Dissecting the Effects of SGD Noise in Distinct Regimes of Deep Learning

Antonio Sclocchi [1] **Mario Geiger** [2] **Matthieu Wyart** [1]

Abstract

Understanding when the noise in stochastic gradient descent (SGD) affects generalization of deep neural networks remains a challenge, complicated by the fact that networks can operate in distinct training regimes. Here we study how the magnitude of this noise T affects performance as the size of the training set P and the scale of initialization α are varied. For gradient descent, α is a key parameter that controls if the network is 'lazy' ($\alpha \gg 1$) or instead learns features ($\alpha \ll 1$). For classification of MNIST and CIFAR10 images, our central results are: *(i)* obtaining phase diagrams for performance in the (α, T) plane. They show that SGD noise can be detrimental or instead useful depending on the training regime. Moreover, although increasing T or decreasing α both allow the net to escape the lazy regime, these changes can have opposite effects on performance. *(ii)* Most importantly, we find that the characteristic temperature T_c where the noise of SGD starts affecting the trained model (and eventually performance) is a power law of P. We relate this finding with the observation that key dynamical quantities, such as the total variation of weights during training, depend on both T and P as power laws. These results indicate that a key effect of SGD noise occurs late in training, by affecting the stopping process whereby all data are fitted. Indeed, we argue that due to SGD noise, nets must develop a stronger 'signal', i.e. larger informative weights, to fit the data, leading to a longer training time. A stronger signal and a longer training time are also required when the size of the training set P increases. We confirm these views in the perceptron model, where signal and noise can be precisely measured. Interestingly, exponents characterizing the effect of SGD depend on the density of data near the decision boundary, as we explain.

1. Introduction

Optimizing the generalization performances of over-parametrized neural networks is one of the main challenges in machine learning. A crucial role is played by gradient-based training algorithms, which converge to solutions which generalize well also when no explicit regularization of the model is used (Zhang et al., 2021). Mini-batch stochastic gradient descent (SGD) is the workhorse algorithm to train modern neural networks. Yet, key aspects of these algorithms are debated.

Effect on performance: A popular idea has been that mini-batch SGD can generalize better than full batch gradient descent (GD) (Heskes & Kappen, 1993; LeCun et al., 2012; Keskar et al., 2016; Hochreiter & Schmidhuber, 1997; Jastrzebski et al., 2017; Chaudhari et al., 2019), yet this view is debated (Hoffer et al., 2017; Dinh et al., 2017; Shallue et al., 2018; Zhang et al., 2019). In fact, comparing SGD and GD at fixed number of training epochs leads to a generalization gap (Keskar et al., 2016) that can be closed by training longer with a fixed number of training steps (Hoffer et al., 2017; Smith et al., 2020). More generally, the choice of the computational budget can affect which algorithm performs better (Shallue et al., 2018; Smith et al., 2020).

Theories for the role of SGD: Several works have argued that larger SGD stochasticity leads the dynamics toward flatter minima of the loss landscape, and it has been argued that this effect leads to improved performances (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Zhang et al., 2018; Smith & Le, 2018; Wu et al., 2018). By contrast, other studies suggest that the SGD noise biases the model in a manner similar to initializing the network with small weights, and helps recovering sparse predictors (Blanc et al., 2020; HaoChen et al., 2021; Pesme et al., 2021).

1.1. This work

In this work, we clarify these two debates by performing systematic empirical studies of how performance is affected

[1]Institute of Physics, École Polytechnique Fédérale de Lausanne, Lausanne, 1015, Switzerland [2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Antonio Sclocchi <antonio.sclocchi@epfl.ch>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

by the noise magnitude of SGD or temperature T (the ratio between the learning rate η and the batch size B (Jastrzebski et al., 2017; Zhang et al., 2019; Smith et al., 2020)), by the initialization scale α, and by the size of the training set P. The initialization scale α was rarely considered in empirical studies so far, yet it governs the training regimes in which nets operate. For large α, tiny changes of weights are sufficient to fit the data: the predictor is approximately linear in its parameters, corresponding to the *kernel* or *lazy* regime (Jacot et al., 2018; Chizat et al., 2019). By contrast for small initialization, networks can learn the relevant features of the task and the dynamics is non-linear, corresponding to the so-called feature-learning regime (Rotskoff & Vanden-Eijnden, 2018; Mei et al., 2018; Sirignano & Spiliopoulos, 2020).

We also deal with the computational budget issue by considering the hinge loss $l(y, \hat{y}) = (1-y\hat{y})^+$, allowing us to train networks until the time t^* where the loss is strictly zero, and the dynamics stops. Importantly, this training methodology is not restrictive, as it yields similar outcomes compared to training with the cross-entropy loss and performing early stopping. [1]

Our central empirical results are:

(i) obtaining phase diagrams for performance in the (α, T) plane. They show that SGD noise can be detrimental or instead useful depending on the training regime, even in the absence of budget constraints. This observation clarifies why different conclusions on the benefits of SGD were previously made.

(ii) Although we find that increasing T or decreasing α both allow the net to escape the lazy regime, these changes can have opposite effects on performance, in disagreement with simple models (Pesme et al., 2021).

(iii) We reveal that several observables characterizing the dynamics follow scaling laws in T and P. Denote by Δw the relative weight variation accumulated after training and t^* the training time defined as the learning rate times the number of training steps required to bring a hinge loss to zero. We find that

$$\Delta w \sim T^{\delta} P^{\gamma}, \quad t^* \sim T P^{b}, \tag{1}$$

where δ, γ, b are exponents depending on the model and the training regime.

(iv) Most importantly, we find that SGD noise starts affecting the trained model at a characteristic temperature scale T_c which depends on the size of the training set P as

$$T_c \sim P^{-a}, \tag{2}$$

where a is a model-dependent exponent. This result can be understood as follows. For the lazy regime $\alpha \gg 1$, T_c is the temperature at which the network exits the lazy regime, i.e. $\Delta w = \mathcal{O}(1)$. Together with 1, it gives $a = \gamma/\delta$ in agreement with our observations. For the feature regime, T_c corresponds to the transition between a low-T regime, where Δw is unaffected by SGD noise and is found to scale as $\Delta w \sim P^{\zeta}$, and a high-T regime where 1 applies. These two empirical relationships imply that $T_c \sim P^{-a}$, with the exponent a satisfying $a = (\gamma - \zeta)/\delta$, consistent with our experimental observations. For fully-connected architectures, we observe that T_c also characterizes the temperature where SGD affects performance. By contrast, for CNNs such a characteristic temperature is hard to extract from the performance curves, while it is clearly identified from the weight variation.

(v) We rationalize these findings using a teacher-student perceptron model, for which Δw and t^* also display power-law dependence on T and P. We show that SGD noise increases weights in directions irrelevant to the task, implying that the correct weights must grow much larger to fit data, thus increasing both t^* and Δw. We compute the dependence of these effects on the size of the training set, and show that this dependence varies qualitatively with the distribution of data near the decision boundary.

Overall, instead of a static view where SGD noise would bias networks toward broader minima of the population loss, these results support a dynamical viewpoint where SGD noise delays the end of training. This effect allows the weights to grow more, affecting performance the most when the network escapes the lazy regime.

1.2. Related works

More related works are indicated in Appendix A.

2. Empirical analysis

2.1. General setting and notation

We consider binary classification on the data $\{\boldsymbol{x}_\mu\}_{\mu=1,\dots,P}$ with labels $\{y_\mu\}_{\mu=1,\dots,P} \in \{-1, +1\}$. P is the size of the training set. Given a predictor $\hat{y}_\mu$, the hinge loss on the sample μ is defined as $l(y_\mu, \hat{y}_\mu) = (1 - y_\mu \hat{y}_\mu)^+$, where $(x)^+ = \max(0, x)$. To control between feature and lazy training, we multiply the model output by α (Chizat et al., 2019). For the hinge loss, this is equivalent to changing the loss margin to $1/\alpha$. Therefore we study the training loss

$$L(\boldsymbol{w}) = \frac{1}{P} \sum_{\mu=1}^{P} (\alpha^{-1} - y_\mu F(\boldsymbol{w}, \boldsymbol{x}_\mu))^+, \tag{3}$$

[1] In Appendix E we verify that the two training methodologies give identical power-law dependencies for all the quantities we analyse in this work.

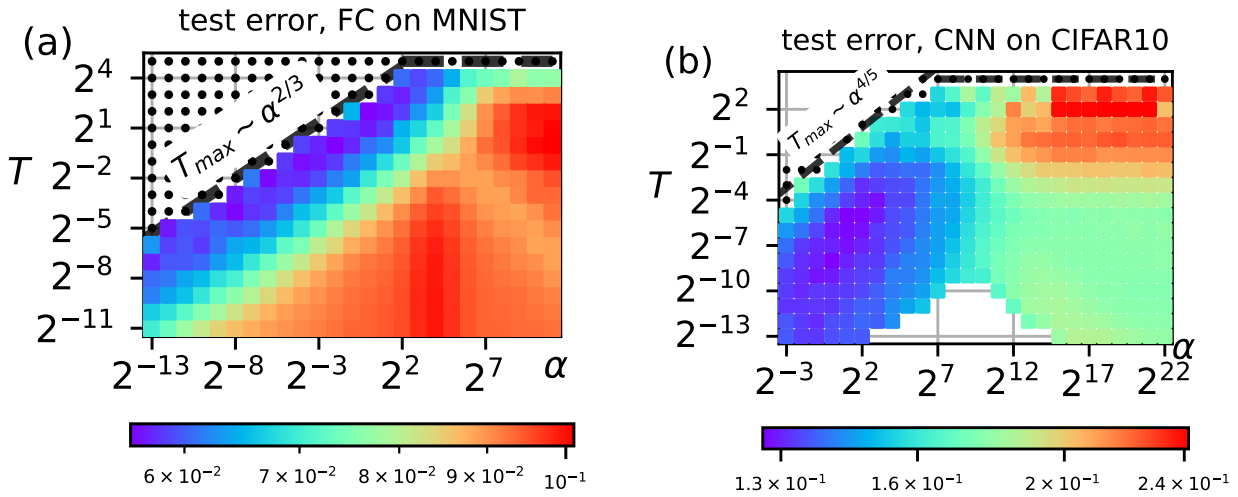

Figure 1. **Test error of deep networks on image data-sets, for varying T and α and fixed $P = 1024$.** Batch size B is kept fixed and learning rate η is varied ($T = \eta/B$). **(a)** 5-hidden layers fully-connected network (FC) on parity MNIST with $B = 16$. **(b)** 9-hidden layers CNN (MNAS) on CIFAR (animals vs the rest) with $B = 64$. Black dots correspond to training runs that do not converge. The black dashed lines indicates the maximal temperatures T_{max}. The lowest test error is achieved in the feature regime ($\alpha \ll 1$), for a temperature $T_{opt} \propto T_{max}$. In the lazy regime ($\alpha \gg 1$), performance is best for the highest T for FC on MNIST. Although it is not apparent here for CNN on CIFAR, it is also the case as the training set increases (see below). In *(a)*, the number of hidden layers is $D = 5$ and $T_{max} \sim \alpha^{\frac{D-1}{D+1}} = \alpha^{\frac{2}{3}}$ (black dashed line) when $\alpha \ll 1$ as argued in 26. Similarly in *(b)*, $D = 9$ and $T_{max} \sim \alpha^{\frac{D-1}{D+1}} = \alpha^{\frac{4}{5}}$ (black dashed line).

where $F(\boldsymbol{w}, \boldsymbol{x}_\mu)$ is the model predictor with weights $\boldsymbol{w}$ on the datum $\boldsymbol{x}_\mu$. The model predictor at time t corresponds to $F(\boldsymbol{w}, \boldsymbol{x}_\mu) = f(\boldsymbol{w}^t, \boldsymbol{x}_\mu) - f(\boldsymbol{w}^0, \boldsymbol{x}_\mu)$, where $f(\boldsymbol{w}^t, \boldsymbol{x}_\mu)$ is the output of a neural net with weights $\boldsymbol{w}^t$ at time t and $\boldsymbol{w}^0$ are the weights at initialization. For a network of width h, the weights are initialized as Gaussian random numbers with standard deviation $1/\sqrt{h}$ for the hidden layers and $1/h$ for the output layer. Such an initialization ensures that the feature learning limit corresponds to $\alpha \ll 1$ while the lazy training limit corresponds to $\alpha \gg 1$, and that every layer has a similar change of weights (Geiger et al., 2020; Yang & Hu, 2021).

The stochastic gradient descent updating equation is:

$$\boldsymbol{w}^{t+\eta} = \boldsymbol{w}^t + \frac{\eta}{B} \sum_{\mu \in \mathbb{B}_t} \theta\left(\alpha^{-1} - y_\mu F(\boldsymbol{w}, \boldsymbol{x}_\mu)\right) y_\mu \nabla_{\boldsymbol{w}} f(\boldsymbol{w}^t, \boldsymbol{x}_\mu) \tag{4}$$

where $\theta(x)$ is the Heaviside step function, $\mathbb{B}_t \subset \{1, ..., P\}$ is the batch at time t and B is its size. The time t corresponds to the number of training steps times the learning rate η. The batch $\mathbb{B}_t$ is randomly selected at each time step among all the P data. The learning rate η is kept constant during training. The end of training is reached when $L(\boldsymbol{w}^{t^*}) = 0$.

The batch size B is taken small enough to be in the "noise dominated" regime (Smith et al., 2020; Zhang et al., 2019), where the dynamics depends on the SGD temperature $T = \eta/B$. Empirical verification of this fact is provided in Appendix G.1.

Below we use a 5-hidden-layers fully-connected (FC) network and a 9-hidden-layers convolutional neural network (CNN) (MNAS architecture (Tan et al., 2019)). In Appendix C we report data also for a 3-hidden layers CNN (simple-CNN). We consider the binary datasets MNIST (even vs odd numbers) and CIFAR10 (animals vs the rest). All the networks use ReLU as activation functions. The code with all the details of the experiments is provided at https://tinyurl.com/mrys4uyp.

2.2. Performance in the (α, T) phase diagram

Fig. 1-(a) shows the test error for a FC network trained on MNIST and Fig. 1-(b) shows the same quantity obtained after training a CNN on CIFAR10. The black dots correspond to training loss exploding to infinity due to too large learning rate. Therefore, the dashed back lines indicate the maximal temperature T_{max} for which SGD converges.

From Fig. 1 we make the following observations:

(i) In the feature regime, both T_{max} and the temperature of optimal performance T_{opt} follow $T_{max} \sim T_{opt} \sim \alpha^k$. In Appendix B, we relate the exponent k to the number D of

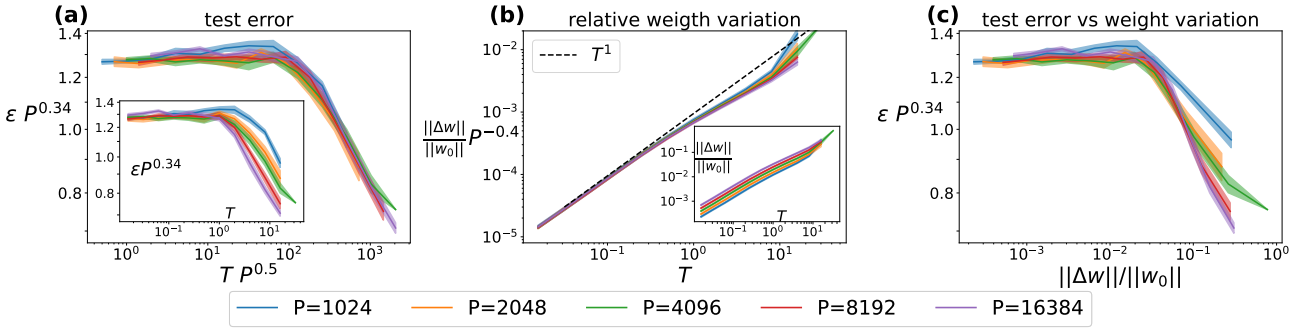

Figure 2. **FC on MNIST, lazy regime,** $\alpha = 32768$**,** $B = 16$**,** $T = \eta/B$**. (a): test error** (ϵ) **vs temperature** (T)**.** *Inset:* ϵ starts improving at a cross-over temperature T_c depending on P. The y-axis is rescaled by P^{β}, with β some fitting exponent, to align ϵ at T_c. *Main:* Rescaling the x-axis by $P^{0.5}$ aligns horizontally the points where ϵ starts improving, suggesting a dependence $T_c \sim P^{-0.5}$. **(b): total weight variation at the end of training normalized with respect to their initialization** ($||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$) **vs** T**.** *Inset:* $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$ increases with both T and P. *Main:* Plotting $\Delta w P^{-\gamma}$ yields a curve increasing approximately as T^{δ}, suggesting $\Delta w \sim T^{\delta} P^{\gamma}$, with $\gamma \approx 0.4$ and $\delta \approx 1$. **(c): test error vs weight variation.** Plotting ϵ vs $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$ for different P aligns the point where ϵ starts improving.

hidden layers of the network as $k = (D-1)/(D+1)$. In the lazy regime, T_{max} and T_{opt} are independent of α.

(ii) In Fig. 1-(a), in the lazy regime (largest α), increasing T leads to an initial slight degradation of the test error followed by an improvement just before reaching the instability T_{max}.

(iii) In Fig. 1-(b), in the lazy regime, increasing T leads to a degradation of the test error before reaching the instability T_{max} (for larger P, a region of good performance appears near T_{max}, see below). In this regime increasing T or decreasing α have opposite effects, showing that in general an increase of SGD noise is not equivalent to making the initialization smaller.

2.3. Role of size of the training set P

This section focuses on the impact of the size of the training set which, surprisingly, determines the SGD noise scale that affects performances.

2.3.1. Lazy regime

Generalization error: Fig. 1 suggests that increasing T leads to a larger test error in the lazy regime. This is evident for the CNN in Fig. 1-(b). However, a detailed analysis for larger P reveals that the test error for the CNN has a non-monotonic behaviour in T. Fig. 3-(a) shows that increasing the number of training points, the performances of the CNN in the lazy regime, after degrading, start improving for increasing T. Also for the FC performances improve for increasing T (Fig. 2-(a)). In both cases, the improvement in performances corresponds to a cross-over temperature T_c that changes with P. In fact, plotting the test error with respect to TP^a, with some fitting exponent a, aligns the point where the test error starts improving (Figs. 3-(a), 2-(a)). This establishes the existence of a characteristic temperature T_c where SGD affects performances, having an asymptotic dependence on P as

$$T_c \sim P^{-a}, \tag{5}$$

with exponent values $a \simeq 0.5$ as reported in Table 1.

Changes of weights: To rationalize this finding, it is useful to consider how the total weight variation relative to their initialization, $\Delta w = \frac{||\boldsymbol{w}^{t^*} - \boldsymbol{w}^0||}{||\boldsymbol{w}^0||}$, increases with T. In Figs. 2-(b),3-(b) we observe an empirical scaling

$$\Delta w \sim T^{\delta} P^{\gamma} \tag{6}$$

with exponents' values $\delta \simeq 1$ (slightly lower for CNNs where $\delta \simeq 0.8, 0.9$) and $\gamma \simeq 0.4$. The values are reported in Table 1.

The dependence of the weight variations on T apparent in Eq. 6 suggests the following hypothesis: the characteristic temperature T_c governing the test error corresponds to the exit from the kernel regime, which occurs when $\Delta w = \mathcal{O}(1)$. We test this hypothesis in two ways. Firstly, if it is true then the test error plotted as a function of Δw should be maximum at the same value of this argument, independently of the size of the training set P. We confirm this result in Figs. 2-(c), 3-(c). Secondly, imposing that $\Delta w = \mathcal{O}(1)$ and using Eq. 6 leads to a characteristic temperature $T_c \sim P^{-\gamma/\delta}$, yielding Eq. 5 with $a = \frac{\gamma}{\delta}$. This prediction is approximately verified, as shown in Table 1.

Convergence time: We expect that a larger change of weights requires a longer training time t^*. We confirm that indeed the increase of T in the lazy regime is accompanied by an increase of the training time t^* (Fig. 9 in Appendix

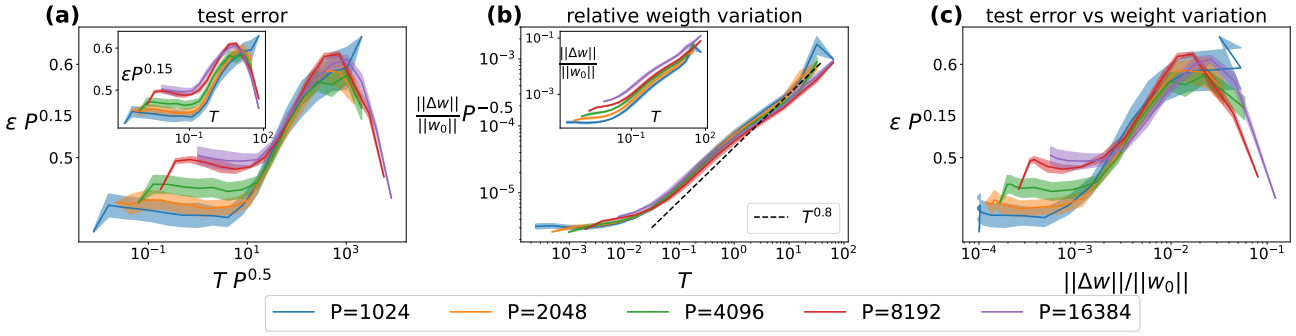

Figure 3. **CNN on CIFAR, lazy regime,** $\alpha = 32768$, $B = 16$, $T = \eta/B$. **(a): test error** (ϵ) **vs temperature** (T). *Inset:* ϵ starts improving at a cross-over temperature T_c depending on P. The y-axis is rescaled by P^{β}, with β some fitting exponent, to align ϵ at T_c. *Main:* Rescaling the x-axis by $P^{0.5}$ aligns horizontally the points where ϵ starts improving, suggesting a dependence $T_c \sim P^{-0.5}$. **(b): total weight variation at the end of training normalized with respect to their initialization** ($||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$) **vs** T. *Inset:* $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$ increases with both T and P. *Main:* Plotting $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0|| P^{-\gamma}$ yields a curve increasing approximately as T^{δ}, suggesting $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0|| \sim T^{\delta} P^{\gamma}$, with $\gamma \approx 0.5$ and $\delta \approx 0.8$. **(c): test error vs weight variation.** Plotting ϵ vs $||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$ for different P approximately aligns the point where ϵ starts improving.

C) and we empirically find the asymptotic behaviour

$$t^* \sim T P^b \tag{7}$$

with values of b around 1.3 (see Table 1).

2.3.2. Feature regime

The power-law behaviours of Eqs. 5, 6, 7 are observed also in the feature-learning regime, with slightly different values of the exponents (see Table 1).

Characteristic temperature: Unlike in the lazy limit, where T_c corresponds to the transition from the linear to the non-linear regime, in the feature regime, we empirically observe that T_c distinguishes between a low T regime where dynamical observables such as Δw remain unaffected by SGD noise and a high T regime where the power-law behaviors of Eqs. 6 and 7 hold. Appendix D contains the data and their detailed discussion.
In particular, the empirical scaling relationships $\Delta w \sim P^{\zeta}$ for $T \ll T_c$ (e.g. for FC on MNIST $\zeta \approx 0.1$) and $\Delta w \sim T^{\delta} P^{\gamma}$ for $T \gg T_c$ imply that $T_c \sim P^{-a}$ with an exponent satisfying $a = (\gamma - \zeta)/\delta$, as we observe (see Table 1). It is worth noting that, while it is straightforward to measure T_c from the behaviour of Δw, this is not always the case from the curve of the test error as a function of T. For instance, in the case of a CNN on CIFAR, the curves of the test error vs T change shape when changing P (Fig. 16-(a)). This change in shape makes it impossible to measure T_c directly from these curves.

In table 1 we report the exponents a, b, γ and δ of the observations $T_c \sim P^{-a}$, $t^* \sim T P^b$ and $\Delta w \sim T^{\delta} P^{\gamma}$. These are extracted from fitting the data in the Figs. 2, 3, 9, 10, 11, 12, 13 for the lazy regime, and Figs. 14, 15, 16 for the feature regime. We observe that the relationships $a = \gamma/\delta$ and $a = (\gamma - \zeta)/\delta$ are approximately verified.

Table 1. Exponents b, γ, δ, a of the empirical observations 5,6,7, including the perceptron model with data distribution parameter χ. The error bar on the fit of the exponents is around ± 0.2 (see Appendix G.2 for further details).

MODEL, lazy regime	b	γ	δ	γ/δ	a
FC on CIFAR	1.4	0.5	1	0.5	0.5
FC on MNIST	1.3	0.4	1	0.4	0.5
MNAS on CIFAR	1.3	0.5	0.8	0.6	0.5
MNAS on MNIST	1.2	0.3	0.75	0.4	0.5
simpleCNN on CIFAR	1.5	0.6	0.9	0.67	0.6
simpleCNN on MNIST	1.4	0.35	0.9	0.45	0.5
perceptron $\chi = 1.5$	1.8	0.4	1	0.4	
perceptron $\chi = 4$	1.4	0.2	1	0.2	

MODEL, feature regime	b	γ	δ	$\frac{\gamma-\zeta}{\delta}$	a
FC on CIFAR	1.4	0.6	0.5	0.9	0.9
FC on MNIST	1.4	0.45	0.5	0.7	0.7
MNAS on CIFAR	1.3	0.5	0.6	0.5	0.5

3. Interpretation of the observations

In this section we provide an understanding for Eq. 6, which justifies Eqs. 5 and 7, based on the local alignment of the model decision boundary with the true one. We then test it in the perceptron model, where relevant quantities can be easily measured.

3.1. Neural networks

Local alignment of decision boundaries. In binary classification, the true decision boundary in data space is the locus of points between $\boldsymbol{x}$'s with different labels $y(\boldsymbol{x}) = \pm 1$, while the decision boundary learnt by the model $F(\boldsymbol{x})$ corresponds to the $\boldsymbol{x}$'s such that $F(\boldsymbol{x}) = 0$. Considering a point $\boldsymbol{x}^*$ where the two boundaries cross and its neighbourhood B_ϵ of diameter ϵ, the local alignment of the model boundary with the true one is given by

$$\frac{||\partial_{\boldsymbol{x}} F_\parallel||}{||\partial_{\boldsymbol{x}} F_\perp||} \tag{8}$$

at linear order in ϵ, where $\partial_{\boldsymbol{x}} F_\parallel$ is the component of the gradient $\partial_{\boldsymbol{x}} F(\boldsymbol{x}^*)$ in the direction perpendicular to the true decision boundary, while $\partial_{\boldsymbol{x}} F_\perp = \partial_{\boldsymbol{x}} F(\boldsymbol{x}^*) - \partial_{\boldsymbol{x}} F_\parallel$ is orthogonal to it (see Fig. 4). The angle between the two boundaries corresponds to $\theta = \text{arccot}\left(\frac{||\partial_{\boldsymbol{x}} F_\parallel||}{||\partial_{\boldsymbol{x}} F_\perp||}\right)$ and perfect learning requires that $\frac{||\partial_{\boldsymbol{x}} F_\parallel||}{||\partial_{\boldsymbol{x}} F_\perp||} \to \infty$.
$\partial_{\boldsymbol{x}} F_\parallel$ identifies the direction that is informative for the task, while $\partial_{\boldsymbol{x}} F_\perp$ is the component in the non-informative directions, which act as noise. It is worth noting that in the lazy regime, the gradient components $\partial_{\boldsymbol{x}} F_\parallel$ and $\partial_{\boldsymbol{x}} F_\perp$ are linear functions of the variation of the weights, as recalled in [2]. This fact allows defining informative and uninformative weight components $\boldsymbol{w}_\parallel$ and $\boldsymbol{w}_\perp$, respectively, around a data point $\boldsymbol{x}^*$. The condition we obtain below on the magnitude of $\|\partial_{\boldsymbol{x}} F_\parallel\|/\|\partial_{\boldsymbol{x}} F_\perp\|$ to fit the data thus corresponds to a bound on $\|\boldsymbol{w}_\parallel\|/\|\boldsymbol{w}_\perp\|$, as shown in Section 3.2 using the example of the perceptron.

Fitting condition. When considering the hinge loss in Eq. 3 with margin α^{-1} defined in Sec. 2.1, a training point $(\boldsymbol{x}^\mu, y^\mu)$ is fitted (i.e. it has zero training loss) when $y^\mu F(\boldsymbol{x}^\mu) \geq \alpha^{-1}$. Having P training points, we call $\boldsymbol{x}^\pm$ the two of them in B_ϵ with $y(\boldsymbol{x}^\pm) = \pm 1$ that have the shortest distances $\delta^\pm$ from the true decision boundary. Their fitting conditions $\pm F(\boldsymbol{x}^\pm) \geq \alpha^{-1}$ imply $F(\boldsymbol{x}^+) - F(\boldsymbol{x}^-) \geq 2\alpha^{-1}$. Assuming $F(\boldsymbol{x})$ is differentiable in B_ϵ, the last inequality can be approximated at linear order in ϵ as

$$\partial_{\boldsymbol{x}} F(\boldsymbol{x}^*) \cdot \left(\boldsymbol{x}^+ - \boldsymbol{x}^-\right) \geq 2\alpha^{-1}. \tag{9}$$

[2] The predictor defined in Sec. 2.1 $F(\boldsymbol{w}, \boldsymbol{x}) = f(\boldsymbol{w}^t, \boldsymbol{x}_\mu) - f(\boldsymbol{w}^0, \boldsymbol{x})$, at linear order in the weight variation $\Delta\boldsymbol{w}$, reads $F(\boldsymbol{w}, \boldsymbol{x}) = \nabla_{\boldsymbol{w}} f(\boldsymbol{w}^0, \boldsymbol{x}) \cdot \Delta\boldsymbol{w}$. Therefore $\partial_{\boldsymbol{x}} F(\boldsymbol{w}, \boldsymbol{x}^*) = \boldsymbol{\mathcal{T}} \Delta\boldsymbol{w}$ with the tensor $\mathcal{T}_{ik} = \partial_{x_i} \nabla_{w_k} f(\boldsymbol{w}^0, \boldsymbol{x}^*)$. Performing a projection of $\boldsymbol{\mathcal{T}}$ onto the informative and uninformative directions in data space, $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}^\parallel + \boldsymbol{\mathcal{T}}^\perp$, we obtain $\partial_{\boldsymbol{x}} F_\parallel = \boldsymbol{\mathcal{T}}^\parallel \Delta\boldsymbol{w} \equiv \boldsymbol{w}_\parallel$ and $\partial_{\boldsymbol{x}} F_\perp = \boldsymbol{\mathcal{T}}^\perp \Delta\boldsymbol{w} \equiv \boldsymbol{w}_\perp$ which corresponds to different components of the weight variation. Therefore Eq. 10 becomes a condition, dependent on $\boldsymbol{x}^*$, on the components of the weights: $\frac{\|\boldsymbol{w}_\parallel\|}{\|\boldsymbol{w}_\perp\|} \geq \frac{1}{\delta_\parallel}\left(\frac{2\alpha^{-1}}{\|\boldsymbol{w}_\perp\|} + c\right)$.

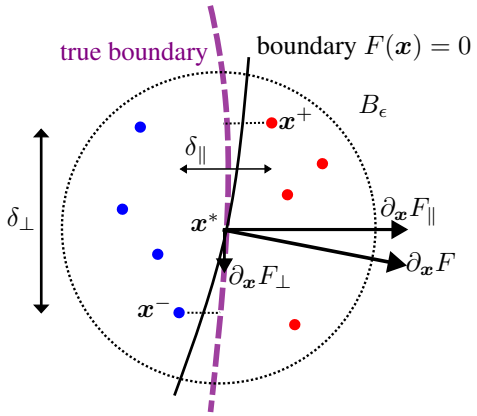

Figure 4. **Pictorial representation of a neighbourhood B_ϵ of the true decision boundary (purple dashed line).** Red (blue) dots are training points with labels $+1$ (-1) and the point $\boldsymbol{x}^+$ ($\boldsymbol{x}^-$) is the closest to the true decision boundary. The decision boundary of the trained model $F(\boldsymbol{x})$ corresponds to the $\boldsymbol{x}$'s such that $F(\boldsymbol{x}) = 0$ (black line). The gradients $\partial_{\boldsymbol{x}} F$ on it quantify the local alignment between the model boundary and the true one: $\partial_{\boldsymbol{x}} F_\parallel$ is the component in the direction of correct alignment, while $\partial_{\boldsymbol{x}} F_\perp$ is orthogonal to it.

Defining $\delta_\parallel$ and c as $\delta_\parallel = \delta^+ + \delta^- = \frac{\partial_{\boldsymbol{x}} F_\parallel}{||\partial_{\boldsymbol{x}} F_\parallel||} \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-)$ and $c = -\frac{\partial_{\boldsymbol{x}} F_\perp}{||\partial_{\boldsymbol{x}} F_\perp||} \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-)$, inequality 9 becomes

$$\frac{||\partial_{\boldsymbol{x}} F_\parallel||}{||\partial_{\boldsymbol{x}} F_\perp||} \geq \frac{1}{\delta_\parallel}\left(\frac{2\alpha^{-1}}{||\partial_{\boldsymbol{x}} F_\perp||} + c\right). \tag{10}$$

Role of the training set size P and of the SGD temperature T. Considering Eq. 10:

(1) we argue that increasing P corresponds to shorter distances $\delta_\parallel$, which require a better alignment of the model decision boundary with the true one, that is a larger $\frac{||\partial_{\boldsymbol{x}} F_\parallel||}{||\partial_{\boldsymbol{x}} F_\perp||}$.

(2) Since increasing T makes the training dynamics more noisy, we propose that a larger T increases the non-informative component $||\partial_{\boldsymbol{x}} F_\perp||$. This implies, according to Eq. 10, a larger informative component $||\partial_{\boldsymbol{x}} F_\parallel||$ to fit the training set.

According to (1) and (2), both T and P increase the gradients magnitude $||\partial_{\boldsymbol{x}} F(\boldsymbol{x}^*)||$, but only increasing P gives a better boundary alignment, that is a larger $||\partial_{\boldsymbol{x}} F_\parallel||/||\partial_{\boldsymbol{x}} F_\perp||$. This effect is illustrated in Fig. 5 for two-dimensional data.

Overall, both increasing P and T require larger gradient magnitudes $||\partial_{\boldsymbol{x}} F(\boldsymbol{x}^*)||$ to fit the training set, which corresponds to a larger relative variation of the weights, in accordance with the observation of Eq. 6. This larger growth of

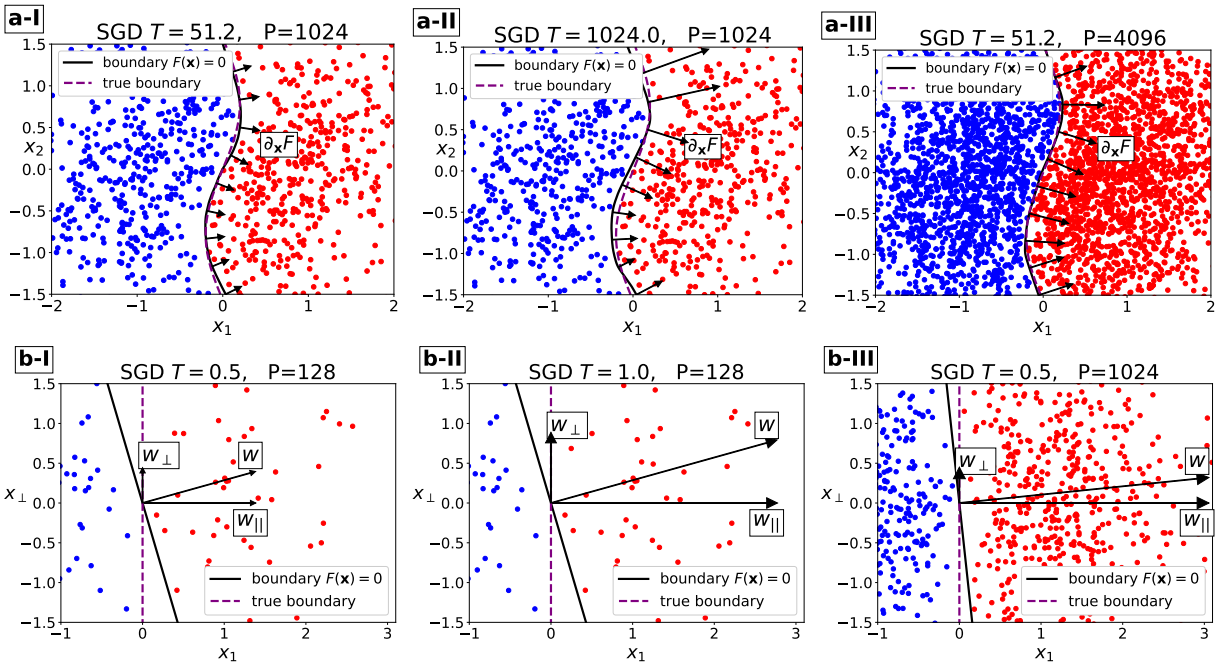

Figure 5. **Decision boundary for binary classification in 2 dimensions: (a) one-hidden-layer FC neural network; (b) perceptron model.** Red (blue) dots are training points with label $+1$ (-1) and the purple dashed line is the true decision boundary. The black line is the decision boundary obtained from training the model $F(\boldsymbol{x})$ with SGD. **(I)-(II).** Increasing the SGD temperature T gives larger gradients $\partial_{\boldsymbol{x}} F$ but not a better alignment between the decision boundaries: it increases the non-informative component ($\boldsymbol{w}_\perp$ for the perceptron). **(I)-(III).** Increasing the number of training points P gives larger gradients $\partial_{\boldsymbol{x}} F$ and a better alignment between the decision boundaries.

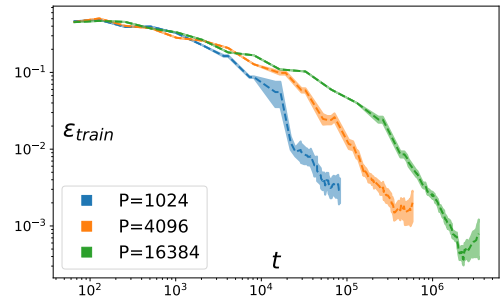

Figure 6. **FC on MNIST: training error in time, fixed T, changing P.** Increasing the training set size P delays the point when the training error goes to zero, while the first part of the dynamics stays unchanged.

the weights requires a longer training time, in accordance with the observation of Eq. 7. In this view, a key effect of increasing P is to diminish the distance between data of different labels, which are the last points to be fitted. We thus expect that changing P affects the dynamics only late in training, as we demonstrate in Fig. 6. Therefore, the hardest data to fit affect both the growth of the weights and the training time.

3.2. Perceptron model

We consider a linearly-separable classification task with high-dimensional data $\boldsymbol{x} \in \mathbb{R}^d$, $d \gg 1$, with labels $y(\boldsymbol{x}) = \pm 1$ given by the signs of the first components:

$$y(\boldsymbol{x}) = \text{sign}(x_1). \tag{11}$$

The true decision boundary in this problem is the hyperplane $x_1 = 0$. We study this problem with a linear classifier, called perceptron:

$$F(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{\sqrt{d}} \boldsymbol{w} \cdot \boldsymbol{x} \tag{12}$$

initialized with $\boldsymbol{w}^0 = 0$.

Although the perceptron is always in the lazy regime[3] and does not have a characteristic temperature of SGD controlling performance, it is of interest because the interpretation discussed in Sec. 3.1 can be tested. In fact, the gradient $\partial_{\boldsymbol{x}} F(\boldsymbol{x}^*)$ corresponds to the perceptron's weights $\boldsymbol{w}/\sqrt{d}$, with the informative and non-informative components respectively $||\partial_{\boldsymbol{x}} F_{||}|| = w_1/\sqrt{d}$ and $||\partial_{\boldsymbol{x}} F_\perp|| = ||\boldsymbol{w}_\perp||/\sqrt{d}$. The alignment of the perceptron decision boundary with the true one is given by the ratio

$$w_1/||\boldsymbol{w}_\perp||. \tag{13}$$

[3] Because it is linear with respect to the weights $\boldsymbol{w}$.

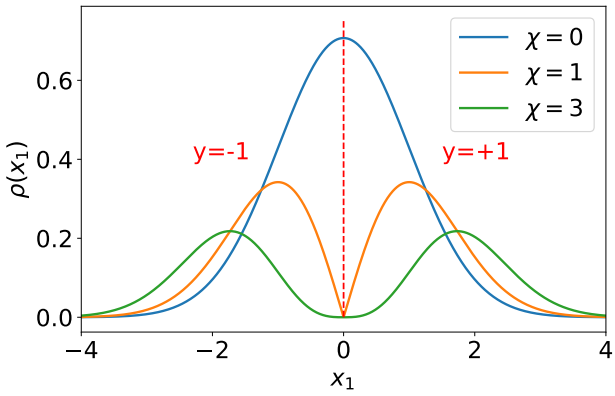

Figure 7. **Perceptron model, data distribution on the x_1 component.** The sign of x_1 determines the class $y = \text{sign}(x_1)$. For $\chi = 0$ the distribution is Gaussian.

The fitting condition on the data point $(\boldsymbol{x}^\mu, y^\mu)$ requires that the weights $\boldsymbol{w} = [w_1; \boldsymbol{w}_\perp]$ satisfy

$$w_1|x_1^\mu| + y^\mu \boldsymbol{w}_\perp \cdot \boldsymbol{x}_\perp^\mu \geq \frac{\sqrt{d}}{\alpha} \tag{14}$$

which, by defining the random quantities $c_\mu = -y^\mu \frac{\boldsymbol{w}_\perp}{||\boldsymbol{w}_\perp||} \cdot \boldsymbol{x}_\perp^\mu$, can be recast as

$$\frac{w_1}{||\boldsymbol{w}_\perp||} \geq \frac{1}{|x_1^\mu|}\left(\frac{\sqrt{d}}{\alpha||\boldsymbol{w}_\perp||} + c_\mu\right). \tag{15}$$

This relationship is a special case of Eq. 10. In fact, increasing P gives smaller values of $|x_1^\mu|$ which require larger $\frac{w_1}{||\boldsymbol{w}_\perp||}$ to fit the training set, while increasing T corresponds to increasing $||\boldsymbol{w}_\perp||$. A qualitative confirmation of this effect is reported in Fig. 5-(b).
In the following, we consider the regime of large T and large α, corresponding to $\frac{\sqrt{d}}{\alpha||\boldsymbol{w}_\perp||} \ll |c_\mu|$, for which condition 15 becomes

$$\frac{w_1}{||\boldsymbol{w}_\perp||} \geq \frac{c_\mu}{|x_1^\mu|}\left(1 + o(1)\right). \tag{16}$$

Data distribution and setting. To control the density of data near the decision boundary $x_1 = 0$, we consider a distribution on the first component x_1 parametrized by $\chi \geq 0$ (Fig. 7):

$$\rho(x_1) = |x_1|^\chi e^{-x_1^2/2}/Z, \tag{17}$$

with $Z = 2^{\frac{1+\chi}{2}}\Gamma(\frac{1+\chi}{2})$ the normalization constant. The other $d-1$ components $\boldsymbol{x}_\perp = [x_i]_{i=2,\dots,d}$ are distributed as standard multivariate Gaussian numbers, i.e. $\boldsymbol{x}_\perp \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d-1})$. $\chi = 0$ corresponds to the Gaussian case. This data distribution has been first considered in Tomasini et al. (2022). The learning setting is defined identically to the one of neural networks in Sec. 2.1. We consider the case $1 \ll d \ll P$, where d is the dimension of the data and the perceptron weights and P is the number of training points. We consider this being a realistic limit when considering the effective dimension d_{eff} of real datasets ($d_{\text{eff}} \approx 15$ for MNIST and $d_{\text{eff}} \approx 35$ for CIFAR-10 (Spigler et al., 2020)) with respect to the number of training samples $P > 10^3$.

Empirical observations. A key result is that the perceptron displays asymptotic behaviours in the change of weights and training time similar to those of neural networks. For the considered perceptron initialized with $\boldsymbol{w}^0 = 0$, the weight variation Δw corresponds to $||\boldsymbol{w}||$. Since $w_1/||\boldsymbol{w}_\perp|| \gg 1$ for large P, we have $\Delta w = ||\boldsymbol{w}|| \simeq w_1$. Eqs. 6 and 7 are verified with exponents reported in Table 1, as shown in Fig. 8-(a,c). These data are produced with $d = 128$, therefore in a high-dimensional setting.
In addition, we observe that $||\boldsymbol{w}_\perp||$ at the end of training is proportional to T and independent of P (Fig. 8-(b)):

$$||\boldsymbol{w}_\perp|| \sim T. \tag{18}$$

This observation is a positive test about the effect of T on $||\partial_{\boldsymbol{x}} F_\perp||$ proposed in Sec. 3.1.

Non-universality of the exponents. Remarkably, the exponents γ and b of P for the perceptron depend on the parameter χ of the data distribution. This finding can be rationalized by considering condition 16 at the end of training. In fact, satisfying 16 for every training point requires $\frac{w_1}{||\boldsymbol{w}_\perp||} \geq \max_\mu \frac{c_\mu}{|x_1^\mu|}$. In Appendix F, classical extreme value theory is used to show that, for large P, the typical value of $\max_\mu \frac{c_\mu}{|x_1^\mu|}$ behaves asymptotically as $\langle \max_\mu \frac{c_\mu}{|x_1^\mu|} \rangle = C P^{\frac{1}{1+\chi}} + o\left(P^{\frac{1}{1+\chi}}\right)$ for some constant C. Therefore we obtain a prediction for the exponent γ:

$$\gamma = \frac{1}{1+\chi}, \tag{19}$$

in excellent agreement with data (Fig 8-(a)). This further confirms that the asymptotic behaviour with respect to P is controlled by the statistics of the points close to the decision boundary. Thus the exponents are non-universal, since they depend directly on the data distribution.
An estimate of the parameter χ for some images datasets is reported in Tomasini et al. (2022) through the study of kernel ridge regression. For binary CIFAR10, $\chi_{CIFAR} = 1.5$ is reported, that according to 19 corresponds to $\gamma = 0.4$, a value compatible with those observed in neural networks (Table 1).

4. Conclusions

In this work we have explored the effect of SGD noise in different training regimes of neural networks using the hinge loss, which is analogous to the widely used

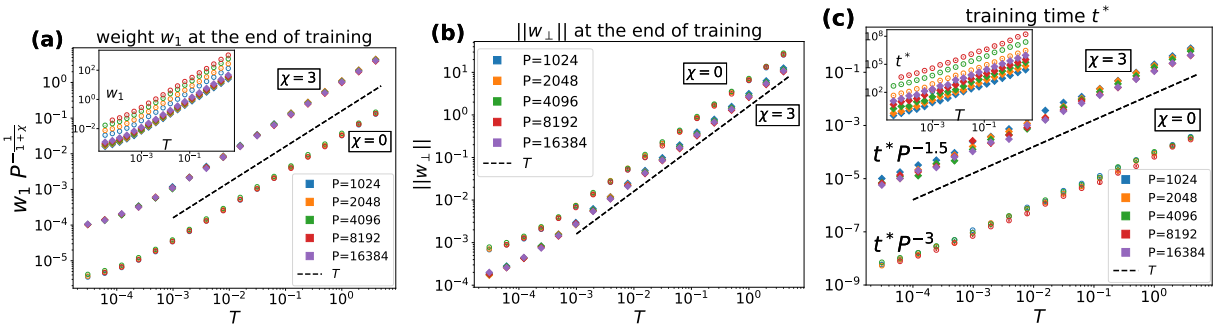

Figure 8. **Perceptron model, $d = 128$, $B = 2$, varying T and P. (a)** *Inset:* Total variation of the weight w_1 at the end of training with respect to SGD noise T and training set size P (colors), for different data distributions $\chi = 0$ (empty circles) and $\chi = 3$ (full diamonds). *Main:* Plotting $w_1 P^{-\frac{1}{1+\chi}}$ gives a curve proportional to T for each value of χ, revealing the asymptotic behaviour $w_1 \sim T P^{\gamma}$ (Eq. 6 for neural networks) with a data-dependent exponent $\gamma = \frac{1}{1+\chi}$ in accordance with prediction 19. **(b)** Total variation of $||w_\perp||$ for the same setting of panel (a). $||w_\perp||$ is proportional to T independently of P, as stated in Eq. 18. **(c)** *Inset:* Total training time t^* for the same setting as panel (a): t^* increases with both T and P. *Main:* Plotting $t^* P^{-b}$, with b depending on χ, gives approximately one curve proportional to T for each value of χ, corresponding to the asymptotic behaviour $t^* \sim T P^b$ as found for neural networks (Eq. 7).

cross-entropy loss and performing early-stopping. Since the hinge loss goes to zero at the end of training, the minima found by the algorithm are always flat: a static view explaining the benefit of SGD in terms of the flatness of minima cannot be applied. Instead, we propose a dynamical view where SGD noise increases the weights of the model in directions that are detrimental for learning, which in turn induces an increase in the useful directions to fit the training set. Fitting is the hardest for data close to the decision boundary, whose statistics depends both on the size of the training set and the distribution of data close to the decision boundary. This view naturally explained our observations that the total weight variation, and the training time, depend on both the SGD noise and the size of the training set. It also rationalizes the puzzling observation that the characteristic SGD temperature for which weight changes become significant and the test error is affected by the noise depends on the training set size. Exponents characterizing this relationship are non-universal. We expect them to depend on the data distribution near the decision boundary, as we demonstrated for the perceptron.

Our work thus clarifies a key effect of SGD, and explains the range of temperatures where SGD noise matters. However, understanding the sign of the effect of this noise on performance (beneficial or detrimental), and how it relates to the data structure and the network architecture, appears to be a particularly vexing question. For example, for the lazy regime of CNNs, we observe a non-monotonic behaviour of the test error, which initially grows and then decays as the SGD noise is increased. What determines this behavior is an open question that requires further investigation.

Acknowledgments

We thank Francesco Cagnetta, Alessandro Favero, Bastien Olivier Marie Göransson, Leonardo Petrini and Umberto Maria Tomasini for helpful discussions. This work was supported by a grant from the Simons Foundation (# 454953 Matthieu Wyart).

References

Blanc, G., Gupta, N., Valiant, G., and Valiant, P. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pp. 483–513. PMLR, 2020.

Chaudhari, P. and Soatto, S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

Gnedenko, B. Sur la distribution limite du terme maximum d'une serie aleatoire. *Annals of mathematics*, pp. 423–453, 1943.

HaoChen, J. Z., Wei, C., Lee, J., and Ma, T. Shape matters: Understanding the implicit bias of the noise covariance. In *Conference on Learning Theory*, pp. 2315–2357. PMLR, 2021.

Heskes, T. M. and Kappen, B. On-line learning processes in artificial neural networks. In *North-Holland Mathematical Library*, volume 51, pp. 199–233. Elsevier, 1993.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in neural information processing systems*, 30, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Leadbetter, M. R., Lindgren, G., and Rootzén, H. *Extremes and related properties of random sequences and processes*. Springer Science & Business Media, 2012.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Li, Q., Tai, C., and Weinan, E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110. PMLR, 2017.

Li, Q., Tai, C., and Weinan, E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *The Journal of Machine Learning Research*, 20(1):1474–1520, 2019.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

Paccolat, J., Petrini, L., Geiger, M., Tyloo, K., and Wyart, M. Geometric compression of invariant manifolds in neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(4):044001, 2021.

Pesme, S., Pillaud-Vivien, L., and Flammarion, N. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34:29218–29230, 2021.

Rotskoff, G. M. and Vanden-Eijnden, E. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *stat*, 1050:22, 2018.

Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

Smith, S., Elsen, E., and De, S. On the generalization benefit of noise in stochastic gradient descent. In *International Conference on Machine Learning*, pp. 9058–9067. PMLR, 2020.

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018.

Spigler, S., Geiger, M., and Wyart, M. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, 2020.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

Tomasini, U. M., Sclocchi, A., and Wyart, M. Failure and success of the spectral bias prediction for laplace kernel ridge regression: the case of low-dimensional data. In *International Conference on Machine Learning*, pp. 21548–21583. PMLR, 2022.

Wu, L., Ma, C., et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31, 2018.

Yang, G. and Hu, E. J. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pp. 11727–11737. PMLR, 2021.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, G., Li, L., Nado, Z., Martens, J., Sachdeva, S., Dahl, G., Shallue, C., and Grosse, R. B. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, 32, 2019.

Zhang, Y., Saxe, A. M., Advani, M. S., and Lee, A. A. Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning. *Molecular Physics*, 116(21-22):3214–3223, 2018.

A. Other related works

As reviewed in the introduction, various works have studied empirically the role of SGD noise on performance. Our work goes beyond these studies by systematically studying the role of initialization scale and size of the training set for a large range of noise magnitude. Some recent studies have analysed the relationship between the implicit bias of SGD and the initialization scale in simple regression models (HaoChen et al., 2021; Pesme et al., 2021), showing that SGD bias the model towards the feature-learning regime. Our work tests this hypothesis for image classification, showing that the effect is not captured by a simple reduction of the initialization scale, but confirming that SGD stochasticity can bring the model outside the kernel regime. Several works have showed that larger SGD stochasticity leads to flatter minima of the loss landscape and it has been argued that this leads to improved performances (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Zhang et al., 2018; Smith & Le, 2018; Wu et al., 2018). Our results show that in some regimes performances can behave non-monotonically with respect to increasing SGD stochasticity, in contradiction with simple arguments based on the flatness of the landscape. Therefore our observations call for a theory of generalization that goes beyond the flatness view and explains at least the sign of change in performances.

The importance of the stopping criterion when evaluating the performances of SGD has already been emphasized (Hoffer et al., 2017; Shallue et al., 2018; Smith et al., 2020). In this work we remove the ambiguity in the choice of the computational budget and show the effect of the size of the training set on the time needed to reach convergence. Moreover, we show how the size of the training set affects the noise scale at which we observe a change in performances. To the best of our knowledge, this relationship constitutes a novelty in the literature.

Previous works have showed that the noise scale of SGD is controlled by the ratio between the learning rate and the batch size when the batch is smaller than some cross-over value (Jastrzebski et al., 2017; Shallue et al., 2018; Smith et al., 2020). On the theoretical side, a description of SGD based on a continuous-time stochastic differential equation (SDE) driven by Gaussian noise was derived (Li et al., 2017; 2019). In our work, we consider SGD in the "small batch regime" where we can describe its noise magnitude by the ratio between learning rate and batch size.

B. Scaling argument for the α dependence of the characteristic temperatures in the feature regime

The covariance of the mini-batch gradients when $B \ll P$ is given by $\boldsymbol{\Sigma}(\boldsymbol{w})/B$ (Chaudhari & Soatto, 2018), with

$$\boldsymbol{\Sigma}(\boldsymbol{w}) = \frac{1}{P}\sum_{\mu=1}^{P} \theta_\mu \, \nabla_{\boldsymbol{w}} f(\boldsymbol{w}, \boldsymbol{x}_\mu) \otimes \nabla_{\boldsymbol{w}} f(\boldsymbol{w}, \boldsymbol{x}_\mu) - \nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \otimes \nabla_{\boldsymbol{w}} L(\boldsymbol{w}), \tag{20}$$

where $\theta_\mu = \theta\left(\alpha^{-1} - y_\mu F(\boldsymbol{w}, \boldsymbol{x}_\mu)\right)$. The stochastic differential equation (SDE) matching the first two moments of the SGD update 4 corresponds to (Smith et al., 2020; Zhang et al., 2019):

$$d\boldsymbol{w}^t = -dt\nabla L(\boldsymbol{w}^t) + \sqrt{T}\sqrt{\boldsymbol{\Sigma}(\boldsymbol{w}^t)}d\boldsymbol{W}^t \tag{21}$$

where $\boldsymbol{W}^t$ is Brownian motion (Ito's convention) and $T = \eta/B$.

Heuristic argument for observation that $T_{max} \sim T_{opt} \sim \alpha^k$: Considering the SDE description 21 of SGD, the corresponding flux $\boldsymbol{J}$ for the weights distribution $\rho(\boldsymbol{w}, t)$ can be written as (Chaudhari & Soatto, 2018)

$$\boldsymbol{J}(\boldsymbol{w}, t) = \rho(\boldsymbol{w}, t)\nabla L(\boldsymbol{w}) + \frac{1}{2}T\nabla \cdot (\boldsymbol{\Sigma}(\boldsymbol{w})\rho(\boldsymbol{w}, t)) \tag{22}$$

where the divergence operator $\nabla\cdot$ is applied column-wise to the matrix $\boldsymbol{\Sigma}\rho$. We notice that the probability flux receives a contribution from both the loss gradient and the covariance divergence. To understand the effect of SGD in the feature regime, we need to compare the scaling of the two terms $\rho(\boldsymbol{w}, t)\nabla L(\boldsymbol{w})$ and $\nabla \cdot (\boldsymbol{\Sigma}(\boldsymbol{w})\rho(\boldsymbol{w}, t))$ in the limit of $\alpha \ll 1$.

In this limit and with the network initialization considered in Sec. 2.1, the variation of the weights in every layer has the same scale w with respect to α. Therefore, for a network of depth $D+1$ with ReLU activation functions, the predictor variation Δf is related to w by $\Delta f \sim w^{D+1}$. To bring the hinge loss to zero, the predictor has to be of the same order of the margin α^{-1}, which corresponds to the scaling $w^{D+1} \sim \alpha^{-1}$ or, equivalently,

$$w \sim \alpha^{-1/(D+1)}. \tag{23}$$

The scaling of ∇L and $\nabla \cdot \mathbf{\Sigma}$ with respect to w is easily obtained by inspecting the definitions of L (Eq. 3) and $\mathbf{\Sigma}$ (Eq. 20). For the hinge loss, we have

$$\nabla L \sim \nabla f \sim w^D \tag{24}$$

and

$$\nabla \cdot \mathbf{\Sigma} \sim \nabla (\nabla f)^2 \sim w^{2D-1}. \tag{25}$$

Therefore, the two terms ∇L and $T\nabla \cdot \mathbf{\Sigma}$ become comparable when $w^D \sim T w^{2D-1}$, that is for a characteristic temperature $T \sim w^{-D+1}$. By using 23, this corresponds to

$$T \sim \alpha^{(D-1)/(D+1)}. \tag{26}$$

For much larger temperatures, the noise term $T\nabla \cdot \mathbf{\Sigma}$ is much larger than the signal ∇L, and we expect the dynamics not to converge. For much smaller temperatures, noise is negligible. These arguments support that $T_{max} \sim T_{opt} \sim \alpha^k$ with $k = (D-1)/(D+1)$, as confirmed in Fig. 1.

In the lazy regime, the network weights are always of $O(1)$ with respect to α, therefore the two terms 24 and 25 don't scale with α for $\alpha \gg 1$. Consequently, the characteristic temperatures in this regime are independent of α.

C. Additional plots in the lazy regime

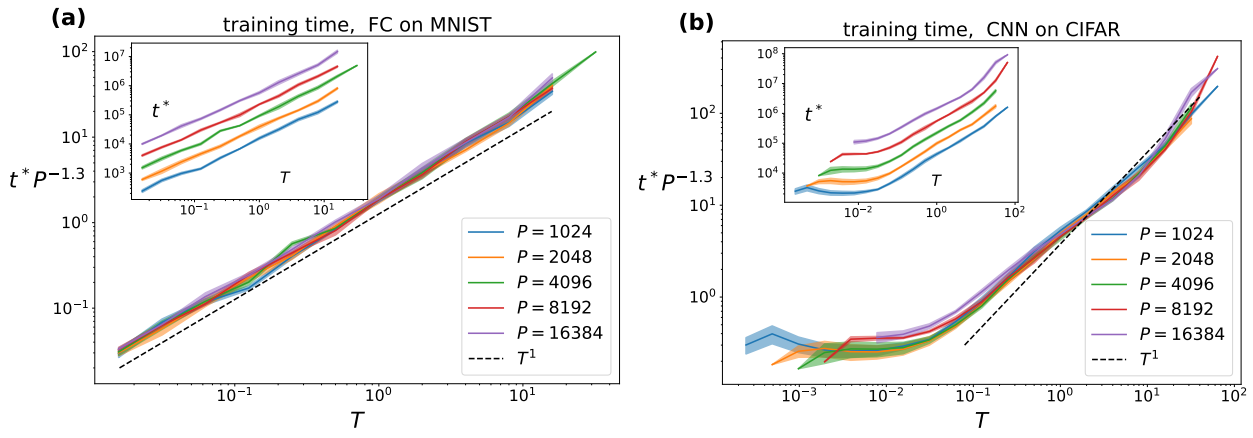

Figure 9. **Training time, lazy regime,** $\alpha = 32768$**,** $B = 16$**, varying** P **and** T**: (a) FC on MNIST, (b) CNN (MNAS) on CIFAR.** *Inset:* t^* increases with both T and P. *Main:* Plotting $t^* P^{-b}$, with b a fitting exponent ($b \approx 1.3$), yields a curve increasing approximately linearly in T, suggesting a dependence $t^* \sim T P^b$.

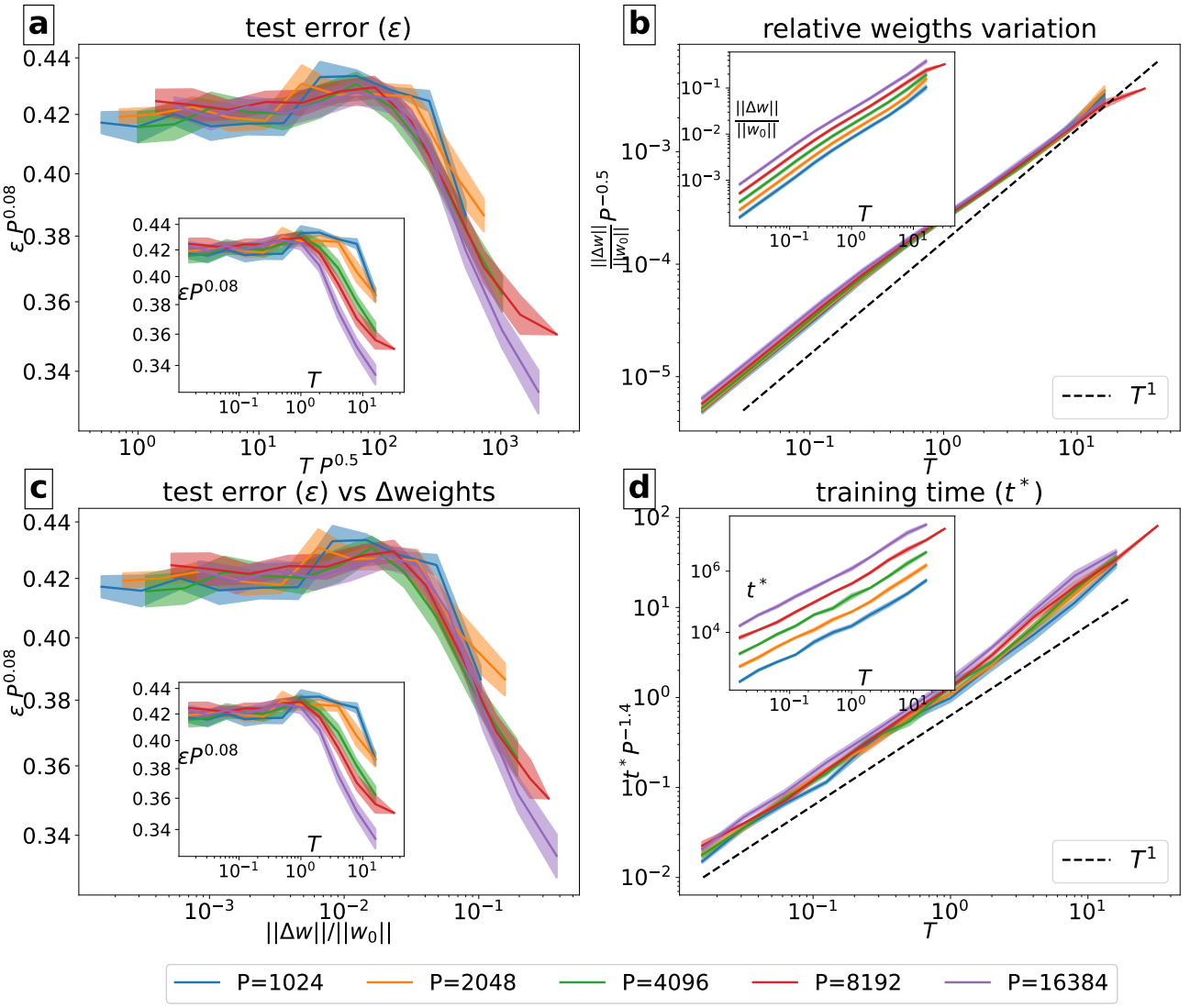

Figure 10. **FC on CIFAR,** $\alpha = 32768$, $B = 16$**, varying** P **and** T**. (a): test error** ϵ**.** *Inset:* ϵ starts improving at a cross-over temperature T_c depending on P. The y-axis is rescaled by P^{β}, with β some fitting exponent, to align ϵ at T_c. *Main:* Rescaling the x-axis by $P^{0.5}$ aligns horizontally the points where ϵ starts improving, suggesting a dependence $T_c \sim P^{-0.5}$. **(b): total weight variation at the end of training normalized with respect to their initialization** (Δw)**.** *Inset:* Δw increases with both T and P. *Main:* Plotting $\Delta w P^{-\gamma}$ yields a curve increasing approximately as T^{δ}, suggesting $\Delta w \sim T^{\delta} P^{\gamma}$, with γ and δ some fitting exponents. **(c): test error vs weight variation.** The point where the test error starts improving shows a better alignment when plotted as a function of the weight variation (*main plots*) rather than temperature alone (*insets*). **(d): training time** t^***.** *Inset:* t^* increases with both T and P. *Main:* Plotting $t^* P^{-b}$, with b a fitting exponent, yields a curve increasing approximately linearly in T, suggesting a dependence $t^* \sim T P^{b}$.

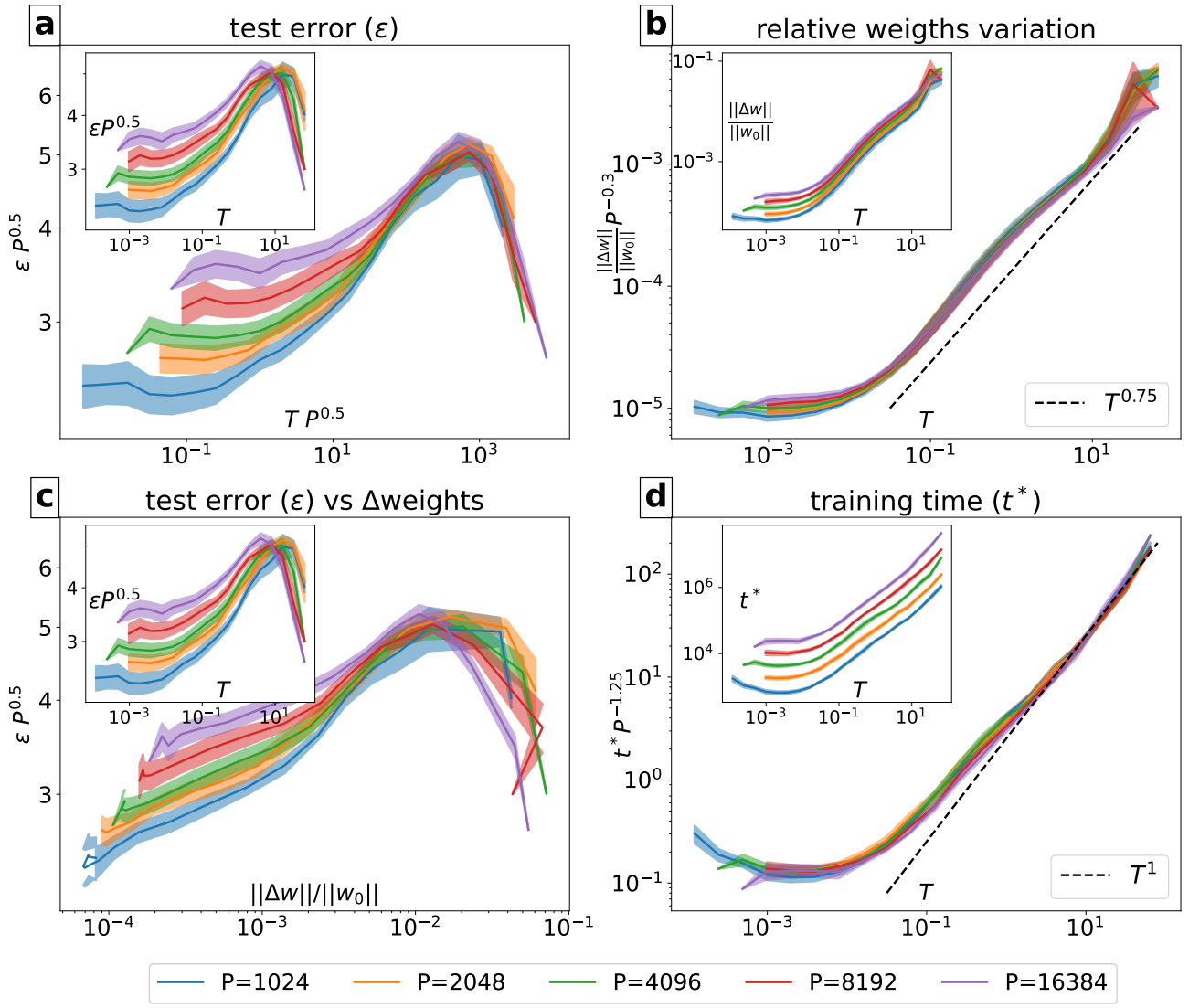

Figure 11. **CNN (MNAS) on MNIST,** $\alpha = 32768$**,** $B = 16$**, varying** P **and** T**: (a) test error, (b) relative weight variation, (c) test error vs relative weight variation, (d) training time.** Same quantities as Fig. 10, see its caption.

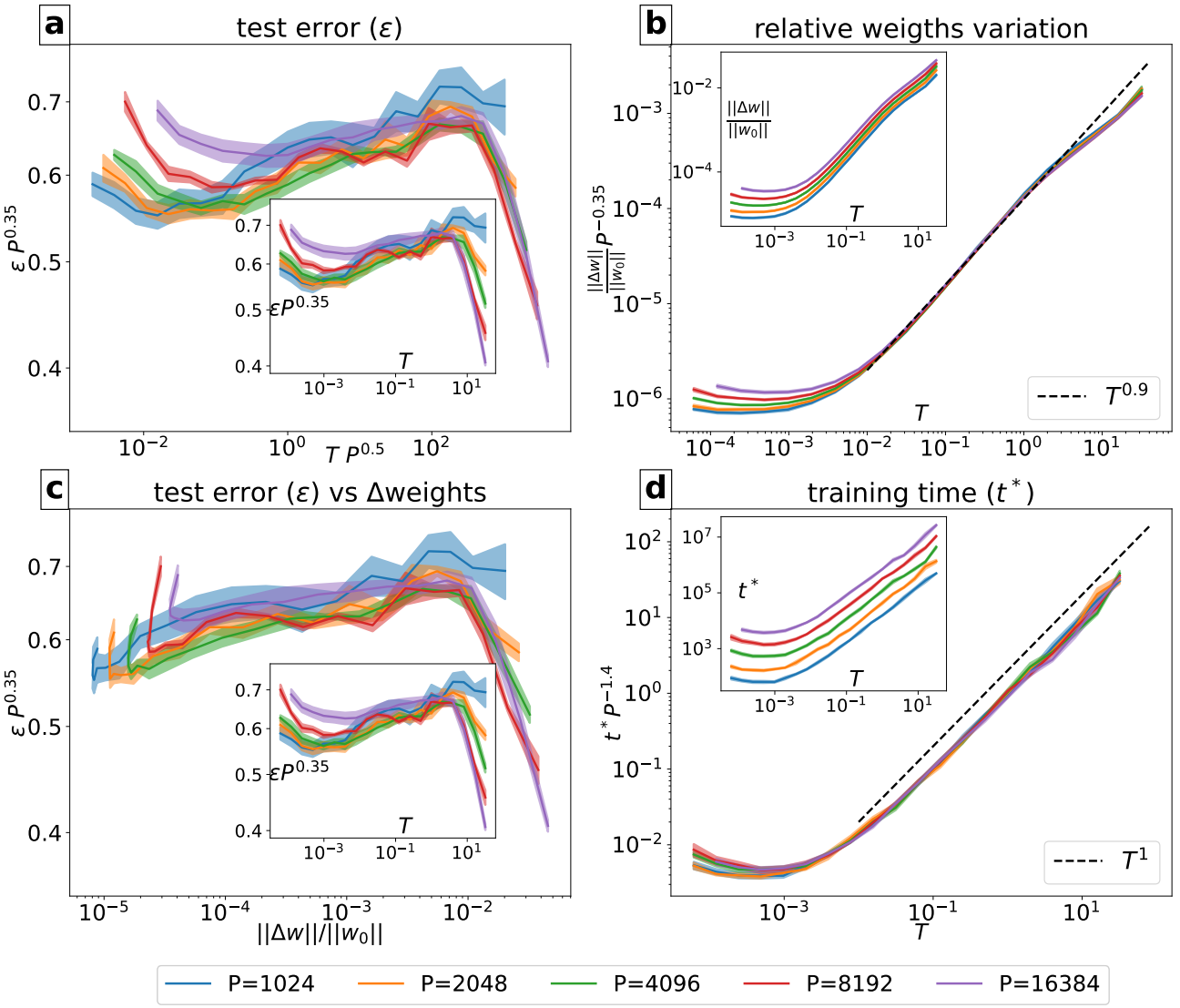

Figure 12. **simpleCNN on MNIST,** $\alpha = 32768$**,** $B = 16$**, varying** P **and** T**: (a) test error, (b) relative weight variation, (c) test error vs relative weight variation, (d) training time.** Same quantities as Fig. 10, see its caption.

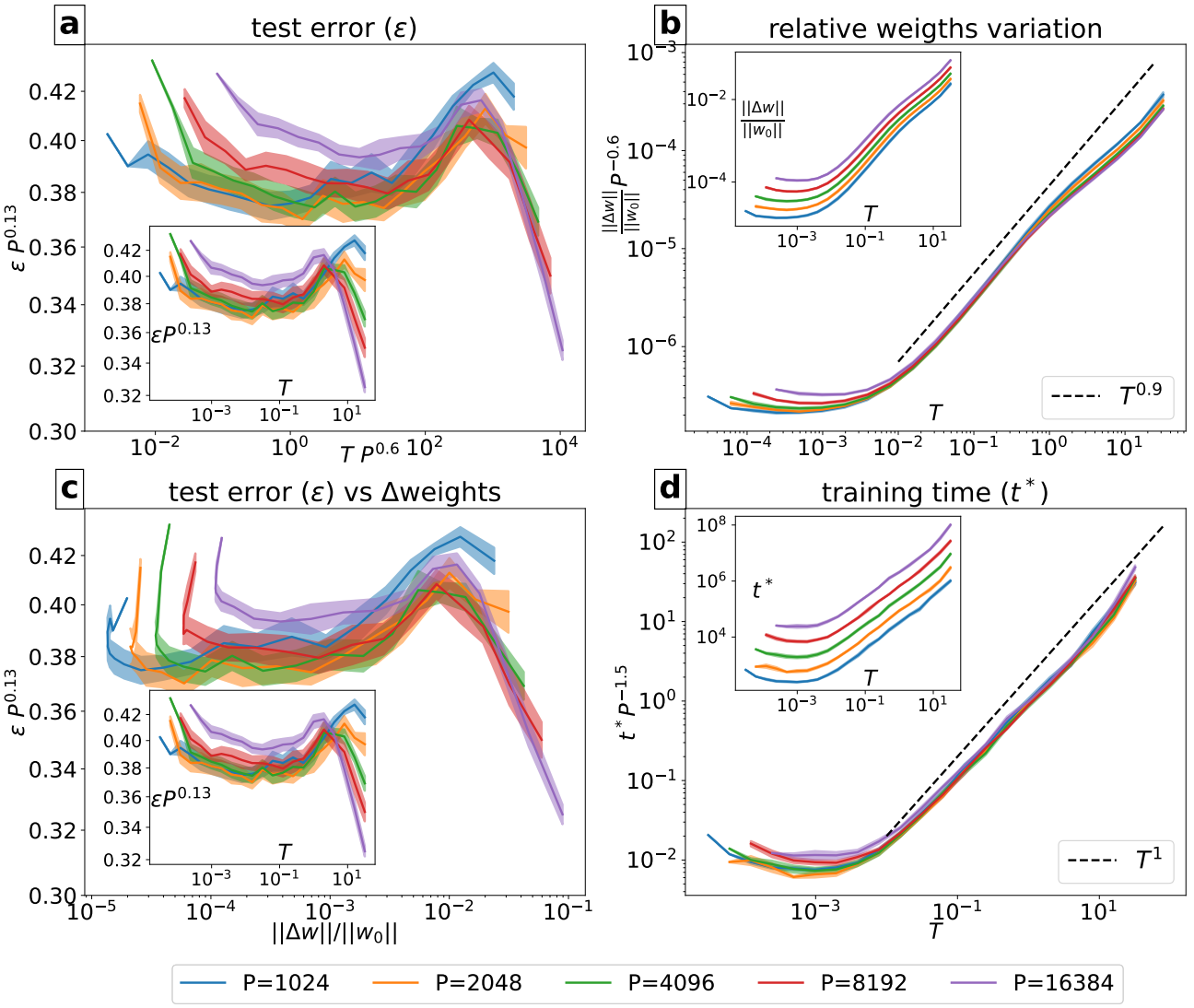

Figure 13. **simpleCNN on CIFAR, $\alpha = 32768$, $B = 16$, varying P and T: (a) test error, (b) relative weight variation, (c) test error vs relative weight variation, (d) training time.** Same quantities as Fig. 10, see its caption.

D. Impact of the training set size P in the feature-learning regime

Empirical observations. In the feature-learning regime, we observe the same scaling behaviors as in the lazy regime, which are discussed in Section 2.3. In particular, the relative change of weights ($\Delta w = \frac{||\boldsymbol{w}^{t^*} - \boldsymbol{w}^0||}{||\boldsymbol{w}^0||}$), the training time ($t^*$) and the characteristic temperature (T_c) where Δw and t^* start being affected by SGD noise exhibit asymptotic behavior as follows:

$$T_c \sim P^{-a}, \qquad \Delta w \sim T^{\delta} P^{\gamma}, \qquad t^* \sim T P^{b}. \tag{27}$$

From the data, we measure T_c as the temperature at which Δw starts increasing with T. In some cases, this also corresponds to the temperature scale where the test error starts improving (e.g. for the FC architecture in Figs. 14-(a-I, a-II) and 15-(a-I, a-II)). In some other cases, instead, the curve of the test error vs T can take different shapes when varying P, and therefore extracting a T_c from it is not possible (e.g. for the CNN architecture in Fig. 16-(a)).

The values of the exponents a, b, γ, δ in Eq. 27 are slightly different from those measured in the lazy regime. For instance, for the fully connected neural network on MNIST in feature learning, we observe $a \approx 0.7$, $\delta \approx 0.5$, $\gamma \approx 0.45$, and $b \approx 1.4$, while in lazy learning, we observe $a \approx 0.5$, $\delta \approx 1.0$, $\gamma \approx 0.4$, and $b \approx 1.3$. Table 1 provides a comparison of the exponents.

Interpretation. The same scaling behaviors of equation 27 are observed in both the feature and lazy regimes. We argue that this similarity comes from the fact that the two training regimes are similar at late times- their main difference corresponds to early times in the dynamics. In the feature regime, the weights need to grow considerably to make the output $\mathcal{O}(1)$[4] and fit the data. At the beginning of the training dynamics, before fitting any data, the weights grow exponentially in time- an initial phase of training that we refer to as an 'inflation period' (Geiger et al., 2020; Paccolat et al., 2021). Afterwards, the network starts fitting the data, and the dynamics is similar to that of the lazy regime, with the exception that the neural tangent kernel has evolved during inflation. In this second part of the dynamics, we expect the arguments presented in Section 3.1 to apply, as supported by our empirical observations.

Characteristic temperature. In feature learning, the characteristic temperature T_c corresponds to the cross-over point between the 'inflation dominated' and the 'noise dominated' dynamics.
Specifically, for $T \ll T_c$, weight variation is mainly concentrated in the initial part of the dynamics, as observed in studies on gradient flow (Geiger et al., 2020) that corresponds to the limit $T \to 0$. In this case, the total weight variation is independent of T and appears to be a function of P as

$$\Delta w_{INFL} \sim P^{\zeta}, \tag{28}$$

where ζ is a fitting exponent. For instance, in a fully connected network on the MNIST dataset, $\zeta \approx 0.1$ (see Figure 14-(b-II)). Conversely, for $T \gg T_c$, most of the weight variation occurs in the later part of the dynamics, when SGD noise becomes relevant. Thus, $\Delta w_{NOISE} \sim T^{\delta} P^{\gamma}$ (equation 27).
Being the cross-over between these two regimes, the characteristic temperature T_c is determined by the condition $\Delta w_{INFL} \sim \Delta w_{NOISE}$, which corresponds to $P^{\zeta} \sim T_c^{\delta} P^{\gamma}$. Therefore,

$$T_c \sim P^{-\frac{\gamma-\zeta}{\delta}} \tag{29}$$

which yields the relationship $T_c \sim P^{-a}$ with an exponent a satisfying

$$a = \frac{\gamma - \zeta}{\delta}, \tag{30}$$

in accordance with the experiments (see Table 1). It should be noted that this relation differs somewhat from the one observed in the lazy regime, where $a = \gamma/\delta$ and the characteristic temperature is determined by comparing weight variation to their initialization.

[4] In our setting, this corresponds to $\alpha F(\boldsymbol{w}, \boldsymbol{x}) \sim \mathcal{O}(1)$.

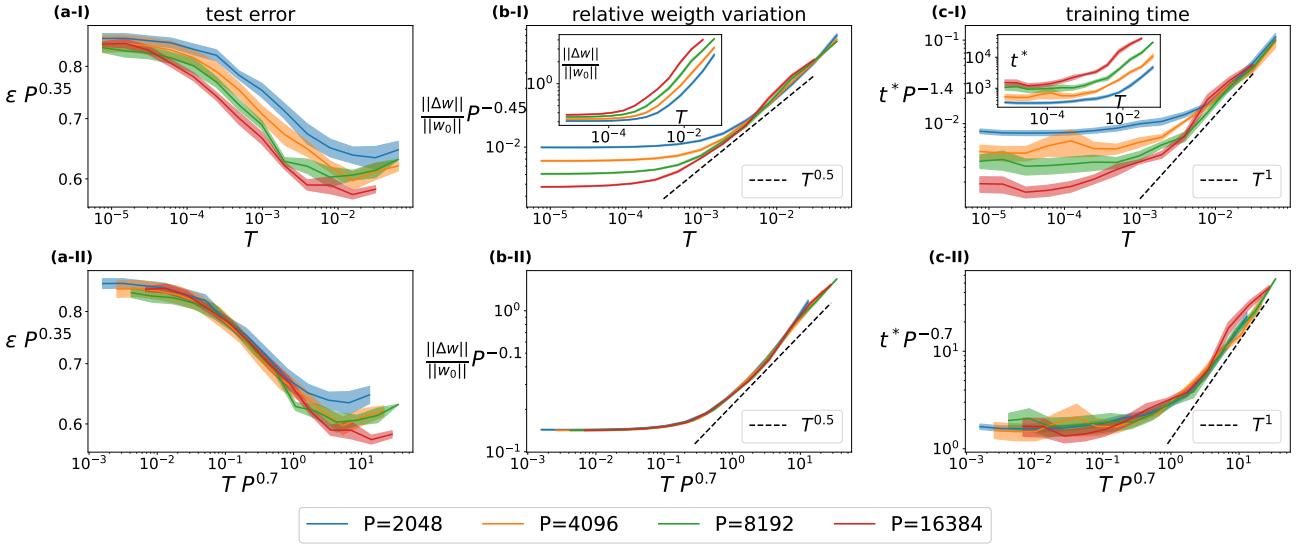

Figure 14. **FC on MNIST, feature regime,** $\alpha = 2^{-10}$**,** $B = 16$**,** $T = \eta/B$**. (a-I, a-II): test error** (ϵ) **vs temperature** (T)**.** *(a-I):* ϵ starts improving at a cross-over temperature T_c depending on P. The y-axis is rescaled by P^{β}, with β some fitting exponent, to align ϵ at the lowest T. *(a-II):* Rescaling the x-axis by $P^{0.7}$ aligns horizontally the points where ϵ starts improving, suggesting a dependence $T_c \sim P^{-0.7}$. **(b-I, b-II): total weight variation at the end of training normalized with respect to their initialization** ($\Delta w = ||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$) **vs** T**.** *(b-I, inset):* Δw increases with both T and P. *(b-I, main):* Plotting $\Delta w P^{-\gamma}$ yields a curve, for large T, increasing approximately as T^{δ}, suggesting $\Delta w \sim T^{\delta} P^{\gamma}$, with $\gamma \approx 0.45$ and $\delta \approx 0.5$. *(b-II):* Rescaling the x-axis by $P^{0.7}$ aligns horizontally the points where T starts having an effect on the weights, corresponding to $T_c \sim P^{-0.7}$. For $T \ll T_c$, the weight variation scale as $\Delta w \sim P^{\zeta}$, with $\zeta \approx 0.1$. **(c-I, c-II): training time** (t^*) **vs temperature** (T)**.** *(c-I, inset):* t^* increases with both T and P. *(c-I, main):* Plotting $t^* P^{-b}$ ($b \approx 1.4$) yields a curve, for large T, increasing approximately linearly in T, suggesting a dependence $t^* \sim T P^b$. *(c-II):* Rescaling the x-axis by $P^{0.7}$ aligns horizontally the points where T starts having an effect on the training time, corresponding to $T_c \sim P^{-0.7}$. For $T \ll T_c$, t^* scales as $t^* \sim P^{0.7}$.

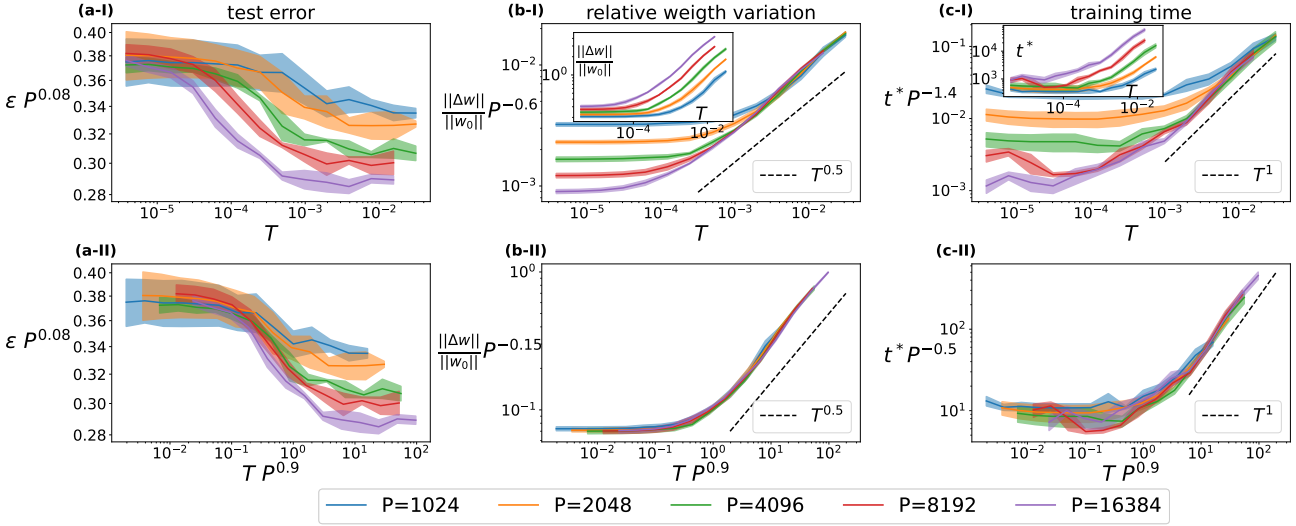

Figure 15. **FC on CIFAR, feature regime,** $\alpha = 2^{-10}$**,** $B = 16$**,** $T = \eta/B$**.** Same quantities as Fig. 14, see its caption.

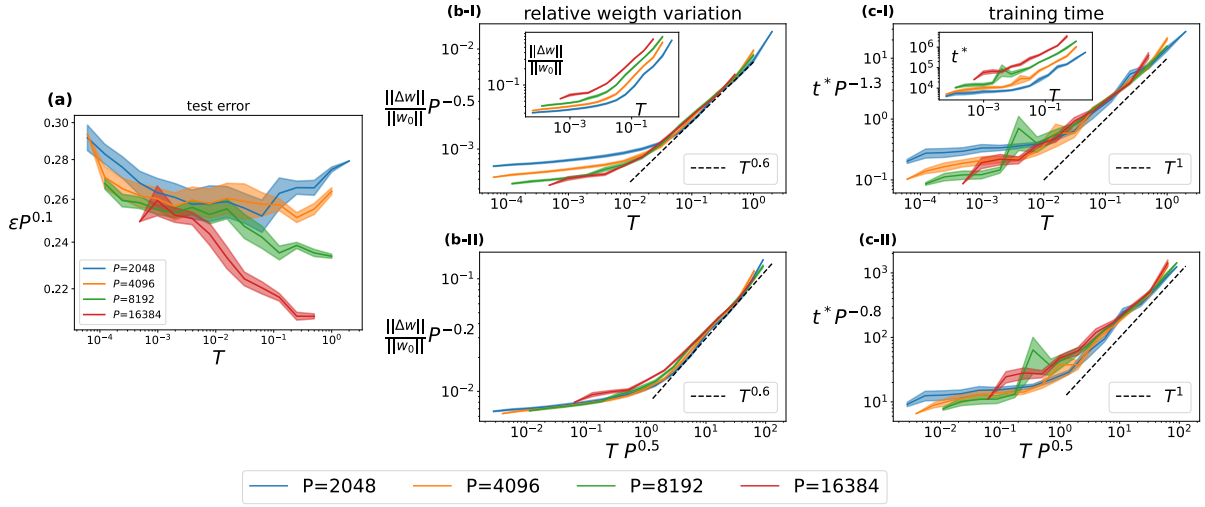

Figure 16. **CNN (MNAS) on CIFAR, feature regime, $\alpha = 1$, $B = 16$, $T = \eta/B$. (a): test error (ϵ) vs temperature (T).** ϵ improves more significantly with T when increasing P. In this case the curves have different shapes and cannot be matched by rescaling the x-axis. The y-axis is rescaled by P^{β}, with β some fitting exponent, to make the curves easier to compare. **(b-I, b-II): total weight variation at the end of training normalized with respect to their initialization ($\Delta w = ||\Delta \boldsymbol{w}||/||\boldsymbol{w}_0||$) vs T.** *(b-I, inset):* Δw increases with both T and P. *(b-I, main):* Plotting $\Delta w P^{-\gamma}$ yields a curve, for large T, increasing approximately as T^{δ}, suggesting $\Delta w \sim T^{\delta} P^{\gamma}$, with $\gamma \approx 0.5$ and $\delta \approx 0.6$. *(b-II):* Rescaling the x-axis by $P^{0.5}$ aligns horizontally the points where T starts having an effect on the weights, corresponding to $T_c \sim P^{-0.5}$. For $T \ll T_c$, the weight variation scales as $\Delta w \sim P^{\zeta}$, with $\zeta \approx 0.2$. **(c-I, c-II): training time (t^*) vs temperature (T).** *(c-I, inset):* t^* increases with both T and P. *(c-I, main):* Plotting $t^* P^{-b}$ ($b \approx 1.3$) yields a curve, for large T, increasing approximately linearly in T, suggesting a dependence $t^* \sim T P^b$. *(c-II):* Rescaling the x-axis by $P^{0.5}$ aligns horizontally the points where T starts having an effect on the training time, corresponding to $T_c \sim P^{-0.5}$. For $T \ll T_c$, t^* scales approximately as $t^* \sim P^{0.8}$.

E. Comparison between hinge loss and cross-entropy loss

This section shows that the setting of our work, using the hinge loss and training until it reaches zero value, is very similar to training with the cross-entropy loss and performing early stopping [5].

Figure 17 shows that the two training procedures give identical power-law dependencies on T and P for all the quantities we analyse, meaning that the exponents of the power-laws are the same. Therefore, our results are relevant for training networks in practical classification tasks.

F. Distribution of the maximum of $c_\mu/|x_1^\mu|$

In this section we compute the distribution of the random variable $M_P = \max_\mu \frac{c_\mu}{|x_1^\mu|}$, $\mu = 1, ..., P$.

Considering that the problem is rotationally invariant in the $(d-1)$-subspace $\boldsymbol{x}_\perp$, since $y^\mu = \text{sign}(x_1^\mu)$ and $\boldsymbol{x}_\perp$ is normally distributed, we make the following assumptions for $c_\mu = -\frac{\boldsymbol{w}_\perp}{||\boldsymbol{w}_\perp||} \cdot \boldsymbol{x}_\perp^\mu y^\mu$ and $|x_1^\mu|$:

[5] In this case, we define the early stopping procedure as follows: *(i)* we store the model weights and the validation error at various checkpoints (e.g. every epoch) during the training dynamics; *(ii)* the training dynamics is considered terminated when the training error is zero and the test error is not improving between consecutive checkpoints; *(iii)* we take as final weights of the network those which gave the lowest validation error during the training dynamics. They correspond to some checkpoint before reaching zero training error, since some over-fitting is observed in the last part of the dynamics.

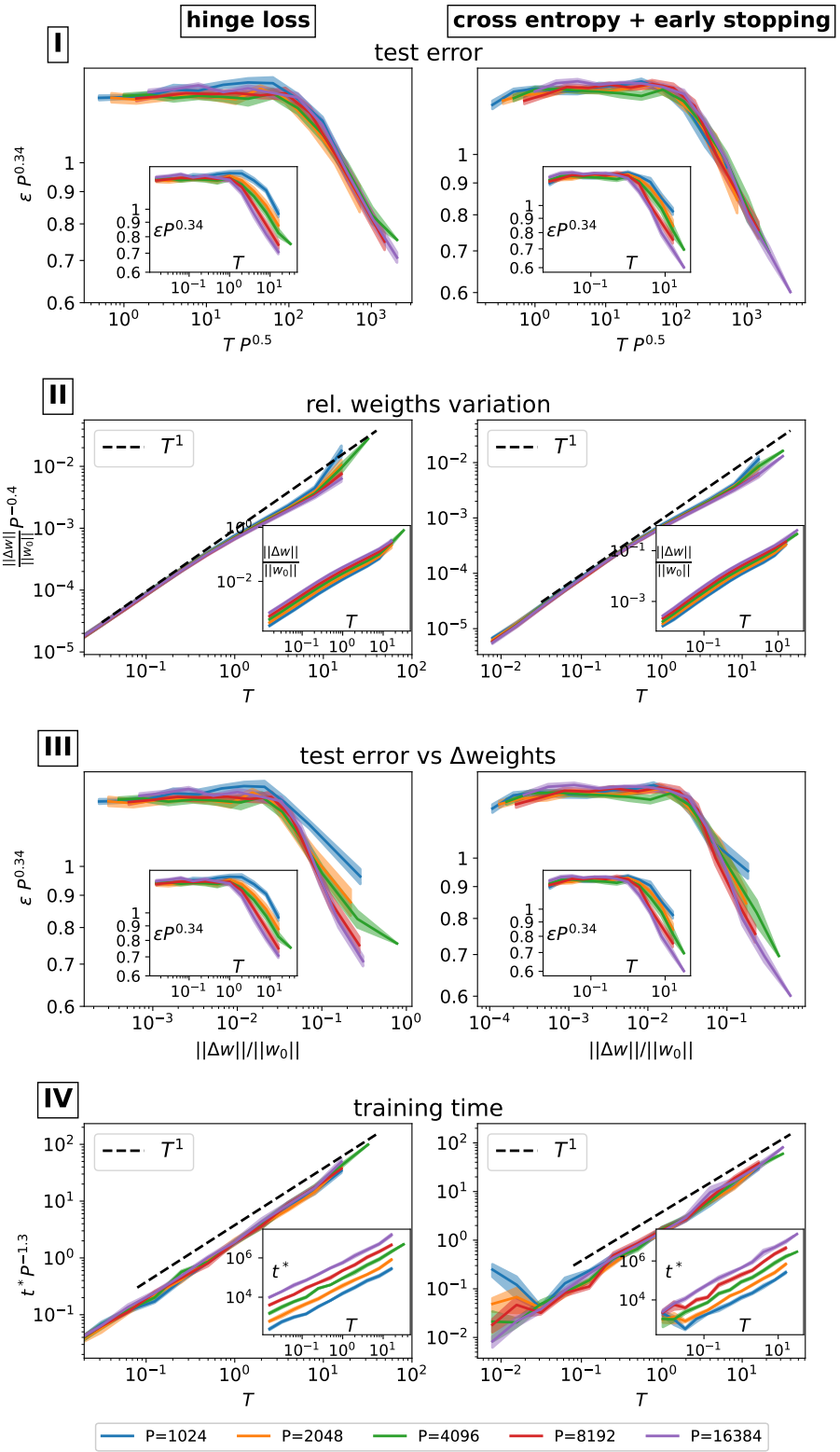

Figure 17. **Comparison of results obtained with the hinge loss trained until 0 loss and the cross-entropy loss performing early-stopping. FC on MNIST, lazy regime, $\alpha = 32768$, $B = 16$, $T = \eta/B$.** On the left column, there are the data obtained with the hinge loss, already presented in Fig. 2 and Fig. 9-(a) (see their captions). On the right column, the same quantities obtained by training with the cross-entropy loss show identical dependence on T and P, with power-laws having the same exponents.

- c_μ are independent and identically distributed (i.i.d.) random variables, whose probability distribution ρ_{c_μ} is Gaussian with zero mean and variance σ^2;

- c_μ and $|x_1^\mu|$ are independent.

Calling $z_\mu = |x_1^\mu|^{-1}$, from 17 the probability distribution of z_μ is given by

$$\rho_{z_\mu}(z) = z^{-\chi-2} e^{1/(2z^2)}/\tilde{Z} \tag{31}$$

with $\tilde{Z}$ the normalization constant.
Since $q_\mu = c_\mu z_\mu = c_\mu |x_1^\mu|^{-1}$ is the product of two independent random variables, its probability distribution is given by the basic formula $\rho_{q_\mu}(q) = \int_0^\infty \rho_{z_\mu}(z)\rho_{c_\mu}(q/z)z^{-1}dz$, which in this case reads:

$$\begin{aligned}
\rho_{q_\mu}(q) &= \int_0^\infty \rho_{z_\mu}(z)\rho_{c_\mu}(q/z)z^{-1}dz = \\
&= \frac{1}{\tilde{Z}\sqrt{2\pi}\sigma}\int_0^\infty z^{-\chi-3} e^{-\frac{1}{2z^2}} e^{-\frac{q^2}{2\sigma^2 z^2}} dz = \\
&= \left(1+\frac{q^2}{\sigma^2}\right)^{-\frac{1}{2}(\chi+2)} \frac{1}{\tilde{Z}\sqrt{2\pi}\sigma}\int_0^\infty z'^{-\chi-3} e^{-\frac{1}{2z'^2}} dz' = \\
&= K\left(1+\frac{q^2}{\sigma^2}\right)^{-\frac{1}{2}(\chi+2)}
\end{aligned} \tag{32}$$

with the normalization constant $K = \frac{2\Gamma\left(\frac{\chi+2}{2}\right)}{\sqrt{\pi}\sigma\Gamma\left(\frac{\chi+1}{2}\right)}$.

Therefore, since in the limit $q \to \infty$ the distribution $\rho_{q_\mu}(q)$ behaves as a power law $\rho_{q_\mu}(q) \sim K\left(\frac{q}{\sigma}\right)^{-(\chi+2)}$, the distribution of the maximum $M_P = \max_\mu q_\mu$, with $\mu = 1, ..., P$, in the limit of large P, converges to the Fréchet distribution (Gnedenko, 1943; Leadbetter et al., 2012):

$$\mathcal{P}\left(a_P M_P < t\right) \underset{P\to\infty}{\to} \exp\left(-t^{-\chi-1}\right), \quad \text{for } t > 0 \tag{33}$$

with

$$a_P = \left(\frac{K\sigma^{\chi+2}}{\chi+1}P\right)^{-\frac{1}{\chi+1}}. \tag{34}$$

Thus, we obtain that the typical value of the maximum $\langle M_P\rangle \propto a_P^{-1}$ behaves asymptotically for $P \to \infty$ as

$$\langle M_P\rangle = CP^{\frac{1}{\chi+1}} + o\left(P^{\frac{1}{\chi+1}}\right) \tag{35}$$

with a constant C.
This asymptotic behaviour can also be found simply by imposing the condition $\int_{\langle M_P\rangle}^\infty \rho_{q_\mu}(q) \sim P^{-1}$ and expanding $\rho_{q_\mu}(q)$ for large q.

G. Additional plots

G.1. Learning rate and batch size

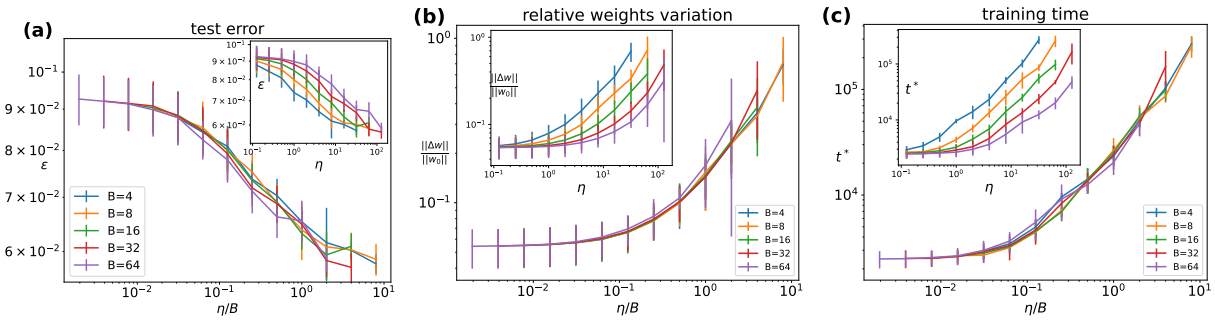

Figure 18. **FC on MNIST, $\alpha = 1$, $P = 1024$: (a) test error, (b) relative weight variation, (c) training time with respect to learning rate η and batch size B.** The represented quantities depend on the ratio η/B.

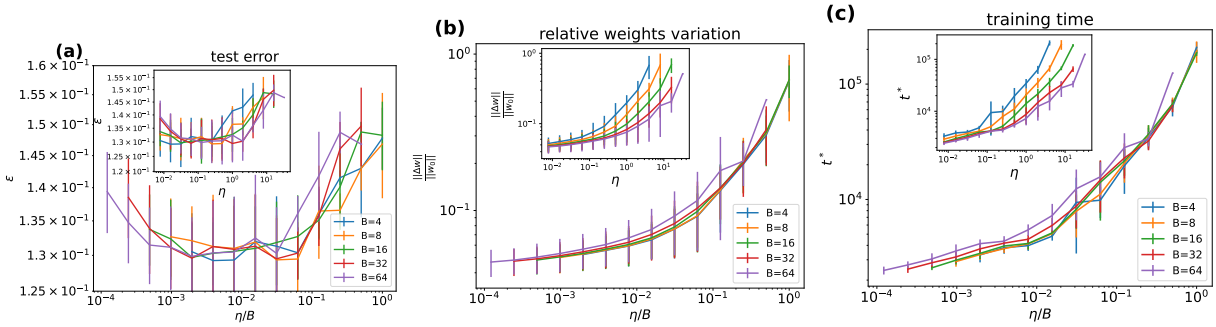

Figure 19. **CNN (MNAS) on CIFAR, $\alpha = 1$, $P = 1024$: (a) test error, (b) relative weight variation, (c) training time with respect to learning rate η and batch size B.** The represented quantities depend on the ratio η/B.

G.2. Error estimation

This section describes the method used to estimate errors on the exponents presented in Table 1. We choose the exponents such that the rescaled curves overlap (i.e. the curves 'collapses'). We estimate the error bars on the exponents based on the quality of this collapse, which we determine to be approximately ± 0.2. To illustrate this process, we consider the data for one example, the fully-connected architecture on MNIST in the lazy regime (Fig. 20).

In the first column of Fig. 20 (A-I, B-I, C-I), we observe that the test error ϵ starts decreasing at a characteristic temperature T_c, which depends on P as $T_c \sim P^{-a}$. Therefore, plotting ϵ versus TP^a should align the curves. We find that $a = 0.5$ produces the best collapse, while $a = 0.3$ (A-I) and $a = 0.7$ (C-I) respectively underestimate and overestimate the value of a. Hence, we estimate a to be 0.5 ± 0.2. The same procedure is used to estimate the errors on the exponents γ, δ of $\Delta w \sim T^\delta P^\gamma$ (Fig. 20 (A-II, B-II, C-II)) and the exponent b of $t^* \sim TP^b$ (Fig. 20 (A-III, B-III, C-III)).

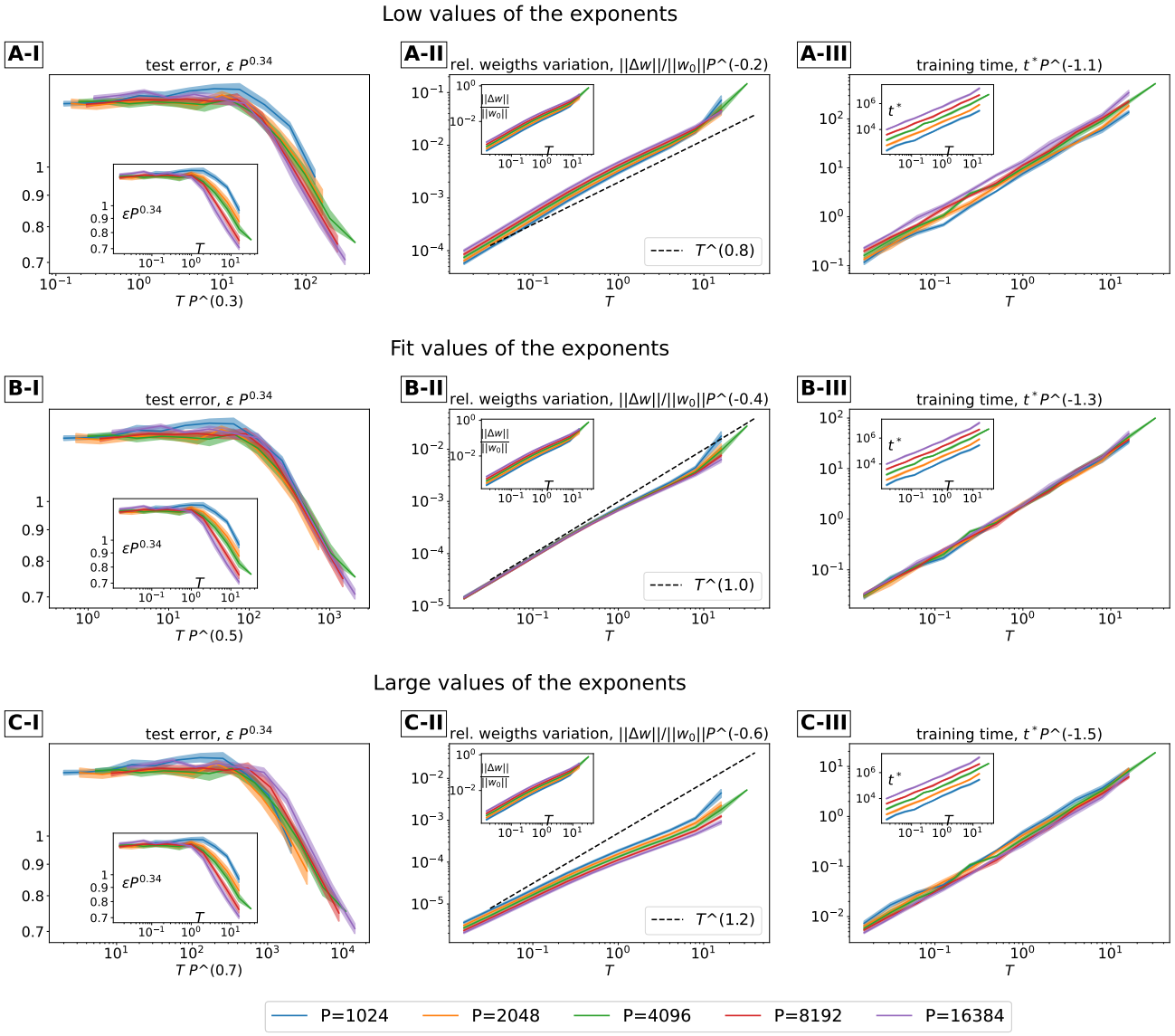

Figure 20. **Error estimation on the exponents, FC on CIFAR, lazy regime,** $\alpha = 32768$**,** $B = 16$**.** *First column (A-I, B-I, C-I): test error* ϵ *vs* T. The insets show that ϵ starts improving at T_c depending on P. The main panels show that the best curves collapse is obtained plotting ϵ vs $TP^{0.5}$ (B-I) rather than $TP^{0.3}$ (A-I) or $TP^{0.7}$ (C-I), indicating $T_c \sim P^{-a}$ with $a = 0.5 \pm 0.2$. *Second column (A-II, B-II, C-II): relative weight variation* Δw *vs* T. The insets show that Δw increases with both T and P. The main panels show that the best curve collapse is obtained plotting $\Delta w P^{-0.4}$ (B-II) rather than $\Delta w P^{-0.2}$ (A-II) or $\Delta w P^{-0.6}$ (C-II). Similarly, the slope of the curve is best matched by T^1 (B-II) rather than $T^{0.8}$ (A-II) or $T^{1.2}$ (C-II). This indicates that $\Delta w \sim P^{\gamma} T^{\delta}$ with $\gamma = 0.4 \pm 0.2$ and $\delta = 1 \pm 0.2$. *Third column (A-III, B-III, C-III): training time* t^* *vs* T. The insets show that t^* increases with both T and P. The main panels show that the best curve collapse is obtained plotting $t^* P^{-1.3}$ (B-III) rather than $t^* P^{-1.1}$ (A-III) or $t^* P^{-1.5}$ (C-III), indicating $t^* \sim TP^b$ with $b = 1.3 \pm 0.2$.

G.3. Perceptron dynamics

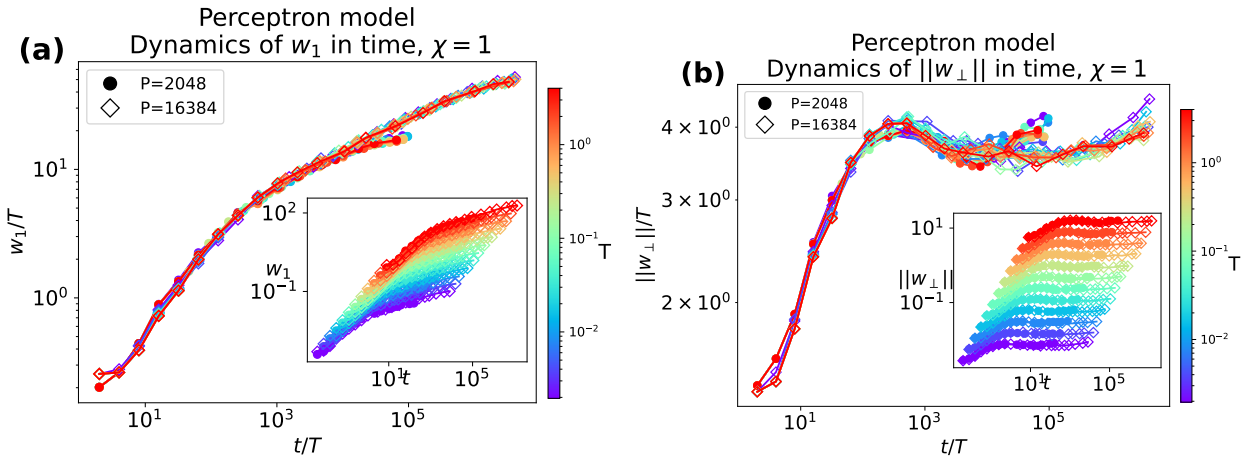

Figure 21. **Training dynamics in the perceptron model.** Data are obtained with data distribution $\chi = 1$, dimension $d = 128$, batch size $B = 2$, varying learning rate η ($T = \eta/B$). **(a)** Evolution of the weight w_1 with respect to time t (=number of steps times learning rate) for different SGD temperatures T (colors) and training set sizes P (symbols). A larger T corresponds to a larger variation of w_1 in a longer time, while the training set size P determines only the end-point of the dynamics. **(b)** Evolution of the norm of the orthogonal weights $||\boldsymbol{w}_\perp||$ in time, for the same setting as panel (a). For larger T, $||\boldsymbol{w}_\perp||$ reaches higher plateau values while P determines only the end-point of the dynamics.