

---

# Continuous-Time Analysis of Adaptive Optimization and Normalization

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Adaptive optimization algorithms, particularly Adam and its variant AdamW,  
2 are fundamental to modern deep learning, however, their training dynamics lack  
3 comprehensive theoretical understanding, with limited insight into why com-  
4 mon practices—such as specific hyperparameter choices and normalization lay-  
5 ers—contribute to successful generalization. This work presents a continuous-time  
6 formulation of Adam and AdamW, facilitating a tractable analysis of training  
7 dynamics that can shed light on such practical questions. We theoretically derive  
8 a stable region for Adam’s hyperparameters  $(\beta, \gamma)$  that ensures bounded updates,  
9 empirically verifying these predictions by observing unstable exponential growth  
10 of parameter updates outside this region. Furthermore, we theoretically justify  
11 the success of normalization layers by uncovering an implicit meta-adaptive ef-  
12 fect of scale-invariant architectural components. This insight leads to an explicit  
13 optimizer, 2-Adam, which we generalize to  $k$ -Adam—an optimizer that applies  
14 an adaptive normalization procedure  $k$  times, encompassing Adam (correspond-  
15 ing to  $k = 1$ ) and Adam with a normalization layer (corresponding to  $k = 2$ ).  
16 Overall, our continuous-time formulation of Adam facilitates a principled analysis,  
17 offering deeper understanding of optimal hyperparameter choices and architectural  
18 decisions in modern deep learning.

## 19 1 Introduction

20 Adaptive optimization algorithms have become an essential component of modern deep learning,  
21 providing significant benefits to the training of neural networks compared to their non-adaptive  
22 counterparts. Among these algorithms, Adam [1] and its variant AdamW [2] have become widely  
23 used in practice, featuring both an adaptive learning rate (in the sense of RMSprop [3]) and an  
24 adaptive gradient direction (in the sense of momentum [4]). Despite their wide success, there is  
25 limited theoretical insight into why common practices – such as typical hyperparameter choice, and  
26 the use of normalization layers (e.g. layer-norm) – contribute towards successful generalization. In  
27 this work, we demonstrate the utility of continuous-time models of optimization in shedding light on  
28 such questions. A continuous-time approach allows for a mathematically tractable analysis that can  
29 leverage the tools of calculus – namely, differential equations.

30 Our contributions can be summarized as:

### 31 1. Continuous-time formulation (Section 2)

- 32 (a) We derive an explicit continuous-time expression for Adam’s (& AdamW’s) parameter update  
33  $u(t)$ , which we directly utilize in Section 3 (discussed in contribution 2).
- 34 (b) We further derive a second-order differential equation modelling the dynamics of Adam (&  
35 AdamW), which we directly utilize in Section 4 (discussed in contribution 3).

36 **2. Theory of adaptive hyperparameters (Section 3)**

- 37 (c) We determine a theoretical stability region for Adam’s adaptive hyperparameters  $(\beta, \gamma)$  that  
 38 ensures bounded parameter updates and stable training, which we verify empirically.  
 39 (d) This result implies that instability may occur when the hyperparameters move outside the stable  
 40 region, a phenomenon we empirically verify, exhibiting *predictable* exponential growth.  
 41 (e) We observe a faster rate of generalization when  $(\beta, \gamma)$  is further from the instability boundary.

42 **3. Implicit effect of scale invariance (Section 4)**

- 43 (f) We perform a theoretical analysis of the *implicit* beneficial effect of scale invariant architectural  
 44 components (e.g. layer-norm), uncovering a *meta-adaptive* normalization effect.  
 45 (g) We convert this implicit effect to an *explicit* optimizer, which we name 2-Adam, and consider  
 46 its extension  $k$ -Adam: a generalization of Adam/AdamW (which corresponds to  $k = 1$ ) that  
 47 performs a normalization procedure  $k$  times successively.  
 48 We discuss related work in Appendix A and future directions in Appendix B.

49 **2 Continuous-time formulation of Adam**

50 In this section we present a continuous-time formulation of the Adam (& AdamW) optimizer which  
 51 forms the theoretical foundation for the rest of the paper: we derive a continuous-time expression  
 52 for the Adam gradient update (utilized in Section 3 to derive a condition for bounded updates), and  
 53 formulate Adam as a second-order differential equation (used in Section 4 to interpret the implicit  
 54 effect of scale invariance and motivate the  $k$ -Adam optimizer). We provide brief descriptions of  
 55 relevant optimizers (momentum, RMSprop, Adam, AdamW) in Appendix C. Experimental details  
 56 are left to Appendix H.

57 **Notation.** We write  $\|x\| := \sqrt{\sum_i x_i^2}$  and  $\|x\|_\infty := \max_i |x_i|$  for  $x \in \mathbb{R}^p$ , and denote the inner  
 58 product  $\langle x, y \rangle \equiv \sum_i x_i y_i$ . We denote elementwise squaring by  $x^{\odot 2}$ . We consider a loss function  
 59  $L : \Theta \rightarrow \mathbb{R}$  where  $\Theta \subset \mathbb{R}^p$ , and  $\theta_n \in \Theta$  denotes a model’s parameters after  $n$  discrete updates, with  
 60 corresponding gradient  $g_n := \nabla_{\theta} L(\theta_n)$ , and initial parameter  $\theta_0$ . We will write  $g_{0:n} \equiv (g_0, \dots, g_n)$ .

61 **2.1 Adam in continuous-time**

62 Neglecting weight decay for the moment (i.e.  $\lambda = 0$ ), Adam & AdamW possess the discrete-time  
 63 update rule

$$\theta_{n+1} = \theta_n - \eta u_n, \quad \text{where } u_n := \frac{\sqrt{1 - \beta^{n+1}}}{1 - \gamma^{n+1}} \frac{m_n}{\sqrt{v_n}} \quad (1)$$

with learning rate  $\eta$ , and *adaptive hyperparameters*  $(\beta, \gamma) \in [0, 1]^2$  with moving-averages

$$m_n := \gamma m_{n-1} + (1 - \gamma) g_n, \quad v_n := \beta v_{n-1} + (1 - \beta) g_n^{\odot 2}$$

64 where  $(m_{-1}, v_{-1}) := (0, 0)$ . We say that  $u_n$  is an *adaptive-normalization* of the gradients  $g_{0:n}$  (see  
 65 Appendix Q for more details). We will work in continuous-time, at a timescale of  $\eta^p$ , such that e.g.  
 66  $m_n = m(t_n)$ , with  $t_n := n\eta^p$  for some  $p \in \mathbb{R}$  (typically  $p = 1$ ). We can then derive continuous-time  
 67 expressions for the moving-averages  $m_n$  and  $v_n$  (details left to Appendix E), which (to leading order  
 68 in  $\eta^p$ ) take the form

$$m(t) = \int_0^t d\tau K_\gamma(\tau, t) g(\tau), \quad v(t) = \int_0^t d\tau K_\beta(\tau, t) g(\tau)^{\odot 2} \quad (2)$$

69 with  $g(\tau) := \nabla_{\theta} L(\theta(\tau))$  the gradient at continuous time  $\tau$  and  $K_\alpha$  defined below. This provides a  
 70 continuous-time expression for Adam/AdamW’s update  $u_n = u(t_n)$ ,

$$u(t) = \frac{\sqrt{1 - \beta^{1+t/\eta^p}}}{1 - \gamma^{1+t/\eta^p}} \frac{\int_0^t d\tau K_\gamma(\tau, t) g(\tau)}{\sqrt{\int_0^t d\tau K_\beta(\tau, t) g(\tau)^{\odot 2}}}, \quad K_\alpha(\tau, t) := \frac{1 - \alpha}{\eta^p \alpha} \exp\left(-\frac{1 - \alpha}{\eta^p \alpha} (t - \tau)\right) \quad (3)$$

71 Including weight decay is simple and is described in Appendix E. As we will see, Equation (3)  
 72 is a central result for Section 3, allowing us to derive theoretical guarantees regarding training

73 stability with respect to Adam/AdamW’s adaptive hyperparameters  $(\beta, \gamma)$ . From Equation (1) we can  
 74 also determine  $\theta_n = \theta(t_n)$  in continuous-time via a governing differential equation (details left to  
 75 Appendix E). To second-order, AdamW with weight decay  $\lambda$  is governed by the differential equation

$$\lambda\eta\theta(t) + \eta^p\dot{\theta}(t) + \frac{\eta^{2p}}{2}\ddot{\theta}(t) = -\eta u(t) \quad (4)$$

76 In combination with the continuous-time expression for  $u(t)$  (Equation (3)), we can numerically  
 77 solve Equation (4) (method described in Appendix I) to obtain a continuous-time parameter trajectory  
 78  $\theta(t)$ . We demonstrate that the continuous-time trajectory (for timescale  $p = 1$ ) associated with  
 79 Equation (4) closely agrees with the true discrete-time trajectory in Appendix K. In Section 4 we will  
 80 use Equation (4) to interpret the implicit effect of scale invariance.

### 81 3 Theory of adaptive hyperparameters

82 In this section we will present a theory-driven account of adaptive hyperparameter choice for  
 83 Adam/AdamW, understanding the values of  $(\beta, \gamma)$  that result in stable training and effective general-  
 84 ization. First we will use the continuous-time expression for Adam’s update (Equation (3)) to derive a  
 85 theoretical region of hyperparameter space  $\mathcal{B}_+$  for which updates are provably bounded (Section 3.1).  
 86 We will then empirically verify that this region indeed exhibits stable training, with unstable training  
 87 in the complementary region  $\mathcal{B}_-$  (exhibiting a *predictable* exponential growth), and observe how  
 88 generalization performance varies in  $\mathcal{B}_+$  (Section 3.2).

#### 89 3.1 Deriving a condition for bounded updates

90 The continuous-time expression for Adam’s update (Equation (3)) has an immediate consequence in  
 91 regards to bounding the *max-update*  $\|u_n\|_\infty \equiv \|\theta_{n+1} - \theta_n\|_\infty/\eta$ . We have the following bound,

92 **Max-update bound.** *The max-update can be bounded as*

$$\|u_n\|_\infty \leq \frac{\sqrt{1 - \beta^{n+1}}}{1 - \gamma^{n+1}} \frac{1 - \gamma}{\gamma} \sqrt{\frac{\beta}{1 - \beta}} B_n(\beta, \gamma) \quad (5)$$

where

$$B_n(\beta, \gamma) := \begin{cases} 1/\sqrt{C(\beta, \gamma)}, & C(\beta, \gamma) > 0, \\ \sqrt{n}, & C(\beta, \gamma) = 0, \\ \exp(n|C(\beta, \gamma)|/2)/\sqrt{|C(\beta, \gamma)|}, & C(\beta, \gamma) < 0 \end{cases}$$

93 defining  $C(\beta, \gamma) := (2\beta(1 - \gamma) - \gamma(1 - \beta))/\beta\gamma$ . We leave the derivation of this bound to Appendix E.

94 We highlight that this bound is only possible because of the specific form of Adam’s update  $u(t)$ :  
 95 a moving-average of  $g$  divided by the square-root of a moving-average of  $g^{\odot 2}$  (Equation (3)), i.e.  
 96 an adaptive-normalization of  $g$  (see Appendix Q), which allows us to apply the Cauchy-Schwarz  
 97 inequality. We can therefore view this result as a theoretical justification for the form of Adam’s  
 98 update.

99 We define the bounded-update region  $\mathcal{B}_+ := \{(\beta, \gamma) : C(\beta, \gamma) > 0\}$ , and the complementary region  
 100  $\mathcal{B}_- := \{(\beta, \gamma) : C(\beta, \gamma) < 0\}$ . From Equation (5), we can bound the max-update by a constant  
 101 independent of  $n$  when  $(\beta, \gamma) \in \mathcal{B}_+$ . Outside of  $\mathcal{B}_+$ , the bound depends on  $n$  and diverges over  
 102 training as  $n \rightarrow \infty$ . This is suggestive that we may observe stable training when  $(\beta, \gamma) \in \mathcal{B}_+$ , and  
 103 that the max-update may grow exponentially (at a rate/exponent proportional to  $|C(\beta, \gamma)|$ ) when  
 104  $(\beta, \gamma) \in \mathcal{B}_-$ . Indeed, we will verify this phenomena empirically in Section 3.2. We comment on  
 105 the case  $(\beta, \gamma) \in \mathcal{B}_0$  in Appendix F. It is easy to show that  $\beta > \gamma$  is a sufficient condition for  
 106  $(\beta, \gamma) \in \mathcal{B}_+$ , meaning that the choice of hyperparameters typically chosen in practice – e.g. the  
 107 PyTorch default values  $(\tilde{\beta}, \tilde{\gamma}) := (0.999, 0.9)$  – lie within  $\mathcal{B}_+$ .

108 For the following, we define the *level curves*  $\mathcal{B}_c := \{(\beta, \gamma) : C(\beta, \gamma) = c\}$ , and consider *normal*  
 109 *curves* perpendicular to these level curves, visualized in Figure 1. We will denote the (unique)  
 110 normal curve passing through the point  $(\beta, \gamma)$  by  $\mathcal{C}_{\beta, \gamma}$ . Note that, by construction,  $C(\beta, \gamma)$  varies  
 111 monotonically along a normal curve.

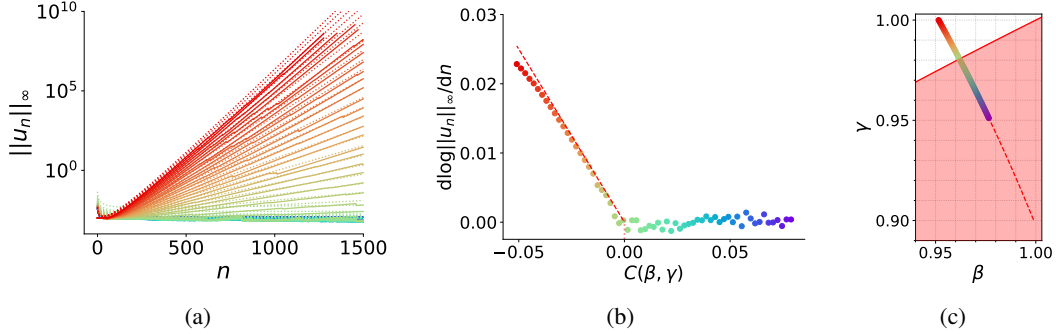


Figure 2: **Max-update bound accurately predicts stable region and unstable exponent of divergence.** We consider 64 hyperparameter points  $(\beta, \gamma)$  taken uniformly along a section of the normal curve  $\mathcal{C}_{\tilde{\beta}, \tilde{\gamma}}$  passing through the point  $(\tilde{\beta}, \tilde{\gamma}) = (0.999, 0.9)$ , which we visualize in (c). For each of these points, we plot (a) the max-update  $\|u_n\|_\infty \equiv \|\theta_n - \theta_{n-1}\|_\infty / \eta$  over training iterations  $n$ , and (b) the slope  $d \log \|u_n\|_\infty / dn$  of the log-max-update at iteration  $n = 500$  (in order to interpret exponential growth). In (a) we visualize the bounds of Equation (5) as dotted lines, and in (b) we denote the predicted slope/exponent  $|C(\beta, \gamma)|/2$  (when  $C(\beta, \gamma) < 0$ ) as a dashed line.

### 112 3.2 Empirical analysis of max-update

113 We now consider the empirical implications of Equation (5), as-  
 114 sessing whether these bounds indeed hold in practice and whether  
 115  $C(\beta, \gamma)$  is predictive of training stability. We train a decoder-only  
 116 transformer model on a shakespeare text dataset (details left to Ap-  
 117 pendix H) and observe how the max-update  $\|u_n\|_\infty$  evolves over  
 118 training iterations  $n$ . Specifically, in Figure 2, we consider the normal  
 119 curve  $\mathcal{C}_{\tilde{\beta}, \tilde{\gamma}}$  passing through the typical hyperparameter values  
 120  $(\tilde{\beta}, \tilde{\gamma}) = (0.999, 0.9)$ , taking 64 points  $(\beta, \gamma)$  uniformly along this  
 121 curve (which we visualize in Figure 2c). For each point  $(\beta, \gamma)$ , we  
 122 plot the max-update trajectory (Figure 2a), finding that the theoret-  
 123 ical bounds of Equation (5) are satisfied in both regions  $\mathcal{B}_+$  and  $\mathcal{B}_-$ ,  
 124 observing well-behaved bounded growth in  $\mathcal{B}_+$ , whereas  $\mathcal{B}_-$  exhibits  
 125 exponential growth at a rate that appears correlated with  $|C(\beta, \gamma)|$   
 126 as predicted by Equation (5). We more closely verify this phenom-  
 127 ena in Figure 2b, finding that in  $\mathcal{B}_-$ , the slope  $d \log \|u_n\|_\infty / dn$  has  
 128 a near-perfect agreement with the theoretically predicted growth  
 129 rate of  $|C(\beta, \gamma)|/2$ . It is surprising that not only are the theoretical  
 130 bounds (Equation (5)) satisfied in practice, but the bounds are very  
 131 accurate models of the true empirical dynamics, with exponential  
 132 growth occurring almost immediately after entering  $\mathcal{B}_-$ . We look  
 133 more closely at the results of Figure 2a in the case of  $(\beta, \gamma) \in \mathcal{B}_+$  in Appendix M. We comment on  
 134 correcting the exponential growth in  $\mathcal{B}_-$  via learning rate annealing in Appendix O.

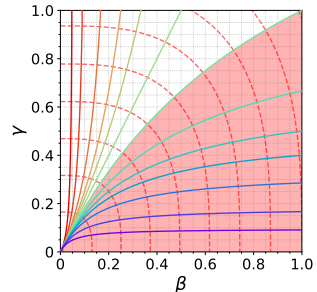


Figure 1: Visualization of level curves  $\mathcal{B}_c$  (solid lines) and normal curves  $\mathcal{C}_{\beta, \gamma}$  (dashed red lines). Level curves are coloured based on their value of  $C(\beta, \gamma)$ , (i.e. purple has most positive value, red most negative). The bounded-update region  $\mathcal{B}_+$  is highlighted in red.

135 These results partly justify the success of typical values  $(\tilde{\beta}, \tilde{\gamma}) = (0.999, 0.9)$  in practice: these  
 136 values lie in  $\mathcal{B}_+$  and hence benefit from theoretical guarantees regarding training stability (which, as  
 137 we have seen, are faithful to practice). Can we further obtain a more fine-grained picture as to what  
 138 choices of  $(\beta, \gamma)$  within the region  $\mathcal{B}_+$  will result in successful generalization? We will now assess  
 139 how generalization performance varies in  $\mathcal{B}_+$ , and particularly, how the value of  $C(\beta, \gamma)$  correlates  
 140 with generalization performance. In Figure 3 we observe how test loss varies along the normal curve  
 141  $\mathcal{C}_{\tilde{\beta}, \tilde{\gamma}}$ , taking 128 points uniformly along the entire curve (visualized in Figure 3c). We see that larger  
 142 values of  $C(\beta, \gamma)$  display faster generalization, as seen in Figure 3a and highlighted at iteration 1000  
 143 in Figure 3b. After sufficient training, i.e. at iteration 3000 in Figure 3b, all points in  $\mathcal{B}_+$  achieve  
 144 roughly the same test loss. The test loss exhibits a rapid increase after entering the region  $\mathcal{B}_-$ , which  
 145 is to be expected given the unstable exponential growth in the max-update shown in Section 3.2.

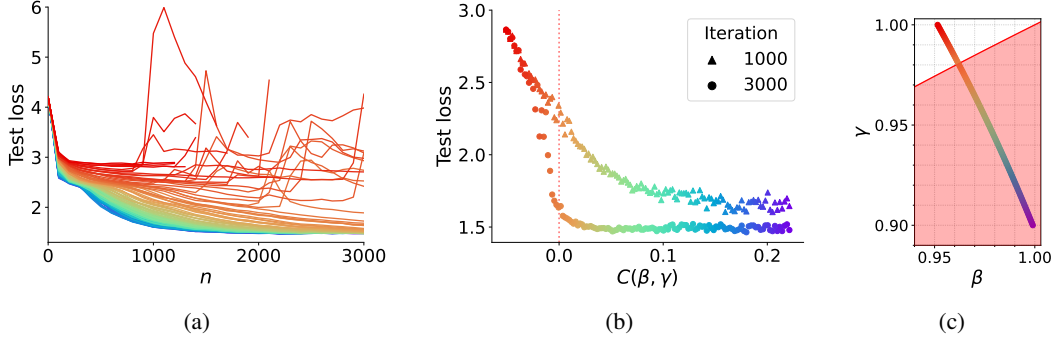


Figure 3: **Our theory accurately predicts the divergence of test loss across Adam’s hyperparameter space.** We consider 128 hyperparameter points  $(\beta, \gamma)$  taken uniformly along the entire normal curve  $C_{\tilde{\beta}, \tilde{\gamma}}$ , visualized in (c). For each point, we plot (a) the test loss over training iterations  $n$ , and (b) the best test loss achieved over the first 1000 and 3000 iterations. The rightmost point in (c) corresponds to  $(\tilde{\beta}, \tilde{\gamma}) = (0.999, 0.9)$ .

146 These results suggest that for a given normal curve, the point  $(\beta, \gamma)$  with the largest value of  $C(\beta, \gamma)$   
 147 generalizes the fastest along the curve. We provide supporting evidence for this claim, finding similar  
 148 results for the normal curve that instead passes through the hyperparameter values  $(0.95, 0.9)$ , in  
 149 Appendix N. This would further justify why the values  $(\tilde{\beta}, \tilde{\gamma})$  succeed in practice (since  $\tilde{\beta} \approx 1$ ,  
 150 hence  $C(\tilde{\beta}, \tilde{\gamma})$  is large relative to other points along its normal curve), however unlike the results  
 151 of Section 3.2, this claim lacks an explicit supporting theoretical result; it is only suggestive from  
 152 Equation (5) that a larger value of  $C(\beta, \gamma)$  may correlate with better training properties. We leave  
 153 direct theoretical guarantees regarding the rate of generalization to future work.

## 154 4 Implicit effect of scale invariance

155 The presence of scale-invariant architectural components has become ubiquitous in modern machine  
 156 learning. One particular instance of such components are normalization layers, such as layer-norm  
 157 [5], batch-norm [6], and qk-norm [7, 8]. It has been observed in practice that such normalization  
 158 provides an implicit beneficial effect to training, compared to e.g. explicitly fixing/constraining the  
 159 weight norm after each update step [9, 10, 11]. In this section we will use the second-order differential  
 160 equation for Adam/AdamW (Equation (4)) to solve for the norm dynamics and uncover an implicit  
 161 *meta-adaptive* effect of scale invariance (Section 4.1). Furthermore, we will introduce the  $k$ -Adam  
 162 optimizer, a generalization of Adam that explicitly models this implicit effect (Section 4.2).

### 163 4.1 Uncovering an implicit meta-adaptive effect

164 A function  $f : \Theta \rightarrow \mathbb{R}$  is *scale-invariant* with respect to a weight  $W \in \mathcal{W} \subset \Theta$  if and only if  
 165  $f(\theta)$  remains unchanged under the transformation  $W \mapsto \alpha W$  for all  $\alpha \in \mathbb{R}$ . One can show (see  
 166 Appendix L) that if  $L$  is scale-invariant with respect to a weight  $W$ , then

$$\langle W(t), g_W(t) \rangle = 0 \quad (6)$$

167 at all times  $t$ , where  $g_W(t) := \nabla_W L(\theta(t))$  is the gradient associated with  $W$ . To theoretically  
 168 analyse the implicit effect of scale invariance, we will use the second-order differential equation for  
 169 AdamW described by Equation (4) and choose a timescale of  $p = 1$  (we verified the accuracy of  
 170 this setup in Appendix K). For an arbitrary scale invariant weight  $W$  (e.g. a key/query matrix under  
 171 qk-norm), the continuous-time dynamics of  $W$  are governed by

$$\frac{1}{2}\eta\ddot{W}(t) + \dot{W}(t) + \lambda W(t) = -u_W(t) \quad (7)$$

172 to second-order, with  $u_W$  defined analogously to  $u$  (replacing  $g$  with  $g_W$  in Equation (3)). From here,  
 173 using Equation (6), we can derive (see Appendix J) the following approximation for the weight norm,

$$\|W(t)\|^2 \approx \|W(0)\|^2 e^{-2\lambda t} + \eta \int_0^t d\tau e^{-2\lambda(t-\tau)} \|u_W(\tau)\|^2 \quad (8)$$

174 Interestingly, Equation (8) has an identical form to an exponential moving-average of  $\|u_W\|^2$  in  
 175 continuous-time (see Appendix J). Further (also see Appendix J), the evolution of the unit direction  
 176  $\hat{W}$  – which is most relevant to consider, given that  $W$  is scale invariant – is governed by

$$\frac{1}{2}\eta\ddot{\hat{W}}(t) + \dot{\hat{W}}(t) + \Lambda\hat{W}(t) = -\frac{1}{\|\hat{W}(t)\|}u_W(t) \quad (9)$$

177 where we define  $\Lambda := \frac{1}{2}\eta r \|\dot{\hat{W}}\|^2$ . Equation (9) has the same form as Equation (7), with weight decay  
 178  $\lambda = \Lambda$  and updating via  $u_W/\|\hat{W}\|$ . With the aforementioned interpretation of  $\|\hat{W}\|$  as the square-  
 179 root of a moving-average of  $\|u_W\|^2$ , we can undo the coarse-graining assumption (Appendix J) and  
 180 view  $\hat{W}$  as updating by an adaptive-normalization of  $u_W$  without additional momentum. This concept  
 181 of *meta-adaptive normalization*, i.e. adaptive-normalization of an already adaptively-normalized  
 182 gradient  $u_W$ , can be converted into an explicit optimizer 2-Adam, which we now describe.

## 183 4.2 The $k$ -Adam optimizer

184 The  $k$ -Adam optimizer captures the concept of meta-adaptivity, applying an adaptive-normalization  
 185 procedure  $k$  times with respect to hyperparameters  $(\beta_{1:k}, \gamma_{1:k})$ , with update rule

$$k\text{-Adam: } \theta_{n+1} := \theta_n - \eta u_n^{(k)} \quad (10)$$

where for  $i = 1, \dots, k$ ,

$$u_n^{(i)} := \frac{\tilde{m}_n^{(i)}}{\sqrt{\tilde{v}_n^{(i)}}}, \quad \tilde{m}_n^{(i)} := \frac{1}{1 - \gamma_i^{n+1}} m_n^{(i)}, \quad \tilde{v}_n^{(i)} := \frac{1}{1 - \beta_i^{n+1}} v_n^{(i)},$$

$$m_n^{(i)} := \gamma_i m_{n-1}^{(i)} + (1 - \gamma_i) u_n^{(i-1)}, \quad v_n^{(i)} := \beta_i v_{n-1}^{(i)} + (1 - \beta_i) (u_n^{(i-1)})^{\odot 2}$$

186 with  $u_n^{(0)} := g_n \equiv \nabla_{\theta} L(\theta_n)$  and  $(m_{-1}^{(i)}, v_{-1}^{(i)}) = (0, 0)$  for all  $i = 1, \dots, k$ . We present an explicit  
 187 algorithm for implementing  $k$ -Adam in Appendix R. A more succinct (but equivalent) description  
 188 of  $k$ -Adam is presented in Appendix Q. Note that 1-Adam is equivalent to Adam. A scale-invariant  
 189 weight trained under Adam is analogous to 2-Adam with  $\gamma_2 = 0$ .

190 **Hyperparameter choice.** We will consider 4 strategies for choosing hyperparameters  $(\beta_{1:k}, \gamma_{1:k})$ ,

Inverse exp:	$\beta_i := 1 - (1 - \tilde{\beta})^{1/k}$ ,	$\gamma_i := 1 - (1 - \tilde{\gamma})^{1/k}$
Exp:	$\beta_i := \tilde{\beta}^k$ ,	$\gamma_i := \tilde{\gamma}^k$
Scaled:	$\beta_i := \tilde{\beta}/k$ ,	$\gamma_i := \tilde{\gamma}/k$
Naive:	$\beta_i := \tilde{\beta}$ ,	$\gamma_i := \tilde{\gamma}$

191 for all  $i = 1, \dots, k$ , where  $(\tilde{\beta}, \tilde{\gamma}) :=$   
 192  $(0.999, 0.9)$  are the typical Adam/AdamW hy-  
 193 perparameter values. In Appendix P we moti-  
 194 vate the inverse exp strategy; the other strategies  
 195 have been chosen heuristically. We note that  
 196 max-update bounds as shown for Adam (Equa-  
 197 tion (5)) hold analogously for each intermediate  
 198 update  $u_n^{(i)}$ , which we describe in Appendix S.  
 199 For the strategies defined above,  $(\beta_i, \gamma_i) \in \mathcal{B}_+$   
 200 for all  $i = 1, \dots, k$  (since  $\beta_i > \gamma_i$ ) hence we  
 201 expect stable training as a result of this bound.

202 **Evaluating  $k$ -Adam.** We motivated interpret-  
 203 ing the implicit effect of scale invariance based  
 204 on its beneficial influence on training [9, 10, 11].

205 We would therefore expect 2-Adam, and perhaps  $k$ -Adam for  $k > 2$ , to outperform Adam; is this  
 206 indeed the case? We train a CNN (architecture details in Appendix H) on the CIFAR10 dataset for  
 207 100 epochs using  $k$ -Adam, for values  $k = 1, \dots, 10$  and using each hyperparameter strategy defined  
 208 above. We display the results in Figure 4, finding that  $k$ -Adam outperforms Adam/AdamW at various  
 209  $k > 1$  under the first three strategies, however the naive strategy performs particularly badly for  
 210  $k > 2$ ; we show this in more detail in Appendix P. We find that 2-Adam with the inverse exp strategy  
 211 and decoupled weight decay performs best. We discuss related future directions in Appendix B.

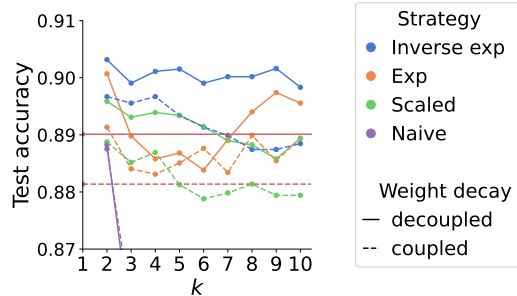


Figure 4: Plot of the best test accuracy against  $k$  after training a CNN for 100 epochs on CIFAR10 using the  $k$ -Adam optimizer. We highlight the  $k = 1$  case in red, corresponding to Adam/AdamW.

## References

- [1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 1, 10
- [2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 1, 10
- [3] T. Tieleman and G. Hinton. Lecture 6.5—rmsprop, coursera: Neural networks for machine learning. Coursera Lecture Slides, 2012. [https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf). 1, 10
- [4] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. 1, 9
- [5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 5
- [6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 5
- [7] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. 5, 13
- [8] Andrea Schioppa Justin Gilmer and Jeremy Cohen. Intriguing properties of transformer training instabilities, 2023. 5, 13
- [9] Hidenori Tanaka and Daniel Kunin. Noether’s learning dynamics: Role of symmetry breaking in neural networks, 2021. 5, 6, 9
- [10] Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Beyond batchnorm: Towards a unified understanding of normalization in deep learning, 2021. 5, 6, 9
- [11] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?, 2019. 5, 6, 9
- [12] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization, 2019. 9
- [13] Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes, 2019. 9
- [14] Dongruo Zhou, Jinghui Chen, Yuan Cao, Ziyang Yang, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization, 2024. 9
- [15] Anas Barakat and Pascal Bianchi. Convergence and dynamical behavior of the adam algorithm for non-convex stochastic optimization, 2020. 9
- [16] André Belotto da Silva and Maxime Gazeau. A general system of differential equations to model first order adaptive algorithms, 2019. 9
- [17] Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics, 2021. 9
- [18] Bo Zhao, Iordan Ganev, Robin Walters, Rose Yu, and Nima Dehmamy. Symmetries, flat minima, and the conserved quantities of gradient flow, 2023. 9
- [19] Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction, 2023. 9

- 257 [20] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond,  
258 2019. 9
- 259 [21] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli,  
260 Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for  
261 deep learning: Training bert in 76 minutes, 2020. 9
- 262 [22] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham,  
263 Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery  
264 of optimization algorithms, 2023. 9
- 265 [23] Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch  
266 normalization provably avoids rank collapse for randomly initialised deep networks, 2020. 9
- 267 [24] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure  
268 attention loses rank doubly exponentially with depth, 2023. 9
- 269 [25] Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and  
270 Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of  
271 rank collapse, 2022. 9
- 272 [26] Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Ming Zhou, Klaus Neymeyr, and Thomas  
273 Hofmann. Exponential convergence rates for batch normalization: The power of length-direction  
274 decoupling in non-convex optimization, 2018. 9
- 275 [27] Ryo Karakida, Shotaro Akaho, and Shun ichi Amari. The normalization method for alleviating  
276 pathological sharpness in wide neural networks, 2019. 9
- 277 [28] Andrej Karpathy. nanogpt. <https://github.com/karpathy/nanoGPT>, 2022. 14
- 278 [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,  
279 Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 14
- 280 [30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document  
281 recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 14



## 282 A Related work

283 Previous work has studied adaptive optimization theoretically [12, 13, 14, 15, 16], but this work  
284 focuses on asymptotic convergence rates and hence does not cover the results presented here. These  
285 works perform a discrete-time analysis, except for [16] which uses the same differential equation  
286 representation of Adam to the one we consider in Section 2, and [15] which considers a different  
287 differential equation for Adam, yet these works do not explicitly solve for Adam’s moving-averages  
288  $m(t)$ ,  $v(t)$  or update  $u(t)$  in continuous-time as we do in this paper.

289 There has been previous work on interpreting the effect of scale invariance on training dynamics  
290 from a theoretical perspective [9, 17, 18, 19, 10]. Most relevant to our work is [9], which – using a  
291 Lagrangian-based approach – shows that scale-invariant weights trained under gradient descent with  
292 momentum exhibits an adaptive effect analogous to RMSprop. Since this work does not consider an  
293 adaptive learning rate, the theoretical analysis is greatly simplified, however the results are less faithful  
294 to practical machine learning where optimizers with adaptive learning rates (i.e. Adam/AdamW) are  
295 widely used; our analysis in Section 4 extends this argument to the case of Adam/AdamW, where we  
296 observe a meta-adaptive effect. We further use our theoretical insights to motivate a novel optimizer,  
297  $k$ -Adam, which captures the implicit role of scale invariance.

298 Various extensions to Adam have been proposed – such as the AMSGrad [20], LAMB [21], and  
299 Lion [22] optimizers – however they differ from our proposed optimizer  $k$ -Adam: AMSGrad extends  
300 Adam by scaling the learning rate by an adaptive factor  $1/\sqrt{\hat{v}_n}$  where  $\hat{v}_n$  is a maximum of all past  
301 moving-averages  $v_n$ ; LAMB uses layer-wise adaptive learning rates and a trust ratio mechanism; Lion  
302 [22] uses the *sign* of a moving-average of the gradient in order to update the parameters.  $k$ -Adam  
303 extends Adam uniquely, instead applying an adaptive normalization procedure  $k$  times in succession,  
304 and is motivated by theoretical insight.

## 305 B Discussion

306 In this work we have presented a continuous-time framework for Adam, facilitating a mathematically  
307 tractable analysis of training dynamics. We have demonstrated the utility of our framework for an-  
308 swering questions of practical interest regarding hyperparameter choice and the role of normalization.

309 We have seen that the derived bound on the max-update (Equation (5)) is surprisingly accurate to  
310 empirical training dynamics, however our bound does not rigorously justify (only suggests) why we  
311 observe a faster rate of generalization for larger  $C(\beta, \gamma)$  along a given normal curve, hence there is  
312 still progress to be made for a complete understanding of adaptive hyperparameter choice.

313 In this paper we do not perform a rigorous analysis of  $k$ -Adam’s performance – i.e. evaluating its  
314 performance across various datasets and architectures – as our focus in this paper is a theory-driven  
315 analysis of training dynamics. Though we find  $k = 2$  to perform best in this paper, it is possible that  
316 a larger  $k$  may be beneficial in certain contexts, e.g. at small batch sizes (to possibly alleviate noisy  
317 gradients). There may also be a more natural, theoretically-motivated hyperparameter strategy than  
318 the strategies we consider here. We leave a study of these aspects to future work.

319 We note that the meta-adaptive effect we describe in this paper is not a complete account of the benefits  
320 of normalization layers. Normalization layers possess additional benefits, such as avoiding rank  
321 collapse [23, 24, 25] and contributing towards the smoothness of the loss landscape [11, 26, 27, 19];  
322 it is unclear whether such phenomena are related to the meta-adaptive effect we describe in this paper.

## 323 C Descriptions of optimizers

Momentum [4] features an adaptive gradient direction – updating by a moving-average of the gradient  
– as described by the update rule,

$$\begin{aligned} \text{Momentum: } \theta_{n+1} &:= \theta_n - \eta m_n, \\ m_n &:= \gamma m_{n-1} + (1 - \gamma) g_n \end{aligned}$$

324 for  $n = 0, 1, 2, \dots$ , with moving-average hyperparameter  $\gamma$  and  $m_{-1} := 0$ .

RMSprop [3] features an adaptive learning rate – normalizing the update gradient  $g_n$  by a moving-average of the squared gradient  $g_n^{\odot 2}$  – as described by the update rule,

$$\begin{aligned} \text{RMSprop: } \theta_{n+1} &:= \theta_n - \eta \frac{g_n}{\sqrt{v_n}}, \\ v_n &:= \beta v_{n-1} + (1 - \beta) g_n^{\odot 2} \end{aligned}$$

325 with  $v_{-1} := 0$ . Note that  $g_n^{\odot 2}$  describes an element-wise squaring, and the division  $g_n/\sqrt{v_n}$  is  
 326 also element-wise. Here we have neglected weight decay; we will discuss weight decay strategies  
 327 (coupled vs decoupled) below.

Adam [1] and its variant AdamW [2] are a combination of momentum and RMSprop, and also use a bias correction factor, which we justify in Appendix D. Neglecting weight decay, Adam & AdamW are equivalent, with update rule

$$\begin{aligned} \text{Adam/AdamW: } \theta_{n+1} &:= \theta_n - \eta \frac{\tilde{m}_n}{\sqrt{\tilde{v}_n}}, \\ \tilde{m}_n &:= \frac{1}{1 - \gamma^{n+1}} m_n, \quad \tilde{v}_n := \frac{1}{1 - \beta^{n+1}} v_n, \\ m_n &:= \gamma m_{n-1} + (1 - \gamma) g_n, \quad v_n := \beta v_{n-1} + (1 - \beta) g_n^{\odot 2} \end{aligned}$$

where a tilde denotes bias correction. The difference between Adam and AdamW comes from how they apply weight decay: Adam uses *coupled* weight decay, equivalent to transforming the loss  $L(\theta) \mapsto L(\theta) + \frac{\lambda}{2} \|\theta\|^2$  such that  $g_n \mapsto g_n + \lambda \theta_n$ , and AdamW uses *decoupled* weight decay, which involves subtracting  $\lambda \eta \theta_n$  from the RHS of the update rule, i.e.

$$\text{AdamW: } \theta_{n+1} := \theta_n - \lambda \eta \theta_n - \eta \frac{\tilde{m}_n}{\sqrt{\tilde{v}_n}}$$

328 We note that the PyTorch implementation of RMSprop uses coupled weight decay, i.e. RMSprop is  
 329 Adam with  $\gamma = 0$ .

## 330 D Motivating bias correction in Adam

331 Consider an exponential moving average of a sequence  $\{x_0, x_1, x_2, \dots\}$  of tensors,

$$\begin{aligned} y_n &:= \beta y_{n-1} + (1 - \beta) x_n \\ &= \dots \\ &= (1 - \beta)(x_n + \beta x_{n-1} + \dots + \beta^n x_0) \end{aligned}$$

In the case that  $\mathbb{E}[x_n]$  is independent of  $n$ ,

$$\mathbb{E}[y_n] = \mathbb{E}[x_n](1 - \beta)(1 + \beta + \dots + \beta^n) = \mathbb{E}[x_n](1 - \beta^{n+1})$$

hence for an unbiased exponential moving average, we should instead consider a *bias-corrected* form of  $y_n$ :

$$\tilde{y}_n := \frac{1}{1 - \beta^{n+1}} y_n$$

such that

$$\mathbb{E}[\tilde{y}_n] = \mathbb{E}[x_n]$$

## 332 E Deriving the parameter update bound

333 First we will derive a continuous-time expression for  $m_n$  by Taylor expanding its definition,

$$\begin{aligned} m(t_n) &= \gamma m(t_n - \eta^p) + (1 - \gamma) g(t_n) \\ &= \gamma m(t_n) - \eta^p \gamma \dot{m}(t_n) + (1 - \gamma) g(t_n) + O(\eta^{2p}) \end{aligned}$$

and so to leading order,

$$\begin{aligned} \dot{m}(t) + \frac{1-\gamma}{\eta^p \gamma} m(t) &= \frac{1-\gamma}{\eta^p \gamma} g(t) \\ \frac{d}{dt} \left[ \exp\left(\frac{1-\gamma}{\eta^p \gamma} t\right) m(t) \right] &= \frac{1-\gamma}{\eta^p \gamma} \exp\left(\frac{1-\gamma}{\eta^p \gamma} t\right) g(t) \\ \implies m(t) &= \frac{1-\gamma}{\eta^p \gamma} \int_0^t d\tau \exp\left(-\frac{1-\gamma}{\eta^p \gamma} (t-\tau)\right) g(\tau) \\ &\equiv \int_0^t d\tau K_\gamma(\tau, t) g(\tau) \end{aligned}$$

334 Similarly for  $v_n$ ,

$$\begin{aligned} \dot{v}(t) + \frac{1-\beta}{\eta^p \beta} v(t) &= \frac{1-\beta}{\eta^p \beta} g(t)^{\odot 2} \\ \implies v(t) &= \frac{1-\beta}{\eta^p \beta} \int_0^t d\tau \exp\left(-\frac{1-\beta}{\eta^p \beta} (t-\tau)\right) g(\tau)^{\odot 2} \\ &\equiv \int_0^t d\tau K_\beta(\tau, t) g(\tau)^{\odot 2} \end{aligned}$$

335 As a result, the ratio of moving-averages present in Adam's update  $u(t)$  takes the form,

$$\frac{m(t)}{\sqrt{v(t)}} = \eta^{-p/2} \frac{1-\gamma}{\gamma} \sqrt{\frac{\beta}{1-\beta}} \frac{\int_0^t d\tau \exp\left(-\frac{1-\gamma}{\eta^p \gamma} (t-\tau)\right) g(\tau)}{\sqrt{\int_0^t d\tau \exp\left(-\frac{1-\beta}{\eta^p \beta} (t-\tau)\right) g(\tau)^{\odot 2}}} \quad (11)$$

Note that this describes an element-wise division of tensors. This expression is particularly amenable to the Cauchy-Schwarz inequality, which says that for (square-integrable) functions  $f, g : \mathbb{R} \rightarrow \mathbb{R}$ ,

$$\left| \int dx f(x)g(x) \right| \leq \sqrt{\int dx f(x)^2} \sqrt{\int dx g(x)^2}$$

336 We can apply the Cauchy-Schwarz inequality (element-wise) to the integral in the numerator of  
337 Equation (11) to provide an upper bound,

$$\begin{aligned} &\left| \int_0^t d\tau \exp\left(-\frac{1-\gamma}{\eta^p \gamma} (t-\tau)\right) g(\tau) \right| \\ &\equiv \left| \int_0^t d\tau \left[ \exp\left(-\frac{C(t-\tau)}{2\eta^p}\right) \right] \left[ \exp\left(-\frac{1-\beta}{2\eta^p \beta} (t-\tau)\right) g(\tau) \right] \right| \\ &\leq \sqrt{\int_0^t d\tau \exp\left(-\frac{C(t-\tau)}{\eta^p}\right)} \sqrt{\int_0^t d\tau \exp\left(-\frac{1-\beta}{\eta^p \beta} (t-\tau)\right) g(\tau)^{\odot 2}} \\ &= \sqrt{\frac{\eta^p}{C} (1 - \exp(-Ct/\eta^p))} \sqrt{\int_0^t d\tau \exp\left(-\frac{1-\beta}{\eta^p \beta} (t-\tau)\right) g(\tau)^{\odot 2}} \end{aligned}$$

where we have defined

$$C := \frac{2\beta(1-\gamma) - \gamma(1-\beta)}{\beta\gamma}$$

338 Note that the second factor is exactly equal to the denominator in the expression for  $m(t)/\sqrt{v(t)}$ ,  
 339 and so they cancel to give the following bound:

$$\begin{aligned} \left| \frac{m(t)}{\sqrt{v(t)}} \right| &\leq \frac{1-\gamma}{\gamma} \sqrt{\frac{\beta}{1-\beta}} \sqrt{\frac{1-\exp(-Ct/\eta^p)}{C}} \\ &\leq \frac{1-\gamma}{\gamma} \sqrt{\frac{\beta}{1-\beta}} \begin{cases} 1/\sqrt{C}, & C > 0, \\ \sqrt{t/\eta^p}, & C = 0, \\ \sqrt{\exp(|C|t/\eta^p)/|C|}, & C < 0 \end{cases} \end{aligned}$$

340 We can then obtain an expression for  $u_n = u(t_n)$  using  $n = t_n/\eta^p$ .

341 **Weight decay.** Note that extending Equation (3) to include weight decay is simple: if we want  
 342 coupled weight decay (as in Adam), we can transform  $g(\tau) \mapsto g(\tau) + k\theta(\tau)$ ; if we want decoupled  
 343 weight decay (as in AdamW), we can subtract  $k\eta\theta_n$  from the RHS of Equation (1) and proceed  
 344 identically.

**Differential equation form.** We can obtain a differential equation for Adam/AdamW by noting that,

$$\theta_{n+1} - \theta_n \equiv \theta(t_n + \eta^p) - \theta(t_n) = \eta^p \dot{\theta}(t_n) + \frac{\eta^{2p}}{2} \ddot{\theta}(t_n) + O(\eta^{3p})$$

345 by Taylor expansion, which we can then substitute into the update rule  $\theta_{n+1} = \theta_n - \eta u_n$ .

**Normal curves.** The level curve  $\mathcal{B}_c$  is determined by

$$f(\beta, \gamma) = c, \quad f(\beta, \gamma) := \frac{2\beta(1-\gamma) - \gamma(1-\beta)}{\beta\gamma} \equiv C(\beta, \gamma)$$

and normal curves  $\gamma = \gamma(\beta)$  satisfy

$$\frac{d\gamma(\beta)}{d\beta} = \frac{\partial f / \partial \gamma}{\partial f / \partial \beta}$$

$$\implies \gamma(\beta) = (k - 2\beta^3)^{1/3}$$

346 for an integration constant  $k$ . The normal curve passing through  $(\beta_0, \gamma_0)$  has integration constant  
 347  $k = 2\beta_0^3 + \gamma_0^3$ .

## 348 F Empirical analysis of $\mathcal{B}_0$

349 Equation (5) suggests that  $\|u_n\|_\infty$  may scale like  $\sqrt{n}$  when  $(\beta, \gamma) \in \mathcal{B}_0$ . Do we observe this in  
 350 practice? In Figure 5 we instead see bounded behaviour analogous to that of the region  $\mathcal{B}_+$  (as seen  
 351 in Figure 2), i.e. there is bounded growth rather than a  $\sqrt{n}$  scaling in the max-update. We note  
 352 that the bound Equation (5) is not violated; the predicted upper bound is indeed met, though the  
 353 prediction is just not *tight* relative to the true dynamics empirically. We suspect that the condition  
 354  $C(\beta, \gamma) = 0$  may be sensitive to numerical precision issues in practice, which is perhaps the reason  
 355 why we observe behaviour analogous to  $C(\beta, \gamma) > 0$  instead.

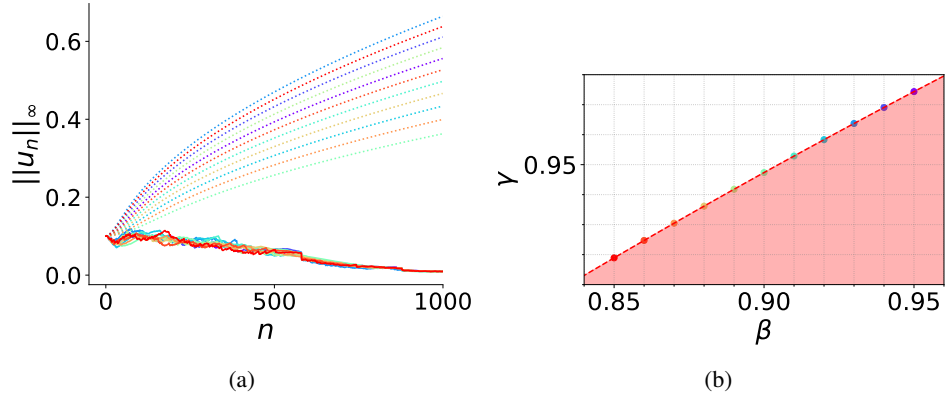


Figure 5: We plot the max-update over training for 11 points along the boundary  $\mathcal{B}_0$ . The theoretically predicted  $\sqrt{n}$ -like scaling is denoted by dotted lines. We use the same experimental setup as used to produce Figure 2, which we describe in Appendix H.

## 356 G Assumptions of Section 4.1

357 **Assumption 1.** To verify Assumption 1 we consider the transformer model setup described in  
 358 Appendix H with qk-norm [7, 8] enabled, such that the loss is scale invariant under each query and  
 359 key matrix. In Figure 6 we plot the cosine similarity  $\text{sim}(W_{100}, g_n) := \frac{\langle W_{100}, g_n \rangle}{\|W_{100}\| \|g_n\|} \in [-1, 1]$  for  
 360 iterations  $n = 1, \dots, 100$ , for each query and key matrix  $W$  in the model (a total of 32 matrices). We  
 361 observe a cosine similarity of  $\approx 0$  at  $n = 100$  (supporting Equation (6)) and an increase in cosine  
 362 similarity as  $n$  decreases, however the cosine similarity still remains negligible. We also demonstrate  
 363 that without qk-norm, the cosine similarity is no longer well behaved.

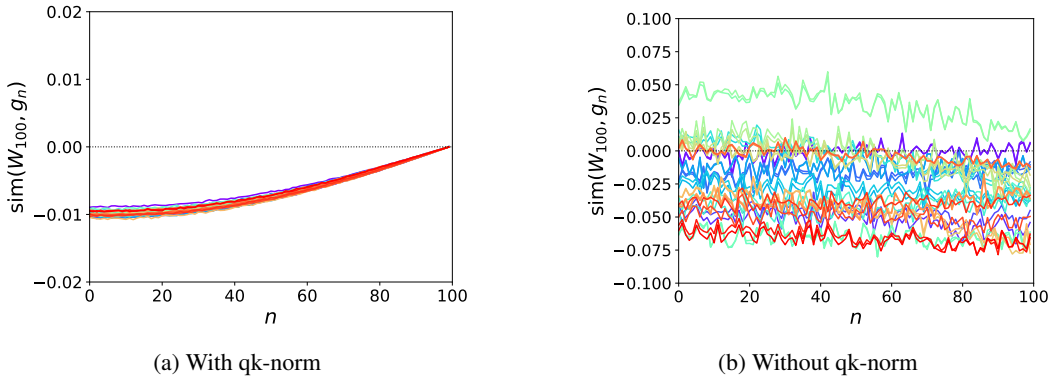


Figure 6: Plot of the cosine similarity between each query/key matrix at iteration  $m = 100$ , with the gradient at all previous iterations  $n$  (x-axis). In (a) we enable qk-norm, whereas in (b) the model does not include qk-norm.

364 **Assumption 2.** To verify Assumption 2 we choose 16 random parameter values from random  
 365 query/key matrices  $W$ , and plot their trajectory in regular training vs. their trajectory when using  
 366 coarse-graining on  $W$ . We display the results in Figure 7, finding a strong agreement.

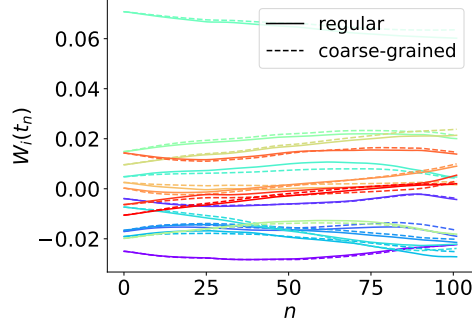


Figure 7: Plot of the trajectories, with and without coarse-graining, for 16 randomly selected parameters from query/key matrices.

367 A consequence of these assumptions is that  $\langle W(t), m_W(t) \rangle \approx 0$  (from Assumption 1 and Equation  
 368 (2)), and hence (in combination with Assumption 2) we have  $\langle W(t), u_W(t) \rangle \approx 0$ . We use this in  
 369 Section 4.1 in order to obtain an adaptive-normalization interpretation of training dynamics. Note that  
 370 though the cosine similarity is larger at earlier iterations in Figure 6, the kernel  $K_\gamma$  present in  $m_W$   
 371 places a lower weight on earlier iterations such that a larger cosine similarity maintains a negligible  
 372 effect.

## 373 H Experimental setups

374 **Architecture details.** Throughout the paper we consider a nanoGPT model [28] – which is a decoder-  
 375 only transformer model [29] – and a CNN model [30]. The nanoGPT model architecture we use has  
 376 6 layers, with 6 attention heads per layer. The embedding dimension is 384. We use a dropout of 0.2.  
 377 When training we use the shakespeare dataset associated with the nanoGPT repository [28], using  
 378 character-level tokenization, for a max input length of 256 tokens. The CNN model architecture we  
 379 use is shown in Figure 8 which we use for Figure 4.

380 **Continuous-time trajectory.** For experiments that require continuous-time trajectories, we use the  
 381 method described in Appendix I.

382 **Figure 1.** We describe the method for plotting normal curves in Appendix E.

383 **Figure 2.** We train the nanoGPT model using Adam at a learning rate of  $10^{-3}$  (and batch size 64)  
 384 for a range of adaptive hyperparameters  $(\beta, \gamma)$  for 1500 iterations, saving the max-update  $\|u_n\|_\infty$   
 385 at every iteration  $n$ . We compute the slope of  $\log \|u_n\|_\infty$  at  $n = 500$  using the values at  $n = 495$  and  
 386  $n = 505$ .

387 **Figure 3.** We use the same setup as Figure 2, and additionally use a learning rate warmup (linear, for  
 388 the first 100 iterations) and after warmup, a cosine decay down to  $10^{-4}$ . We train for 3000 iterations  
 389 in order for training to converge such that we can assess generalization performance. We evaluate on  
 390 a test split of the shakespeare dataset (on 200 iterations worth of test data) every 100 iterations to  
 391 obtain the test losses.

392 **Figure 4.** We train a CNN (using architecture in Figure 8) on CIFAR10 for 100 epochs and for each  
 393  $k \in \{1, \dots, 10\}$ , we sweep over learning rates  $\eta \in \{6e-5, 1e-4, 3e-4, 6e-4, 1e-3, 3e-3\}$  and weight  
 394 decays  $\lambda \in \{1e-6, 1e-5, 1e-4, 1e-3, 1e-2\}$  and plot the best test accuracy across this sweep. We use  
 395 a linear warmup for the first 2 epochs, and a cosine decay to  $\eta/10$  for the remaining epochs. We  
 396 evaluate on the test split every 5 epochs. We use  $(\tilde{\beta}, \tilde{\gamma}) = (0.999, 0.9)$  and a batch size of 512.

## 397 I Numerically solving Equation (4)

398 **Numerically solving for continuous dynamics.** Defining  $\psi(t) := \dot{\theta}(t)$ , we can write Equation (4)  
 399 as two coupled first-order DEs,

$$\dot{\theta}(t) = \psi(t)$$

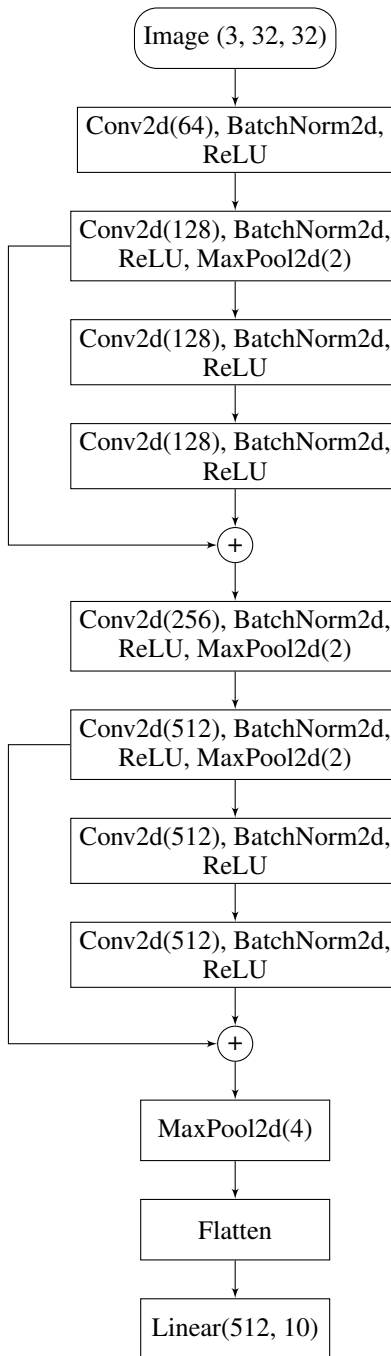


Figure 8: CNN model architecture used in Figure 4. All convolutional layers use a kernel size of  $3 \times 3$  and a padding of 1.

$$\dot{\psi}(t) = -\frac{2}{\eta^{2p}} \left( \lambda\eta\theta(t) + \eta^p\psi(t) + \eta \frac{\sqrt{1 - \beta^{1+t/\eta^p}}}{1 - \gamma^{1+t/\eta^p}} \frac{m(t)}{\sqrt{v(t)}} \right)$$

400 and numerically solve via Euler's method,

$$\theta((n+1)\Delta t) \approx \theta(n\Delta t) + \Delta t\psi(n\Delta t)$$

$$\text{with } \psi(n\Delta t) \approx \psi((n-1)\Delta t) + \Delta t\dot{\psi}((n-1)\Delta t)$$

401 defining  $\Delta t := \eta^p/K$ . In our numerical validation experiments, we use the timescale  $p = 1$  and  
402  $K = 100$ .

403 To compute  $\dot{\psi}(n\Delta t)$  via the above DE we must compute

$$\begin{aligned} m(t) &= \int_0^t d\tau M(\tau, t)g(\tau) = \int_0^{t-\Delta t} d\tau M(\tau, t)g(\tau) + \int_{t-\Delta t}^t d\tau M(\tau, t)g(\tau) \\ &= \exp\left(-\frac{1-\gamma}{\eta^p\gamma}\Delta t\right) \left[ \underbrace{\int_0^{t-\Delta t} d\tau M(\tau, t-\Delta t)g(\tau)}_{\text{from } n-1 \text{th update step}} + \underbrace{\int_{t-\Delta t}^t d\tau M(\tau, t-\Delta t)g(\tau)}_{\approx \Delta t \frac{1-\gamma}{\eta^p\gamma} g(t-\Delta t)} \right] \end{aligned}$$

404 **Empirically validating  $\varphi(t)$ .** We note that we can write Equation (12) approximately as

$$\varphi(n\Delta t) \approx \varphi_0 e^{-2\lambda n\Delta t} + \Delta t \sum_{m=0}^{n-1} e^{-2\lambda(n-m)\Delta t} f(m\Delta t)$$

where

$$f(t) := \eta \|u_W(t)\|^2 - 2 \langle W(t), u_W(t) \rangle$$

405 We then compare this expression to the true value of  $\|W(n\Delta t)\|^2$  to validate Equation (12) in  
406 Appendix K.

## 407 J Implicit meta-adaptive effect

**Deriving an expression for  $\|W(t)\|^2$ .** Applying the inner product  $\langle W(t), \cdot \rangle$  to both sides of Equation (7), we arrive at the differential equation

$$\frac{1}{2}\eta\ddot{\varphi} + \dot{\varphi} + 2\lambda\varphi \approx \eta\|\dot{W}\|^2 - 2\langle W, u_W \rangle$$

where we have defined the squared norm  $\varphi(t) := \|W(t)\|^2$ . One can show (see further below) that to leading order,  $\|\dot{W}\|^2 \approx \|u_W\|^2$  and  $\eta\ddot{\varphi} = O(\eta^2) + O(\lambda\eta)$ , hence we neglect the  $\ddot{\varphi}$  term, with the resulting first-order differential equation,

$$\dot{\varphi} + 2\lambda\varphi \approx \eta\|u_W\|^2 - 2\langle W, u_W \rangle$$

408 which when integrated, gives

$$\|W(t)\|^2 \approx \|W(0)\|^2 e^{-2\lambda t} + \int_0^t d\tau e^{-2\lambda(t-\tau)} (\eta\|u_W(\tau)\|^2 - 2\langle W(\tau), u_W(\tau) \rangle) \quad (12)$$

409 We empirically verify this expression in Appendix K.

410 **A further approximation.** In order to derive Equation (12) we did not need to use the property  
411 of scale invariance. We will now make use of this property, in combination with the following two  
412 simplifying assumptions:



**Assumption 1.** We will assume that

$$\langle W(t), g_W(\tau) \rangle \approx 0 \quad \forall \tau \leq t$$

**Assumption 2.** We will use a coarse-graining approximation for  $v_W(t)$  (using Equation (2)):

$$v_W(t) \equiv \int_0^t d\tau K_\beta(\tau, t) g_W(\tau)^{\odot 2} \approx \int_0^t d\tau \tilde{K}_\beta(\tau, t) \|g_W(\tau)\|^2 =: \tilde{v}_W(t)$$

413 with  $\tilde{K}_\beta(\tau, t) := K_\beta(\tau, t)/N$  where  $N$  is the total number of entries of  $W$ , i.e. we replace the entries  
414 of  $g_W(\tau)^{\odot 2}$  with the average entry across the tensor.

Assumption 1 is intuitive given the properties of high-dimensional spaces<sup>1</sup>, but also Equation (6) is suggestive of such a result. We provide empirical support for this assumption in Appendix G. We also show (in Appendix G) that replacing  $v_W$  with the coarse-grained version  $\tilde{v}_W$  has little effect on training dynamics. A consequence of these assumptions is that  $\langle W(t), u_W(t) \rangle \approx 0$ , allowing us to approximate,

$$\|W(t)\|^2 \approx \|W(0)\|^2 e^{-2\lambda t} + \eta \int_0^t d\tau e^{-2\lambda(t-\tau)} \|u_W(\tau)\|^2$$

415  $\|W(t)\|^2$  as a moving average. Restating Equation (8),

$$\|W(t)\|^2 \approx \|W(0)\|^2 e^{-2\lambda t} + \eta \int_0^t d\tau e^{-2\lambda(t-\tau)} \|u_W(\tau)\|^2$$

which takes an analogous form to a moving-average of  $\|u_W\|^2$  in continuous-time. Specifically, consider a moving-average of  $(x_0, x_1, x_2, \dots)$ ,

$$w_n = (1 - \alpha)w_{n-1} + \alpha x_n$$

416 Then in continuous time, with  $w_n = w(t_n)$  and  $x_n = x(t_n)$ , we have shown in Appendix E that

$$w(t) = w(0) \exp\left(-\frac{1-\alpha}{\eta^p \alpha} t\right) + \frac{1-\alpha}{\eta^p \alpha} \int_0^t d\tau \exp\left(-\frac{1-\alpha}{\eta^p \alpha} (t-\tau)\right) x(\tau)$$

417 Comparing this expression to Equation (8) allows us to view  $\|W\|^2$  as a moving-average of  $\|u_W\|^2$ .

418 **Weight direction dynamics.** To obtain Equation (9) from Equation (7), we write  $W = r\hat{W}$  with  
419  $r := \|W\|$  (and  $\|\hat{W}\| = 1$ ), which after substituting into Equation (7) becomes

$$\left(\frac{1}{2}\eta\ddot{r} + \dot{r} + \lambda r\right)\hat{W} + (\eta\dot{r} + r)\dot{\hat{W}} + \frac{1}{2}\eta r\ddot{\hat{W}} = -u_W \quad (13)$$

Note that taking the derivative of  $\langle \hat{W}, \hat{W} \rangle = 1$  implies  $\langle \dot{\hat{W}}, \hat{W} \rangle = 0$ , and further  $\langle \ddot{\hat{W}}, \hat{W} \rangle = -\|\dot{\hat{W}}\|^2$ . As a result, applying  $\langle \cdot, \hat{W} \rangle$  to Equation (13) gives

$$\frac{1}{2}\eta\ddot{r} + \dot{r} + \lambda r = \frac{1}{2}\eta r \|\dot{\hat{W}}\|^2$$

Substituting this expression back into Equation (13) and neglecting higher-order terms (noting that from above,  $\eta\dot{r} = O(\eta^2)$ ), we arrive at

$$\frac{1}{2}\eta\ddot{\hat{W}} + \dot{\hat{W}} + \Lambda\hat{W} = -\frac{1}{r}u_W$$

420 where we define  $\Lambda := \frac{1}{2}\eta r \|\dot{\hat{W}}\|^2$ .

---

<sup>1</sup>For example the Johnson–Lindenstrauss lemma implies that in high-dimensional spaces, randomly chosen vectors are likely to be almost orthogonal.

**Leading order approximations.** We can write

$$\dot{W} = -\lambda W - \frac{1}{2}\eta\ddot{W} - u_W$$

Squaring both sides and using  $\langle W, u_W \rangle \approx 0$  (a consequence of Assumption 1 and Assumption 2) results in

$$\|\dot{W}\|^2 = \|u_W\|^2 + \lambda\eta\langle W, \ddot{W} \rangle + \eta\langle \ddot{W}, u_W \rangle + \frac{\eta^2}{4}\|\ddot{W}\|^2 + \lambda^2\|W\|^2$$

421 hence to leading order in  $(\eta, \lambda)$ ,  $\|\dot{W}\|^2 \approx \|u_W\|^2$ .

## 422 K Empirical validation of continuous dynamics

423 In Figure 9 we empirically verify the differential equation form for AdamW (Equation (4)) and the  
 424 prediction for  $\|W(t)\|^2$  (Equation (12)). We use the method described in Appendix I – with  $K = 100$ ,  
 425  $p = 1$  and  $\eta = 10^{-3}$  – to numerically solve Equation (4), obtaining a continuous-time trajectory in  
 426 parameter space. We train a nanoGPT model for 100 iterations using setup in Appendix H, with  
 427  $(\beta, \gamma) = (0.999, 0.9)$ . To produce Figure 9a we randomly select 16 parameters from the model and  
 428 plot their trajectories. For Figure 9b we randomly select 16 query/key matrices  $W$  and plot the  
 429 trajectory of  $\|W\|^2$  for each.

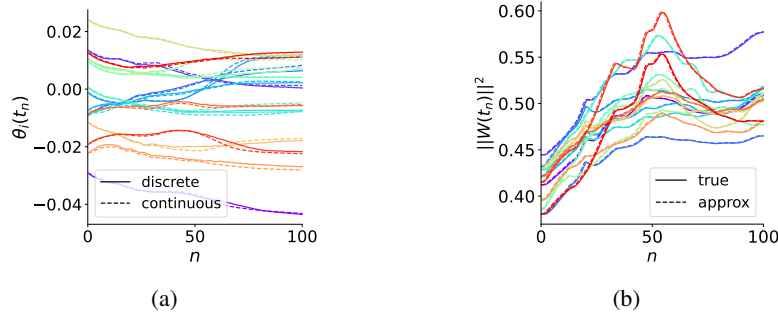


Figure 9: (a) Parameter trajectories and (b) norm trajectories over 100 iterations.

## 430 L Symmetries of the transformer architecture

Say  $\Phi_s : \Theta \rightarrow \Theta$  describes some continuous invariance parameterised by some continuous parameter  $s \in \mathbb{R}^S$ , i.e.

$$L(\Phi_s(\theta)) = L(\theta) \quad \forall s \in \mathbb{R}^S$$

431 with  $\Phi_0(\theta) = \theta \quad \forall \theta \in \Theta$ . Differentiating the above with respect to  $s_i$  and evaluating at  $s = 0$ , we  
 432 obtain

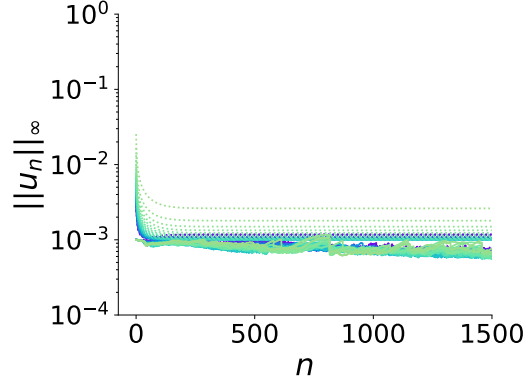
$$\left\langle \frac{\partial \Phi_s}{\partial s_i} \Big|_{s=0}, \nabla_{\theta} L(\theta) \right\rangle = 0 \quad \forall i = 1, \dots, S \quad (14)$$

**Scale invariance.** In the special case of scale invariance with respect to a weight  $W$ , if we write  $\theta = (W, \phi)$  then the relevant transformation is  $\Phi_s(\theta) = (sW, \phi)$ , and Equation (14) becomes,

$$\langle W, \nabla_W L(\theta) \rangle = 0$$

## 433 M Max-update trajectory in $\mathcal{B}_+$

434 To visualize the theoretical bounds in the case of  $(\beta, \gamma) \in \mathcal{B}_+$  in more detail, we plot the results of  
 435 Figure 2a but restricted to  $(\beta, \gamma) \in \mathcal{B}_+$  in Figure 10. We indeed see that the theoretically predicted  
 436 bounds (dotted lines) are approximately satisfied.

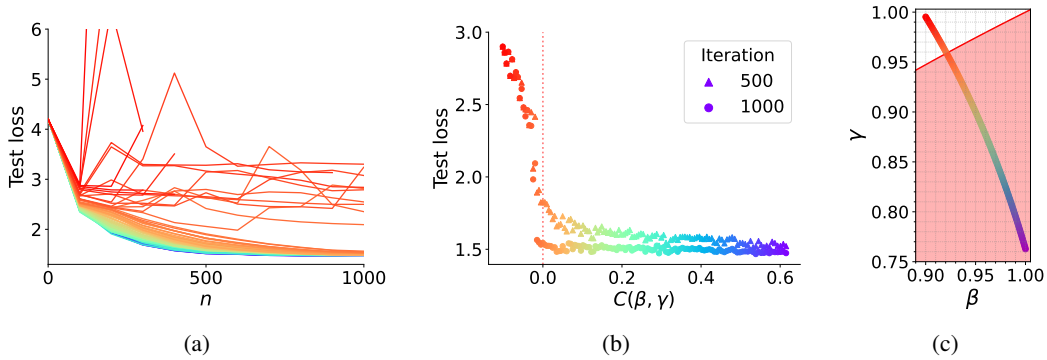


(a)

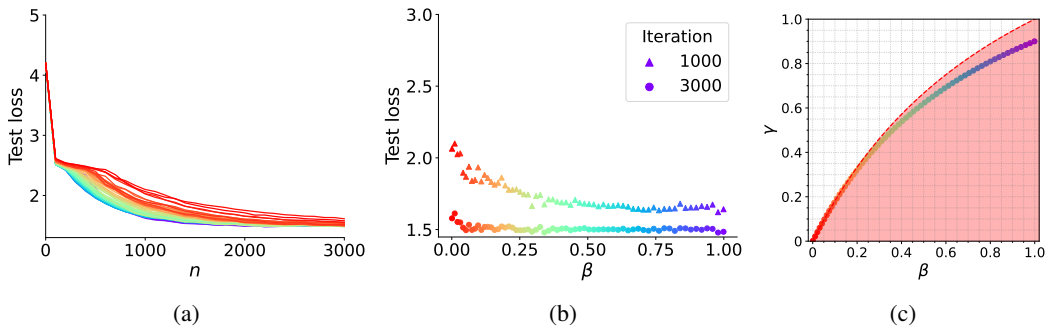
Figure 10: We plot the results of Figure 2a but restrict to  $(\beta, \gamma)$  that lie in  $\mathcal{B}_+$ .

### 437 N Supporting data for Section 3.2

438 In Section 3.2 we found that a larger value of  $C(\beta, \gamma)$  correlated with a faster generalization. We  
 439 further verify this in Figure 11 by instead taking points along the normal curve through  $(0.95, 0.9)$ ,  
 440 but otherwise an identical experimental setup to the one used to produce Figure 3.

Figure 11: Reproduced Figure 3 but instead for the normal curve through the point  $(0.95, 0.9)$ .

441 We also consider the level curve  $\mathcal{B}_{C(\bar{\beta}, \bar{\gamma})}$ , the results of which we plot in Figure 12. We see a slower  
 442 rate of generalization for points closer to the origin, as expected, given that such points will have a  
 443 smaller value of  $C(\beta, \gamma)$  relative to other points along their normal curve.

Figure 12: Reproduced Figure 3 but instead for the level curve through the point  $(0.999, 0.9)$ .

444 **O Attempting to correct exponential growth with learning rate annealing**

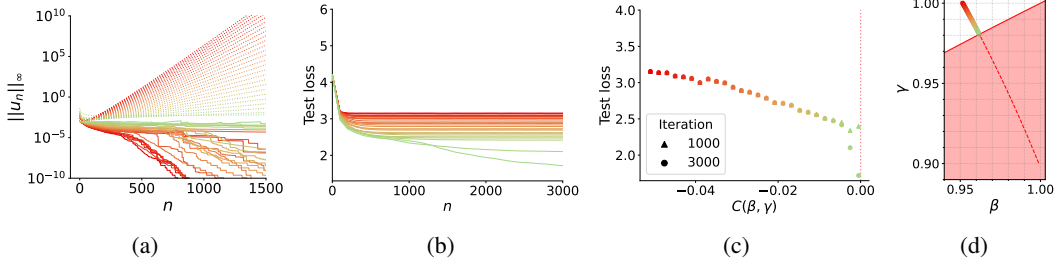


Figure 13: Reproduced Figure 2 and Figure 3 restricted to the region  $\mathcal{B}_-$  and using learning rate annealing described below.

445 Since the parameter update is  $\theta_{n+1} - \theta_n \equiv -\eta u_n$ , it appears that a well chosen step-dependent  
 446 learning rate  $\eta_n$  may correct the exponential growth in the case of  $C(\beta, \gamma) < 0$  described by  
 447 Equation (5), which occurs via the factor  $\exp(n|C(\beta, \gamma)|/2)$ . Particularly, given the tightness of the  
 448 bound seen in Figure 2b, we could consider the learning rate

$$\eta_n = \eta_0 \exp(-n|C(\beta, \gamma)|/2) \quad (15)$$

449 at step  $n$ , which results in a bound on  $\|\theta_{n+1} - \theta_n\|_\infty \equiv \eta_n \|u_n\|_\infty$  that is equivalent to the case  
 450 of  $C(\beta, \gamma) > 0$  with learning rate  $\eta_0$ . However when we run this learning rate annealing strategy  
 451 (Equation (15)) in practice, we find that it is too strong, as shown in Figure 13; the max-update  
 452 becomes very small, and we do not observe good generalization performance. Hence, even though  
 453 this annealing strategy for  $\mathcal{B}_-$  results in the regions  $\mathcal{B}_+$  and  $\mathcal{B}_-$  possessing the same bounds on  
 454  $\|\theta_{n+1} - \theta_n\|_\infty$ , their training properties are still quite different. Analysis of weaker annealing  
 455 methods are out of scope for this work. It is unclear whether correcting such exponential growth in  
 456  $\mathcal{B}_-$  by choosing an appropriate annealing method would provide any benefits to simply choosing  
 457  $(\beta, \gamma) \in \mathcal{B}_+$ .

458 **P Choice of  $k$ -Adam hyperparameters**

459 The moving-average  $m_n^{(k)}$  will have a prefactor of  $(1 - \beta_1) \cdots (1 - \beta_k)$ , and we can consider a uniform strategy  
 460  $\beta_1 = \cdots = \beta_k$  and enforce that this prefactor is equal to the typical prefactor for Adam/AdamW, i.e.  
 462

$$(1 - \beta_i)^k = 1 - \tilde{\beta} \\ \implies \beta_i = 1 - (1 - \tilde{\beta})^{1/k}$$

463 which we can do similarly for  $\gamma_i$ .

464 **Naive strategy.** In Figure 14 we show a zoomed-out version of Figure 4 to show the performance of the naive  
 465 strategy across all  $k$ , emphasizing the importance of well-selected  $k$ -Adam hyperparameters.  
 466

467 We also found the strategy  $\beta_i = \tilde{\beta}^{1/k}, \gamma_i = \tilde{\gamma}^{1/k}$  to perform badly.

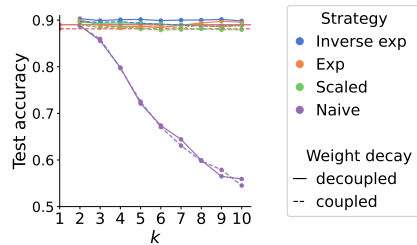


Figure 14: Zoomed out version of Figure 4 to show performance of the naive strategy.

468 **Q Adaptive normalization and  $k$ -Adam**

469 For a sequence of tensors  $x_{0:n} \equiv (x_0, \dots, x_n)$ , we define the bias-corrected moving-average operator  
 470  $M_\gamma$ ,

$$M_\gamma(x_{0:n}) := \frac{1-\gamma}{1-\gamma^{n+1}}(x_n + \gamma x_{n-1} + \dots + \gamma^n x_0)$$

471 We justify bias correction in Appendix D. We then define the *adaptive-normalization* operator  $A_{\gamma,\beta}$ ,

$$A_{\gamma,\beta}(x_{0:n}) := \frac{M_\gamma(x_{0:n})}{\sqrt{M_\beta(x_{0:n}^{\odot 2})}}$$

472 **Adaptive optimization.** We can describe Adam/AdamW with hyperparameters  $(\beta, \gamma)$  simply as

$$\text{Adam/AdamW: } \theta_{n+1} = \theta_n - \eta A_{\gamma,\beta}(g_{0:n})$$

473 i.e. updating by the adaptive-normalization of gradients  $g_{0:n}$ , where  $g_i \equiv \nabla_\theta L(\theta_i)$ .

474  $k$ -Adam extends Adam/AdamW by applying the adaptive normalization operator  $k$ -times with  
475 hyperparameters  $(\beta_{1:k}, \gamma_{1:k})$ ,

$$k\text{-Adam: } \theta_{n+1} = \theta_n - \eta (A_{\gamma_k, \beta_k} \circ \dots \circ A_{\gamma_1, \beta_1})(g_{0:n})$$

476 which is equivalent to Equation (10). Specifically, the intermediate updates take the form  $u_n^{(i)} \equiv$   
477  $(A_{\gamma_i, \beta_i} \circ \dots \circ A_{\gamma_1, \beta_1})(g_{0:n})$  (meaning that  $u_n^{(i+1)}$  is an adaptive-normalization of  $u_n^{(i)}$ ).

**Continuous-time properties.** We move into continuous-time, with  $x_n \equiv x(t_n)$  where  $t_n = n\eta^p$ . Appendix E has shown that

$$M_\gamma(x_{0:n}) = \int_0^{t_n} d\tau K_\gamma(\tau, t)x(\tau), \quad K_\gamma(\tau, t) := \frac{1-\gamma}{\eta^p \gamma} \exp\left(-\frac{1-\gamma}{\eta^p \gamma}(t-\tau)\right)$$

478 to first-order. Then using Cauchy-Schwarz on  $A_{\gamma,\beta}$ , Appendix E has also shown that

$$\|A_{\gamma,\beta}(x_{0:n})\|_\infty \leq \frac{\sqrt{1-\beta^{n+1}}}{1-\gamma^{n+1}} \frac{1-\gamma}{\gamma} \sqrt{\frac{\beta}{1-\beta}} B_n(\beta, \gamma) \quad (16)$$

479 where  $B_n$  is defined as in Section 3.1. Note that this bound is in general independent of the elements  
480  $x_{0:n}$ . Since each  $k$ -Adam sub-update  $u_n^{(i)} = A_{\gamma_i, \beta_i}((A_{\gamma_{i-1}, \beta_{i-1}} \circ \dots \circ A_{\gamma_1, \beta_1})(g_{0:n}))$ , this bound  
481 holds analogously for  $u_n^{(i)}$ .

## 482 R Algorithm for $k$ -Adam

483 We provide an explicit algorithm for the  $k$ -Adam optimizer below.

---

**Algorithm 1**  $k$ -Adam update rule

---

**given** learning rate  $\eta \in \mathbb{R}$ , weight decay  $\lambda \in \mathbb{R}$ , *coupled*  $\in \{\text{False}, \text{True}\}$ , hyperparameters  $(\beta_{1:k}, \gamma_{1:k})$ , epsilon  $\epsilon$  ( $10^{-30}$  by default)  
**initialize** step count  $n \leftarrow 0$ , initial parameter  $\theta_0 \in \mathbb{R}^p$ ,  $(m_i, v_i) \leftarrow (0, 0)$  for  $i = 1, \dots, k$   
**repeat**  
   $g \leftarrow \nabla_{\theta} L(\theta_n)$   
  **if** *coupled* **then**  
     $g \leftarrow g + \lambda \theta_n$   
   $\hat{g} \leftarrow g$   
  **for**  $i = 1, \dots, k$  **do**  
     $m_i \leftarrow \gamma_i m_i + (1 - \gamma_i) \hat{g}$   
     $v_i \leftarrow \beta_i v_i + (1 - \beta_i) \hat{g}^{\odot 2}$   
     $C \leftarrow \sqrt{1 - \beta_i^{n+1}} / (1 - \gamma_i^{n+1})$   
     $\hat{g} \leftarrow C m_i / (\sqrt{v_i} + \epsilon)$   
   $u \leftarrow \hat{g}$   
  **if not** *coupled* **then**  
     $u \leftarrow u + \lambda \theta_n$   
   $\theta_{n+1} \leftarrow \theta_n - \eta u$   
   $n \leftarrow n + 1$   
**return** parameter trajectory  $(\theta_0, \theta_1, \dots)$ 

---

484 **S**  $k$ -Adam update bounds

485 Theoretical guarantees in the form of bounds – as shown for Adam/AdamW in Section 3 – hold  
486 analogously for  $k$ -Adam. Specifically, since  $u_n^{(i)}$  is an adaptive-normalization of  $u_{0:n}^{(i-1)}$  for each  
487  $i = 1, \dots, k$ , we have the bound

$$\|u_n^{(i)}\|_{\infty} \leq \frac{\sqrt{1 - \beta_i^{n+1}}}{1 - \gamma_i^{n+1}} \frac{1 - \gamma_i}{\gamma_i} \sqrt{\frac{\beta_i}{1 - \beta_i}} B_n(\beta_i, \gamma_i)$$

488 with  $B_n$  defined as in Equation (5). As a result, we can view  $k$ -Adam (for  $k > 1$ ) as possessing  
489 stronger stability guarantees than Adam – e.g.  $k$ -Adam updates by  $u_n^{(k)}$ , an adaptive-normalization of  
490 a *bounded* quantity  $u_{0:n}^{(k-1)}$  due to the above, whereas Adam updates by an adaptive-normalization of  
491  $g_{0:n}$ , however the gradients  $g_{0:n}$  have no such bound guarantees.