3}\, m/(n_1 m_2) \geq q\, T_0\, d\, \eta/\big(4n_1(1+\epsilon)(1+\delta)\big) =: \lambda_{\mathrm{floor}}(\eta)$.*
- *(b)* Noise leaves the head. *The degree-2 head stores a share $m\nu/n$ of the fitted noise ($\psi\nu$ of the part not taken by degrees ≤ 1). Its energy at test inputs, $\sigma^2\psi\nu^2/(1-\psi\nu^2)$, falls as λ grows.*
- *(c)* Risk. *The population risk $\mathcal{R}_n(\lambda)$ of this ridge regression decreases in λ on $[0, \lambda^\star]$, $\lambda^\star = \psi\sigma_{\mathrm{eff}}^2/r^2$; its deterministic equivalent is $\mathcal{R}_{\mathrm{ridge}}(\lambda) = \frac{n}{n_1}\big(r^2(1-\nu)^2 + \sigma_{\mathrm{eff}}^2\big)/(1-\psi\nu^2) - \sigma^2$. Hence, if the plateau satisfies the same assumptions with ridge λ_{nat}, $\mathcal{R}(\boldsymbol{\theta}_\eta) \leq \mathcal{R}_n(\lambda_{\mathrm{floor}}(\eta)) < \mathcal{R}_{\mathrm{plateau}}$ whenever $\lambda_{\mathrm{nat}} < \lambda_{\mathrm{floor}}(\eta) \leq \lambda \leq \lambda^\star$.*

The monotonicity in (c) holds at finite n; the closed forms are asymptotic and, at our size, within 1.4% of the exact risk for $\lambda \geq 0.5$ (App. B.7 separates exact from asymptotic statements). For a tangent model the cap also bounds the head's part of the seed-to-seed variance by m_2, which suggests the direction of the initialization terms but not their size. On 15 held-out SiLU settings the predicted gain is 0.69–1.79 times the measured one (median 1.04; Fig. 5d, Table 2).

Steps ❶–❷ also hold for ReLU and GELU (App. B.6). Step ❸ holds along the whole SiLU branch. GELU follows the closed form at low rates but reaches its right flank before the model's optimum, and for ReLU the tail shrinks and the kernel is depleted at the training inputs (App. B.7). See Section B.5) for further discussion.

5 Implications for hyperparameter transfer (μP failure)

We now turn to Question 3. At width 64 the optimal rate lies in Dip 1; at every width from 128 to 2048 it lies in Dip 2, more than two decades higher (fig. 6a). To see why, we use clean labels and fixed data, so initialization is the only source of randomness. We use the following notation: L (Dip 1) and H (Dip 2).

μP failure model: Each variance and the bias difference follow simple laws (fig. 6c,d; bias law motivated in Section B.8, (a_j, b_j, c_j) are fitted constants):

$$R_H - R_L \simeq \underbrace{\frac{a_H}{w + c_H} - \frac{a_L}{w + c_L}}_{\Delta V = V_H - V_L} + \underbrace{b_\infty - \frac{b_0}{1 + (w/w_B)^2}}_{\Delta B = B_H - B_L}. \tag{6}$$

Dip 2 is optimal iff $R_H - R_L < 0$. The crossover is a root of this sum, not a fitted parameter.

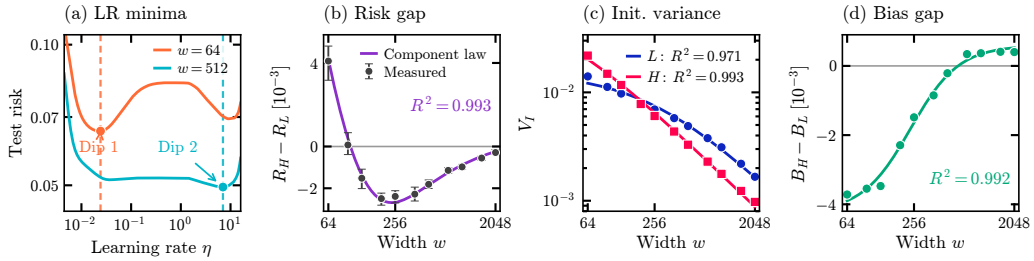

Figure 6: Clean ReLU teacher–student ($n = 1024$, $\sigma = 0$, $T = 10^4$, 16 initializations). **(a)** Test risk at $w = 64$ and 512; dashed lines mark the optimal rate. **(b)** Risk gap between the best Dip-2 and Dip-1 solutions; the curve is the sum of the fits in (c) and (d). Error bars: 95% paired bootstrap over initializations. **(c)** Initialization variance per branch. **(d)** Squared-bias gap, $B = R - V$. R^2 values are in-sample. Dip-1 risks are read off gradient flow at $t = \eta T$.

Both variances decay roughly as $1/w$, as expected under μP (Bordelon & Pehlevan, 2023), but with different constants. The high-rate branch starts noisier but decays from the smallest widths, while the low-rate variance stays flat up to $w \approx c_L$ and then decays with a larger amplitude ($a_L > a_H$). Over the fitted range, the high-rate branch's bias advantage shrinks and turns into a small disadvantage above $w \approx 600$. A crossover needs $\Delta V = -\Delta B$, so it can only occur where the two gaps have opposite signs. Between widths 64 and 128 the excess variance of Dip 2 falls below its bias advantage and Dip 2 takes over, *before* the variances themselves cross.

Why transfer fails. A small proxy ranks the two dips largely by initialization variance, which μP drives down with width, whereas at larger widths the mean predictor counts more. The optimal rate jumps whenever the two rankings disagree; no label noise is needed. The jump holds over a width window rather than asymptotically. Dip 2's advantage peaks around $w \approx 200$ and shrinks toward $w = 2048$. Extrapolated, the fitted law predicts a second reversal near $w \approx 3 \cdot 10^3$. We show the extended fits in Appendix B.8).

6 Discussion and Limitations

In the feature-learning regime, the learning rate does more than set how far training travels along the gradient-flow path: GD at rate η can only converge to interpolants with $\lambda_{\max}(\boldsymbol{K}/n) \leq 2/\eta$. Imposing this ceiling at stable step sizes retains the second dip, so the instability through which native GD reaches it is not needed. In SiLU teacher–student networks the ceiling acts as a ridge that moves label noise into the tail of the kernel, away from test inputs. In practice, learning-rate sweeps spanning two or three decades can miss the better minimum, and a small proxy can rank the two minima differently from the target model, even under μP.

Limitations. Our models are small: teacher–student MLPs, ResNet-18 on CIFAR-10, and a transformer that does not interpolate at our budget. They are trained without schedules or weight decay, and tuned weight decay removes the double dip in our MLPs. An interpolating second dip needs $n < D$, so the effect does not concern compute-optimal LLM pretraining. It appears with GD, SGD and Adam, but not with Muon. The theory is quantitative only for noisy SiLU networks (and for GELU up to the onset of its right flank); it does not quantify the initialization terms, and does not explain the mean-predictor gain on CIFAR-10. Finally, the μP failure model is fitted on one clean ReLU task and describes a width window (64–2048); wider runs suggest a second reversal (App. B.8), so how the optimal rate behaves at larger widths needs further study. Our central aim here is to expose the failure mode.

AI USE STATEMENT

We used large language models to assist with code development, obtain feedback on possible improvements to the paper, check mathematical proofs, and improve the clarity and readability of the manuscript. These uses supported the research workflow and manuscript preparation; LLMs are not components of the proposed method or the models evaluated in our experiments. The authors take full responsibility for the final content of this work, including its theoretical claims, proofs, experimental results, and code.

ETHICS STATEMENT

This work investigates LR double descent in neural networks through theoretical analysis and experiments. It does not involve human participants, crowdsourcing, or personal data, and does not introduce or release high-risk models or scraped datasets. Given the foundational nature of the research, we do not identify direct, foreseeable negative societal impacts specific to the theoretical results and synthetic experiments presented here.

REPRODUCIBILITY STATEMENT

We provide detailed algorithmic and experimental specifications to support independent reproduction of our results. Appendix B contains the theoretical derivations, assumptions, and proofs. Appendix C.5 specifies the capped gradient descent update rule, while the remainder of Appendix C comprehensively describes the network architectures, data configurations, and hyperparameters for the main experiments. Appendix D reports on further experiments, detailing controls for weight decay, alternative optimizers such as Adam and Muon, and edge-of-stability diagnostics. The implementation primarily uses standard neural-network training operations, with the method-specific updates explicitly documented in the appendices. These descriptions are intended to enable independent implementation before the code release which we will do for the camera-ready version.

REFERENCES

Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition. *ArXiv*, abs/2011.03321, 2020a. URL https://api.semanticscholar.org/CorpusID:226278106.

Ben Adlam and Jeffrey Pennington. Understanding double descent requires a fine-grained bias-variance decomposition, 2020b. URL https://arxiv.org/abs/2011.03321.

Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428 – 446, 2017. URL https://api.semanticscholar.org/CorpusID:38293418.

Kwangjun Ahn, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via the "edge of stability". *ArXiv*, abs/2212.07469, 2022. URL https://api.semanticscholar.org/CorpusID:254685942.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on edge of stability in deep learning. *ArXiv*, abs/2205.09745, 2022. URL https://api.semanticscholar.org/CorpusID:248887658.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. *ArXiv*, abs/2111.00034, 2021. URL https://api.semanticscholar.org/CorpusID:240354190.

Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *ArXiv*, abs/2205.01445, 2022. URL https://api.semanticscholar.org/CorpusID:248505757.

David GT Barrett and Benoit Dherin. Implicit gradient regularization. *arXiv preprint arXiv:2009.11162*, 2020.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117:30063 – 30070, 2019. URL https://api.semanticscholar.org/CorpusID:195700154.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. On the benefits of large learning rates for kernel methods. In *Annual Conference Computational Learning Theory*, 2022. URL https://api.semanticscholar.org/CorpusID:247158390.

Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks, 2023. URL https://arxiv.org/abs/2304.03408.

Lei Chen and Joan Bruna. Beyond the edge of stability via two-step gradient updates. 2022. URL https://api.semanticscholar.org/CorpusID:260165112.

Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:189928159.

Jeremy Cohen, Alex Damian, Ameet Talwalkar, Zico Kolter, and Jason D. Lee. Understanding optimization in deep learning with central flows. *ArXiv*, abs/2410.24206, 2024. URL https://api.semanticscholar.org/CorpusID:273707747.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *arXiv preprint arXiv:2209.15594*, 2022.

Stéphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent : Bias and variance(s) in the lazy regime. *ArXiv*, abs/2003.01054, 2020. URL https://api.semanticscholar.org/CorpusID:211678117.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:7636159.

Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification, 2015. URL https://arxiv.org/abs/1507.03003.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, et al. Scaling exponents across parameterizations and optimizers. *arXiv preprint arXiv:2407.05872*, 2024.

Niclas Alexander Göring, Charles London, Abdurrahman Hadi Erturk, Chris Mingard, Yoonsoo Nam, and Ard A Louis. Feature learning is decoupled from generalization in high capacity neural networks. *arXiv preprint arXiv:2507.19680*, 2025.

Ouns El Harzli, Bernardo Cuenca Grau, Guillermo Valle Pérez, and Ard A. Louis. Double-descent curves in neural networks: a new perspective using gaussian processes. *ArXiv*, abs/2102.07238, 2021. URL https://api.semanticscholar.org/CorpusID:231924904.

Trevor J. Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50:949 – 986, 2019. URL https://api.semanticscholar.org/CorpusID:84186305.

Soufiane Hayou. A proof of learning rate transfer under p. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3322–3330. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/hayou26a.html.

R Heckel and FF Yilmaz. Early stopping in deep networks: Double descent and how to eliminate it. arxiv 2020. *arXiv preprint arXiv:2007.10099*.

David Holzmüller. On the universality of the double descent peak in ridgeless regression. *ArXiv*, abs/2010.01851, 2020. URL https://api.semanticscholar.org/CorpusID:222133956.

Hong Hu and Yue M. Lu. Sharp asymptotics of kernel ridge regression beyond the linear regime, 2022. URL https://arxiv.org/abs/2205.06798.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *ArXiv*, abs/1806.07572, 2018. URL https://api.semanticscholar.org/CorpusID:49321232.

Kaiqi Jiang, Jeremy Cohen, and Yuanzhi Li. Understanding the evolution of the neural tangent kernel at the edge of stability. *ArXiv*, abs/2507.12837, 2025. URL https://api.semanticscholar.org/CorpusID:280278251.

Dayal Singh Kalra and Maissam Barkeshli. Learning rates do not transfer across double descent. In *High-dimensional Learning Dynamics 2026*.

Dayal Singh Kalra, Tianyu He, and Maissam Barkeshli. Universal sharpness dynamics in neural network training: Fixed point analysis, edge of stability, and route to chaos. *ArXiv*, abs/2311.02076, 2023. URL https://api.semanticscholar.org/CorpusID:265019099.

Atli Kosson, Jeremy Welborn, Yang Liu, Martin Jaggi, and Xi Chen. Weight decay may matter more than μp for learning rate transfer in practice. In *International Conference on Learning Representations*, volume 2026, pp. 67012–67048, 2026.

Jaerin Lee and Kyoung Mu Lee. The road taken: The role of optimizers at the edge of stability. 2026. URL https://api.semanticscholar.org/CorpusID:291217879.

Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. First-order methods almost always avoid saddle points, 2017. URL https://arxiv.org/abs/1710.07406.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Predictable scale: Part i, step law – optimal hyperparameter scaling law in large language model pretraining, 2025. URL https://arxiv.org/abs/2503.04715.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:195874215.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 48(3), 2020. ISSN 0090-5364. doi: 10.1214/19-aos1849. URL http://dx.doi.org/10.1214/19-AOS1849.

Tongtong Liang, Alexander Cloninger, Rahul Parhi, and Yu-Xiang Wang. Generalization below the edge of stability: The role of data geometry. *ArXiv*, abs/2510.18120, 2025. URL https://api.semanticscholar.org/CorpusID:282246553.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75, 2019. URL https://api.semanticscholar.org/CorpusID:199668852.

Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A Louis. Is sgd a bayesian sampler? well, almost. *Journal of Machine Learning Research*, 22(79):1–64, 2021.

Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A Louis. Deep neural networks have an inbuilt occam's razor. *Nature Communications*, 16(1):220, 2025.

Theodor Misiakiewicz. Spectrum of inner-product kernel matrices in the polynomial regime and multiple descent phenomenon in kernel ridge regression, 2022. URL https://arxiv.org/abs/2204.10425.

Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi. Completed hyperparameter transfer across modules, width, depth, batch and duration, 2025. URL https://arxiv.org/abs/2512.22382.

Andrea Montanari and Pierfrancesco Urbani. Dynamical decoupling of generalization and overfitting in large two-layer networks. *Advances in Neural Information Processing Systems*, 38: 26530–26608, 2026.

Vidya K. Muthukumar, Kailas Vodrahalli, and Anant Sahai. Harmless interpolation of noisy data in regression. *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2299–2303, 2019. URL https://api.semanticscholar.org/CorpusID:84843966.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto. Super consistency of neural network landscapes and learning rate transfer. *Advances in Neural Information Processing Systems*, 37:102696–102743, 2024.

Ioannis Panageas and Georgios Piliouras. Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions. In Christos H. Papadimitriou (ed.), *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 2:1–2:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.2. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2017.2.

Guillermo Valle Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *ArXiv*, abs/1805.08522, 2018. URL https://api.semanticscholar.org/CorpusID:43939886.

Ildus Sadrtdinov, Maxim Kodryan, Eduard Pokonechny, Ekaterina Lobacheva, and Dmitry P. Vetrov. Where do large learning rates lead us? *ArXiv*, abs/2410.22113, 2024. URL https://api.semanticscholar.org/CorpusID:273662145.

James B. Simon, Madeline Dickens, Dhruva Karkada, and Michael R. DeWeese. The eigenlearning framework: A conservation law perspective on kernel regression and wide neural networks, 2023. URL https://arxiv.org/abs/2110.03922.

Dayal Singh Kalra and Maissam Barkeshli. Quantifying hyperparameter transfer and the importance of embedding layer learning rate. *arXiv e-prints*, pp. arXiv–2605, 2026.

Mario Tuci, Caner Korkmaz, Umut cSimcsekli, and Tolga Birdal. Generalization at the edge of stability. *ArXiv*, abs/2604.19740, 2026. URL https://api.semanticscholar.org/CorpusID:287636709.

Lawrence Wang and Stephen J. Roberts. Can stability be detrimental? better generalization through gradient descent instabilities. *ArXiv*, abs/2412.17613, 2024. URL https://api.semanticscholar.org/CorpusID:274982153.

Yuqing Wang, Minshuo Chen, Tuo Zhao, and Molei Tao. Large learning rate tames homogeneity: Convergence and balancing effect. *ArXiv*, abs/2110.03677, 2021. URL https://api.semanticscholar.org/CorpusID:238418995.

Yuqing Wang, Zhenghao Xu, Tuo Zhao, and Molei Tao. Good regularity creates large learning rate implicit biases: edge of stability, balancing, and catapult. *J. Mach. Learn. Res.*, 26:273:1–273:68, 2023. URL https://api.semanticscholar.org/CorpusID:264491141.

Blake E. Woodworth, Suriya Gunasekar, J. Lee, Edward Moroshko, Pedro H. P. Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. *ArXiv*, abs/2002.09277, 2019. URL https://api.semanticscholar.org/CorpusID:211252492.

Jingfeng Wu, Vladimir Braverman, and Jason D. Lee. Implicit bias of gradient descent for logistic regression at the edge of stability. *ArXiv*, abs/2305.11788, 2023. URL https://api.semanticscholar.org/CorpusID:258822914.

Lei Wu, Chao Ma, and E. Weinan. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:55701518.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:235825390.

Greg Yang, James B Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. *ArXiv*, abs/2002.11328, 2020. URL https://api.semanticscholar.org/CorpusID:211505957.

Fatih Furkan Yilmaz and Reinhard Heckel. Regularization-wise double descent: Why it occurs and how to eliminate it. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 426–431. IEEE, 2022.

Shuofeng Zhang, Isaac Reid, Guillermo Valle Pérez, and Ard Louis. Why flatness does and does not correlate with generalization for deep neural networks. *arXiv preprint arXiv:2103.06219*, 2021.

Libin Zhu, Chaoyue Liu, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Catapults in sgd: spikes in the training loss and their impact on generalization through feature learning. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:259108432.

A Appendix

Appendix overview

Section B states and proves the theoretical results of the main text: the necessary conditions (Theorem B.2), the first- versus second-order effect of steps and constraints on the gradient-flow limit, the selection law along the ceiling that gives the sign and slope of the onset, Lemma B.8, the proof of Theorem 4.1, and the motivation of the width laws of section 5. Section C gives the architectures, parametrizations, data and protocols of every experiment in the main text. Section D contains further experiments: why common practice hides the double dip (including weight decay), other optimizers, edge-of-stability diagnostics, kernel degree masses and data size. Section E is an extended related-work section.

B THEORY AND PROOFS

B.1 SETTING AND NOTATION

We write $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{y} \in \mathbb{R}^n$ for the training inputs and labels, $f_{\boldsymbol{\theta}}(\boldsymbol{X}) \in \mathbb{R}^n$ for the training outputs, and $\boldsymbol{\theta} \in \mathbb{R}^P$ for all trainable parameters. The training loss is the half squared error of equation 1, $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\|f_{\boldsymbol{\theta}}(\boldsymbol{X}) - \boldsymbol{y}\|^2$, GD is equation 2, and gradient flow (GF) is $\dot{\boldsymbol{\theta}} = -\nabla\mathcal{L}(\boldsymbol{\theta})$ with flow map Φ_τ. The *GF limit map* is $\Pi(\boldsymbol{\theta}) = \lim_{\tau\to\infty}\Phi_\tau\boldsymbol{\theta}$, so $\boldsymbol{\theta}_F = \Pi(\boldsymbol{\theta}_0)$, $\Lambda_F = \lambda_{\max}(\nabla^2\mathcal{L}(\boldsymbol{\theta}_F))$ and $\eta_c = 2/\Lambda_F$. The training Jacobian is $\boldsymbol{J} = \partial f_{\boldsymbol{\theta}}(\boldsymbol{X})/\partial\boldsymbol{\theta} \in \mathbb{R}^{n\times P}$, the empirical NTK Gram matrix is $\boldsymbol{K} = \boldsymbol{J}\boldsymbol{J}^\top$, and $\mathcal{M} = \{\boldsymbol{\theta} : f_{\boldsymbol{\theta}}(\boldsymbol{X}) = \boldsymbol{y}\}$ is the set of interpolants. Where $\boldsymbol{J}$ has full row rank, $\mathcal{M}$ is a $(P-n)$-dimensional manifold with tangent space $\ker\boldsymbol{J}$, the orthogonal projector onto it is $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{J}^\top\boldsymbol{K}^{-1}\boldsymbol{J}$, and on $\mathcal{M}$ the loss Hessian is $\nabla^2\mathcal{L} = \boldsymbol{J}^\top\boldsymbol{J}/n$, whose nonzero spectrum is that of $\boldsymbol{K}/n$. We write $\lambda(\boldsymbol{\theta}) = \lambda_{\max}(\nabla^2\mathcal{L}(\boldsymbol{\theta}))$ for the sharpness; on $\mathcal{M}$, $\lambda = \lambda_{\max}(\boldsymbol{K}/n)$.

Assumption B.1 (Smooth regime)**.** On an open set U containing the GD iterates and the segments between consecutive iterates, $\mathcal{L} \in C^3$, $\nabla\mathcal{L}$ is Lipschitz, and $\Pi \in C^2$ with $\|D^2\Pi\| \le \kappa$.

Near a part of $\mathcal{M}$ where $\boldsymbol{J}$ has full row rank, $\mathcal{M}$ is a normally attracting manifold of equilibria of GF and Π is as smooth as the flow (stable-foliation theory). For ReLU networks $\mathcal{L}$ is piecewise polynomial in $\boldsymbol{\theta}$, so Assumption B.1 holds inside an activation region and fails at gate crossings. This is the reason why smooth activations follow the quantitative predictions below and ReLU follows only their signs (section B.5).

B.2 NECESSARY CONDITIONS (THEOREM B.2)

Theorem B.2 (Necessary conditions)**.** *Let $\eta > 0$.*

(i) Endpoint selection by stability. *(a) Under Assumption B.1, if every GD step satisfies $\eta\,\lambda_{\max}(\nabla^2\mathcal{L}) \le 2-\delta$ on the segment $[\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1}]$ for some $\delta \in (0,2]$, then $\|\Pi(\boldsymbol{\theta}_T) - \boldsymbol{\theta}_F\| \le \kappa\,\eta\,\mathcal{L}(\boldsymbol{\theta}_0)/\delta$ for every T. If moreover $\boldsymbol{\theta}_t \to \boldsymbol{\theta}_\infty \in \mathcal{M}$, then $\|\boldsymbol{\theta}_\infty - \boldsymbol{\theta}_F\| \le \kappa\,\eta\,\mathcal{L}(\boldsymbol{\theta}_0)/\delta$. (b) Let the GD map $G(\boldsymbol{\theta}) = \boldsymbol{\theta} - \eta\nabla\mathcal{L}(\boldsymbol{\theta})$ be C^1 with $\det DG(\boldsymbol{\theta}) \neq 0$ for Lebesgue-almost every $\boldsymbol{\theta}$, and let $E \subseteq \mathcal{M}$ be a set of interpolants $\boldsymbol{\theta}_*$ with $\eta\lambda(\boldsymbol{\theta}_*) > 2$ at each of which $DG(\boldsymbol{\theta}_*) = \boldsymbol{I} - \eta\nabla^2\mathcal{L}(\boldsymbol{\theta}_*)$ is invertible (no eigenvalue of $\nabla^2\mathcal{L}(\boldsymbol{\theta}_*)$ equals $1/\eta$). Then the set of initializations from which GD converges to a point of E is Lebesgue-null. In particular, if $\eta > \eta_c$ and $DG(\boldsymbol{\theta}_F)$ is invertible, GD converges to $\boldsymbol{\theta}_F$ only from a null set of initializations.*

(ii) The kernel must move. *(a) If $f_{\boldsymbol{\theta}}$ is affine in $\boldsymbol{\theta}$ (a fixed kernel) and $\boldsymbol{K}$ is invertible, then GD converges to $\boldsymbol{\theta}_F$ for every $0 < \eta < \eta_c$, and for $\eta > \eta_c$ it diverges unless the initial residual is orthogonal to the top eigenvector of $\boldsymbol{K}$. The converged predictor therefore does not depend on the rate. (b) Let $\eta > \eta_c$. If GD converges to an interpolant $\boldsymbol{\theta}_\infty$ that is linearly stable, $\eta\lambda(\boldsymbol{\theta}_\infty) \le 2$ (by (i)(b) this holds for almost every initialization whenever the unstable interpolants satisfy its hypotheses), then*

$$\|\boldsymbol{K}(\boldsymbol{\theta}_\infty) - \boldsymbol{K}(\boldsymbol{\theta}_F)\|_{\mathrm{op}} \;\ge\; (1-\eta_c/\eta)\,\lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_F)).$$

(iii) Overparameterization. *Let $A_{\boldsymbol{\theta}} : \boldsymbol{v} \mapsto \langle\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\cdot), \boldsymbol{v}\rangle$ be the population tangent map and $D(\boldsymbol{\theta}) = \operatorname{rank} A_{\boldsymbol{\theta}}$. (a) If $\operatorname{rank}\boldsymbol{J}(\boldsymbol{\theta}) = n < D(\boldsymbol{\theta})$, then $\dim A_{\boldsymbol{\theta}}(\ker\boldsymbol{J}(\boldsymbol{\theta})) = D(\boldsymbol{\theta}) - n$: there are $D(\boldsymbol{\theta}) - n$ independent directions along $\mathcal{M}$ that change the population function. (b) Let F be a set on which $f_{\boldsymbol{\theta}}(\boldsymbol{X})$ is constant and in which any two points are joined by a piecewise C^1 path that stays in F. If f is continuously differentiable in $\boldsymbol{\theta}$ on a neighbourhood of F and $\operatorname{rank}\boldsymbol{J}(\boldsymbol{\theta}) = D(\boldsymbol{\theta})$ on F, then $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the same for all $\boldsymbol{\theta} \in F$ and all $\boldsymbol{x}$.*

Proof. Flow invariance. Since $\Pi(\Phi_\tau\boldsymbol{\theta}) = \Pi(\boldsymbol{\theta})$ for all $\tau \ge 0$, differentiating at $\tau = 0$ gives

$$D\Pi(\boldsymbol{\theta})\,\nabla\mathcal{L}(\boldsymbol{\theta}) = 0. \tag{7}$$

(i)(a). Write $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$. Taylor's formula with integral remainder along $s \mapsto \Pi(\boldsymbol{\theta}_t - s\eta\boldsymbol{g}_t)$ gives $\Pi(\boldsymbol{\theta}_{t+1}) = \Pi(\boldsymbol{\theta}_t) - \eta D\Pi(\boldsymbol{\theta}_t)\boldsymbol{g}_t + \eta^2\int_0^1(1-s)D^2\Pi(\boldsymbol{\theta}_t - s\eta\boldsymbol{g}_t)[\boldsymbol{g}_t,\boldsymbol{g}_t]\,ds$. The linear term vanishes

by equation 7. Summing over t,

$$\Pi(\boldsymbol{\theta}_T)-\boldsymbol{\theta}_F = \eta^2 \sum_{t<T} \int_0^1 (1-s)\, D^2\Pi(\boldsymbol{\theta}_t - s\eta \boldsymbol{g}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t]\, ds, \qquad \text{so} \qquad \|\Pi(\boldsymbol{\theta}_T)-\boldsymbol{\theta}_F\| \le \tfrac{\kappa}{2}\, \eta^2 \sum_{t<T} \|\boldsymbol{g}_t\|^2. \tag{8}$$

The same expansion of $\mathcal{L}$ along the segment gives $\mathcal{L}(\boldsymbol{\theta}_{t+1}) \le \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\boldsymbol{g}_t\|^2 + \frac{\eta^2}{2} \sup_s \lambda_{\max}\|\boldsymbol{g}_t\|^2 \le \mathcal{L}(\boldsymbol{\theta}_t) - \frac{\eta\delta}{2}\|\boldsymbol{g}_t\|^2$. Summing and using $\mathcal{L} \ge 0$, $\eta^2 \sum_t \|\boldsymbol{g}_t\|^2 \le 2\eta\, \mathcal{L}(\boldsymbol{\theta}_0)/\delta$, which gives the first bound. If $\boldsymbol{\theta}_t^{\cdot} \to \boldsymbol{\theta}_\infty \in \mathcal{M}$, then $\Pi(\boldsymbol{\theta}_\infty) = \boldsymbol{\theta}_\infty$, and continuity of Π gives the second.

(i)(b). G fixes every $\boldsymbol{\theta}_* \in \mathcal{M}$, and $DG(\boldsymbol{\theta}_*)$ has the eigenvalue $1 - \eta\lambda(\boldsymbol{\theta}_*) < -1$. Since $DG(\boldsymbol{\theta}_*)$ is invertible, G is a local diffeomorphism near $\boldsymbol{\theta}_*$, and the centre-stable manifold theorem for local diffeomorphisms gives a neighbourhood V of $\boldsymbol{\theta}_*$ in which the points whose forward orbit stays in V lie on an embedded disc of dimension $< P$, a Lebesgue-null set W_V. Countably many such V cover E (second countability). If GD converges to a point of E, its orbit eventually stays in one of these V, so the initialization lies in $\bigcup_V \bigcup_{m\ge 0} G^{-m}(W_V)$. Because G is C^1 with $\det DG \ne 0$ almost everywhere, the preimage of a null set under G is null (on the open set where $\det DG \ne 0$, G is a local diffeomorphism; the remaining set is null), so the countable union is null. This is the argument of Lee et al. (2017) and Panageas & Piliouras (2017) for strict saddles, with the invertibility assumed only near the endpoints of interest. For ReLU networks $\mathcal{L}$ is smooth only inside activation regions, and the statement applies to endpoints in the interior of a region.

(ii)(a). For $f_{\boldsymbol{\theta}} = f_{\boldsymbol{\theta}_0} + \boldsymbol{J}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$ with $\boldsymbol{J}$ fixed, the residual $\boldsymbol{r}_t = f_{\boldsymbol{\theta}_t}(\boldsymbol{X}) - \boldsymbol{y}$ obeys $\boldsymbol{r}_{t+1} = (\boldsymbol{I} - \eta\boldsymbol{K}/n)\boldsymbol{r}_t$, and $\boldsymbol{\theta}_t - \boldsymbol{\theta}_0$ stays in the row space of $\boldsymbol{J}$. If $0 < \eta\lambda_{\max}(\boldsymbol{K}/n) < 2$, every factor $|1 - \eta\lambda_i(\boldsymbol{K}/n)|$ is below one (all $\lambda_i > 0$ because $\boldsymbol{K}$ is invertible) and the iterates converge to $\boldsymbol{\theta}_0 - \boldsymbol{J}^\top \boldsymbol{K}^{-1}\boldsymbol{r}_0$, which is also the GF limit. If $\eta\lambda_{\max}(\boldsymbol{K}/n) > 2$, the component of the residual along the top eigenvector grows by the factor $|1 - \eta\lambda_{\max}(\boldsymbol{K}/n)| > 1$ per step.

(ii)(b). On $\mathcal{M}$, $\nabla^2\mathcal{L} = \boldsymbol{J}^\top\boldsymbol{J}/n$, so $\lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_\infty)/n) \le 2/\eta$, while $\lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_F)) = n\Lambda_F$. Weyl's inequality $\lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_F)) \le \lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_\infty)) + \|\boldsymbol{K}(\boldsymbol{\theta}_F) - \boldsymbol{K}(\boldsymbol{\theta}_\infty)\|_{\mathrm{op}}$ gives $\|\boldsymbol{K}(\boldsymbol{\theta}_\infty) - \boldsymbol{K}(\boldsymbol{\theta}_F)\|_{\mathrm{op}} \ge n(\Lambda_F - 2/\eta) = (1 - \eta_c/\eta)\, \lambda_{\max}(\boldsymbol{K}(\boldsymbol{\theta}_F))$.

(iii)(a). Every row of $\boldsymbol{J}$ is a value $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ of the population tangent map, so $\ker A_{\boldsymbol{\theta}} \subseteq \ker \boldsymbol{J}$. Hence $\dim A_{\boldsymbol{\theta}}(\ker \boldsymbol{J}) = \dim\ker \boldsymbol{J} - \dim\ker A_{\boldsymbol{\theta}} = (P - n) - (P - D(\boldsymbol{\theta})) = D(\boldsymbol{\theta}) - n$.

(iii)(b). The row space of $\boldsymbol{J}(\boldsymbol{\theta})$ lies in the range of $A^*_{\boldsymbol{\theta}}$, which has dimension $D(\boldsymbol{\theta})$. If $\operatorname{rank}\boldsymbol{J} = D(\boldsymbol{\theta})$ the two coincide, so $\ker \boldsymbol{J} = \ker A_{\boldsymbol{\theta}}$. Take two points of F and a piecewise C^1 path $\boldsymbol{\theta}(\tau)$ in F joining them. On each C^1 piece, constancy of $f_{\boldsymbol{\theta}}(\boldsymbol{X})$ gives $\boldsymbol{J}\dot{\boldsymbol{\theta}} = 0$, hence $A_{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}} = 0$, i.e. $\frac{d}{d\tau} f_{\boldsymbol{\theta}(\tau)}(\boldsymbol{x}) = 0$ for every $\boldsymbol{x}$; by continuity $f_{\boldsymbol{\theta}(\tau)}(\boldsymbol{x})$ is constant along the whole path. □

Remarks. (i)(a) bounds a displacement in parameter space, not a risk difference, and degrades as $\delta \to 0$. It states that the *only* way GD can end at a different interpolant than GF is the accumulated second-order dose $\eta^2 \sum_t \|\boldsymbol{g}_t\|^2$ in equation 8; this dose need not be small when δ is small. (i)(b) excludes an *endpoint*; it does not prescribe the curvature history, and in particular it does not imply that GD above η_c must take steps with $\eta\lambda_{\max} \ge 2$. For example, for $\mathcal{L}(x, y) = \frac{1}{2}(xy - 1)^2$ from $(x_0, y_0) = (1, 0.1)$ with $\eta = 0.9$, the GF endpoint $(1.2692, 0.7879)$ has $\eta\Lambda_F = 2.008 > 2$, yet GD converges to $(1.0883, 0.9188)$ while every update segment has $\eta\lambda_{\max}(\nabla^2\mathcal{L}) \le 1.861$. Crossing the nominal boundary, sustained edge-of-stability oscillation and catapult spikes of the loss are therefore empirical properties of our native runs (section D.3), not consequences of (i). (ii)(a) concerns converged predictors at a constant rate; early-stopped kernel methods can depend on the rate (Beugnot et al., 2022). (iii) is local: (a) assumes full row rank, and (b) compares only interpolants joined by regular paths inside a constant-output set. It says nothing about interpolants in different components, which can compute different functions even when $n \ge D$ (e.g. $f_\theta(x) = \theta^2 + x\theta$ with the single training pair $(0, 1)$ has the interpolants $\theta = \pm 1$, both of full rank $1 = D$, with different predictions off the training input). For bias-free ReLU networks, rescaling the incoming weights of a hidden unit by $c > 0$ and its outgoing weights by $1/c$ leaves f unchanged, so $D \le P - \sum_l w_l$; for two hidden layers of width w and input dimension d this is $D \le w(w + d - 1)$. Smooth activations such as SiLU and GELU have no continuous symmetry of this kind.

No interpolation above D. The main text also uses that for $n > D$ generic noisy labels cannot be interpolated.

Proposition B.3. *Let $\boldsymbol{\theta} \mapsto f_{\boldsymbol{\theta}}(\boldsymbol{X})$ be C^∞, or continuous and polynomial on each of finitely many open cells whose closures cover $\mathbb{R}^P$ (ReLU networks with fixed inputs are of the second kind; there $D(\boldsymbol{\theta})$ and $\operatorname{rank} \boldsymbol{J}$ are defined inside the cells). Let $D := \sup_{\boldsymbol{\theta}} \operatorname{rank} \boldsymbol{J}(\boldsymbol{\theta})$ over the points where $f_{\boldsymbol{\theta}}(\boldsymbol{X})$ is differentiable (so $D \le \sup_{\boldsymbol{\theta}} D(\boldsymbol{\theta})$). If $n > D$ and $\boldsymbol{y}$ has a density given $\boldsymbol{X}$ (e.g. Gaussian label noise), then* $\Pr\left(\exists\, \boldsymbol{\theta} :\ f_{\boldsymbol{\theta}}(\boldsymbol{X}) = \boldsymbol{y}\right) = 0$.

Proof. Write $\mathcal{F}(\boldsymbol{\theta}) = f_{\boldsymbol{\theta}}(\boldsymbol{X}) \in \mathbb{R}^n$; its rank is at most $D < n$ wherever it is differentiable. In the C^∞ case every $\boldsymbol{\theta}$ is a critical point, and Sard's theorem makes $\mathcal{F}(\mathbb{R}^P)$ null. In the piecewise-polynomial case, on each open cell C_k, $\mathcal{F} = \mathcal{F}_k$ for a polynomial map $\mathcal{F}_k$; all $(D+1)$-minors of $D\mathcal{F}_k$ vanish on the open set C_k, hence identically, so $\mathcal{F}_k(\mathbb{R}^P)$ is null by Sard. Every $\boldsymbol{\theta}$ lies in the closure of some cell, where $\mathcal{F}(\boldsymbol{\theta}) = \mathcal{F}_k(\boldsymbol{\theta})$ by continuity, so $\mathcal{F}(\mathbb{R}^P) \subseteq \bigcup_k \mathcal{F}_k(\mathbb{R}^P)$ is null. Since $\boldsymbol{y}$ has a density, $\Pr(\boldsymbol{y} \in \mathcal{F}(\mathbb{R}^P)) = 0$. □

For $n > D$ the ceiling can therefore only trade training fit for a sharpness-constrained fit, and for $n \ge D$ interpolants joined by full-rank paths compute one function, so along such paths it can only rebalance parameters. How deep a resulting non-interpolating valley can be is not a consequence of Proposition B.3. A heuristic bound: if the GF fit is a regular, identifiable and consistent least-squares estimator, classical M-estimation asymptotics give an excess risk $\approx \operatorname{tr}(\mathcal{A}^{-1}\mathcal{B})/n$ ($\approx \sigma^2 D/n$ in the well-specified case), which bounds the gain of any valley and vanishes as $n/D \to \infty$. We use this only as a guide for fig. 8; a generic nonlinear GF fit need not satisfy these conditions.

Measurements. *Kernel movement (ii)(b).* On 1,189 converged branch endpoints (15 settings, 4 seeds, all rates above η_c) the measured $\|\boldsymbol{K}_\infty - \boldsymbol{K}_F\|_{\mathrm{op}}/\lambda_{\max}(\boldsymbol{K}_F)$ exceeds the bound $1 - \eta_c/\eta$ in every case and lies on it to within a few percent, and $\eta\lambda_{\max}(\boldsymbol{K}_\infty)/n = 2$ at the end of every branch run (fig. 7c,d): the kernel moves by what the ceiling forces and not more. At the canonical ReLU Dip 2, which here lies at the end of the trainable branch ($\eta_2 = \eta_{\mathrm{edge}} = 9.86 \approx 15.9\,\eta_c$ with $\eta_c = 0.62$), the kernel has moved by 94% of $\lambda_{\max}(\boldsymbol{K}_F)$, against a bound of $1 - 1/15.9 = 0.937$. *Branch length.* Without a second hidden layer or a bias, the high-rate branch ends at $\eta_{\mathrm{edge}} \approx 1.7\,\eta_c$ (ReLU, SiLU and GELU at widths 256 and 1024) and the Dip-2 gain is 0–10%; a first-layer bias restores branches of 5–12 η_c (gains 1–30%); depth two gives 12–21 η_c and depth three 28 η_c (gains 30–54%) (fig. 7a,b). *Data (iii).* For bias-free depth-2 ReLU nets the Jacobian rank at initialization equals $w(w + d - 1)$ exactly at all 11 tested widths from 6 to 64 (singular-value gap 10^{11}–10^{13} at the cut). In a sweep over $n/D \in [0.125, 8]$ (fig. 8; full-batch GD, $\sigma = 0.5$, $T = 4 \cdot 10^4$, 4–8 seeds) an *interpolating* second dip occurs only for $n/D \le 0.5$; for $1 < n/D \le 1.8$ only non-interpolating valleys remain, whose bottoms are 3.5–6.6 times worse than Dip 1 and near the zero predictor; no cell with $n \ge 2.5D$ has a double dip. The cut-off $2.5D$ is empirical; the heuristic above only suggests an $O(D/n)$ decay. $n < D$ is not sufficient: width 128 has a single dip at $n/D = 0.75$.

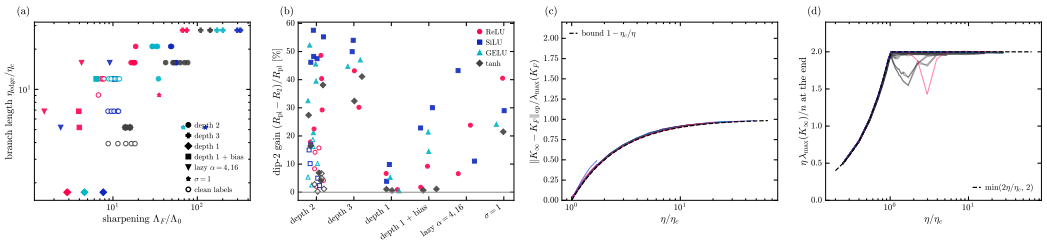

Figure 7: **The necessary conditions across 72 settings** (ReLU, SiLU, GELU, tanh; depths 1–3 with and without a first-layer bias; widths 64–1024; $\sigma \in \{0, 0.5, 1\}$; lazy scaling $\alpha \in \{4, 16\}$; 8 seeds each). (a) Branch length $\eta_{\mathrm{edge}}/\eta_c$ against the sharpening Λ_F/Λ_0 of the GF interpolant. (b) Dip-2 gain $(\mathcal{R}_{\mathrm{pl}} - \mathcal{R}_2)/\mathcal{R}_{\mathrm{pl}}$ by architecture. (c) Kernel movement $\|\boldsymbol{K}_\infty - \boldsymbol{K}_F\|_{\mathrm{op}}/\lambda_{\max}(\boldsymbol{K}_F)$ against η/η_c with the bound $1 - \eta_c/\eta$ of Theorem B.2(ii)(b) (dashed). (d) $\eta\lambda_{\max}(\boldsymbol{K}_\infty)/n$ at the end of training against η/η_c; dashed: $\min(2\eta/\eta_c, 2)$. Open markers: clean labels.

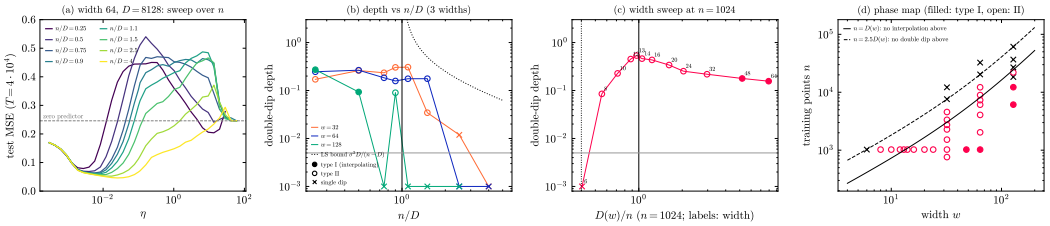

Figure 8: **How much data removes the double dip** (depth-2 bias-free ReLU, $d = 64$, $\sigma = 0.5$, full-batch GD, $T = 4 \cdot 10^4$). (a) Width 64: test risk against η for n/D from 0.25 to 4. (b) Depth of the double dip against n/D at widths 32, 64, 128; filled: the second minimum interpolates (training MSE $< 10^{-3}$); open: it does not; $\times$: single dip; dotted: least-squares bound $\sigma^2 D/(n-D)$. (c) Width sweep at $n = 1024$: the depth peaks at the interpolation threshold $D(w) = n$ ($w = 13$) and vanishes at $D/n = 0.4$ ($w = 6$). (d) All cells in the (w, n) plane with the curves $n = D(w)$ and $n = 2.5D(w)$. A double dip is counted when two local minima are separated by a maximum, each with prominence above $\max(3\,\mathrm{SE}, 0.005)$.

B.3 Below the edge, GD tracks gradient flow at time ηT

Section 2 states that Dip 1 depends only on ηT. We state the precise consequence of a uniform comparison between GD and GF, and keep what we only observe separate.

Assumption B.4 (Uniform flow-time comparison)**.** There are a set Θ_0 that carries the whole initialization law and constants $\bar{\eta}, C > 0$ such that for every $\boldsymbol{\theta}_0 \in \Theta_0$, every $0 < \eta \le \bar{\eta}$ and every $T \ge 0$, $\|\boldsymbol{\theta}_T - \Phi_{\eta T}\boldsymbol{\theta}_0\| \le C\eta$.

When it holds (sketch, not proved here). On a finite horizon $[0, \tau_0]$ Euler's global error is $O(\eta)$. If, uniformly on Θ_0, both trajectories then stay in a tube around a part of $\mathcal{M}$ where $\boldsymbol{J}$ has full row rank, $\mathcal{L}$ is Morse–Bott and the normal attraction rate is bounded below, one decomposes $\boldsymbol{\theta} = \Pi(\boldsymbol{\theta}) + (\boldsymbol{\theta} - \Pi(\boldsymbol{\theta}))$: the tangential part moves by $O(\eta)$ uniformly in T by equation 8, and the normal parts contract linearly for both, $(\boldsymbol{I} - \eta\boldsymbol{H})^t$ versus $e^{-\boldsymbol{H}\eta t}$, which differ by $O(\eta)$ mode by mode, with summable nonlinear corrections. The constants are not explicit, and $\bar{\eta}$ can be much smaller than η_c, because the edge of the GF endpoint does not bound the curvature earlier on the path.

Proposition B.5 (Consequences of Assumption B.4)**.** *Let $\mathcal{R}$ be $L_{\mathcal{R}}$-Lipschitz, and write $\bar{\mathcal{R}}(\eta, T) = \mathbb{E}\,\mathcal{R}(\boldsymbol{\theta}_T)$ and $\bar{\mathcal{R}}_F(\tau) = \mathbb{E}\,\mathcal{R}(\Phi_\tau\boldsymbol{\theta}_0)$ (expectations over initializations). (a) For $0 < \eta \le \bar{\eta}$ and all T, $|\bar{\mathcal{R}}(\eta, T) - \bar{\mathcal{R}}_F(\eta T)| \le L_{\mathcal{R}} C\eta$. (b)* Persistence of Dip 1 (one-way). *Suppose $\bar{\mathcal{R}}_F$ has a separated interior minimum: there are $\tau_1 < \tau^\star < \tau_2$ and $\Delta > 0$ with $\bar{\mathcal{R}}_F(\tau_i) \ge \bar{\mathcal{R}}_F(\tau^\star) + \Delta$ for $i = 1, 2$. If $\tau_2/T \le \bar{\eta}$ and $2L_{\mathcal{R}} C\,\tau_2/T < \Delta$, then $\eta \mapsto \bar{\mathcal{R}}(\eta, T)$ has a local minimum at some η_1 with $\eta_1 T \in (\tau_1, \tau_2)$. (c)* Plateau. *If $|\bar{\mathcal{R}}_F(\tau) - \bar{\mathcal{R}}_F(\infty)| \le \varepsilon$ for $\tau \ge \tau_{\mathrm{fit}}$, then $|\bar{\mathcal{R}}(\eta, T) - \bar{\mathcal{R}}_F(\infty)| \le \varepsilon + L_{\mathcal{R}} C\eta$ for $\tau_{\mathrm{fit}}/T \le \eta \le \bar{\eta}$. (d)* Oracle early stopping, per initialization. *If moreover $\|\nabla\mathcal{L}\| \le G$ along the flow, then for every $\boldsymbol{\theta}_0 \in \Theta_0$ and $0 < \eta \le \bar{\eta}$, $\left|\min_{t \le T} \mathcal{R}(\boldsymbol{\theta}_t) - \min_{\tau \le \eta T} \mathcal{R}(\Phi_\tau\boldsymbol{\theta}_0)\right| \le L_{\mathcal{R}}(C+G)\eta$. The oracle-stopped risk is thus within $O(\eta)$ of a function that is non-increasing in η, and it has no interior minimum deeper than $O(\eta)$ on $(0, \bar{\eta}]$.*

Proof. (a) Take expectations of $|\mathcal{R}(\boldsymbol{\theta}_T) - \mathcal{R}(\Phi_{\eta T}\boldsymbol{\theta}_0)| \le L_{\mathcal{R}} C\eta$, which holds on Θ_0 with a uniform constant. (b) By (a), $\bar{\mathcal{R}}(\tau^\star/T, T) \le \bar{\mathcal{R}}_F(\tau^\star) + L_{\mathcal{R}} C\tau^\star/T < \bar{\mathcal{R}}_F(\tau_i) - L_{\mathcal{R}} C\tau_i/T \le \bar{\mathcal{R}}(\tau_i/T, T)$ for $i = 1, 2$. The map $\eta \mapsto \boldsymbol{\theta}_T$ is continuous for fixed T, so $\eta \mapsto \bar{\mathcal{R}}(\eta, T)$ is continuous on $[\tau_1/T, \tau_2/T]$, and its minimum there is attained in the interior. (c) Immediate from (a). (d) The GD iterates at times $t \le T$ are within $C\eta$ of the flow at the grid times ηt, and the flow's risk changes by at most $L_{\mathcal{R}} G\eta$ between grid times. □

Proposition B.5 is one-way and restricted to $\eta \le \bar{\eta}$. It locates Dip 1 at $\eta_1 \approx \tau^\star/T$ only up to the width of the separating interval, and it concerns oracle stopping. That the plateau is flat all the way up to η_c and that *validation-selected* stopping removes Dip 1 are empirical findings: the risk at matched ηT collapses to $\le 8 \cdot 10^{-6}$ across rates below η_c; below η_c the curve is flat to within 0.2% of the plateau in 72 of 72 settings (section B.5); Dip 1 sits at $\eta_1 T = \tau^\star = 68$–$94$ at every budget

from 10^2 to $4 \cdot 10^5$; and validation-selected per-rate early stopping makes the curve monotone below η_c in 72 of 72 settings (fig. 18a, fig. 19).

B.4 Why a stable constrained flow does not contradict (i)

Section 3 ("Is a catapult or EoS necessary") argues that a gradient step moves the GF limit only at second order in the step, whereas a constraint force moves it at first order. The precise statement is:

Proposition B.6 (First versus second order)**.** *Let $\boldsymbol{\theta}_* \in \mathcal{M}$ with $\boldsymbol{J}(\boldsymbol{\theta}_*)$ of full row rank, and let Assumption B.1 hold near $\boldsymbol{\theta}_*$. (a) $D\Pi(\boldsymbol{\theta}_*) = \boldsymbol{P}(\boldsymbol{\theta}_*)$, the orthogonal projector onto $T_{\boldsymbol{\theta}_*}\mathcal{M} = \ker \boldsymbol{J}(\boldsymbol{\theta}_*)$. (b) For any update $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - h(\nabla\mathcal{L}(\boldsymbol{\theta}) + \boldsymbol{\zeta})$ whose segment lies in the domain of Assumption B.1,*

$$\big\|\Pi(\boldsymbol{\theta}^+) - \Pi(\boldsymbol{\theta}) + h\, D\Pi(\boldsymbol{\theta})\,\boldsymbol{\zeta}\big\| \le \tfrac{\kappa}{2}\, h^2 \|\nabla\mathcal{L}(\boldsymbol{\theta}) + \boldsymbol{\zeta}\|^2.$$

(c) Consequently, with $\bar{\boldsymbol{\theta}} = \Pi(\boldsymbol{\theta})$ the foot point of $\boldsymbol{\theta}$ on $\mathcal{M}$,

$$\Pi(\boldsymbol{\theta}^+) = \Pi(\boldsymbol{\theta}) - h\,\boldsymbol{P}(\bar{\boldsymbol{\theta}})\,\boldsymbol{\zeta} + O\big(h\,\|\boldsymbol{\zeta}\|\,\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|\big) + O\big(h^2\|\nabla\mathcal{L}(\boldsymbol{\theta}) + \boldsymbol{\zeta}\|^2\big).$$

For plain GD ($\boldsymbol{\zeta} = 0$) only the second-order term remains. For the capped step of section C.5, $\boldsymbol{\zeta} = \sum_j \mu_j \nabla\lambda_j$ with $\mu_j \ge 0$, and the leading motion of the limit is $-h\sum_j \mu_j\, \boldsymbol{P}(\bar{\boldsymbol{\theta}})\nabla\lambda_j$: first order in h, along the descent direction of the sharpness within *the interpolation manifold.*

Proof. (a) Π is the identity on $\mathcal{M}$, so $D\Pi(\boldsymbol{\theta}_*)\boldsymbol{v} = \boldsymbol{v}$ for $\boldsymbol{v} \in \ker \boldsymbol{J}$. Differentiating equation 7 at $\boldsymbol{\theta}_*$ in a direction $\boldsymbol{v}$ gives $D^2\Pi(\boldsymbol{\theta}_*)[\boldsymbol{v}, \nabla\mathcal{L}(\boldsymbol{\theta}_*)] + D\Pi(\boldsymbol{\theta}_*)\nabla^2\mathcal{L}(\boldsymbol{\theta}_*)\boldsymbol{v} = 0$; since $\nabla\mathcal{L}(\boldsymbol{\theta}_*) = 0$, $D\Pi(\boldsymbol{\theta}_*)$ vanishes on the range of $\nabla^2\mathcal{L}(\boldsymbol{\theta}_*) = \boldsymbol{J}^\top\boldsymbol{J}/n$, which is $(\ker \boldsymbol{J})^\perp$. Hence $D\Pi(\boldsymbol{\theta}_*) = \boldsymbol{P}(\boldsymbol{\theta}_*)$. (b) Taylor's formula with integral remainder for Π along the step, and $D\Pi(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta}) = 0$ by equation 7. (c) $\|D\Pi(\boldsymbol{\theta}) - D\Pi(\bar{\boldsymbol{\theta}})\| \le \kappa\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|$ and $D\Pi(\bar{\boldsymbol{\theta}}) = \boldsymbol{P}(\bar{\boldsymbol{\theta}})$ by (a). □

So the stability ceiling can be imposed *without* unstable steps, and then it selects at first order. Native GD has no constraint force. Theorem B.2(i)(b) excludes its convergence to the GF interpolant above η_c but does not prescribe how it leaves (see the remark after Theorem B.2). In our measured native runs the trajectory crosses the nominal curvature threshold, with a catapult followed by sustained edge-of-stability oscillation, before reaching the high-rate endpoint (section D.3). The capped experiments of section 3 and section C.5 test whether the instability is necessary: in the verified regime the gain is retained without measured instability. This is consistent with the ceiling being what matters, but it does not establish the scalar curvature as the causal mediator, nor that capped and native GD reach the same function.

B.5 Selection along the ceiling: sign and slope of the onset

Above η_c the ceiling excludes interpolants with $\lambda > 2/\eta$. The simplest way to obey it is to descend the sharpness inside $\mathcal{M}$. We call this the *transport* $\dot{\boldsymbol{\theta}} = -\boldsymbol{g}_\Lambda$ with $\boldsymbol{g}_\Lambda = \boldsymbol{P}\nabla\Lambda$ and $\Lambda = \lambda_{\max}(\boldsymbol{K}/n)$; by Proposition B.6 it is the direction in which the constraint force moves the GF limit.

Proposition B.7 (Selection along the ceiling)**.** *Let $\boldsymbol{J}$ have full row rank and Λ be a simple eigenvalue on an open subset of $\mathcal{M}$, so that $\boldsymbol{P}$ and Λ are differentiable there. (a) The transport keeps every training output fixed ($\boldsymbol{J}\boldsymbol{P} = 0$), and along it*

$$\dot{\Lambda} = -\|\boldsymbol{g}_\Lambda\|^2, \qquad \dot{\mathcal{R}} = -\Gamma, \qquad \Gamma = \langle \boldsymbol{P}\nabla\mathcal{R}, \boldsymbol{g}_\Lambda\rangle = 2\,\mathbb{E}_{\boldsymbol{x}}\big[\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_*(\boldsymbol{x})\big)\,\delta f(\boldsymbol{x})\big], \quad (9)$$

*where $\delta f(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{g}_\Lambda$ vanishes on the training inputs. Where $\boldsymbol{g}_\Lambda \ne 0$, $d\mathcal{R}/d\Lambda = \Gamma/\|\boldsymbol{g}_\Lambda\|^2$ along the transport. (b) (*Local, fixed subcritical rate.*) Let $\mathcal{L} \in C^4$ near $\boldsymbol{\theta}_F$, the top eigenvalue Λ_F of $\boldsymbol{H} = \nabla^2\mathcal{L}(\boldsymbol{\theta}_F)$ be simple with unit eigenvector $\boldsymbol{u}$, and fix η with $\rho := \max_{\lambda_i \ne 0}|1 - \eta\lambda_i| < 1$ over the nonzero eigenvalues λ_i of $\boldsymbol{H}$ (so $\eta < \eta_c$). Start GD and GF from $\boldsymbol{\theta}_F + a\boldsymbol{u}$. Then their limits differ by $-c_\eta \boldsymbol{g}_\Lambda + O(a^3)$ and their risks by $-c_\eta\Gamma + O(a^3)$, with $c_\eta = \eta a^2/\big(4(2 - \eta\Lambda_F)\big)$. The $O(a^3)$ constants grow like $(1-\rho)^{-1}$, so the expansion is not uniform as $\eta \uparrow \eta_c$. (c) (*Onset slope; conditional.*) Assume, in addition, that $\boldsymbol{g}_\Lambda(\boldsymbol{\theta}_F) \ne 0$ and that for $\eta \downarrow \eta_c$ the GD endpoint $\boldsymbol{\theta}_\eta$ is, up to $o(\varepsilon)$ with $\varepsilon = \Lambda_F - 2/\eta$, the point at which the transport started at $\boldsymbol{\theta}_F$ first reaches $\Lambda = 2/\eta$. Then*

$$\frac{d\,\mathcal{R}(\boldsymbol{\theta}_\eta)}{d\log\eta}\bigg|_{\eta_c^+} = -\Lambda_F\,\frac{\Gamma}{\|\boldsymbol{g}_\Lambda\|^2}\bigg|_{\boldsymbol{\theta}_F},$$

so the second branch starts downward iff $\Gamma(\boldsymbol{\theta}_F) > 0$.

The contact assumption in (c) is what the self-stabilization / central-flow description of GD at the edge of stability (Damian et al., 2022; Cohen et al., 2024) gives when the ceiling first becomes active in an $O(\varepsilon)$ neighbourhood of $\mathcal{M}$ and the constrained endpoint depends differentiably on ε through $\boldsymbol{\theta}_F$: by Proposition B.6(c) the constraint force then moves the limit along $-\boldsymbol{g}_\Lambda$ at first order. It fails if the trajectory meets the ceiling far from $\mathcal{M}$ or approaches it degenerately; we do not prove it for native GD.

Proof. (a) $\boldsymbol{J}\boldsymbol{P} = 0$, so the transport keeps every training output. Along $\dot{\boldsymbol{\theta}} = -\boldsymbol{g}_\Lambda$, $\dot{\Lambda} = -\langle \nabla\Lambda, \boldsymbol{P}\nabla\Lambda\rangle = -\|\boldsymbol{g}_\Lambda\|^2$ and $\dot{\mathcal{R}} = -\langle \nabla\mathcal{R}, \boldsymbol{P}\nabla\Lambda\rangle = -\Gamma$, because $\boldsymbol{P}$ is symmetric and idempotent; with $\nabla\mathcal{R} = 2\mathbb{E}_{\boldsymbol{x}}[(f_{\boldsymbol{\theta}} - f_*)\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}]$ this is equation 9. (b) Write $\boldsymbol{\theta} = \boldsymbol{\theta}_F + \boldsymbol{h} + \boldsymbol{w}$ with $\boldsymbol{h}$ in the range of $\boldsymbol{H}$ and $\boldsymbol{w} \in \ker \boldsymbol{J}$. Then $\nabla\mathcal{L}(\boldsymbol{\theta}_F + \boldsymbol{v}) = \boldsymbol{H}\boldsymbol{v} + \frac{1}{2}\nabla^3\mathcal{L}[\boldsymbol{v},\boldsymbol{v}] + O(\|\boldsymbol{v}\|^3)$ and $\boldsymbol{P}\boldsymbol{H} = 0$. From $\boldsymbol{h}_0 = a\boldsymbol{u}$, GD gives $\boldsymbol{h}_t = a(1-\eta\Lambda_F)^t\boldsymbol{u} + O(a^2\rho^t)$, and the tangential motion is $\boldsymbol{w}_{t+1} - \boldsymbol{w}_t = -\frac{\eta}{2}\boldsymbol{P}\nabla^3\mathcal{L}[\boldsymbol{h}_t,\boldsymbol{h}_t] + O(a^3\rho^t)$; the remainders are summable, with sum $O(a^3/(1-\rho))$. For a simple eigenvalue (Hellmann–Feynman) $\partial_i\Lambda = \boldsymbol{u}^\top(\partial_i\boldsymbol{H})\boldsymbol{u} = \nabla^3\mathcal{L}[\boldsymbol{u},\boldsymbol{u}]_i$. Summing the geometric series with $1-(1-x)^2 = x(2-x)$ gives $\boldsymbol{w}_\infty = -\frac{a^2}{2\Lambda_F(2-\eta\Lambda_F)}\boldsymbol{g}_\Lambda + O(a^3)$. For GF, $\boldsymbol{h}(\tau) = ae^{-\Lambda_F\tau}\boldsymbol{u}$ gives $\boldsymbol{w}_\infty = -\frac{a^2}{4\Lambda_F}\boldsymbol{g}_\Lambda + O(a^3)$. The difference is $-c_\eta\boldsymbol{g}_\Lambda$, and expanding $\mathcal{R}$ to first order gives the risk difference. (c) By (a), the transport from $\boldsymbol{\theta}_F$ reaches $\Lambda = 2/\eta$ at time $t(\varepsilon) = \varepsilon/\|\boldsymbol{g}_\Lambda\|^2 + o(\varepsilon)$, and the risk there is $\mathcal{R}(\boldsymbol{\theta}_F) - \Gamma\, t(\varepsilon) + o(\varepsilon)$. By assumption this is $\mathcal{R}(\boldsymbol{\theta}_\eta)$ up to $o(\varepsilon)$. Finally $\frac{d\varepsilon}{d\log\eta} = 2/\eta \to \Lambda_F$ as $\eta \downarrow \eta_c$. □

The ceiling specifies a direction of selection, not a generalization objective: a sharpness-reducing move improves the risk when the change it induces in the population predictor is aligned with the prediction error. The factor $1/(2-\eta\Lambda_F)$ in (b) grows towards the stability edge, so selection concentrates there, although (b) itself holds only at fixed $\eta < \eta_c$. The identities (a) are exact for the declared transport; their identification with the native GD endpoint is the assumption of (c). Averaged over seeds at a fixed training set, with $v = \delta f/\|\boldsymbol{g}_\Lambda\|^2$ and bars for seed means,

$$\mathbb{E}_{\text{seed}}\left[\Gamma/\|\boldsymbol{g}_\Lambda\|^2\right] = 2\,\mathbb{E}_{\boldsymbol{x}}\left[(\bar{f} - f_*)\,\bar{v}\right] + 2\,\mathbb{E}_{\boldsymbol{x}}\,\mathrm{Cov}_{\text{seed}}(f, v). \tag{10}$$

The first term improves the mean predictor, the second removes seed-specific structure.

Structural zeros. $\Gamma = 0$ (i) for fixed features, since then $\nabla\Lambda \equiv 0$ (Theorem B.2(ii)(a)); (ii) for moves along symmetry orbits that leave $f_{\boldsymbol{\theta}}$ invariant, which are orthogonal to $\boldsymbol{P}\nabla\mathcal{R}$; and (iii) when the data identify the function locally, $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}) \in \operatorname{row}\boldsymbol{J}$ for all $\boldsymbol{x}$, since then $\boldsymbol{P}\nabla\mathcal{R} = 0$ (Theorem B.2(iii)(b)). A nonzero Γ therefore needs a nonlinear model whose population function is not determined by the data.

Pre-registered test. In 72 settings (listed in fig. 7; protocol in section C.8) we computed $\eta_c = 2/\Lambda_F$ and the left side of equation 10 from the small-rate interpolant and froze them before any high-rate run. Results (fig. 9):

- Below η_c the risk is flat to within 0.2% of the plateau (median 0.055%, maximum absolute drift $3.9 \cdot 10^{-4}$).
- $\Gamma > 0$ in all 72 settings, and the sign of the post-onset change is right in 72 of 72.
- The measured onset slope (least squares over $\eta_c < \eta \le 1.5\eta_c$) is 0.94–1.21 times the prediction of (c) in 40 of 41 smooth-activation settings of depth ≥ 2 (median 1.07; the exception, a clean tanh net of width 1024 with a vanishing branch, gives 0.76). ReLU at depth ≥ 2 gives 0.37–0.74 (median 0.64), consistent with gate crossings violating Assumption B.1. At depth one, where the branch ends at $1.7\eta_c$, the first-order law is unreliable (0.01–1.56).
- In the 36 label-noise settings of depth ≥ 2, 51–90% (median 80%) of Γ is the covariance term of equation 10.

Negative control. Because $\Gamma > 0$ everywhere, we also changed its sign on the same runs. At each seed's $\boldsymbol{\theta}_F$ we formed δf on the test inputs and the teachers $f_{*,s} = f_{\boldsymbol{\theta}_F} + s\,\delta f$. Since $\delta f(\boldsymbol{X}) = 0$, each fits every training label exactly, and $\Gamma_s = -2s\,\mathbb{E}_{\boldsymbol{x}}[\delta f^2]$. With $s = m\,t_{\max}$, $m \in \{\pm1, \pm2, \pm4\}$

and $t_{\max} = (\Lambda_F/3)/\|\boldsymbol{g}_\Lambda\|^2$ (the transport time at $\eta = 1.5\eta_c$), the transport predicts $\mathcal{R}_s - \mathcal{R}_s(\boldsymbol{\theta}_F) = \mathbb{E}[\delta f^2]\,(2st + t^2)$ with $t = (\Lambda_F - 2/\eta)_+/\|\boldsymbol{g}_\Lambda\|^2$ and no fitted constant. The measured sign follows the prediction in 24 of 24 cases (four activations, six offsets); the part odd in s is 1.10–1.13 times the prediction for SiLU, GELU and tanh and 0.63 for ReLU (fig. 10).

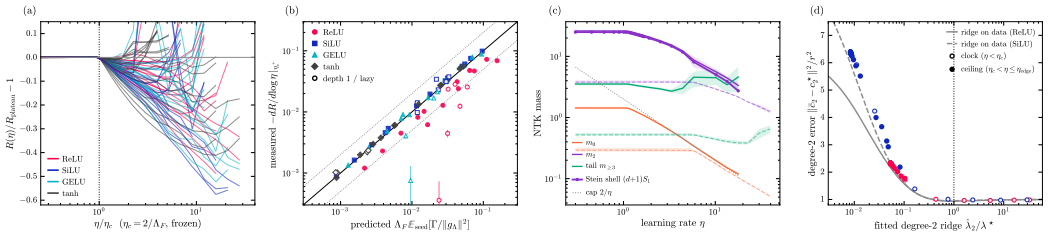

Figure 9: **Pre-registered test of Proposition B.7** (72 settings, 8 seeds each; η_c and $\mathbb{E}_{\text{seed}}[\Gamma/\|\boldsymbol{g}_\Lambda\|^2]$ frozen from a small-rate run before any high-rate run). (a) Risk relative to the plateau against η/η_c, each setting up to the end of its trainable branch. (b) Measured onset slope $-d\mathcal{R}/d\log\eta\big|_{\eta_c^+}$ against the prediction $\Lambda_F\,\mathbb{E}_{\text{seed}}[\Gamma/\|\boldsymbol{g}_\Lambda\|^2]$; open markers: depth 1 or lazy; bars: seed bootstrap; dotted: factor 3. (c) SiLU branch (width 256; $\sigma = 0.5$ solid, clean dashed; bands: seed range): the degree-0 mass follows the cap $2/\eta$, the degree-2 mass follows the Stein shell $(d+1)S_1$ of Lemma B.8, the tail $m_{\geq 3}$ does not. (d) Degree-2 error of the seed-mean predictor against its fitted degree-2 ridge $\hat{\lambda}_2$, in units of the ridge model ($\lambda^\star$, r^2; ReLU and SiLU, width 256, $\sigma = 0.5$): below η_c (open) and on the branch (filled) the network moves along the data's own ridge curve (grey).

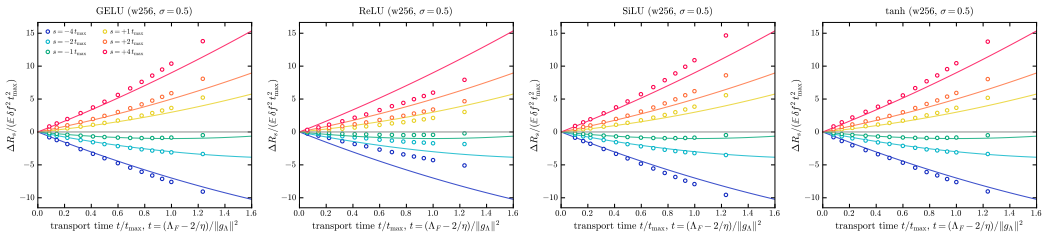

Figure 10: **Negative control for the sign of Γ.** Risk change for the teachers $f_{*,s} = f_{\boldsymbol{\theta}_F} + s\,\delta f$, which fit the same training labels, against the transport time $t/t_{\max}$ (width 256, $\sigma = 0.5$). Lines: the prediction $(2st + t^2)\,\mathbb{E}\delta f^2$ with no fitted constant; points: measured, one colour per offset s.

B.6 The ceiling caps the head, not the tail (Lemma B.8)

We use the notation of equation 4: for $p = \mathcal{N}(0, \boldsymbol{I}_d)$, $\boldsymbol{j}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_k \boldsymbol{j}_k(\boldsymbol{x})$ is the Hermite expansion of the tangent features, $m_k = \mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{j}_k(\boldsymbol{x})\|^2$ is the degree-k mass of the kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{j}(\boldsymbol{x}), \boldsymbol{j}(\boldsymbol{x}')\rangle$, and $m_{\geq 3} = \mathbb{E}_{\boldsymbol{x}} K(\boldsymbol{x}, \boldsymbol{x}) - m_0 - m_1 - m_2$.

Lemma B.8 (The ceiling caps the head, not the tail)**.** *(i) $m_0 = \|\mathbb{E}_{\boldsymbol{x}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x})\|^2$, and its design version $\mathbf{1}^\top \boldsymbol{K}\mathbf{1}/n^2$ is at most $\lambda_{\max}(\boldsymbol{K}/n)$. (ii) Write the first-layer block as $K_1(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}')\,\boldsymbol{b}(\boldsymbol{x})^\top \boldsymbol{b}(\boldsymbol{x}')$ with $\boldsymbol{b} = \partial f/\partial(\boldsymbol{W}_1 \boldsymbol{x})$, and let $\boldsymbol{B}_1 = \mathbb{E}[\boldsymbol{b}(\boldsymbol{x})\boldsymbol{x}^\top]$ and $S_1 = \|\boldsymbol{B}_1\|_F^2$. Then $m_0(K_1) = S_1 \leq m_0$ and $m_2(K_1) = (d+1)S_1 + \epsilon_3$, where the correction ϵ_3 consists of the contribution of the degree-3 Hermite part of $\boldsymbol{b}$ and its cross term with the degree-1 part; it vanishes when the degree-3 part is zero. (iii) If the tail's part of $\boldsymbol{K}$ on the training inputs is $\beta \boldsymbol{I}_n$, it adds exactly β/n to every eigenvalue of $\boldsymbol{K}/n$, so the ceiling bounds it only by $\beta \leq 2n/\eta$.*

Proof. (i) $\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'} K(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{x}} \nabla f(\boldsymbol{x})^\top \mathbb{E}_{\boldsymbol{x}'} \nabla f(\boldsymbol{x}')$, and the degree-0 projection of K is this constant. The design version is the Rayleigh quotient of $\boldsymbol{K}/n$ at the unit vector $\mathbf{1}/\sqrt{n}$. (ii) $\nabla_{\boldsymbol{W}_1} f(\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x})\boldsymbol{x}^\top$, so $K_1 = \sum_{i=1}^d \boldsymbol{u}_i(\boldsymbol{x})^\top \boldsymbol{u}_i(\boldsymbol{x}')$ with $\boldsymbol{u}_i = x_i \boldsymbol{b} \in \mathbb{R}^w$, and the degree-$k$ projection acts coordinatewise: $m_k(K_1) = \sum_i \|P_k \boldsymbol{u}_i\|^2$. First, $P_0 \boldsymbol{u}_i = \mathbb{E}[x_i \boldsymbol{b}] = \boldsymbol{B}_1 \boldsymbol{e}_i$, so $m_0(K_1) = S_1$; this is at most m_0 because m_0 is the sum of the nonnegative degree-0 masses $\|\mathbb{E}\nabla_{\boldsymbol{\theta}_l} f\|^2$ of all parameter

blocks (Stein's lemma: $\mathbb{E}[\underline{\boldsymbol{b}}(\boldsymbol{x})x_i] = \mathbb{E}[\partial_i \boldsymbol{b}]$, so $\boldsymbol{B}_1$ is the mean Jacobian of the back-propagated signal). Next, expand $\boldsymbol{b} = \bar{\boldsymbol{b}} + \boldsymbol{B}_1\boldsymbol{x} + \boldsymbol{b}_2 + \boldsymbol{b}_3 + \boldsymbol{b}_{\geq 4}$ in Hermite degrees; the degree-1 coefficient is $\mathbb{E}[\boldsymbol{b}\boldsymbol{x}^\top] = \boldsymbol{B}_1$. Since $x_i \mathrm{He}_\alpha = \mathrm{He}_{\alpha+\boldsymbol{e}_i} + \alpha_i \mathrm{He}_{\alpha-\boldsymbol{e}_i}$, a degree-$k$ term of $\boldsymbol{b}$ contributes to degree 2 of $\boldsymbol{u}_i$ only if $k \in \{1, 3\}$. From $k = 1$, $P_2(x_i \boldsymbol{B}_1\boldsymbol{x}) = \sum_{j\neq i} \boldsymbol{B}_1\boldsymbol{e}_j\, x_i x_j + \boldsymbol{B}_1\boldsymbol{e}_i\,(x_i^2 - 1)$, whose squared norm is $\sum_{j\neq i} \|\boldsymbol{B}_1\boldsymbol{e}_j\|^2 + 2\|\boldsymbol{B}_1\boldsymbol{e}_i\|^2 = S_1 + \|\boldsymbol{B}_1\boldsymbol{e}_i\|^2$; summed over i this is $(d+1)S_1$. The degree-3 part adds $\boldsymbol{c}_i = P_2(x_i\boldsymbol{b}_3)$, so $m_2(K_1) = (d+1)S_1 + \epsilon_3$ with $\epsilon_3 = \sum_i \big(2\langle P_2(x_i\boldsymbol{B}_1\boldsymbol{x}), \boldsymbol{c}_i\rangle + \|\boldsymbol{c}_i\|^2\big)$, which can have either sign (for a single ReLU unit, $\epsilon_3 = -3/(4\pi)$; for $d = 1$ and $b(x) = ax + c\,\mathrm{He}_3(x)$, $m_2(K_1) = 2(a+3c)^2 = 2a^2 + 12ac + 18c^2$, so ϵ_3 depends on both parts). (iii) $\lambda_i(\boldsymbol{K}_\mathrm{h}/n + \beta\boldsymbol{I}/n) = \lambda_i(\boldsymbol{K}_\mathrm{h}/n) + \beta/n$, and $\lambda_{\max}(\boldsymbol{K}/n) \leq 2/\eta$ at a stable interpolant. □

Collecting ϵ_3 and the degree-2 mass of the other parameter blocks in $1 + \epsilon = m_2/((d+1)S_1)$ and writing $1 + \delta = m_0/\lambda_{\max}(\boldsymbol{K}/n)$ gives step ❷ of section 4.2: $m_2 \leq (1+\epsilon)(1+\delta)(d+1)\,2/\eta$. Measured on the SiLU branch, $\epsilon = 0.04$–0.21 and $1 + \delta \leq 1.05$ once the cap binds (table 1); the Stein shell $(d+1)S_1$ carries 81–98% (ReLU), 81–96% (SiLU) and 76–96% (GELU) of m_2 (8 seeds). Odd networks such as tanh have $m_0 = m_2 = 0$ exactly, so this channel does not apply to them.

Why depth or a bias is needed. For $f(\boldsymbol{x}) = \sum_j a_j \phi(\boldsymbol{w}_j^\top \boldsymbol{x})$ with $\phi(z) - \phi(-z) = z$ (ReLU, GELU, SiLU), $\phi'(z) + \phi'(-z) = 1$ and symmetry of $\boldsymbol{w}^\top\boldsymbol{x}$ give $\mathbb{E}_{\boldsymbol{x}}\phi'(\boldsymbol{w}_j^\top\boldsymbol{x}) = \frac{1}{2}$ for all weights; for ReLU, $\nabla_{\boldsymbol{w}_j} f = a_j \mathbb{1}[\hat{\boldsymbol{w}}_j^\top\boldsymbol{x} > 0]\,\boldsymbol{x}$ and the degree-k mass of each unit's kernel is rotation-invariant, so $m_k(\boldsymbol{K}_1) = c_k(d)\sum_j a_j^2$ with constants depending only on (d, k) (e.g. $c_0 = 1/(2\pi)$, $c_2 = (2d-1)/(4\pi)$). A one-hidden-layer bias-free network can therefore change its kernel's degree profile only through the balance $\sum_j a_j^2 : \sum_j \|\boldsymbol{w}_j\|^2$, not by closing gates. A bias turns the gate into $\mathbb{1}[\hat{\boldsymbol{w}}^\top\boldsymbol{x} > t]$, whose Hermite profile moves towards high degrees as the threshold rises, and a second hidden layer lets the first layer's back-propagated signal carry the second layer's gates, whose thresholds move with the mean activation. This is consistent with the short branches of depth-1 networks in fig. 7. (Monte Carlo check at $d = 8$: unit masses 0.1593 and 1.181 against $1/(2\pi) = 0.1592$ and $15/(4\pi) = 1.194$; mean gate 0.500 ± 0.002 for ReLU, SiLU and GELU at $\|\boldsymbol{w}\| \in \{0.3, 1, 3\}$.)

B.7 Proof of Theorem 4.1, tests and assumption diagnostics

Precise setting. Let $\boldsymbol{\Phi}_1 \in \mathbb{R}^{n\times(d+1)}$ and $\boldsymbol{\Phi}_2 \in \mathbb{R}^{n\times m}$, $m = d(d+1)/2 = 2080$, hold the normalized Hermite features $\phi_1(\boldsymbol{x})$, $\phi_2(\boldsymbol{x})$ of degrees ≤ 1 and 2 at the training inputs, and assume $n > d + 1$ with $\boldsymbol{\Phi}_1$ of full column rank; $n_1 = n - d - 1$ and $\psi = m/n_1$. Let $\boldsymbol{Q} = \boldsymbol{I}_n - \boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^+$ project off the resolved features, $\boldsymbol{S} = \boldsymbol{\Phi}_2^\top\boldsymbol{Q}\boldsymbol{\Phi}_2/n_1$, and $\boldsymbol{G} = \boldsymbol{\Phi}_1^+\boldsymbol{\Phi}_2$. Theorem 4.1 is shorthand for the following statements under these assumptions:

- (S) $\boldsymbol{\theta}_\eta$ is an interpolating GD limit that is linearly stable, $\eta\lambda_{\max}(\boldsymbol{K}/n) \leq 2$ (by Theorem B.2(i)(b) this is the generic case).
- (A1) $f_{\boldsymbol{\theta}_\eta}(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^\top\boldsymbol{K}^{-1}\boldsymbol{y}$ for all $\boldsymbol{x}$, with $\boldsymbol{k}(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{X})$ its own final tangent kernel; in particular no initialization offset is present (its effect is discussed below).
- (A2) On the training inputs $\boldsymbol{K} = A_1\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^\top + A\,\boldsymbol{\Phi}_2\boldsymbol{\Phi}_2^\top + \beta\boldsymbol{I}_n$ and $\boldsymbol{k}(\boldsymbol{x}) = A_1\boldsymbol{\Phi}_1\phi_1(\boldsymbol{x}) + A\,\boldsymbol{\Phi}_2\phi_2(\boldsymbol{x})$, with $A = m_2/m > 0$, $\beta = q\,m_{\geq 3} > 0$, and $A_1/A \to \infty$ (resolved degrees).
- (A3) $m_{\geq 3} \geq T_0 > 0$.
- (M) *Averaging model.* The clean target is $f_* = \phi_1^\top\boldsymbol{\theta}_1^\star + \phi_2^\top\boldsymbol{\theta}_2^\star + f_{\geq 3}$ with $\mathbb{E}_{\boldsymbol{x}} f_{\geq 3}^2 = B_{\geq 3}$ and $f_{\geq 3}$ orthogonal in $L^2(p)$ to degrees ≤ 2. Risks are averaged over $\boldsymbol{\theta}_2^\star$, drawn isotropically with $\mathbb{E}\boldsymbol{\theta}_2^\star\boldsymbol{\theta}_2^{\star\top} = (r^2/m)\boldsymbol{I}$, and over the noise. On the design, the label noise plus the unlearned part, $\boldsymbol{\varepsilon} = \sigma\boldsymbol{z} + f_{\geq 3}(\boldsymbol{X})$, is modelled as independent of $\boldsymbol{\theta}_2^\star$, mean zero, with covariance $\sigma_{\mathrm{eff}}^2\boldsymbol{I}_n$, $\sigma_{\mathrm{eff}}^2 = \sigma^2 + B_{\geq 3}$. The design and the kernel parameters A, β are held fixed in the average. Treating $f_{\geq 3}(\boldsymbol{X})$ as independent noise, and the learned, label-dependent kernel as fixed, are modelling assumptions, not consequences of (A1)–(A3).

Write $\mathcal{R}_n(\lambda)$ for the averaged population risk, at this fixed design, of the ridge predictor defined in (a) below.

(a), ridge (exact). The kernel interpolant is the minimum-norm solution of $\min \|\boldsymbol{\theta}_1\|^2/A_1 + \|\boldsymbol{\theta}_2\|^2/A + \|\boldsymbol{v}\|^2/\beta$ subject to $\boldsymbol{\Phi}_1\boldsymbol{\theta}_1 + \boldsymbol{\Phi}_2\boldsymbol{\theta}_2 + \boldsymbol{v} = \boldsymbol{y}$. Eliminating the tail variable $\boldsymbol{v}$ gives $\min \|\boldsymbol{y} - \boldsymbol{\Phi}_1\boldsymbol{\theta}_1 - \boldsymbol{\Phi}_2\boldsymbol{\theta}_2\|^2/\beta + \|\boldsymbol{\theta}_2\|^2/A + \|\boldsymbol{\theta}_1\|^2/A_1$. Since the tail extends nothing, the prediction is $\phi_1(\boldsymbol{x})^\top\hat{\boldsymbol{\theta}}_1 + \phi_2(\boldsymbol{x})^\top\hat{\boldsymbol{\theta}}_2$. As $A_1/A \to \infty$, $\boldsymbol{\theta}_1$ is unpenalized: $\hat{\boldsymbol{\theta}}_2 = (\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}_2^\top\boldsymbol{Q}\boldsymbol{y}/n_1$ is ridge regression of $\boldsymbol{y}$ on $\boldsymbol{\Phi}_2$ after projecting out $\operatorname{span}\boldsymbol{\Phi}_1$, with $\lambda = \beta/(n_1 A) = q\, m_{\geq 3}\, m/(n_1 m_2)$, and $\hat{\boldsymbol{\theta}}_1 = \boldsymbol{\Phi}_1^+(\boldsymbol{y} - \boldsymbol{\Phi}_2\hat{\boldsymbol{\theta}}_2)$.

(a), floor (exact). By Lemma B.8(ii), $m_2 = (1+\epsilon)(d+1)S_1$ and $S_1 \leq m_0$. By Lemma B.8(i) and (S), $m_0 = (1+\delta)\lambda_{\max}(\boldsymbol{K}/n) \leq (1+\delta)\, 2/\eta$. Hence $m_2 \leq (1+\epsilon)(1+\delta)(d+1)\, 2/\eta$. With (A3) and $m/(d+1) = d/2$: $\lambda \geq q\, T_0\, d\, \eta/\big(4n_1(1+\epsilon)(1+\delta)\big) = \lambda_{\text{floor}}(\eta)$.

Exact finite-design risk. Write $\boldsymbol{e} = \hat{\boldsymbol{\theta}}_2 - \boldsymbol{\theta}_2^\star$. Then $\boldsymbol{e} = -\lambda(\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\theta}_2^\star + (\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}_2^\top\boldsymbol{Q}\boldsymbol{\varepsilon}/n_1$ and $\hat{\boldsymbol{\theta}}_1 - \boldsymbol{\theta}_1^\star = \boldsymbol{\Phi}_1^+\boldsymbol{\varepsilon} - \boldsymbol{G}\boldsymbol{e}$. By population orthonormality of the Hermite features the risk is $\|\hat{\boldsymbol{\theta}}_1 - \boldsymbol{\theta}_1^\star\|^2 + \|\boldsymbol{e}\|^2 + B_{\geq 3}$. The cross term $\mathbb{E}\,\boldsymbol{\varepsilon}^\top\boldsymbol{\Phi}_1^{+\top}\boldsymbol{G}\boldsymbol{e}$ vanishes because $\boldsymbol{Q}\boldsymbol{\Phi}_1 = 0$, so with $\boldsymbol{W} = \boldsymbol{I} + \boldsymbol{G}^\top\boldsymbol{G}$,

$$\mathcal{R}_n(\lambda) = \sigma_{\text{eff}}^2 \operatorname{tr}\big((\boldsymbol{\Phi}_1^\top\boldsymbol{\Phi}_1)^{-1}\big) + B_{\geq 3} + \frac{r^2\lambda^2}{m} \operatorname{tr}\big(\boldsymbol{W}(\boldsymbol{S}+\lambda\boldsymbol{I})^{-2}\big) + \frac{\sigma_{\text{eff}}^2}{n_1} \operatorname{tr}\big(\boldsymbol{W}\boldsymbol{S}(\boldsymbol{S}+\lambda\boldsymbol{I})^{-2}\big). \quad (11)$$

The matrix $\boldsymbol{W}$ accounts for the unpenalized degree-≤ 1 coefficients, which absorb part of the degree-2 error on a finite design.

(c), monotonicity (exact, finite n). Since $\boldsymbol{S}$ commutes with $(\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}$, $\partial_\lambda[\lambda^2(\boldsymbol{S} + \lambda\boldsymbol{I})^{-2}] = 2\lambda\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-3}$ and $\partial_\lambda[\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-2}] = -2\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-3}$, so

$$\mathcal{R}_n'(\lambda) = \frac{2r^2}{m}\,(\lambda - \lambda^\star)\,\operatorname{tr}\big(\boldsymbol{W}\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-3}\big), \qquad \lambda^\star = \frac{m\,\sigma_{\text{eff}}^2}{n_1 r^2} = \frac{\psi\,\sigma_{\text{eff}}^2}{r^2}.$$

The trace of a product of two positive semidefinite matrices is nonnegative, and it is positive when $\boldsymbol{S} \neq 0$. Hence $\mathcal{R}_n$ decreases on $[0, \lambda^\star]$ for every design. (Without $\boldsymbol{W}$, i.e. counting only the degree-2 coefficients, the same sign result holds; the $\boldsymbol{W}$-weighting is needed for the full risk.)

(b), noise in the head (exact, finite n). The label noise reaches test inputs through degree 2 with energy $\frac{\sigma^2}{n_1}\operatorname{tr}\big(\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-2}\big)$, whose derivative in λ is $-\frac{2\sigma^2}{n_1}\operatorname{tr}\big(\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-3}\big) < 0$. Of the fitted noise vector $\boldsymbol{\varepsilon}$, the degree-2 head stores (in expectation) the share $\pi_2 = \frac{1}{n}\operatorname{tr}\big(\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}\big)$, i.e. $\frac{n_1}{n}\cdot\frac{1}{n_1}\operatorname{tr}(\cdot)$; the resolved degrees store $(d+1)/n$ and the tail the rest. Both π_2 and the head's share of the part orthogonal to the resolved degrees, $\frac{n}{n_1}\pi_2$, decrease in λ.

Conclusion (exact under the model). If the plateau satisfies the same assumptions with ridge λ_{nat}, then under (S), (A1)–(A3) and (M), $\mathcal{R}(\boldsymbol{\theta}_\eta) = \mathcal{R}_n(\lambda(\eta))$ and $\mathcal{R}_{\text{plateau}} = \mathcal{R}_n(\lambda_{\text{nat}})$, and $\lambda_{\text{nat}} < \lambda_{\text{floor}}(\eta) \leq \lambda(\eta) \leq \lambda^\star$ gives $\mathcal{R}(\boldsymbol{\theta}_\eta) \leq \mathcal{R}_n(\lambda_{\text{floor}}(\eta)) < \mathcal{R}_n(\lambda_{\text{nat}}) = \mathcal{R}_{\text{plateau}}$. □

Closed forms (asymptotic, not exact). The coloured formulas of Theorem 4.1 are the deterministic equivalents of these finite-design quantities. They require an additional random-design assumption: as $n, m \to \infty$ with ψ fixed, the projected degree-2 design behaves like an isotropic Gaussian design (Gaussian-feature universality), so that the spectrum of $\boldsymbol{S}$ follows the Marchenko–Pastur law (Dobriban & Wager, 2015; Simon et al., 2023). Then each degree-2 mode has learnability $\nu = 1/(1+\kappa)$, where the effective regularization κ solves $\psi\nu + \lambda\nu/(1-\nu) = 1$; multiplying by $1-\nu$ gives $\psi\nu^2 - (1+\psi+\lambda)\nu + 1 = 0$, whose root in $(0,1)$ is the ν of the theorem. In the eigenlearning framework the test error is $\frac{n}{n - \sum_i L_i^2}\big(\sum_i (1-L_i)^2 v_i^2 + \sigma^2\big)$ over all modes with learnabilities L_i and target energies v_i^2; with $d+1$ resolved modes ($L_i = 1$), m degree-2 modes ($L_i = \nu$, total energy r^2) and the degrees ≥ 3 ($L_i = 0$, energy $B_{\geq 3}$), $n - \sum L_i^2 = n_1(1 - \psi\nu^2)$, and subtracting the label noise of a fresh test point gives $\mathcal{R}_{\text{ridge}}(\lambda) = \frac{n}{n_1}\big(r^2(1-\nu)^2 + \sigma_{\text{eff}}^2\big)/(1-\psi\nu^2) - \sigma^2$. The factor n/n_1 is the asymptotic form of the resolved-degree terms in equation 11 (the λ-independent term and the rank-$(d+1)$ correction $\boldsymbol{G}^\top\boldsymbol{G}$). In the same limit the degree-2 noise energy is $\sigma^2\psi\nu^2/(1-\psi\nu^2)$ and $\frac{1}{n_1}\operatorname{tr}(\boldsymbol{S}(\boldsymbol{S} + \lambda\boldsymbol{I})^{-1}) \to \psi\nu$, so the head stores the share $\pi_2 = m\nu/n$ of the fitted noise vector, equivalently $\psi\nu$ of its part orthogonal to the resolved degrees. These closed forms are approximations at finite n, with no sign guarantee on their error; the strict inequality above is a statement

about $\mathcal{R}_n$, not about $\mathcal{R}_{\mathrm{ridge}}$. At our size the approximation error is quantified below ($\le 1.41\%$ for $\lambda \ge 0.5$, 5.31% overall).

Why (A2) is natural. For rotation-invariant kernels with $n \asymp d^2$, the reduction of (A2) is a theorem: the kernel matrix is a resolved low-degree part, plus a Wishart-like degree-2 block, plus the trace of the higher degrees times $\boldsymbol{I}_n$, so the higher degrees act as a ridge with $q = 1$ (Misiakiewicz, 2022; Hu & Lu, 2022). The learned tail is lumpier: about half of it acts as a ridge ($q = 0.4$–0.6 on the SiLU branch). Storage in the tail is cheap because a minimum-norm fit of energy E with $N > n$ isotropic features leaves $\approx E\,n/(N-n)$ at test inputs: with $N = m = 2080$ this is $E/(\psi - 1) \approx 0.86E$ in the head, but with $N = N_3 = \binom{d+2}{3} = 45{,}760$ it is $\approx 0.02E$ in degree 3.

Initialization terms (remark after Theorem 4.1). With resolved degrees projected out, the head's posterior covariance on its coefficients is $A\boldsymbol{I} - A^2\boldsymbol{\Phi}_2^\top(A\boldsymbol{\Phi}_2\boldsymbol{\Phi}_2^\top + \beta\boldsymbol{I})^{-1}\boldsymbol{\Phi}_2 = (A^{-1}\boldsymbol{I} + \boldsymbol{\Phi}_2^\top\boldsymbol{\Phi}_2/\beta)^{-1}$ (Woodbury). Its trace in the eigenbasis of $\boldsymbol{\Phi}_2^\top\boldsymbol{\Phi}_2/n_1$ is $\sum_i A\lambda/(s_i+\lambda)$; each term increases in A, and the sum is at most $mA = m_2$, which the ceiling bounds by (a). For a tangent model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{j}(\boldsymbol{x})^\top\boldsymbol{\theta}$ with $\boldsymbol{J}$ fixed and $\boldsymbol{\theta}_0 \sim \mathcal{N}(0, s^2\boldsymbol{I})$, GF converges to $\boldsymbol{\theta}_0 + \boldsymbol{J}^\top\boldsymbol{K}^{-1}(\boldsymbol{y} - \boldsymbol{J}\boldsymbol{\theta}_0)$, so $f(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^\top\boldsymbol{K}^{-1}\boldsymbol{y} + \boldsymbol{j}(\boldsymbol{x})^\top\boldsymbol{P}\boldsymbol{\theta}_0$ and $\mathrm{Var}_{\boldsymbol{\theta}_0} f(\boldsymbol{x}) = s^2\big(K(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^\top\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x})\big)$, the posterior variance. This is why the remark after Theorem 4.1 speaks of a *direction*: at a fixed design and a fixed tail β the head's posterior mass increases in A and is bounded by m_2, which the ceiling lowers. When the tail and the design statistics also change with η, a decreasing upper bound alone does not determine the change in the actual initialization variance, and trained networks are not tangent models with isotropic offsets. We therefore claim neither the size nor, in general, the sign of the initialization terms of fig. 4 from this argument.

Finite-n check of the closed form. We drew $\boldsymbol{X}$ with $d = 64$, $n = 1024$, built all 2080 degree-2 Hermite features, and computed the exact risk $\mathcal{R}_n(\lambda)$ of the interpolant with kernel $A_1\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^\top + A\boldsymbol{\Phi}_2\boldsymbol{\Phi}_2^\top + \beta\boldsymbol{I}$ under the averaging model (M) (teacher rotations and spherical noise, in closed form via traces), on the 36-point grid $\sigma^2 \in \{1/16, 1/4, 1\}$, $m_2 \in [1.5, 26.3]$, $\beta \in \{1.7, 3.6\}$. The closed form deviates from it by at most 1.41% for $\lambda \ge 0.5$ and 5.31% overall, at the smallest ridge ($\lambda = 0.14$), where the degree-2 design is furthest from Marchenko–Pastur (fig. 12). This checks the deterministic-equivalent approximation on this grid; it is neither a uniform error bound nor a test of the modelling assumption (M), since the check itself treats the unlearned teacher part as spherical noise.

Measured inputs (canonical SiLU, $\sigma = 0.5$). Plateau masses $m_2^{\mathrm{pl}} = 26.3$, $m_{\ge 3}^{\mathrm{pl}} = 3.6$ (the tail stays within 2.8–4.5 on the branch, apart from 6.1 at $\eta = 7.3$); $\epsilon = 0.045$ on the plateau rising to 0.2 at the edge; $\delta \le 0.05$ once the cap binds; teacher energies $r^2 = 0.0229$ and $B_{\ge 3} = 0.0060$. The first layer's share of the cap, $\kappa_S = S_1/(2/\eta)$, rises from 0.21 on the plateau to 0.40 ± 0.01 for $\eta \ge 10$; the same share appears for rebalanced initializations and along the sharpness transport, so it is an attractor rather than inherited from the plateau. The tail's ridge efficiency $q = 0.47$ is fitted on the canonical net; $\kappa_S = 0.40$ and $q = 0.47$ are the only two constants in fig. 5. The network's ridge exceeds the proven floor (evaluated at $q = 1$, $\delta = 0$) iff $q \ge \kappa_S$. With these inputs, $\lambda^\star \approx 24$ and the risk-optimal rate is $\eta^\star \approx 360$, far beyond the largest trainable rate (≈ 20). The closed form switches the head cap on sharply at η_{on}, where $(1+\epsilon)(d+1)\kappa_S\, 2/\eta_{\mathrm{on}} = m_2^{\mathrm{pl}}$, whereas the network ramps κ_S from 0.21 to 0.40 between $\eta \approx 1.3$ and 10; it therefore descends earlier and more gently than predicted.

Estimand and uncertainty. Every test uses the population risk of a single trained network, estimated against the clean target on 16,384 test inputs and summarized by the median over seeds; only interpolating runs (training MSE $\le 10^{-2}$) enter. Relative to the plateau the canonical risk is $0.91, 0.86, 0.77, 0.64, 0.59, 0.52, 0.45$ at $\eta = 2.6, 3.7, 5.2, 7.3, 10.3, 14.5, 20.4$; predicted, $0.94, 0.85, 0.75, 0.66, 0.56, 0.48, 0.41$. Error bars in fig. 5 are seed ranges. Bands are envelopes over each seed's plateau inputs (m_2, $m_{\ge 3}$, ϵ) $\times$ $q \in \{0.40, 0.47, 0.60\} \times \kappa_S \in \{0.38, 0.40, 0.43\}$, the ranges measured on the branch. We report two coverage numbers. *Point coverage*, the fraction of plotted seed medians inside the band, is 30%, 20% and 50% in panels (a)–(c) and 8–17% per held-out curve in (d). *Gain coverage*, whether the measured plateau-to-branch-end gain of a held-out setting lies inside its predicted band, is 4 of 15 SiLU settings (table 2). Both are low, so the misses are model error, not input uncertainty. *At the rates of fig. 4* ($\eta = 0.118 \to 3.645$; 16

initializations $\times$ 8 datasets $\times$ 8 noise draws) the closed form (a model prediction) gives a risk ratio of 0.851 (measured 0.845) and a noise-free part that changes by 2% (pooled bias $+0.0003$). This agreement of the total is partly a cancellation of errors: the model books the whole gain to the noise term $\sigma^2[\frac{n}{n_1}(1-\psi\nu^2)^{-1}-1]$, -0.029, against -0.021 measured in the terms that contain the noise draw, and it has no initialization term, whereas the initialization terms without the noise fall by 0.016. We therefore do not count the match of the totals as evidence for the model.

Held-out settings. All depth-2 SiLU cells of a separate sweep (widths 64–1024, $\sigma \in \{0.25, 0.5, 1\}$, 3 seeds, $T = 10^4$; section C.7) are scored with the same estimand, using only their plateau masses and the canonical κ_S, q (table 2). The measured gain is taken at the last rate at which all seeds interpolate, not at the best high-rate minimum. The predicted gain is 0.69–1.79 times the measured one (median 1.04), and the measured gain lies inside the band in 4 of 15 settings. The ratio rises with width, from 0.69–0.79 at width 64 to 1.30–1.79 at widths 512–1024, the expected signature of the missing initialization channel, which is largest at small width.

GELU (table 3 and fig. 11). We repeated the calibration on the canonical GELU net: GD at the rates of the SiLU branch plus $\eta = 12.1$, with clean twins, the same kernel-mass and ridge-efficiency pipeline, and the same rule (section C.7). The rule reproduces SiLU's constants ($q = 0.48$, $\kappa_S = 0.40$) and gives nearly the same ones for GELU: $q = 0.46$ and $\kappa_S = 0.42$. Along the GELU branch q stays at 0.36–0.60 up to $\eta = 16.8$ and κ_S at 0.39–0.45 once the cap binds, as for SiLU; only at the trainability edge ($\eta = 20.4$) does q collapse to 0.20, while the tail inflates ($m_{\geq 3} = 9.6$ against 7.2 on the plateau; the GELU tail is about twice SiLU's throughout). Re-scoring the 15 held-out GELU cells with GELU's own constants therefore changes little: ratios 0.62–3.21 (median 1.45; with SiLU's constants 0.64–3.25, median 1.48), 3 of 15 gains inside the band. The misfit is not a calibration problem. It is the same pattern as SiLU's, amplified: accurate at widths 64–128 (ratios 0.62–1.03), too large from width 256 on (1.45–3.21). On the canonical GELU net the closed form follows the measured risk up to $\eta \approx 7$ and then separates: beyond $\eta \approx 10$ the measured risk stops falling (0.56–0.69 of the plateau, not monotone) while the closed form continues to 0.44 (fig. 11a). GELU's branch thus reaches its right flank, where the tail grows and stops acting as a ridge, before the closed form's optimum; the closed form has no right flank. At $\sigma = 0.25$ and widths 512 and 1024 the last interpolating rate already lies on that flank, so the measured gain there is negative; since the gain is read at that rate, this does not show that GELU lacks a beneficial second dip. We therefore claim quantitative agreement for SiLU and, for GELU, up to the onset of its right flank.

Assumption diagnostics (table 1). On the same canonical runs (medians over interpolating seeds): *(A1)*: the noise field $\Delta f = f_{\text{noisy}} - f_{\text{clean}}$ (same seed and rate) against the extension of its training values through the noisy net's own final kernel, $\boldsymbol{k}(\boldsymbol{x})^\top \boldsymbol{K}^{-1}\Delta f(\boldsymbol{X})$, on 16,384 test inputs; relative error and energy ratio. *(A2), head*: participation ratio of the exact Stein-shell spectrum $\{\mu_p + \mu_q\}_{p\leq q}$ (μ: eigenvalues of $\boldsymbol{B}_1^\top \boldsymbol{B}_1$) divided by the 2080 degree-2 modes ($= 1$ for an isotropic head). *(A2), tail extension*: share of the noise field's test energy in Hermite degrees ≥ 3 ($= 0$ in the model). *(A2), design side*: the head's share of the fitted noise vector, $\pi_2 = \langle \Delta f_2(\boldsymbol{X}), \Delta f(\boldsymbol{X})\rangle / \|\Delta f(\boldsymbol{X})\|^2$ (denominator: the full n-dimensional training vector). Here Δf is regressed on all Hermite features of degree ≤ 2 on the test inputs, and Δf_2 is the part carried by the 2080 degree-2 coefficients. The model predicts $\pi_2 = m\nu/n$, i.e. $(n_1/n)\psi\nu$; the degree-≤ 1 coefficients give $\pi_{\leq 1}$, predicted $(d+1)/n = 0.063$ and measured 0.053–0.073. At the canonical plateau $m\nu/n = 0.844$, whereas $\psi\nu = 0.901$ is the share of the part orthogonal to the resolved degrees. A direct estimate of the tail's block of the kernel on the design is not feasible at $d = 64$: the head's design mass is about 20 times the tail's, and subtracting a Monte Carlo estimate of the head from two probe sets of 2^{15} and 2^{14} inputs leaves eigenvalues of ± 400 against a tail of 3. The head is not isotropic: its participation ratio is 0.72 of the 2080 degree-2 modes on the plateau and falls to 0.53 on the branch, whereas (A2) assumes 1. The closed form treats the head as isotropic, which is one source of its error (an anisotropic head concentrates the fitted noise in fewer directions). (A1) holds in energy, but its pointwise error grows from 4% to 20–26% along the branch; (A2) is approximate (the tail extends up to 24% of the fitted noise, 14–24% for $\eta \geq 7.3$); (A3) holds with $T_0 = 0.77\, m^{\text{pl}}_{\geq 3}$.

Where the theorem does not apply. *GELU*: see above; the closed form holds with GELU's own (SiLU-like) constants up to the onset of GELU's right flank, which comes earlier than for SiLU and

is not in the model. *ReLU*: the tail shrinks about threefold between the plateau and Dip 2 (A3), and the kernel is depleted at the training inputs, to 0.69–0.70 of its population value, so the network does not extend noise through its own kernel (A1). *Clean labels*: the noise channel is small (the closed form puts the onset at $\eta_{\rm on} \approx 16$), and the clean SiLU dip runs through the initialization terms (table 8). *tanh*: no even-degree mass.

η	A1 ext. error	A1 energy ratio	A2 head PR/m	A2 tail share of noise	π_2 meas.	π_2 pred.	A3 $m_{\geq 3}/m_{\geq 3}^{\rm pl}$	κ_S	ϵ
0.118	0.04	0.97	0.72	0.03	0.80	0.84	1.00	0.21	0.05
1.59	0.04	0.99	0.73	0.03	0.80	0.84	0.92	0.26	0.04
2.62	0.05	1.03	0.72	0.04	0.79	0.82	0.86	0.35	0.04
3.69	0.11	1.03	0.68	0.05	0.77	0.79	1.04	0.38	0.04
5.19	0.18	0.98	0.60	0.10	0.69	0.75	1.12	0.38	0.06
7.31	0.26	0.94	0.55	0.24	0.57	0.70	1.64	0.36	0.12
10.3	0.22	0.98	0.57	0.21	0.56	0.64	1.21	0.39	0.12
14.5	0.20	1.03	0.56	0.14	0.54	0.58	0.83	0.40	0.15
16.81	0.22	1.03	0.56	0.14	0.53	0.55	0.77	0.40	0.17
20.4	0.25	0.99	0.53	0.16	0.48	0.51	0.78	0.40	0.21

Table 1: **Assumption diagnostics for Theorem 4.1** on the canonical SiLU runs ($\sigma = 0.5$, width 256; medians over interpolating seeds). A1: relative error and energy ratio of the own-kernel extension of the noise field. A2: participation ratio of the head over the 2080 degree-2 modes; share of the noise field's test energy in degrees ≥ 3; the degree-2 share π_2 of the fitted noise vector (denominator n), measured and predicted ($m\nu/n$). A3: tail mass relative to the plateau. Also the first layer's cap share κ_S and the non-Stein degree-2 share ϵ.

σ	width	$\eta_{\rm end}$	$\mathcal{R}_{\rm pl}$	gain (meas.)	gain (pred.)	band	ratio
0.25	64	28.7	0.120	0.054	0.038	[0.034, 0.040]	0.69
0.25	128	28.7	0.096	0.039	0.033	[0.030, 0.035]	0.83
0.25	256	28.7	0.078	0.026	0.027	[0.025, 0.029]	1.02
0.25	512	28.7	0.068	0.018	0.024	[0.022, 0.026]	1.30
0.25	1024	28.7	0.064	0.016	0.023	[0.021, 0.024]	1.41
0.5	64	7.3	0.367	0.149	0.111	[0.094, 0.130]	0.74
0.5	128	14.5	0.293	0.157	0.148	[0.137, 0.160]	0.94
0.5	256	20.4	0.233	0.122	0.137	[0.130, 0.144]	1.12
0.5	512	14.5	0.203	0.081	0.106	[0.098, 0.113]	1.30
0.5	1024	14.5	0.190	0.076	0.100	[0.093, 0.107]	1.31
1	64	2.6	1.363	0.604	0.480	[0.402, 0.560]	0.79
1	128	3.7	1.044	0.476	0.494	[0.434, 0.551]	1.04
1	256	2.6	0.845	0.318	0.318	[0.280, 0.361]	1.00
1	512	2.6	0.727	0.157	0.280	[0.247, 0.317]	1.79
1	1024	2.6	0.681	0.199	0.265	[0.234, 0.302]	1.33

Table 2: **Out-of-sample test of Theorem 4.1** on the depth-2 SiLU sweep cells (full-batch GD, $T = 10^4$, 3 seeds). Inputs: plateau masses only; $\kappa_S = 0.40$ and $q = 0.47$ are fixed on the canonical net. Risk is per network (median over the 3 seeds). Gain: plateau risk minus the risk at the last rate $\eta_{\rm end}$ at which all seeds interpolate (not the best high-rate minimum). Band: envelope of the predicted gain over the seeds' plateau inputs, $q \in [0.40, 0.60]$ and $\kappa_S \in [0.38, 0.43]$. Ratio: predicted over measured gain.

B.8 The width laws of the μP failure model

The model of section 5 fits the two initialization variances and the squared-bias difference separately,

$$V_j(w) \simeq \frac{a_j}{w + c_j}\ (j \in \{L, H\}), \qquad B_H(w) - B_L(w) \simeq b_\infty - \frac{b_0}{1 + (w/w_B)^2},$$

and reconstructs the gap equation 6. Let P denote the interpolating plateau between the dips. We motivate each term and test each identity on a separate measurement, never on the gap itself.

(i) Width. Under μP a trained network fluctuates around a width-independent limit with variance $\propto 1/w$ (Bordelon & Pehlevan, 2023). The plateau obeys $V_P = a/(w+c)$; the shift c resums the first finite-width correction and keeps V_P finite for narrow networks. *Measured:* $V_P = 3.50/(w + 30)$

σ	width	η_{end}	$\mathcal{R}_{\text{pl}}$	gain (meas.)	SiLU constants pred.	SiLU constants ratio	GELU's own constants pred.	GELU's own constants band	GELU's own constants ratio
0.25	64	20.4	0.141	0.050	0.032	0.64	0.031	[0.025, 0.033]	0.62
0.25	128	20.4	0.111	0.033	0.029	0.87	0.028	[0.023, 0.029]	0.85
0.25	256	28.7	0.091	0.008	0.027	3.25	0.027	[0.024, 0.028]	3.21
0.25	512	28.7	0.079	-0.006	0.024	–	0.024	[0.022, 0.025]	–
0.25	1024	28.7	0.073	-0.004	0.023	–	0.022	[0.020, 0.023]	–
0.5	64	14.5	0.353	0.185	0.169	0.91	0.166	[0.135, 0.175]	0.89
0.5	128	14.5	0.272	0.133	0.139	1.04	0.136	[0.113, 0.141]	1.02
0.5	256	14.5	0.215	0.075	0.111	1.48	0.109	[0.092, 0.112]	1.45
0.5	512	14.5	0.182	0.053	0.095	1.78	0.093	[0.079, 0.096]	1.75
0.5	1024	10.3	0.169	0.037	0.079	2.15	0.077	[0.060, 0.080]	2.10
1	64	2.6	1.198	0.549	0.492	0.90	0.466	[0.297, 0.551]	0.85
1	128	2.6	0.893	0.367	0.395	1.08	0.376	[0.244, 0.423]	1.03
1	256	1.9	0.697	0.131	0.234	1.78	0.218	[0.132, 0.254]	1.66
1	512	1.9	0.590	0.096	0.200	2.09	0.187	[0.114, 0.218]	1.95
1	1024	2.6	0.546	0.115	0.246	2.15	0.235	[0.161, 0.262]	2.05

Table 3: **GELU held-out cells** scored as in table 2, with SiLU's constants ($q = 0.47$, $\kappa_S = 0.40$) and with GELU's own constants measured on the canonical GELU net ($q = 0.46$, $\kappa_S = 0.42$). Band: envelope over the seeds' plateau inputs and the ranges of the per-rate medians of q and κ_S on the GELU branch. "–": the measured gain is negative because the last interpolating rate lies on the right flank.

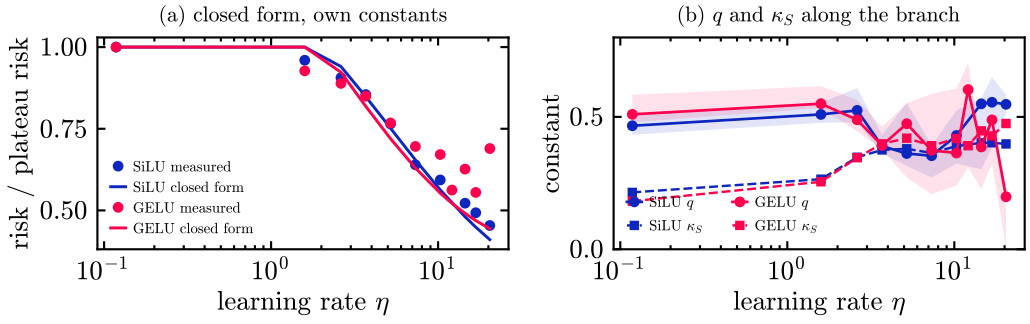

Figure 11: **Calibrating GELU.** Canonical nets (width 256, $\sigma = 0.5$, 4 seeds, $T = 2 \cdot 10^4$). (a) Risk relative to the plateau: seed medians (points) and the closed form of Theorem 4.1 from plateau inputs with each activation's own constants (lines). (b) Ridge efficiency q (circles; band: seed range) and cap share $\kappa_S = S_1/\lambda_{\max}$ (squares) along the branch. The constants agree between the activations; GELU departs from the closed form beyond $\eta \approx 10$, and its q collapses at the trainability edge.

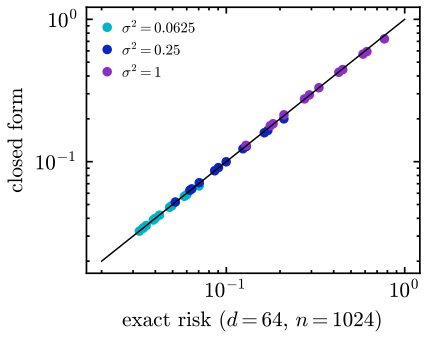

Figure 12: The closed form of Theorem 4.1(c) against the exact risk of the kernel interpolant on the actual design ($d = 64$, $n = 1024$, all 2080 degree-2 Hermite features), averaged in closed form over teacher rotations and noise.

(test MSE units; e.g. $V_P(64) = 0.0372$) to within 0.8% over widths 64–2048, and without refitting it predicts widths 4096 and 8192 to within 1.4% and 2.0%.

(ii) Large rate. At a fixed rate the stability ceiling, not the width, sets the regularization. Over the fitted range it keeps a width-independent fraction κ of the plateau variance and adds a roughly width-independent mean error, so $a_H = \kappa a$ and $c_H = c$. *Measured:* $\kappa = V_H/V_P = 0.53 \pm 0.03$ at every width; the fitted $a_H = 1.82 \approx \kappa a = 1.86$ and $c_H = 26 \approx c = 30$; the large rate's mean cost $B_H - B_P$ stays between 0.4 and $1.7 \cdot 10^{-3}$.

(iii) Early stopping. Under μP the GF path is width-independent: the mean error $\beta(\tau) = B(\tau) - B_P$ and the retained share of plateau variance $\rho(\tau) = V(\tau)/V_P$ do not depend on w. Stopping at τ costs $B_P + \beta(\tau) + \rho(\tau)V_P(w)$, so the optimal stopping time solves

$$p(\tau_L) = V_P(w), \qquad p(\tau) := -\beta'(\tau)/\rho'(\tau),$$

the mean error paid per unit of variance removed. GF fits high signal-to-noise structure first, so p falls along the path (checked at all 11 widths: 89–100% of steps decrease). Hence $B_L - B_P = \beta\big(p^{-1}(V_P(w))\big)$ is monotone and bounded in $\log V_P$, i.e. a sigmoid in $\log w$: flat while V_P exceeds p at the end of signal fitting, and vanishing, together with Dip 1, once V_P drops below the smallest p on the path. In a mode picture p is the signal-to-noise density of the mode being fitted, and the sigmoid is the cumulative signal over that density. With $V_P = a/(w + c)$, the logistic form $b_0/(1 + (p_0/V_P)^2)$, which is the case in which the logarithm of this density is logistic with scale $\frac{1}{2}$, equals $b_0/\big(1 + (p_0(w + c)/a)^2\big)$; it reduces to the Lorentzian $b_0/(1 + (w/w_B)^2)$ with $w_B = a/p_0$ only for $w \gg c$ ($c \approx 30$), so the Lorentzian in w is a large-width approximation of this argument. Because early stopping removes less and less variance as w grows, $a_L = a$ and $c_L > c$. *Measured:* $a_L = 3.77 \approx a = 3.50$; the selected stopping time is $\tau_L \approx 230$–250 up to width 192 and rises to 834 at width 2048; $B_L - B_P$ is the Lorentzian ($R^2 = 0.999$, $w_B = 241$). Calibrated on a single width (any of 512–2048), with no width-sweep fit, (iii) predicts the exponent (2.2–2.35 against 2), the corner ($w_B = 195$–204 against 241), the level ($b_0 = 7.4$–$7.6 \cdot 10^{-3}$ against $5.9 \cdot 10^{-3}$), $B_L - B_P$ to within 15% at widths 192–512, V_L/V_P to within 0.06 at every width, and the end of Dip 1 at width 1400–1900 (measured between 1536 and 2048). Narrow networks traverse the path more slowly than predicted ($\tau_L \approx 230$–250 against 126–200) and pay 20–27% less mean error, so b_0 is predicted only to about 25%.

With a, c, κ fixed by these separate measurements the constrained law still has $R^2 = 0.974$ on the gap (free fit: 0.993) and a lower root at width 95 (measured crossover ≈ 97). The fitted constants are in table 4.

Wider networks: a width window, not an asymptote. The law is a finite-range description of widths 64–2048, and we make no claim about $w \to \infty$. Within that range Dip 2's advantage peaks at about $2.5 \cdot 10^{-3}$ (4–5% of the risk) around $w \approx 200$–256 and shrinks to $0.29 \cdot 10^{-3}$ at $w = 2048$. Extrapolated, the fitted law has an upper root at $w = 2997$. We checked widths 4096 and 8192 with 256 fresh initializations, selecting the large rate and the stopping time on validation risk of disjoint pilot seeds over extended searches (fig. 13):

- At $w = 4096$ the validation-selected Dip 1 wins ($R_H - R_L = +0.088 \cdot 10^{-3}$, simultaneous 95% interval $[0.071, 0.104] \cdot 10^{-3}$) at the validation-selected $\eta = 4.83$. On the 16 pilot seeds, however, $\eta = 2$ had lower test risk than the low branch ($-0.044 \cdot 10^{-3}$), and validation and test disagree by ≈ 3 standard errors at this resolution, so which branch wins at its best rate is not settled.
- At $w = 8192$ the validation-selected large rate falls to $\eta = 0.18$, on the plateau, and the two branches nearly coincide: the gap is $+2.6 \cdot 10^{-7}$ (simultaneous 95% interval $[0.66, 4.5] \cdot 10^{-7}$, which excludes zero but is more than 300 times smaller than at $w = 4096$) and the variances differ by $-9.7 \cdot 10^{-8}$.

The fitted constant b_∞ is therefore not an asymptote: the bias gap vanishes once the large rate stops regularizing. The supported statement is a width window: the optimal rate jumps from Dip 1 to Dip 2 between widths 64 and 128, stays there up to width 2048, and the advantage becomes negligible or reverses by width 4096. Since $B = R - V$, the components sum to the measured gap by construction.

parameter	meaning	law	fit
a_H	high-branch variance amplitude	$V_H = a_H/(w + c_H)$	1.8249
c_H	high-branch width offset		26.20
a_L	low-branch variance amplitude	$V_L = a_L/(w + c_L)$	3.7667
c_L	low-branch width offset		246.3
b_∞	large-width level of the bias-gap fit	$B_H - B_L = b_\infty - b_0/(1 + (w/w_B)^2)$	$5.879 \cdot 10^{-4}$
b_0	narrow-width bias advantage of Dip 2		$4.809 \cdot 10^{-3}$
w_B	half-transition width of the bias gap		238.0
roots of equation 6: lower / upper (extrapolated)			101.2 / 2997
measured crossover (log-linear interpolation of the gap)			97.3
in-sample R^2: V_L / V_H / ΔB / ΔR			0.971 / 0.993 / 0.992 / 0.993
leave-one-width-out gap RMSE (winner correct)			$3.6 \cdot 10^{-4}$ (11/11)

Table 4: **Fitted constants of the μP failure model** (fig. 6; clean ReLU, $n = 1024$, $T = 10^4$, 16 initializations, widths 64–2048). Variances are fitted by squared relative residuals, the bias gap by squared absolute residuals; the crossover is a root of the reconstructed gap and is never fitted. These are phenomenological parameters valid on the fitted range; b_∞ is not an infinite-width limit (fig. 13).

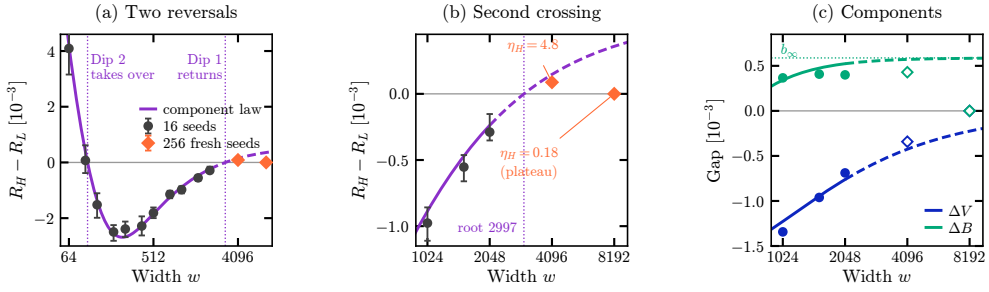

Figure 13: **Wider networks.** Same problem as fig. 6. (a) Risk gap $R_H - R_L$ across widths. Black: 16 initializations, 95% paired bootstrap intervals. Orange: 256 fresh initializations at $w = 4096$ and 8192 with extended rate and stopping-time searches, simultaneous 95% intervals. The curve is equation 6, fitted on $w \leq 2048$ (solid) and extrapolated (dashed); dotted lines mark its roots. (b) Zoom on the second crossing, with the selected large rate η_H. (c) Variance and bias gaps at large width against their fitted laws; at $w = 8192$ both nearly vanish instead of approaching b_∞, so the law does not extrapolate.

C EXPERIMENTAL DETAILS

All runs use a constant learning rate, no momentum (unless stated for Adam/Muon), no weight decay (unless stated), no gradient clipping, no warm-up, no schedule, no dropout and no data augmentation, and report the final iterate after exactly T updates. Diverged runs (non-finite parameters or a training loss above a blow-up guard) are recorded as failures. Unless a protocol below states otherwise, a mean over seeds is shown only if every seed is finite, and failed seeds are not replaced. The exceptions are stated where they occur: the capped-GD study and the tests of Theorem 4.1 summarize over interpolating seeds only, and the capped-GD study adds seed batches for native GD until three seeds interpolate (section C.5); every attempt is reported in the fit counts. Test risks are always measured against clean targets on held-out data.

C.1 PARAMETRIZATION AND ARCHITECTURES

Teacher–student MLP. The student with L hidden layers of width w and input dimension d is

$$\boldsymbol{z}_1 = \sqrt{w/d}\,\boldsymbol{W}_1\boldsymbol{x}, \qquad \boldsymbol{z}_l = \boldsymbol{W}_l\,\phi(\boldsymbol{z}_{l-1})\ (2 \le l \le L), \qquad f_{\boldsymbol{\theta}}(\boldsymbol{x}) = w^{-1/2}\,\boldsymbol{v}^\top\phi(\boldsymbol{z}_L),$$

without biases, with all raw weights drawn i.i.d. from $\mathcal{N}(0, 1/w)$ and a single learning rate η for all raw parameters. In effective-weight coordinates the initialization variances are $1/d$, $1/w$ and $1/w^2$ and the effective SGD rates are $(w/d)\eta$, η and η/w for the input, hidden and readout layers. In the (a, b, c) notation of Yang & Hu (2021) ($\boldsymbol{W}^l = w^{-a_l}\boldsymbol{\omega}^l$, $\boldsymbol{\omega}^l_{ij} \sim \mathcal{N}(0, w^{-2b_l})$, learning rate ηw^{-c}) this is $a = (-\frac{1}{2}, 0, \frac{1}{2})$, $b = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, $c = 0$, i.e. μP for SGD, with the fixed input dimension entering only as a constant. At the canonical width $w = 256$ and $d = 64$ the forward multipliers are $(2, 1, 1/16)$. Activations are ReLU, SiLU, GELU or tanh; depth is two hidden layers unless stated. Where stated, a first-layer bias is implemented by appending a constant input κ to $\boldsymbol{x}$ with a zero-initialized weight column, and *lazy* scaling by α trains $\alpha(f_{\boldsymbol{\theta}} - f_{\boldsymbol{\theta}_0})$ with model step η/α^2, so that $\eta\lambda$ remains the stability variable (Chizat et al., 2018). Training is full-batch GD on the half MSE in float32 with TF32 disabled (float64 where stated), metric reductions in float64. Analytic batched gradients were checked against autograd (maximum error $\le 10^{-9}$ in float32, $7 \cdot 10^{-18}$ in float64), batched updates against independent `torch.optim.SGD` lanes, and CUDA-graph replay against eager updates.

ResNet-18 (CIFAR-10). CIFAR ResNet-18 with stage widths 64/128/256/512 (11,173,962 parameters, affine BatchNorm), in μP via the official `mup` package with base width 16 and target width 64 (multiplier $m = 4$). Convolutions use fan-in He initialization; the readout input is divided by m. `MuSGD` gives the input convolution, the BatchNorm affine parameters and the raw readout the rate 4η, and hidden convolutions and the readout bias the rate η. BatchNorm running statistics are kept per trajectory and are not updated by evaluation. Training uses BF16 autocast with float32 parameters, gradients and optimizer state.

Transformer (WikiText). A causal pre-norm transformer with 4 blocks, width 256 (base width 128, $m = 2$), 4 heads, feed-forward width 1024, ReLU MLPs, learned token and position embeddings, a byte vocabulary of 256, and parameter-free RMSNorm ($\epsilon = 10^{-5}$, no gain or bias) in place of every LayerNorm (3,319,040 parameters). μP follows the official implementation: embedding and raw readout initialization SD 0.02, hidden matrices $0.02/\sqrt{m}$, zero-initialized query weights, attention scale $\sqrt{d_{\text{head,base}}}/d_{\text{head}}$ (the μP $1/d_{\text{head}}$ rule calibrated at the base width), readout input divided by m. With `MuSGD`, embeddings, hidden biases and the raw readout use 2η and hidden matrices η. Training is float32 with compensated summation of the SGD updates.

C.2 DATA

Teacher–student regression. Inputs $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_{64})$; $n = 1024$ training, 4096 validation and $16{,}384$ test inputs (8,192 test inputs in the crossover study of section 5). The teacher is a fixed bias-free network with two hidden layers of width 256 and the student's activation, $f_*(\boldsymbol{x}) = \frac{1}{16}\boldsymbol{v}_*^\top\phi\big(\frac{1}{16}\boldsymbol{W}_{*2}\phi(\frac{1}{8}\boldsymbol{W}_{*1}\boldsymbol{x})\big)$ with all entries of $\boldsymbol{W}_{*1} \in \mathbb{R}^{256\times 64}$, $\boldsymbol{W}_{*2} \in \mathbb{R}^{256\times 256}$, $\boldsymbol{v}_* \in \mathbb{R}^{256}$ i.i.d. $\mathcal{N}(0, 1)$. Training labels are $y_i = f_*(\boldsymbol{x}_i) + \sigma z_i$ with $z_i \sim \mathcal{N}(0, 1)$; $\sigma = 0.5$ unless stated. For the ReLU teacher $\mathbb{E}f_*^2 = 0.2467$ (the risk of the zero predictor) and the signal-to-noise ratio at

$\sigma = 0.5$ is 0.70 (SiLU: 0.48). Teacher, inputs and noise are drawn from separate recorded random streams and are shared, bit for bit, by all rates, seeds and arms of an experiment, so that comparisons across rates are paired; seeds vary only the student initialization, except in the bias–variance study (section C.6), where training sets and noise draws are resampled. The larger training sets of fig. 3d are nested prefixes of one extension of the original set, and the original 1,024 inputs, targets and noise values are preserved exactly.

CIFAR-10. All 50,000 training and 10,000 test images; minibatches of 128 images sampled with replacement; the loss is cross-entropy. The training curves of fig. 2b evaluate a fixed probe of 2,048 training images.

WikiText. A byte-level next-token task on WikiText-103. The training set consists of 64 fixed windows of 129 bytes (8,256 source bytes) drawn without replacement from the training text; in each window the first 32 prediction positions are context only, leaving $64 \times 96 = 6{,}144$ scored training targets, all with distinct causal contexts (checked exhaustively; without the context-only prefix, identical short prefixes with different next bytes would make exact interpolation impossible). Minibatches contain 32 windows sampled with replacement. The held-out risk is evaluated on nonoverlapping windows of the full WikiText-103 test stream with the same 32-position context-only mask. The set is interpolatable: at the plateau rate, 62 of 64 clean trajectories of the decomposition study (section C.6) fit every training target.

C.3 Learning-rate sweeps of fig. 2

- *(a) MLP.* ReLU student and teacher, width 256, two hidden layers, $\sigma = 0.5$, full-batch GD, 40 learning rates from 10^{-7} to 20, 3 initializations, $T = 4 \cdot 10^5$ updates with 2,083 evaluation times per trajectory (240 trajectories with the SiLU twin).
- *(b) CIFAR-10.* Width-64 μP ResNet-18, SGD with batch size 128, 33 learning rates from 10^{-5} to 50, 2 initializations (each with its own minibatch stream, shared across rates), $T = 10^5$ updates (256 epochs); clean labels.
- *(c) WikiText.* The transformer above with `MuSGD`, 97 learning rates from 10^{-8} to 50 (dense spacing 0.025 on $[0.15, 1.2]$), 5 initializations, clean targets, $5 \cdot 10^4$ updates; the top panel is the exact slice at $T = 5{,}000$. At this budget the plateau interpolates while Dip 1 and the high-rate valley do not.

Heatmaps show the seed-mean test loss over (η, T) on the measured grid; the dashed line in the bottom panels is constant ηT through Dip 1, and the dotted line in (a) is $\eta_c = 2/\Lambda_F$ computed from the small-rate interpolant (section C.8).

C.4 Controls of fig. 3

Unless stated: ReLU student and teacher of width 256, $d = 64$, $n = 1024$, $\sigma = 0.5$, full-batch GD for $T = 2 \cdot 10^4$ updates, 64 log-spaced rates in $[10^{-4}, 14]$ (12.5 per decade), 8 initializations; error bars are $\pm$SEM over initializations. No trajectory diverged on this grid for $k = 1$. The first minimum is 0.0785 at $\eta = 3.56 \cdot 10^{-3}$, the plateau 0.1792, and the high-rate minimum 0.1252 at $\eta = 9.61$.

(a) Gradient flow at matched time. GF depends only on the integrated time, so one trajectory per initialization covers every rate: we read it off at $t = \eta T$ for all 64 rates, up to $t = 2.8 \cdot 10^5$. We integrate with explicit Euler steps $h = 0.125$ (plotted) and $h = 0.25$ from the same initializations; the two agree to $5.8 \cdot 10^{-5}$ MSE for $\eta > 0.5$ (largest difference $1.7 \cdot 10^{-3}$ at $t \approx 2$, from rounding the evaluation time to whole steps). GF matches GD through Dip 1 (0.0785 for both) and then stays on the plateau: at the time of the high-rate minimum it has risk 0.1792, against 0.1252 for GD.

(b) Matched-time substepping. For each rate η and $k \in \{1, 2, 4, 8, 16\}$ the arm runs kT updates at rate η/k from the same initialization, so ηT is identical across arms. Below $\eta \approx 0.5$ the arms agree to within $2.5 \cdot 10^{-3}$ and Dip 1 is unchanged (0.0785 at $\eta = 3.56 \cdot 10^{-3}$ for $k = 1$ and $k = 16$). At the high-rate minimum the arms separate monotonically (table 5); the effect is ≈ 23

SEM, and at $k = 16$ the step $9.61/16 = 0.60$ is below $\eta_c \approx 0.62$ and the risk equals the plateau. Near the stability boundary ($\eta \gtrsim 12$) the ordering reverses because the $k = 1$ arm itself approaches divergence; we do not interpret that region.

arm	test MSE at $\eta = 9.61$	paired difference from $k = 1$ [95% CI]
$k = 1$	0.1252	—
$k = 2$	0.1411	$+0.0158$ $[0.0114, 0.0193]$
$k = 4$	0.1573	$+0.0320$ $[0.0271, 0.0360]$
$k = 8$	0.1679	$+0.0426$ $[0.0378, 0.0468]$
$k = 16$	0.1792	$+0.0540$ $[0.0489, 0.0583]$

Table 5: Substepping at the high-rate minimum. Differences are paired by initialization; intervals are 95% bootstrap intervals over the 8 initializations (4,000 resamples).

(c) Fixed kernel. We use the infinite-width NTK Θ_∞ of the same two-hidden-layer bias-free ReLU architecture in the NTK parameterization, computed with the arc-cosine recursion $\Sigma_0(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'/d$, $\Sigma_h = \mathbb{E}[\phi(u)\phi(v)]$, $\dot{\Sigma}_h = \mathbb{E}[\phi'(u)\phi'(v)]$ with (u, v) Gaussian of covariance given by Σ_{h-1}, and $\Theta_\infty = \Sigma_2 + \dot{\Sigma}_2(\Sigma_1 + \dot{\Sigma}_1\Sigma_0)$; empirical kernels of NTK-parametrized networks converge to it at rate width$^{-1/2}$ (relative Frobenius error $0.095, 0.047, 0.023$ at widths 1024, 4096, 16,384). Kernel GD on the half MSE moves the residual as $\boldsymbol{r}_t = (\boldsymbol{I} - \frac{\eta}{n}\Theta_\infty)\boldsymbol{r}_{t-1}$, so after T steps mode i carries the filter $1 - (1 - \eta\lambda_i/n)^T$; it is stable for $\eta < 2n/\lambda_{\max} = 11.41$ ($\lambda_{\max} = 179.4$). The initial output of the infinite-width network is a Gaussian process $f_0 \sim \mathcal{GP}(0, \Sigma_2)$, independent of the labels; the plotted curve is the exact expectation of the risk over f_0 (hence no error bars), for the same noise draw as the network. No ridge is used: the only regularization is the finite-time filter. The kernel has a single minimum, 0.1186 at $\eta = 0.087$, where the network is on its plateau, and then converges to the rate-independent interpolant (0.1271) until it diverges. Replacing Θ_∞ by each width-256 network's empirical NTK at its own initialization gives the same single-minimum shape (best 0.0811 at $\eta = 0.041$, converged 0.1211, stable up to $\eta \approx 11.2$).

Relation to the fixed-kernel baseline. At convergence the fixed kernels are about as good as the network's Dip 2 on this noisy problem: 0.1271 for the infinite-width NTK with its random initial output and 0.1211 for each network's own initial NTK, against 0.1252 at Dip 2 and 0.1792 on the plateau. So the high-rate branch does not beat the lazy predictor here. What it undoes is the damage that feature learning does on the plateau, which is 48% worse than the network's own initial kernel. Three observations locate the difference. (1) Dip 2 is not a return to the initial kernel: its NTK has moved by 94% of $\lambda_{\max}(\boldsymbol{K}_F)$ (section B.2). (2) In the SiLU analysis (section D.4) the plateau's degree-2 head is inflated about $265\times$ from initialization while the tail grows about $138\times$, which lowers the effective ridge of Theorem 4.1; the ceiling lowers the head again. (3) Against the network's own initial kernel, feature learning helps where the network early-stops: Dip 1 (0.0785) beats the best early-stopped initial NTK (0.0811). Comparisons with the infinite-width NTK depend on its initial-output convention. With its random initial output its best early-stopped risk is 0.1186 and its converged clean-label risk 0.0927; with a zero initial output they are 0.0691 and 0.0480, better than the network's Dip 1 (0.0785) and its best clean high-rate risk (0.0526). Our network's initial output is small (SD 0.022 on the training inputs). We therefore do not claim that the network beats kernel regression on this teacher; the claim is about the rate dependence, which a fixed kernel does not have (Theorem B.2(ii)(a)).

(d) Training-set size. Width-64 ReLU student (two hidden layers; $P = 8{,}256$, $D = w(w + d - 1) = 8{,}128$) and the fixed width-256 ReLU teacher, $\sigma = 0.5$, $n \in \{256, 512, 1024, 4096, 16384, 65536, 98304\}$ (n/D from 0.03 to 12.1), full-batch GD for $T = 2 \cdot 10^4$ updates, 124 learning rates from 10^{-7} to 64, 3 initializations shared across n. For $n > 1024$ each update accumulates exact chunked full-batch gradients before a single step. A rate is plotted only if all three seeds are finite and have test MSE ≤ 1; crosses mark rates where any seed diverged. The full (η, T) maps are in fig. 24.

(e) Stability cap (SiLU). Canonical SiLU problem, $T = 2 \cdot 10^4$; native GD at η versus capped GD at $h = \eta/2$ with target $c = 2/\eta$ (section C.5); median over interpolating seeds (training MSE $< 10^{-6}$), bars min–max. At every plotted capped point $h\lambda_{\max}(H_L) < 2$ at all recorded checkpoints

of all seeds, and the training loss never exceeds its initial value. "Native: no fit" marks rates at which no native seed out of 20 interpolates.

(f) Label-noise control. The $k = 1$ sweep repeated with the same grid, horizon, inputs, initializations and noise vector z, changing only σ. The gain Δ is the paired difference between the plateau maximum and the best rate with $\eta \geq 1$ (table 6). Fitting $a + b\sigma^2$ to $\sigma = 0.25$ and 0.5 predicts 0.0028 at $\sigma = 0$ against 0.0049 measured, so about 91% of the gain at $\sigma = 0.5$ is noise-driven; the noiseless gain is resolved at ≈ 16 SEM.

σ	plateau	high-rate minimum	paired gain Δ
0	0.0575 at $\eta = 0.22$	0.0526 at $\eta = 7.96$	-0.0049 ± 0.0003
0.25	0.0869 at $\eta = 0.11$	0.0712 at $\eta = 11.6$	-0.0156 ± 0.0005
0.5	0.1792 at $\eta = 0.39$	0.1252 at $\eta = 9.61$	-0.0540 ± 0.0026

Table 6: Label-noise control of fig. 3f (clean test MSE; ±SEM over 8 initializations).

C.5 Capped gradient descent

For a target rate η we run GD with the stable update rate $h = \eta/2$ and constrain the top eigenvalues of $\boldsymbol{K}/n$ to $c = 2/\eta$, the level that native GD at rate η must satisfy at a converged interpolant (Theorem B.2(i)(b)). If the constraint held exactly, $h\lambda_{\max}(\boldsymbol{K}/n) \leq 1$, far below the GD threshold 2; because the full loss Hessian also contains a residual term, we verify stability independently (below).

Update rule. Let $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$ and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ be cached orthonormal top eigenvectors of $\boldsymbol{K}(\boldsymbol{\theta}_t)/n$. For each cached vector we recompute the Rayleigh quotient differentiably, $\lambda_j = \|\boldsymbol{J}_t^\top \boldsymbol{v}_j\|^2/n$, with normal $\boldsymbol{a}_j = \nabla_{\boldsymbol{\theta}}\lambda_j$ (the eigenvector held fixed). Mode j is *active* if $\lambda_j \geq 0.97c$. The update direction is the Euclidean projection of the gradient onto a polyhedron, the intersection of one half-space per active mode,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - h\,\boldsymbol{d}_t, \qquad \boldsymbol{d}_t = \arg\min_{\boldsymbol{d}} \|\boldsymbol{d} - \boldsymbol{g}_t\|^2 \quad \text{s.t.} \quad \boldsymbol{a}_j^\top \boldsymbol{d} \geq \frac{\gamma}{h}\,[\lambda_j - c]_+ \text{ for all active } j, \tag{12}$$

with $\gamma = 0.5$. By the KKT conditions $\boldsymbol{d}_t = \boldsymbol{g}_t + \sum_j \mu_j \boldsymbol{a}_j$ with multipliers $\mu_j \geq 0$ solving a small linear complementarity problem. To first order the step changes each *active* cached Rayleigh quotient by $-h\boldsymbol{a}_j^\top \boldsymbol{d}_t \leq -\gamma[\lambda_j - c]_+$: an active quotient below the cap does not increase at first order, and an overshoot contracts by the factor $1 - \gamma$. This is a first-order statement about the cached quotients: inactive modes may increase, the cached vectors need not remain exact eigenvectors between refreshes, and finite steps are not guaranteed to be feasible. The actual curvature is therefore checked separately (below). With no active mode the update is plain GD at rate h. We use two implementations.

- *Smooth activations (SiLU, GELU; fig. 3e).* $k = 4$ modes, refreshed by subspace iteration with a residual check (exact eigendecomposition as fallback); the multipliers are found by enumerating all 2^k active sets in one batched solve. float64 gradients. For $\eta \leq 0.3$ the cap is inactive and the run is plain float64 GD, with the kernel checked at every measurement (the shortcut is rejected if $\lambda_{\max}$ reaches $0.85c$).
- *ReLU.* A relaxed, cached single-mode controller ($k = 1$): the top eigenvector is refreshed every 2 updates when $\tilde{\lambda}/c \geq 0.85$, every 20 updates away from the boundary and every 100 updates in the inactive low-rate regime; if the cached estimate is below $0.85c$ the step is plain GD, otherwise $\mu = [(\gamma[\lambda - c]_+/h - \boldsymbol{a}^\top \boldsymbol{g})/\|\boldsymbol{a}\|^2]_+$.

Both hold the state fixed once the training MSE is below 10^{-12} and $\|\boldsymbol{g}_t\| < 10^{-9}$.

Diagnostics. The cap acts on the Gauss–Newton part $\boldsymbol{K}/n$ only. We therefore evaluate the full loss Hessian $H_L = \boldsymbol{J}^\top \boldsymbol{J}/n + \frac{1}{n}\sum_i (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ independently, in float64 by residual-checked Lanczos, at every recorded checkpoint (every update to 200, every 5 to 1,000, every 50 to 10^4, every 500 thereafter). A target rate counts as *verified stable* if $\max_{t,s} h\lambda_{\max}(H_L(\boldsymbol{\theta}_{t,s})) <$

2 over all recorded checkpoints t and seeds s; this certifies the recorded checkpoints, not every intermediate update or segment. It counts as free of a catapult if $\max_{t>0,s} \mathcal{L}_{t,s}/\mathcal{L}_{0,s} \le 1$, with the loss tracked at every capped update for SiLU and at the recorded checkpoints for ReLU. A loss that never exceeds its initial value need not decrease monotonically. *Edge-of-stability occupancy* is the fraction of checkpoints with $|h\lambda_{\max}(H_L)/2 - 1| \le 0.05$.

Protocol and results. Same data, teacher and initialization streams as native GD; three arms per rate: native GD at η, capped GD at $h = \eta/2$, uncapped GD at $h = \eta/2$ (the matched control for the smaller step). Seeds 0–4 are run first; for native GD further batches of five are added until three interpolate (training MSE $< 10^{-6}$) or 20 have been attempted, and every attempt is retained; grid refinement used training convergence only. Budget $5 \cdot 10^4$ updates; fig. 3e and fig. 14 show the $2 \cdot 10^4$ horizon. For target rates $\eta \ge 0.30$ every capped run has converged by then (identical values at $2 \cdot 10^4$ and $5 \cdot 10^4$); at $\eta = 0.118$ no capped run interpolates within $2 \cdot 10^4$ updates, so that rate has no plotted point. Table 7 gives SiLU. At every capped rate from 0.30 to 40, $h\lambda_{\max}(H_L)$ stays below 2 at all recorded checkpoints (maximum 1.81 at $\eta = 40$) and the training loss never exceeds its initial value; the best capped risk is 0.0967 at $\eta = 32$, 58.5% below the plateau 0.2333, where native GD fits in only 3 of 20 attempts and is lost entirely from $\eta = 36$. At $\eta = 48$ and 64 single checkpoints in the first 25 updates exceed 2 (2.19, 2.35), so those rates are not counted. ***ReLU*** (40 target rates, 3 seeds, $5 \cdot 10^4$ updates, fig. 15): at $\eta = 8.87$ the capped run reaches 0.1080 against 0.1229 for native GD and a plateau of 0.1806, with $\max h\lambda_{\max}(H_L) = 1.668$ and $\max \mathcal{L}_t/\mathcal{L}_0 = 0.847$; over $\eta \in [5.63, 10.49]$ the capped median is 0.1331–0.1016 with $\max h\lambda_{\max}(H_L) \in [1.145, 1.928]$. The ReLU controller is not uniformly feasible over the whole grid: 1.84% of all recorded cells exceed $h\lambda_{\max}(H_L) = 2$ (maximum 4.21) and 40.2% violate the nominal $\eta\lambda_{\max}(\boldsymbol{K}/n) \le 2$ (maximum 7.38), so for ReLU we claim non-necessity only on the verified band, and only at the resolution of the recorded checkpoints.

target η	native GD	fits	capped, $h = \eta/2$	$\max h\lambda_{\max}(H_L)$	uncapped, $h = \eta/2$	fits
0.30	0.2333	5/5	0.2334	0.28	0.2334	5/5
3.64	0.1977	5/5	0.1960	0.99	0.2212	5/5
8.38	0.1499	3/5	0.1520	1.00	0.1930	5/5
11.99	0.1297	5/5	0.1294	1.06	0.1682	5/5
16.81	0.1168	5/5	0.1111	1.16	0.1556	4/10
20.44	0.1102	4/5	0.1036	1.26	0.1373	5/5
27.65	0.1303	4/5	0.0972	1.47	0.1248	5/5
32	0.1174	3/20	0.0967	1.54	0.1190	4/5
36	—	0/20	0.0989	1.69	0.1095	4/5
40	—	0/20	0.1028	1.81	0.1076	4/5

Table 7: **Capped GD on SiLU** (canonical problem, $\sigma = 0.5$; median test MSE over interpolating seeds at $T = 2 \cdot 10^4$; "fits": interpolating/attempted seeds). All listed capped runs interpolate (5/5), and their training loss never exceeds its initial value. The uncapped control at $h = \eta/2$ is much worse than capped GD at the same step, so the gain is not an effect of the smaller step.

C.6 Bias–variance decomposition

Estimand. For each endpoint (Dip 1, plateau, Dip 2) we train one network for every combination of the levels of the random factors $\mathcal{S} \subseteq \{\mathsf{I}, \mathsf{D}, \mathsf{N}, \mathsf{M}\}$ present in a setting (initialization, training set, label-noise draw, minibatch stream), in a fully crossed balanced design, and evaluate its predictor F on a fixed probe set. On the empirical product measure over the levels, define for $A \subseteq \mathcal{S}$

$$g_A = \sum_{B \subseteq A} (-1)^{|A \setminus B|}\, \bar{F}_B, \qquad \bar{F}_B = \text{average of } F \text{ over the factors not in } B,$$

so that $F = \bar{F}_\varnothing + \sum_{A \neq \varnothing} g_A$ with mutually orthogonal effects (each g_A averages to zero over any one of its factors). Hence the *exact* identity

$$\overline{\|F - f_*\|^2_{\text{pop}}} = \|\bar{F}_\varnothing - f_*\|^2_{\text{pop}} + \sum_{\varnothing \neq A \subseteq \mathcal{S}} \overline{\|g_A\|^2_{\text{pop}}},$$

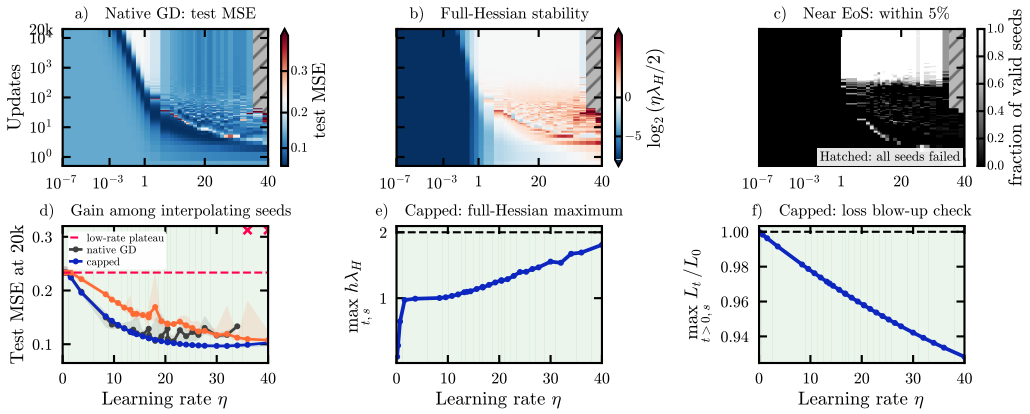

Figure 14: **Native versus capped GD for SiLU** (canonical problem, $\sigma = 0.5$, $T = 2 \cdot 10^4$). Top: native GD across rate and time, (a) test MSE, (b) full-Hessian stability ratio $\eta\lambda_{\max}(H_L)$, (c) edge-of-stability occupancy. High-rate native runs pass through an irregular instability-dominated phase and then spend most of the time at the stability threshold; hatched cells: every attempted seed blew up. Bottom: (d) final test MSE of native and capped GD (medians, bars: range over interpolating seeds); (e) maximum capped-run $h\lambda_{\max}(H_L)$ over all checkpoints and seeds; (f) maximum post-initialization training loss relative to the initial loss. In the shaded band capped GD keeps the Dip-2 gain while $h\lambda_{\max}(H_L)$ stays below the stability threshold at all recorded checkpoints and the loss never exceeds its initial value.

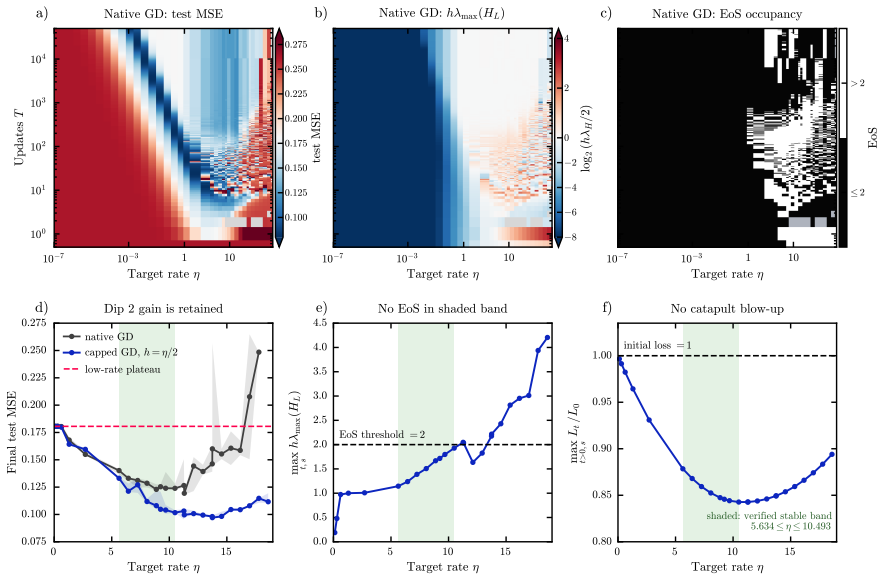

Figure 15: **Native versus capped GD for ReLU** (canonical problem, 40 target rates, 3 seeds, $T = 5 \cdot 10^4$). Top: native GD test MSE, $h\lambda_{\max}(H_L)$ and edge-of-stability occupancy over (η, T). Bottom: final test MSE of native and capped GD, the capped runs' maximum $h\lambda_{\max}(H_L)$, and their maximum $\mathcal{L}_t/\mathcal{L}_0$. The shaded band is the range verified at all recorded checkpoints.

the empirical version of equation 3: pooled bias^2 plus all main effects and interactions. It closes to the directly measured mean risk to $\leq 6 \cdot 10^{-16}$ (MLP), $4 \cdot 10^{-16}$ (CIFAR-10) and at numerical precision (WikiText). For the MLP $\|\cdot\|^2_{\text{pop}}$ is the mean square over 8,192 fixed Gaussian probes; for CIFAR-10 and WikiText F is the softmax probability vector and $\|\cdot\|^2_{\text{pop}}$ is the multiclass Brier risk (sum over classes, mean over 10,000 test images or 8,192 held-out byte positions) against the clean one-hot target. The bars of fig. 4 show pooled bias, the initialization, dataset, label-noise and minibatch main effects, the dataset×label-noise interaction, and all other interactions summed.

Uncertainty. Whiskers are 95% percentile intervals of the *total* risk from a product-factor bootstrap that resamples the levels of every factor independently, with the same resampled indices at all three endpoints (2,000 resamples for the MLP, 5,000 for CIFAR-10). The plug-in components are empirical quantities of the declared finite design; as estimates of their population counterparts they carry finite-level bias (e.g. with 4 levels a main effect absorbs a quarter of the higher-order variance), and their percentile intervals need not contain the plug-in value. We therefore report intervals only for the total and state component shares as properties of the design.

Designs.

- *MLP* (canonical problem, width 256, two hidden layers, full-batch GD, float64 metrics). 16 initializations × 8 independently sampled training sets of $n = 1024$ Gaussian inputs (× 8 label-noise draws at $\sigma = 0.5$ for noisy labels), with the teacher fixed: 128 (clean) or 1024 (noisy) networks per activation and endpoint, the same levels at all endpoints. Clean: plateau $\eta = 0.3$ and Dip 2 $\eta = 10.49$ (ReLU, $T = 10^4$) or 16.81 (SiLU, $T = 4 \cdot 10^4$, because the SiLU plateau twin is not fully fitted at 10^4). Noisy: plateau $\eta = 0.1181$ and high-rate endpoint $\eta = 8.064$ (ReLU) or $\eta = 3.645$ (SiLU), $T = 2 \cdot 10^4$; these are the largest rates on the native noisy branches at which all lanes interpolate, and for SiLU $\eta = 3.645$ lies on the descending branch, not at its minimum ($\eta \approx 20$). The "Dip 1" row of every MLP panel is a low-rate reference, $\eta = 1.73 \cdot 10^{-4}$ for $4 \cdot 10^5$ updates (flow time 69, at Dip 1's flow time $\tau^\star$), not a separately optimized matched-budget minimum; it is strongly underfit (training MSE ≈ 0.04 clean, 0.22–0.25 noisy). All clean plateau and Dip-2 lanes interpolate below training MSE 10^{-8}; for noisy ReLU at Dip 2, 1003 of 1024 lanes are below 10^{-8} and the maximum is $1.3 \cdot 10^{-3}$; no finite lane was discarded.
- *CIFAR-10* (width-64 μP ResNet-18, SGD, batch 128, $T = 2 \cdot 10^4$ updates). 4 initializations × 4 class-stratified 90% training subsets × 4 independent with-replacement minibatch streams (× 4 draws of 20% symmetric label corruption for noisy labels): $3 \times (4^3 + 4^4) = 960$ trajectories. Rates from the clean $2 \cdot 10^4$-update slice of the source sweep: Dip 1 $1.13 \cdot 10^{-3}$, plateau $9.26 \cdot 10^{-3}$, Dip 2 0.2168. Dip 1 does not interpolate (median training error 28% clean, 44% noisy); the plateau and Dip 2 interpolate essentially every trajectory. Recomputing BatchNorm statistics with one clean training pass changes the total risk by less than 0.0028 at every endpoint.
- *WikiText* (transformer, `MuSGD`, batch 32, $T = 5{,}000$). 4 initializations × 4 independently drawn training sets of 64 windows × 4 minibatch streams (× 4 label-noise draws replacing exactly 20% of the scored targets by a different random byte): 960 trajectories. Rates at the local extrema of the clean 5,000-update slice: $9.65 \cdot 10^{-4}$, 0.275, 0.6. Only the plateau interpolates (62/64 clean, 254/256 noisy members); Dip 1 and the high-rate valley do not.

Results. Tables 8 and 9 list every component. From the plateau to Dip 2 the MLP gain is variance reduction: clean ReLU and SiLU variance falls by 0.00721 and 0.00256 while pooled bias worsens by 0.00239 and 0.00029 (net gains 0.00482 and 0.00227); with noisy labels variance falls by 0.0621 (ReLU; 107% of the gain, bias +0.0042) and 0.0393 (SiLU; 99%). For noisy SiLU, the terms that contain the noise draw carry 53% of the gain (dataset×noise alone 33%), the initialization terms without it 40% (initialization main effect 23%), the dataset main effect 6% and pooled bias 1%. On CIFAR-10 the pooled mean improves: clean risk falls by 0.2352 [0.2240, 0.2474], of which 58.7% is pooled bias and 41.3% variance; with 20% label noise it falls by 0.2111 [0.2039, 0.2187], 72.4% pooled bias and 27.6% variance. All minibatch-containing terms together supply 19.6% (clean) and 9.3% (noisy) of the CIFAR gain, all initialization-containing terms 35.5% and 29.6% (these groupings overlap). For WikiText the high-rate valley reduces variance (96.9% clean, 100.7% noisy of a gain of 0.462 and 0.438), but it does not interpolate. Two statements do not hold uniformly

across settings. First, reaching the plateau lowers pooled bias for the MLPs and CIFAR-10 but *raises* it for WikiText ($0.856 \to 0.885$ clean, $0.865 \to 0.884$ noisy). Second, the MLP "Dip 1" row is the $4{\cdot}10^5$-update low-rate reference, and for clean labels it is worse than the plateau (ReLU 0.0646 vs 0.0586, SiLU 0.0476 vs 0.0257), so these rows do not show a clean double dip by themselves; the clean MLP double dip is shown by the rate sweeps of fig. 18 and fig. 9. The ReLU and SiLU MLP panels are shown side by side in fig. 16.

setting	endpoint	$\eta\,(T)$	bias^2	init.	data	noise	data×noise	other int.	variance	risk
MLP ReLU, clean	Dip 1	$1.7{\cdot}10^{-4}$ $(4{\cdot}10^5)$	0.05744	0.00344	0.00329	–	–	0.00038	0.00711	0.06455
	plateau	0.3 (10^4)	0.03761	0.00581	0.00960	–	–	0.00556	0.02097	0.05859
	Dip 2	10.49 (10^4)	0.04000	0.00209	0.00851	–	–	0.00316	0.01377	0.05376
MLP SiLU, clean	Dip 1	$1.7{\cdot}10^{-4}$ $(4{\cdot}10^5)$	0.04340	0.00238	0.00167	–	–	0.00013	0.00419	0.04758
	plateau	0.3 $(4{\cdot}10^4)$	0.01373	0.00246	0.00659	–	–	0.00290	0.01195	0.02568
	Dip 2	16.81 $(4{\cdot}10^4)$	0.01402	0.00131	0.00623	–	–	0.00185	0.00939	0.02341
MLP ReLU, noisy	Dip 1	$1.7{\cdot}10^{-4}$ $(4{\cdot}10^5)$	0.05453	0.00407	0.00509	0.00171	0.01129	0.00150	0.02367	0.07820
	plateau	0.118 $(2{\cdot}10^4)$	0.03849	0.01750	0.01987	0.00975	0.06759	0.04014	0.15484	0.19333
	Dip 2	8.064 $(2{\cdot}10^4)$	0.04266	0.00214	0.01474	0.00719	0.05010	0.01861	0.09279	0.13545
MLP SiLU, noisy	Dip 1	$1.7{\cdot}10^{-4}$ $(4{\cdot}10^5)$	0.03864	0.00260	0.00277	0.00099	0.00601	0.00059	0.01297	0.05161
	plateau	0.118 $(2{\cdot}10^4)$	0.01700	0.01257	0.02528	0.01925	0.13402	0.04720	0.23832	0.25532
	Dip 2	3.645 $(2{\cdot}10^4)$	0.01666	0.00337	0.02284	0.01734	0.12088	0.03458	0.19902	0.21568

Table 8: **Exact functional-ANOVA ledger behind fig. 4, MLP** (population MSE; 16 initializations $\times$ 8 datasets, $\times$ 8 noise draws for noisy labels). "Other int." sums all interactions except dataset×noise; variance is the sum of all effects; bias^2 plus variance equals the risk (to rounding). The "Dip 1" rows are a low-rate reference ($4 \cdot 10^5$ updates), not a matched-budget minimum.

setting	endpoint	$\eta\,(T)$	bias^2	init.	data	noise	minib.	data×noise	other int.	variance	risk
CIFAR-10, clean	Dip 1	0.00113	0.4708	0.0256	0.0015	–	0.0012	–	0.0125	0.0408	0.5117
	plateau	0.00926	0.3460	0.0442	0.0095	–	0.0078	–	0.1423	0.2038	0.5498
	Dip 2	0.2168	0.2078	0.0063	0.0070	–	0.0051	–	0.0883	0.1067	0.3146
CIFAR-10, 20% noise	Dip 1	0.00113	0.5777	0.0136	0.0003	0.0016	0.0002	0.0004	0.0078	0.0239	0.6017
	plateau	0.00926	0.4536	0.0348	0.0045	0.0120	0.0035	0.0107	0.2682	0.3337	0.7873
	Dip 2	0.2168	0.3006	0.0041	0.0047	0.0205	0.0031	0.0107	0.2325	0.2756	0.5762
WikiText, clean	Dip 1	$9.7{\cdot}10^{-4}$	0.8563	0.0037	0.0292	–	0.0003	–	0.0089	0.0421	0.8985
	plateau	0.275	0.8854	0.0215	0.1427	–	0.0195	–	0.3588	0.5424	1.4278
	high-rate	0.6	0.8711	0.0047	0.0045	–	0.0042	–	0.0816	0.0951	0.9661
WikiText, 20% noise	Dip 1	$9.7{\cdot}10^{-4}$	0.8649	0.0023	0.0177	0.0018	0.0002	0.0049	0.0070	0.0340	0.8989
	plateau	0.275	0.8839	0.0045	0.1398	0.0195	0.0037	0.0577	0.2905	0.5158	1.3996
	high-rate	0.6	0.8871	0.0015	0.0012	0.0009	0.0007	0.0027	0.0677	0.0747	0.9617

Table 9: **Exact functional-ANOVA ledger behind fig. 4, CIFAR-10 and WikiText** (multiclass Brier risk; $4 \times 4 \times 4$ levels of initialization, dataset and minibatch stream, $\times 4$ label-noise draws for 20% noise; CIFAR-10 $T = 2 \cdot 10^4$, WikiText $T = 5{,}000$). Columns as in table 8. The WikiText high-rate valley does not interpolate.

C.7 Tests of Theorem 4.1 (fig. 5)

Canonical runs (panels a–c): SiLU, width 256, $\sigma = 0.5$, full-batch GD in float64, $T = 2 \cdot 10^4$, 4 seeds at $\eta \in \{0.118, 1.59, 2.62, 3.69, 5.19, 7.31, 10.3, 14.5, 16.81, 20.4\}$; only interpolating runs (training MSE $\le 10^{-2}$) enter, which excludes one run at $\eta = 20.4$ and one diverged run at $\eta = 7.3$. *Kernel masses.* For the Gaussian input law, the degree-k mass is $m_k = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[K(\boldsymbol{x},\boldsymbol{x}')P_k(\boldsymbol{x},\boldsymbol{x}')]$, where P_k is the reproducing kernel of the degree-k Hermite space (a function of $\boldsymbol{x}^\top\boldsymbol{x}'$, $\|\boldsymbol{x}\|^2$ and $\|\boldsymbol{x}'\|^2$). We estimate it with two independent probe sets of 8,192 (or 16,384) Gaussian inputs, $\boldsymbol{x} \in$ set A and $\boldsymbol{x}' \in$ set B, which makes the estimator unbiased, block by block (first-layer, second-layer and readout parameters); $m_{\ge 3} = \mathbb{E}_{\boldsymbol{x}} K(\boldsymbol{x},\boldsymbol{x}) - m_0 - m_1 - m_2$ and $S_1 = \|\boldsymbol{B}_1\|_F^2$ is computed from the two probe sets' cross product. The ridge in (a) is $\lambda = q\, m_{\ge 3} m/(n_1 m_2)$ from each run's own final masses; the degree-2 energy of the fitted noise in (b) is the degree-2 Hermite energy of $\Delta f = f_{\text{noisy}} - f_{\text{clean}}$ (same seed and rate, clean twin trained on $f_*(\boldsymbol{X})$). *GELU calibration*: the canonical GELU net was trained with the same protocol at $\eta \in \{0.118, 1.59, 2.62, 3.69, 5.19, 7.31, 10.3, 12.1, 14.5, 16.81, 20.4\}$ (clean, $\sigma = 0.5$ and the rough-target condition; 4 seeds), and q (the median over the branch of the ridge efficiency that

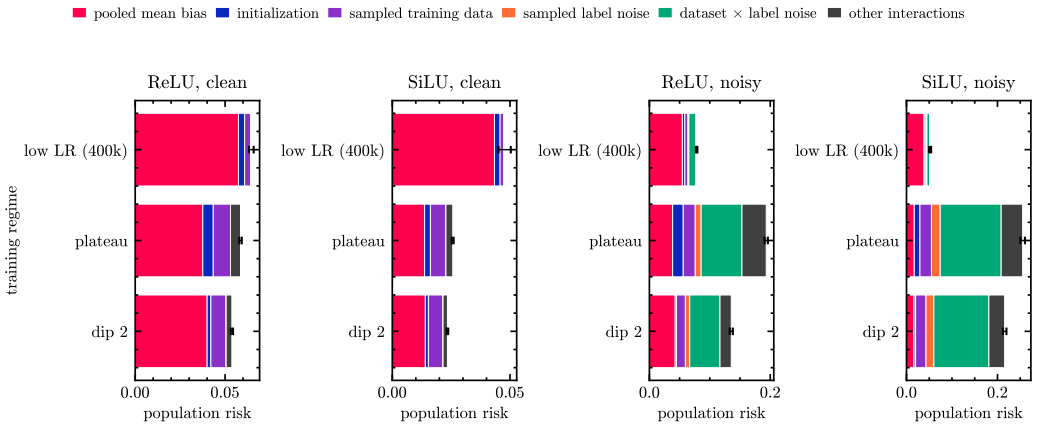

Figure 16: **MLP decomposition for both activations** (ReLU and SiLU; clean and $\sigma = 0.5$). Same design and estimand as fig. 4a. The row labelled "low LR (400k)" is the row labelled "dip 1" in fig. 4: the low-rate reference ($\eta = 1.7 \cdot 10^{-4}$, $4 \cdot 10^5$ updates) described in section C.6. The noisy SiLU high-rate row is $\eta = 3.645$.

matches each run's measured degree-≤ 2 noise energy) and κ_S (the median of $S_1/\lambda_{\max}(\boldsymbol{K}/n)$ for $\eta \geq 10$) were obtained by the rule that reproduces SiLU's constants. *Held-out networks* (panel d, table 2): the depth-2 cells of a separate 344-configuration sweep (activation $\times$ depth $\times$ width $\times$ noise $\times$ teacher, 32 rates $\times$ 3 seeds, $T = 10^4$, float32); for each cell the plateau nets were retrained with the sweep's own trainer to measure their masses, and the prediction uses only those masses with the canonical $\kappa_S = 0.40$, $q = 0.47$.

C.8 Pre-registered sweep for section B.5

72 settings of the canonical problem ($d = 64$, $n = 1024$, network teacher, μP, full-batch GD): ReLU/SiLU/GELU/tanh $\times$ widths 64–1024 at depth 2 with $\sigma \in \{0, 0.5\}$; depth 1 (widths 256 and 1024) with and without a first-layer bias; depth 3 (widths 128 and 256); $\sigma = 1$; lazy $\alpha \in \{4, 16\}$; 8 seeds each. *Stage 1* trains to the GF interpolant at a small rate (to training MSE $< 10^{-6}$; clean SiLU at widths 64/128 needs $\approx 8 \cdot 10^4$ steps) and computes Λ_F, $\boldsymbol{g}_\Lambda$ and Γ in float64 with exact Jacobians. The prediction is written with a SHA-256 hash and a time stamp before *stage 2*, which sweeps 42 rates (12 of them in $(\eta_c, 1.5\eta_c]$) on a grid built from the frozen η_c, $T = 10^4$; no stage-2 output existed at any freeze. The onset slope is the least-squares slope of $\mathcal{R}$ against $\log \eta$ over $\eta_c < \eta \leq 1.5\eta_c$, with seed-bootstrap error bars. Per-rate early stopping selects each lane's checkpoint on the 4,096-point validation set. The same runs provide the kernel movement of fig. 7. Weight-decay runs (section D.1) reuse this trainer with raw-parameter decay $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta(\nabla\mathcal{L} + \lambda_{\text{wd}}\boldsymbol{\theta})$, width 256, $\sigma = 0.5$, $T = 10^5$, 4 seeds.

C.9 Width crossover (fig. 6)

Clean ReLU teacher–student problem ($\sigma = 0$, $n = 1024$, $d = 64$, two hidden layers, bias-free, μP as in section C.1), widths $\{64, 96, 128, 192, 256, 384, 512, 768, 1024, 1536, 2048\}$, 16 initializations, validation set 4,096, clean test set 8,192, float32 with TF32 disabled and float64 metrics. *Low branch (Dip 1):* the risk depends on η and T only through $\tau = \eta T$ (at matched τ, rates from 0.04 to 0.16 agree to about 10^{-6}), so three small rates $\{0.04, 0.08, 0.16\}$ are run for 10^4 steps with a test evaluation every 50 steps, the curve is read in τ (which is gradient flow at time τ to this accuracy), and $\tau^\star$ minimizes validation risk; this avoids quantizing the early-stopping time by a rate grid, and the Dip-1 risks of fig. 6 are plotted at the flow-equivalent rate $\tau/10^4$. *High branch (Dip 2):* native GD endpoints after $T = 10^4$ updates at $\eta \in \{2, 3, \ldots, 12, 14, 16, 20, 24, 32\}$, η chosen on validation risk; a rate at which any seed diverges is excluded as a whole. *Components:* V is the unbiased across-seed function variance on the test set and $B = R - V$; four disjoint four-seed groups give replicate estimates. Error bars on the gap are 95% paired-seed bootstrap intervals that

include branch reselection. The component laws are fitted separately (table 4). *Wide confirmation* (fig. 13): 256 fresh initializations at widths 4096 and 8192, with rate and stopping time selected on validation risk of disjoint pilot seeds over extended searches; simultaneous 95% intervals.

C.10 Our networks are in μP, and μP behaves as intended

Parameterization. All students are bias-free two-hidden-layer ReLU networks on $d = 64$ inputs,

$$f(x) \;=\; \tfrac{1}{\sqrt{w}}\, W_3\, \phi\big(W_2\, \phi(\sqrt{w/d}\, W_1 x)\big), \qquad (W_\ell)_{ij} \sim \mathcal{N}(0, 1/w),$$

trained by full-batch gradient descent on the half mean-squared error with a single global learning rate η shared by all layers. In the abc-notation of Yang & Hu (2021) this is exactly μP: the input multiplier $\sqrt{w/d}$, the hidden layer without multiplier, and the $1/\sqrt{w}$ readout, with no width-dependent learning-rate scaling. As a control we use standard parameterization (SP): the same architecture without multipliers and with $\mathcal{N}(0, 1/\text{fan-in})$ initialization.

Why we check. Our central observation (Fig. **??**) is that the optimal learning rate changes structure with width. A trivial explanation would be a parameterization that is not really μP, since under SP every learning-rate scale moves as $1/w$. Fig. 17 and Table 10 rule this out for every width we use ($w = 16$ to 8192).

Coordinate check (Fig. 17a–c). Following Yang et al. (2021), we train each width for four full-batch steps at one fixed learning rate and record how much each layer moves per coordinate: the root-mean-square change from initialization of the layer-1 and layer-2 pre-activations and of the output. The rate is $\eta = 1$ for μP, about $20\times$ below its stability edge, and $\eta = 0.05$ for SP, which is stable at $w = 64$; this mimics tuning a small proxy and reusing the rate. We use the first $1{,}024$ training inputs and average over 3 seeds. Under μP every curve is flat in width: from $w = 64$ to 8192 the change after four steps stays at ≈ 0.009 (layer 1), ≈ 0.07 (layer 2) and ≈ 0.14 (output). Under SP the same rate fails in both known ways. First, the layer-1 update shrinks with width ($0.009 \to 0.002$ from $w = 16$ to 256), so feature learning fades. Then, from $w \approx 512$, the network blows up: the layer-2 change grows from 1.2 to 4.8×10^5 between $w = 512$ and 8192.

Stability scale (Fig. 17d). We compute the top Hessian eigenvalue $\lambda_{\max}$ of the training loss at initialization by power iteration on Hessian–vector products (60 iterations, first $4{,}096$ training inputs, 3 seeds). Gradient descent is stable only for $\eta \lesssim 2/\lambda_{\max}$, so $2/\lambda_{\max}$ sets the network's learning-rate scale. Under μP it is width-independent (7.3–9.9 for $w = 16$ to 8192). Under SP it falls as $1/w$, from 0.43 to 0.0014, a factor of 300. Its crossing with SP's fixed rate $\eta = 0.05$ near $w = 256$ explains the blow-up in panels (a)–(c). The same holds in the actual training runs: the largest learning rate at which every seed of the $n = 32{,}768$ sweep trained stays between 12 and 26 for $w = 16$ to 1024 (Table 10), with no trend in width.

Consequence. The learning-rate scale of our networks does not move with width, which is what μP promises. The width dependence in Fig. **??** is therefore not a parameterization artefact. Within a fixed, width-independent range of learning rates, the landscape itself changes: a single optimum splits into Dip 1 and Dip 2 once the network is over-parameterized.

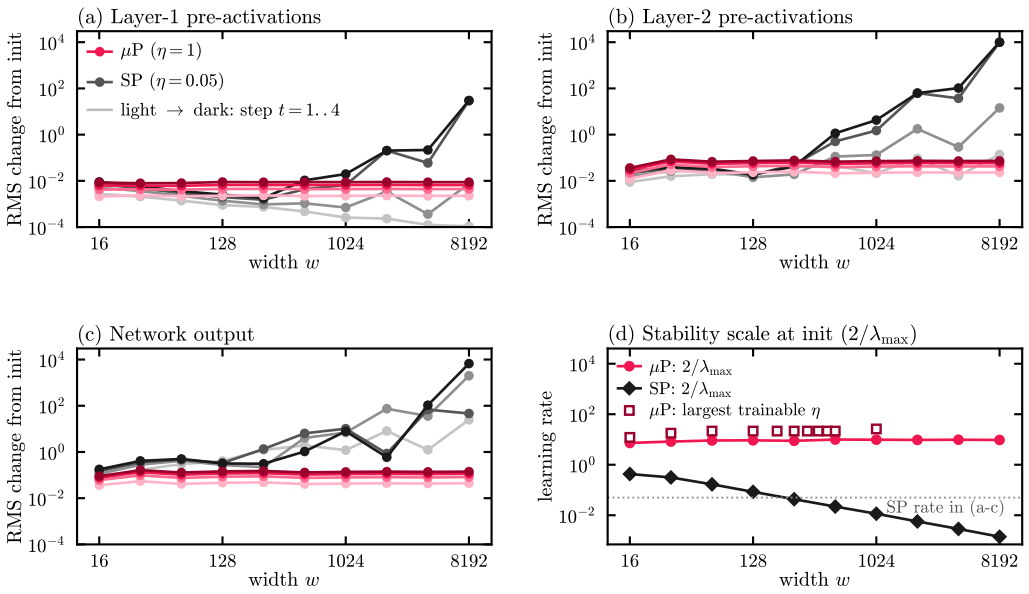

Figure 17: **Our μP implementation behaves as μP should.** Coordinate check (Yang et al., 2021) for the networks of Fig. **??**: widths 16–8192, μP (red) vs. standard parameterization (SP, grey), four full-batch GD steps at one fixed learning rate per parameterization (μP: $\eta = 1$; SP: $\eta = 0.05$, stable at $w = 64$), mean over 3 seeds. **(a)–(c)** RMS change per coordinate of the layer-1 and layer-2 pre-activations and of the output after $t = 1, \ldots, 4$ steps. Under μP every curve is flat in width. Under SP the first-layer update fades with width, and the network blows up once η exceeds SP's stability limit. **(d)** Stability scale $2/\lambda_{\max}$ of the loss Hessian at initialization: width-independent under μP (7.3–9.9) and $\propto 1/w$ under SP ($0.43 \to 0.0014$). Squares: largest learning rate that trained in the $n = 32{,}768$ sweep, also flat.

Table 10: Stability numbers behind Fig. 17 ($n = 32{,}768$). $D = w(w + 63)$. Largest trainable η is resolved up to $w = 1024$; wider networks were sampled only up to the value shown. Last two columns: RMS change per coordinate of the layer-2 pre-activations after $t = 4$ steps (μP: $\eta = 1$; SP: $\eta = 0.05$).

		$2/\lambda_{\max}$ at init		μP largest	layer-2 change, $t = 4$	
width w	D/n	μP	SP	trainable η	μP	SP
16	0.039	7.3	0.43	12.1	0.036	0.027
32	0.093	8.3	0.31	17.8	0.085	0.039
64	0.25	9.1	0.17	21.5	0.068	0.034
128	0.75	9.2	0.084	21.5	0.072	0.019
256	2.5	8.9	0.043	21.5	0.077	0.043
512	9.0	9.9	0.022	21.5	0.067	1.2
1024	34	9.8	0.011	26.1	0.072	4.3
2048	130	9.6	0.0057	≥ 22	0.074	63
4096	520	9.8	0.0029	≥ 18	0.072	103
8192	2100	9.6	0.0014	≥ 10	0.073	4.8×10^5

D FURTHER EXPERIMENTS

D.1 WHY COMMON PRACTICE HIDES THE DOUBLE DIP

Figure 18 tests the practices listed in section 3 on the pre-registered settings of section C.8.

- *Per-rate early stopping* (panel a). Selecting each run's checkpoint on validation risk makes the rate curve monotone below η_c in 72 of 72 settings and removes Dip 1. This is an empirical finding; Proposition B.5(d) proves the analogue only for oracle stopping and for $\eta \le \bar{\eta}$. What remains is a single descent towards the stability edge, whose optimum is Dip 2 or the flow's minimum.
- *Weight decay* (panel b; raw-parameter decay λ_{wd}, width 256, $\sigma = 0.5$, $T = 10^5$, 4 seeds). Only decay strong enough to regularize the plateau removes the second descent. SiLU: $\lambda_{\mathrm{wd}} = 10^{-4}$ keeps plateau and Dip 2 ($0.223 \to 0.115$), 10^{-3} lowers the plateau and keeps a second descent ($0.200 \to 0.136$), and 10^{-2} gives a flat curve at 0.086, the level of Dip 1. ReLU: $0.145 \to 0.131$, $0.134 \to 0.132$ and a flat 0.090. An independent ReLU sweep over $\lambda_{\mathrm{wd}} \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 5\}$ at $T = 2.5 \cdot 10^5$ (3 seeds, 32 rates) agrees (fig. 20). The flat curve at $\lambda_{\mathrm{wd}} = 10^{-2}$ (0.086 SiLU, 0.090 ReLU) is *lower* than Dip 2 without decay (0.115 SiLU, 0.131 ReLU) and at the level of Dip 1: on this problem tuned weight decay is the best choice, and the double dip is a property of unregularized training.
- *Lazy training* (panel c; output scale α, section C.1). Near-NTK training shortens the branch and reduces the gain, but does not remove it: at $\alpha = 16$ the branch ends at $6.9\eta_c$ (ReLU) and $5.2\eta_c$ (SiLU) instead of $15.9\eta_c$, and the gain falls from 29–47% to 7–11%, as Theorem B.2(ii) predicts for a kernel that cannot move.
- *Grid width* (panel d). Where both minima exist, they lie $\log_{10}(\eta_2/\eta_1) = 2.3$–$3.7$ decades apart at $T = 10^4$ (61 settings with an interior Dip 1). Since $\eta_1 = \tau^\star/T$ while η_2 does not move for $T \ge 10^3$, the separation grows by one decade per decade of budget, to 4–5 decades at $T = 4 \cdot 10^5$ (fig. 19).
- *Short budgets.* Two dips need $T > T_c$, the budget at which the flow has overfitted by $\eta_c T$. T_c predicted from the flow curve, η_c and the converged branch value is within a factor 2.1 of the measured threshold in 9 of 9 cells (SiLU within 5%; ReLU and GELU reach two dips 1.3–2.1 times later because the branch has not converged at such budgets).
- *Data much larger than capacity* removes the interpolating Dip 2 altogether (section B.2, fig. 3d).

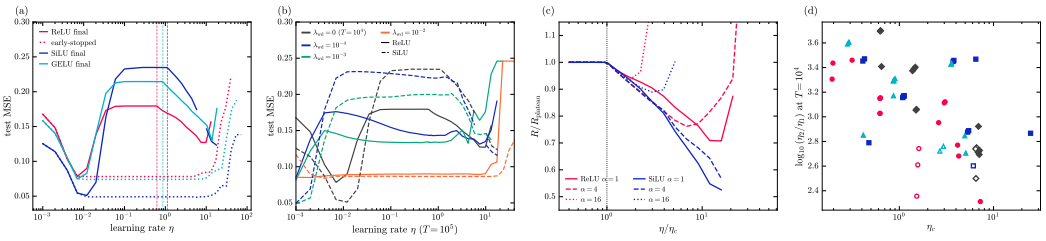

Figure 18: **Why standard training hides the double dip.** (a) Final-iterate (solid) versus per-rate early-stopped (dotted) test risk, $T = 10^4$; vertical lines: η_c. (b) Raw-parameter weight decay $\lambda_{\mathrm{wd}} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$, $T = 10^5$ (ReLU solid, SiLU dashed). (c) Lazy scaling $\alpha \in \{1, 4, 16\}$: risk relative to the plateau against η/η_c. (d) Distance between the two minima, $\log_{10}(\eta_2/\eta_1)$ at $T = 10^4$, against η_c (colours: activations; open: clean labels).

D.2 OTHER OPTIMIZERS: ADAM AND MUON

Setup. A twin of the training-set-size study of fig. 3d in which only the optimizer changes: width-64 ReLU student (μP multipliers as in section C.1), fixed width-256 ReLU teacher, $\sigma = 0.5$, the same nested training sets $n \in \{256, \ldots, 98{,}304\}$, the same three initializations, full-batch updates, $T = 2 \cdot 10^4$, no schedule, warm-up, clipping or weight decay. *Adam*: `torch.optim.Adam` with

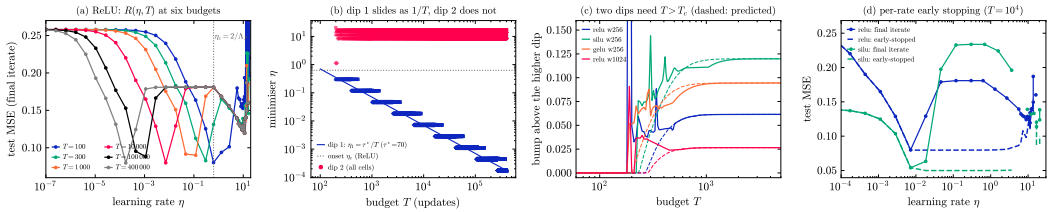

Figure 19: **Budget** ($4 \cdot 10^5$-update sweep, $\sigma = 0.5$, 3 seeds, 40 rates, 2,083 evaluation times). (a) ReLU, width 256: final-iterate risk at six budgets; dotted: $\eta_c = 2/\Lambda_F$. (b) Location of the two minima against T in all cells: Dip 1 follows $\eta_1 = \tau^\star/T$ ($\tau^\star \approx 70$), Dip 2 does not move. (c) Height of the bump above the higher dip against T, measured (solid) and predicted from the flow curve, η_c and the converged branch value (dashed). (d) Final-iterate (solid) versus per-rate validation early-stopped (dashed) risk at $T = 10^4$.

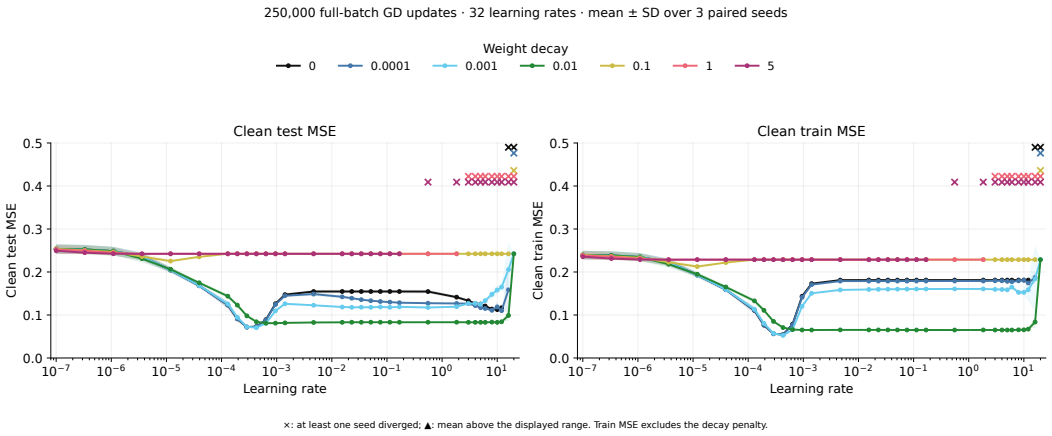

Figure 20: **Weight decay, long budget** (canonical ReLU problem, width 256, $\sigma = 0.5$, $T = 2.5 \cdot 10^5$, 3 seeds, 32 rates; raw-parameter decay on all layers, reported MSE excludes the penalty). Left: clean test MSE; right: clean training MSE. Crosses: at least one seed diverged; means are never taken over surviving seeds only.

default $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$ on all three raw weight matrices with one learning rate. *Muon*: `torch.optim.Muon` on both 64×64 matrices (momentum 0.95, Nesterov, five quintic Newton–Schulz steps in float32, update scaled by $0.2\sqrt{64}$ to match the RMS of AdamW so that one rate is comparable with Adam); the 1×64 readout uses Adam at the same rate. 101 rates per optimizer (10 per decade from 10^{-8} to $10^{0.5}$, plus half steps on $[10^{-1.95}, 10^{-0.55}]$), 4,242 trajectories, none diverged. Above $\eta \approx 0.3$ (Adam) or 0.6 (Muon) every network dies (output identically zero, test MSE $0.2467 = \mathbb{E}f_*^2$). Each batched Adam and Muon update was checked against the PyTorch optimizers ($\leq 10^{-6}$), and every finite endpoint was re-evaluated in float64.

Results (fig. 21, table 11). Local minima are identified by a fixed rule: a 3-point running median over rates, a prominence of at least 10%, and confirmation by every seed; the table reports seed means at the selected rates. The rule was fixed after inspecting the data (without the running median, single-rate wiggles in Muon's noisy pre-collapse region counted as dips), so we checked its sensitivity. Over 18 variants of the rule (running-median window 1, 3 or 5; prominence 5, 10 or 20%; seed confirmation on or off), GD and Adam have a second minimum at every $n \leq 1{,}024$ in every variant, at the same rates (GD: $\eta = 11.3$–12.3, test 0.166–0.170); no variant finds a second minimum that interpolates (training MSE $< 10^{-3}$) and beats the zero predictor for $n \geq 4{,}096$ (GD, Adam) or for Muon at any n. The only variant dependence is whether GD's $n = 1{,}024$ minimum counts as interpolating: its training MSE is $2.5 \cdot 10^{-4}$–$1.2 \cdot 10^{-3}$ depending on the selected rate. *Adam* has the double dip where GD has it: for $n \leq 1{,}024$ a low-rate minimum, an interpolation peak, and a second, interpolating minimum (training MSE $\leq 3.4 \cdot 10^{-5}$) at $\eta \approx 0.010$–0.014 that lies 1.3–1.8 times below the peak; as for GD at this width, it does not beat Dip 1. At $n = 4{,}096$ Adam still has a second valley, but it is worse than the zero predictor. At $n \geq 65{,}536$ ($n/D \geq 8$) Adam has a non-interpolating valley near the edge of trainability (training MSE $\approx \sigma^2$), which GD does not have. *Muon* has no useful second dip at any n: its high-rate local minima are worse than the zero predictor (0.35–1.34) and it does not interpolate at high rates. The learning-rate double dip is therefore not specific to GD, but it is not universal across optimizers.

optimizer	n	Dip 1: $\eta \to$ test	peak: $\eta \to$ test	Dip 2: $\eta \to$ test (train MSE)	Dip 2 < 0.247
GD	256	$1.8 \cdot 10^{-3} \to 0.132$	$0.075 \to 0.240$	$8.87 \to 0.178$ ($7 \cdot 10^{-16}$)	yes
	512	$2.8 \cdot 10^{-3} \to 0.107$	$0.625 \to 0.267$	$10.4 \to 0.170$ ($2 \cdot 10^{-7}$)	yes
	1,024	$3.2 \cdot 10^{-3} \to 0.094$	$0.118 \to 0.329$	$12.1 \to 0.170$ ($1 \cdot 10^{-3}$)	yes
	4,096	$7.2 \cdot 10^{-3} \to 0.069$	–	–	–
	16,384	$0.024 \to 0.053$	–	–	–
	65,536	$0.10 \to 0.035$	–	–	–
	98,304	$0.13 \to 0.031$	–	–	–
Adam	256	$2.5 \cdot 10^{-6} \to 0.123$	$7.9 \cdot 10^{-6} \to 0.208$	$0.010 \to 0.165$ ($3 \cdot 10^{-5}$)	yes
	512	$2.5 \cdot 10^{-6} \to 0.102$	$1.0 \cdot 10^{-5} \to 0.228$	$0.013 \to 0.158$ ($2 \cdot 10^{-5}$)	yes
	1,024	$2.5 \cdot 10^{-6} \to 0.093$	$1.6 \cdot 10^{-5} \to 0.292$	$0.014 \to 0.161$ ($6 \cdot 10^{-6}$)	yes
	4,096	$3.2 \cdot 10^{-6} \to 0.068$	$5.0 \cdot 10^{-4} \to 0.587$	$0.016 \to 0.420$ ($2 \cdot 10^{-4}$)	no
	16,384	$5.0 \cdot 10^{-6} \to 0.053$	–	–	–
	65,536	$7.9 \cdot 10^{-6} \to 0.037$	$0.018 \to 0.118$	$0.063 \to 0.083$ (0.24)	yes
	98,304	$1.0 \cdot 10^{-5} \to 0.032$	$0.020 \to 0.087$	$0.063 \to 0.051$ (0.24)	yes
Muon	256	$1.3 \cdot 10^{-5} \to 0.154$	$2.5 \cdot 10^{-3} \to 1.13$	$0.016 \to 0.388$ ($2 \cdot 10^{-4}$)	no
	512	$1.6 \cdot 10^{-5} \to 0.141$	–	–	–
	1,024	$2.0 \cdot 10^{-5} \to 0.130$	$0.126 \to 132$	$0.20 \to 1.34$ (0.03)	no
	4,096	$2.5 \cdot 10^{-5} \to 0.109$	$0.089 \to 3.18$	$0.22 \to 0.457$ (0.03)	no
	16,384	$3.2 \cdot 10^{-5} \to 0.073$	$1.6 \cdot 10^{-3} \to 0.446$	$7.9 \cdot 10^{-3} \to 0.345$ (0.07)	no
	65,536	$4.0 \cdot 10^{-5} \to 0.044$	–	–	–
	98,304	$4.0 \cdot 10^{-5} \to 0.039$	–	–	–

Table 11: **Local minima of the rate curves** (width 64, $\sigma = 0.5$, $T = 2 \cdot 10^4$, full batch; seed means over 3 seeds at the rates selected by the rule in the text). "–": the rule finds no second minimum. The last column compares the second minimum with the zero-predictor risk $\mathbb{E}f_*^2 = 0.247$. No second minimum beats Dip 1 for any optimizer.

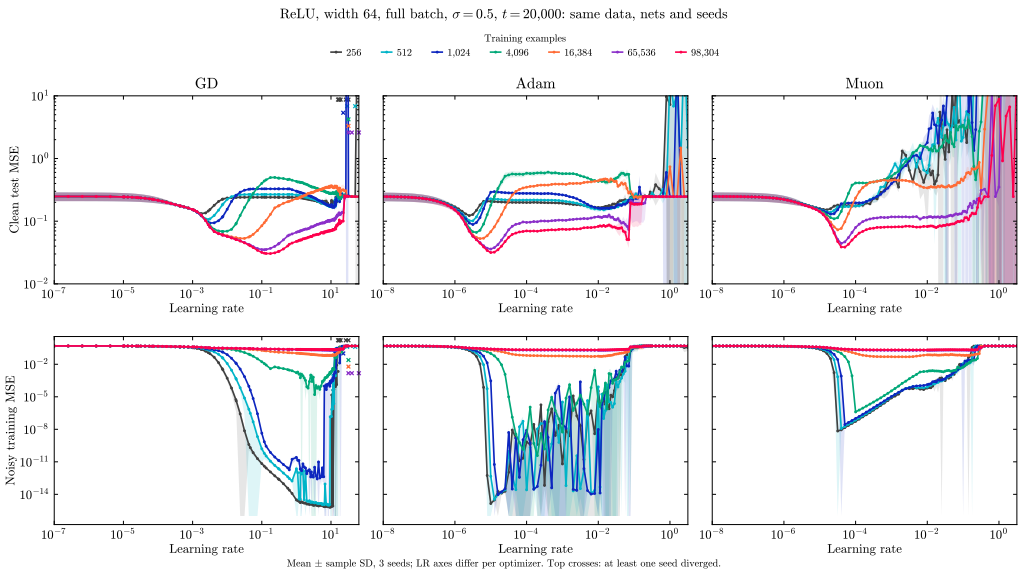

Figure 21: **GD, Adam and Muon** on the width-64 training-set-size problem ($\sigma = 0.5$, $T = 2 \cdot 10^4$, full batch, 3 seeds; learning-rate axes differ per optimizer). GD curves are those of fig. 3d. Bands: sample SD over seeds (shown only when all three seeds are finite); crosses: at least one seed diverged; a collapsed network appears as the zero-predictor risk 0.2467.

D.3 Edge of stability and catapults in native GD

We retrained every rate of the $4 \cdot 10^5$-update sweep (ReLU and SiLU, width 256, $\sigma = 0.5$, 40 rates $\times$ 3 seeds) to $T = 10^4$ and logged, at 721 times (every update to $T = 200$, every 5 to 10^3, every 25 to 10^4), the top eigenvalue of the full loss Hessian H_L (float64 Lanczos) and of the Gauss–Newton part $\boldsymbol{K}/n$ (fig. 22; the native (η, T) maps of test error, $h\lambda_{\max}(H_L)$ and edge-of-stability occupancy are the top panels of figs. 14 and 15).

- *The two dips lie on opposite sides of the cap.* The plateau sharpness S_{pl} (the net at $\eta = 0.118$, $T = 10^4$) gives $\eta_c = 2/S_{\mathrm{pl}} = 0.629$ (ReLU) and 1.117 (SiLU), consistent with $\eta_c = 2/\Lambda_F = 0.62$ and 1.11 from the GF interpolant (section C.8); the largest rate that never crosses $\eta\lambda_{\max}(H_L) = 2$ is 0.62 (ReLU) and 0.69 (SiLU), and the smallest that ever crosses is 1.30 and 1.58. Dip 1 ($\eta = 0.0072$) stays a factor 400 (ReLU) and 1800 (SiLU) below the cap for all 10^4 updates (peak $\eta\lambda_{\max} = 0.0051$, 0.0011); Dip 2 is above it from the first update (peak 8.4, 7.1).
- *Sustained EoS.* For ReLU at $\eta \in [1.3, 13.7]$ and SiLU at $\eta \in [1.59, 24]$, 100% of the measurements after $T = 2000$ lie within 5% of $\eta\lambda_{\max} = 2$ (medians 1.998 and 1.994): $\lambda_{\max}$ falls from the plateau value to $2/\eta$ and stays there.
- *Catapult.* For $\eta \approx 1.3$–5 the cap is crossed once, after 115 (ReLU) or 225 (SiLU) updates. For $\eta \gtrsim 9$ (ReLU) or $\gtrsim 11$ (SiLU) the network is above the cap at initialization and oscillates between $\eta\lambda_{\max} \approx 0.1$ and 10 for 50–150 updates, with transient excursions up to $4.3 \cdot 10^3$ (ReLU, $\eta = 16.2$) and $7.9 \cdot 10^5$ (SiLU, $\eta = 10.8$), before settling at 2. The fraction of the first 10^3 updates spent above the cap rises from 0.35 at $\eta = 1.3$ to 0.94 at $\eta = 12.1$. In the SiLU runs of section C.5 the native training loss rises to 2.5–4.7 times its initial value at every rate ≥ 3.6 at which all seeds interpolate (and to non-finite values in the seeds that blow up), whereas the capped runs never exceed their initial loss.
- *Hessian versus NTK.* $\lambda_{\max}(H_L)/\lambda_{\max}(\boldsymbol{K}/n)$ has median 1.32 (ReLU) and 1.68 (SiLU) at initialization and up to 79 during the first 10^3 updates, but equals 1 to eight or nine decimals after $T \approx 2000$, once the networks interpolate. Stability statements before interpolation therefore use the full Hessian.

These observations establish that native Dip 2 runs pass through a catapult and then sit at the edge of stability; the capped experiments (section C.5) show that neither is necessary for the gain.

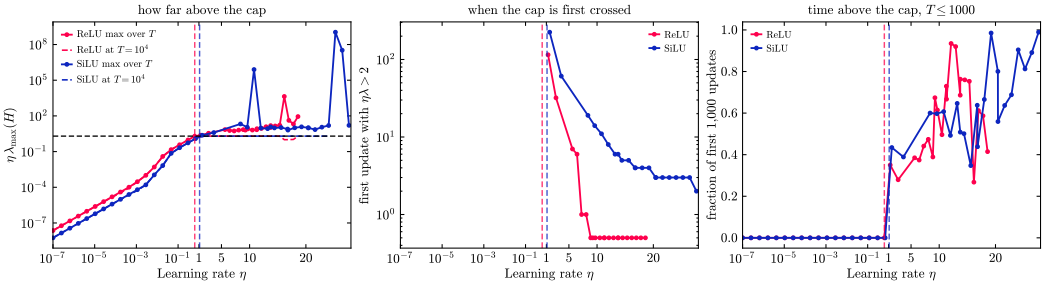

Figure 22: **Crossing the cap** (ReLU and SiLU, width 256, $\sigma = 0.5$). (a) Maximum of $\eta\lambda_{\max}(H_L)$ over the run; (b) update at which the cap is first crossed; (c) fraction of the first 10^3 updates above the cap.

D.4 Kernel degree masses across the dips

Figure 23 shows the Hermite-degree masses $m_0, m_1, m_2, m_{\ge 3}$ of the tangent kernel of the canonical SiLU net (width 256, $\sigma = 0.5$, full-batch GD in float64, $T = 2 \cdot 10^4$, 4 seeds; estimator as in section C.7 with two probe sets of 16,384 inputs) at initialization, at Dip 1 ($\eta = 0.0055$, the best of $\{0.0039, 0.0055, 0.0078\}$), on the plateau ($\eta = 0.118$) and at Dip 2 ($\eta = 20.4$; 3 interpolating seeds). Training inflates the head more than the tail: from initialization to the plateau m_0 grows $46\times$, m_1 $208\times$, m_2 $265\times$ and $m_{\ge 3}$ $138\times$. These are the primary estimates (4 seeds, $2 \times 16{,}384$ probes); an independent estimate from 3 initializations and $2 \times 8{,}192$ probes gives $265\times$ and $138\times$ for m_2 and $m_{\ge 3}$. Rounded, the head grows about $260\times$ and the tail about $140\times$. Dip 1 stops early: m_2 has grown only $16\times$ and $m_{\ge 3}$ $9\times$. From the plateau to Dip 2 the ceiling pulls the head back, m_0 by $14\times$ ($1.45 \to 0.10$, close to the cap $2/\eta = 0.098$) and m_2 by $9\times$ ($26.3 \to 3.0$), while the tail changes by only $1.3\times$ ($3.56 \to 2.79$), so the ratio $m_{\ge 3}/m_2$, and with it the ridge of Theorem 4.1(a), grows from 0.13 to 0.92. The same masses along the whole SiLU branch are shown in fig. 9c.

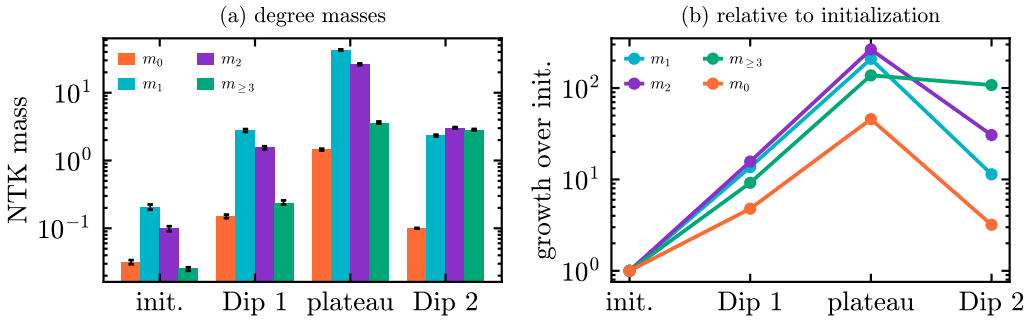

Figure 23: **NTK mass by Hermite degree** (canonical SiLU net, $\sigma = 0.5$). (a) m_0, m_1, m_2 and $m_{\ge 3}$ at initialization, Dip 1, the plateau and Dip 2 (median over seeds, bars: range). (b) The same masses relative to initialization. The head (degrees ≤ 2) is inflated more than the tail during training and is pulled back by the ceiling at Dip 2, while the tail stays.

D.5 Training-set size

Figure 24 shows the full (η, T) test-error maps behind fig. 3d (width 64, $D = 8{,}128$). With the rule of section D.2, GD has an interpolating second minimum only for $n \le 1{,}024$ ($n/D \le 0.13$). For $n \ge 4{,}096$ ($n/D \ge 0.5$) the rule finds a single minimum, which moves to larger rates and lower risk as n grows (0.069 at $n = 4{,}096$, 0.031 at $n = 98{,}304$; table 11); at $n = 4{,}096$ a small raw local

valley remains near $\eta \approx 16$, but it does not interpolate (training MSE 0.22) and does not pass the rule. Dip 1 deepens and moves right with n at fixed T, as a longer flow time is needed before the flow starts to overfit. This agrees with the $T = 4 \cdot 10^4$ data-threshold sweep of fig. 8 (which uses its own detection rule): at width 64 its second minimum interpolates only at $n = 1{,}024$ ($n/D = 0.13$) and is non-interpolating from $n/D = 0.25$ on; the interpolating second dips up to $n/D = 0.5$ in that sweep are at width 128.

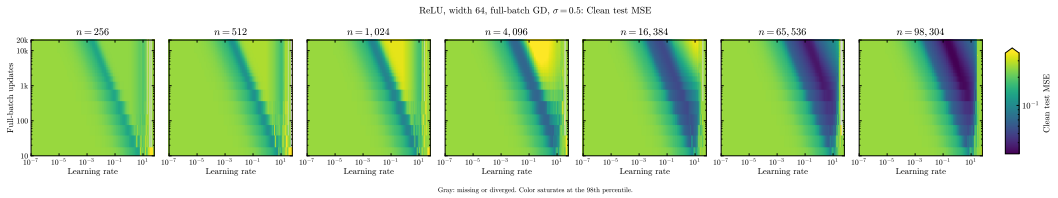

Figure 24: **Test error over (η, T) for each training-set size** (width-64 ReLU student, width-256 ReLU teacher, $\sigma = 0.5$, full-batch GD, 3 seeds). Gray: missing or diverged cells.

E EXTENDED RELATED WORK

Double descent across axes. Modern work links interpolation to both benign generalization and double-descent peaks (Belkin et al., 2019; Hastie et al., 2019; Liang & Rakhlin, 2020; Holzmüller, 2020). In deep networks, double descent has been studied through model size, training duration, effective complexity, and fine-grained bias–variance decompositions (Nakkiran et al., 2021; d'Ascoli et al., 2020; Adlam & Pennington, 2020a; Yang et al., 2020). Epoch-wise double descent has further been connected to superposed bias–variance tradeoffs and the separation of feature-learning and overfitting time scales (Heckel & Yilmaz; Advani et al., 2017; Montanari & Urbani, 2026). Non-monotone curves also arise as explicit regularization varies, while random-feature and neural-network Gaussian-process analyses relate them to spectral structure and finite-width effects (Yilmaz & Heckel, 2022; Mei & Montanari, 2019; Harzli et al., 2021). Complementary results characterize when interpolating noisy labels can remain benign (Bartlett et al., 2019; Muthukumar et al., 2019). These works vary capacity, sample size, training time, or explicit regularization. We instead hold the architecture, data, and update budget fixed within each sweep and vary the learning rate itself. Although the low-rate minimum is an early-stopping effect, our matched-time gradient-flow and substepping controls show that the high-rate minimum is not obtained merely by advancing farther along the same continuous-time trajectory.

Finite-step and large-learning-rate dynamics. A finite learning rate can change more than the speed at which an optimizer traverses a gradient-flow trajectory. Backward-error and spectral-filtering analyses show that discrete steps introduce learning-rate-dependent implicit regularization (Barrett & Dherin, 2020; Beugnot et al., 2022). Prior work identified qualitatively distinct small- and large-learning-rate phases and found that initial or sustained large rates can select better-generalizing solutions (Lewkowycz et al., 2020; Li et al., 2019; Sadrtdinov et al., 2024).

Within high-rate training, a *catapult* is a transient instability: when an update crosses the stability boundary of the current model, the training loss can increase abruptly before the evolving network reduces its curvature and recovers (Lewkowycz et al., 2020; Zhu et al., 2023). By contrast, the *edge of stability* (EoS) is a sustained regime in which the leading loss-Hessian curvature remains near the conventional quadratic threshold, $\eta\lambda_{\max}(H_L) \simeq 2$, and the loss oscillates over individual updates while decreasing over longer time scales (Cohen et al., 2021; Wu et al., 2018; Arora et al., 2022). Large-step dynamics near this boundary can additionally exhibit self-stabilization, balancing, period-two behavior, and transitions toward chaos (Damian et al., 2022; Wang et al., 2021; Chen & Bruna, 2022; Kalra et al., 2023; Wang et al., 2023). Central-flow and constrained-flow models provide complementary continuous descriptions of the time-averaged large-step trajectory (Cohen et al., 2024; Arora et al., 2022).

In our native Dip 2 runs, a catapult occurs as an early transient and is followed by extended EoS occupancy. These observations establish an association with Dip 2, not that either behavior causes its generalization gain. Our contribution is more specific than identifying large-step instability: we identify a minimum–plateau–minimum structure in a single fixed-budget learning-rate sweep and study the origin of its high-rate branch.

Large learning rates, representations, and generalization. Several works connect training dynamics to the alignment of learned features or tangent kernels with the target (Zhu et al., 2023; Atanasov et al., 2021; Ba et al., 2022; Jiang et al., 2025). Large steps have also been associated with selecting favorable basins, rotating curvature eigendirections, learning threshold-like features, and retaining max-margin implicit bias (Sadrtdinov et al., 2024; Wang & Roberts, 2024; Ahn et al., 2022; Wu et al., 2023). The distinction between frozen-kernel and evolving-feature regimes has been developed through NTK, lazy-training, and rich-regime analyses; recent work also cautions that the magnitude of feature evolution need not determine the quality of the learned features (Jacot et al., 2018; Chizat et al., 2018; Woodworth et al., 2019; Yang & Hu, 2021; Göring et al., 2025). Complementary function-space attributes part of neural-network generalization to architecture-induced simplicity biases and studies how these biases interact with SGD and structured data (Pérez et al., 2018; Mingard et al., 2021; 2025).

Raw parameter-space sharpness is not invariant to function-preserving reparametrizations, and its correlation with generalization can depend on the optimizer (Dinh et al., 2017; Zhang et al., 2021). Moreover, good generalization has been studied both below and at EoS, while recent work questions

whether $\eta\lambda_{\max}$ alone captures the stability actually realized along an optimizer's update direction (Liang et al., 2025; Tuci et al., 2026; Lee & Lee, 2026). We therefore use sharpness and the nominal stability threshold as trajectory diagnostics rather than universal explanations of generalization. Our curvature intervention addresses the narrower non-necessity question. In the settings studied here, the capped trajectory reaches comparably low test error without a measured loss blow-up and remains below the conventional full-Hessian threshold at the recorded checkpoints. This shows that catapult and measured EoS behavior are not necessary for the gain; it does not establish that scalar curvature is its causal mediator or that the controlled and native trajectories reach the same function.

Feature-learning parametrizations and hyperparameter transfer. Tensor Programs IV characterizes width-dependent parametrizations that retain nontrivial feature learning at infinite width, while complementary spectral analysis recovers feature-learning scaling prescriptions from operator-norm conditions (Yang & Hu, 2021; Yang et al., 2023). Tensor Programs V introduced μTransfer, and subsequent work generalized or diagnosed learning-rate transfer using alternative parametrizations, optimizers, and width-consistent sharpness dynamics (Yang et al., 2021; Everett et al., 2024; Noci et al., 2024). The defining theoretical property of μP is maximal, order-one feature evolution; invariance of the test-optimal learning rate is an empirical transfer property rather than a universal guarantee.

Subsequent results prove transfer in idealized settings while showing that practical transfer can depend strongly on weight decay, optimizer details, and the embedding-layer learning rate (Hayou, 2026; Kosson et al., 2026; Singh Kalra & Barkeshli, 2026). Most closely related, prior work shows that the test-optimal learning rate can change sharply when width crosses the model-wise double-descent transition (Kalra & Barkeshli; Singh Kalra & Barkeshli, 2026). We therefore do not claim the first failure of μTransfer across model-wise double descent. Our focus is the within-width risk landscape underlying such switches: the separation of its low-rate and high-rate branches, the finite-step origin of the latter, and the training dynamics that select between them.