

PAPER • OPEN ACCESS

## Shift-curvature, SGD, and generalization

To cite this article: Arwen V Bradley *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 045002

View the [article online](#) for updates and enhancements.

You may also like

- [Establishing functional correlations between multiscale areal curvatures and coefficients of friction for machined surfaces](#)  
T Bartkowiak, J Berglund and C A Brown
- [Stochastic stabilization of cosmological photons](#)  
C P Dettmann, J P Keating and S D Prado
- [Porous Thick Film Low Temperature Lanthanum Strontium Ferrite Oxygen Surface Exchange Coefficient Measurements By Curvature Relaxation](#)  
Qing Yang and Jason D. Nicholas



## PAPER

## Shift-curvature, SGD, and generalization

## OPEN ACCESS

RECEIVED  
1 August 2022REVISED  
8 September 2022ACCEPTED FOR PUBLICATION  
16 September 2022PUBLISHED  
10 October 2022

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.

Arwen V Bradley<sup>\*,1</sup> , Carlos A Gomez-Uribe<sup>1</sup> and Manish Reddy Vuyyuru 

Apple Inc., 1 Apple Park Way, Cupertino, CA, United States of America

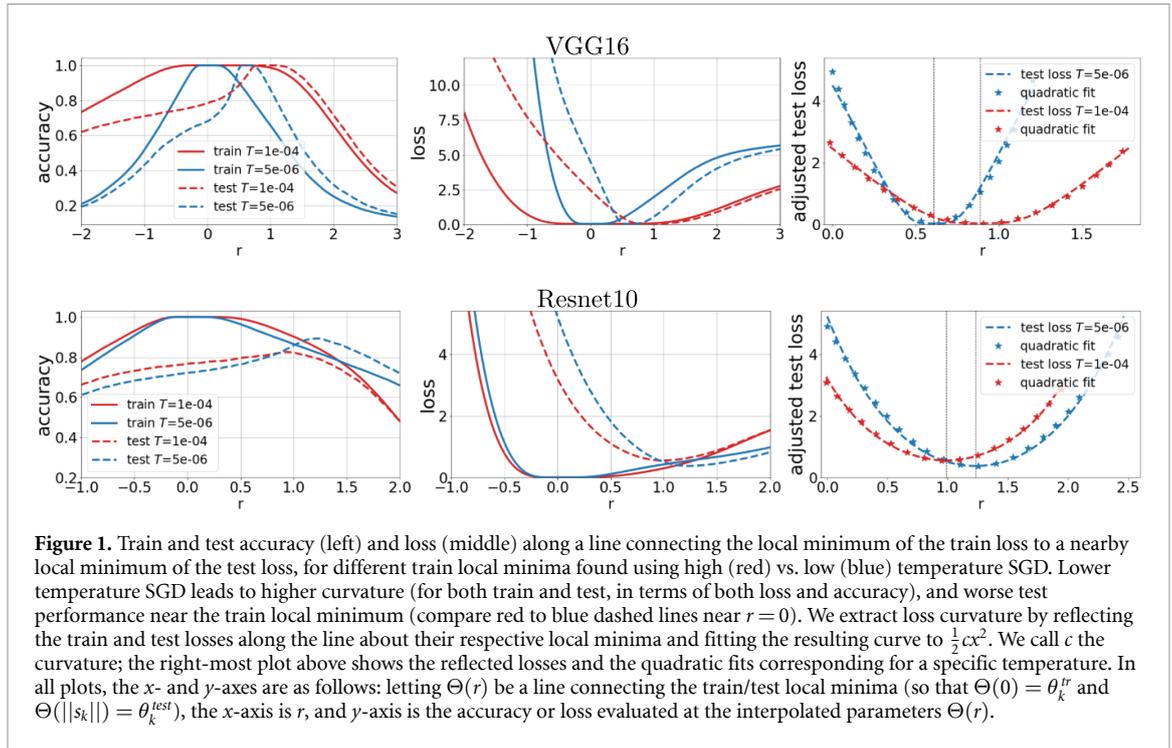
\* Author to whom any correspondence should be addressed.

<sup>1</sup> Authors contributed equally.E-mail: [arwen\\_bradley@apple.com](mailto:arwen_bradley@apple.com)**Keywords:** generalization, curvature, stochastic gradient descent noise, nonconvex optimization, diffusion approximation**Abstract**

A longstanding debate surrounds the related hypotheses that low-curvature minima generalize better, and that stochastic gradient descent (SGD) discourages curvature. We offer a more complete and nuanced view in support of both hypotheses. First, we show that curvature harms test performance through two new mechanisms, the shift-curvature and bias-curvature, in addition to a known parameter-covariance mechanism. The shift refers to the difference between train and test local minima, and the bias and covariance are those of the parameter distribution. These three curvature-mediated contributions to test performance are reparametrization-invariant even though curvature itself is not. Although the shift is unknown at training time, the shift-curvature as well as the other mechanisms can still be mitigated by minimizing overall curvature. Second, we derive a new, explicit SGD steady-state distribution showing that SGD optimizes an effective potential related to but different from train loss, and that SGD noise mediates a trade-off between low-loss versus low-curvature regions of this effective potential. Third, combining our test performance analysis with the SGD steady state shows that for small SGD noise, the shift-curvature is the dominant of the three mechanisms. Our experiments demonstrate the significant impact of shift-curvature on test loss, and further explore the relationship between SGD noise and curvature.

**1. Introduction**

Understanding generalization remains one of the key questions in machine learning. In typical machine learning applications, we train a model on a training dataset hoping that the model will generalize well to a test dataset, which is a proxy for unseen data. However, our theoretical understanding of how datasets, models, and learning algorithms combine to determine test performance is still incomplete, particularly in the case of overparametrized models that defy classical expectations about overfitting (Belkin 2021). A decades-old debate focuses on the related hypotheses that low curvature of the loss function results in better test performance, and that stochastic gradient descent (SGD) favors lower-curvature local minima. There is strong evidence for these hypotheses. First, many studies have shown that increasing SGD noise (by increasing the ratio of learning rate to batch size) can improve test performance (Goyal *et al* 2017, Hoffer *et al* 2017, You *et al* 2017, Golmant *et al* 2018, McCandlish *et al* 2018, Shallue *et al* 2018, He *et al* 2019a, Smith *et al* 2020). Our experiment in figure 8 also confirms this. Second, Keskar *et al* (2016), Li *et al* (2017a), Jiang *et al* (2019) offer empirical evidence that lower-curvature local minima tend to generalize better, an idea often attributed to Hochreiter and Schmidhuber (1997). Third, Jastrzebski *et al* (2017) make direct theoretical connections between test performance, curvature, and SGD noise (albeit under strong assumptions). They show that under the SGD steady-state distribution, the expected test loss near a given local minimum depends on the trace of curvature times parameter covariance, assuming a constant Hessian equal to SGD noise. They also show that the probability of SGD landing near a particular local minimum is inversely related to its curvature, with noise mediating a depth/curvature tradeoff, assuming constant, isotropic SGD noise and valid Laplace approximation. On the other hand, there are several arguments



against the idea that curvature predicts test performance. Dinh *et al* (2017) point out that model reparametrization can arbitrarily change curvature without changing test performance. Further, a number of recent works argue that Jastrzebski *et al* (2017) and others present an oversimplified view of curvature, since loss landscapes often (e.g. when there are many more model parameters than training datapoints) have many directions of near-zero curvature rather than locally strictly-convex valleys; in particular, the SGD noise matrix and Hessian of the loss are different, and both are highly non-isotropic with many near-zero eigenvalues (Draxler *et al* 2018, Sagun *et al* 2018, Li *et al* 2020). Our work aims to help reconcile these points of view by removing unrealistic assumptions such as constant, isotropic Hessian and loss gradient covariance, identifying new and possibly more significant mechanisms by which curvature affects test performance, and showing that these curvature-dependent contributions to test performance are reparametrization invariant.

At a high level, we find that the intuitive connections between test performance, curvature, and SGD noise still hold, but in a more complete and nuanced way. We show that the test loss averaged over an arbitrary parameter distribution depends on the *shift-curvature* (in addition to the *bias-curvature* and the previously-identified covariance-curvature). The shift-curvature is equal to the curvature of the loss in the direction of the *shift* between nearby train and test local minima, as shown in figure 1; such a shift may be caused by dataset sampling or possibly by test and train datasets coming from different distributions. However, the shift is unknown at training time, so it is beneficial to reduce curvature in all directions. Second, we show that the SGD steady-state distribution favors train local minima with low curvature in all directions, although the curvature is not that of the train loss, but rather of a related *effective potential* which we obtain by refining existing results about the steady state of SGD. Third, we find that the shift-curvature may be the most significant curvature term when SGD noise is small. Our empirical results show that shift-curvature indeed has a significant impact on test loss.

In more detail, we refine existing results for the steady-state distribution of SGD as follows. Starting from a continuous-time diffusion approximation of SGD due to Li *et al* (2017b) (for which we also offer an alternative, constructive derivation which may be helpful for analyzing other algorithms), we derive a steady-state distribution showing that SGD minimizes an explicit effective potential that is related to but different from the training loss. Our expression is more explicit than the result in Chaudhari and Soatto (2018), and more general than that in Jastrzebski *et al* (2017) and Seung *et al* (1992) since we remove their assumptions of constant and isotropic SGD noise.

This paper is organized as follows. Sections 2 and 3 present the setup and main results for the simpler underparametrized case (more data samples than model parameters). Section 4 discusses the SGD diffusion approximation and its steady state in more detail, describing how the results in section 3 are obtained. Section 5 extends the results to the more complex overparametrized case. The appendices contain derivations and additional experiments.

**Table 1.** Notation reference guide.

Symbol	Definition	Notes
$U(x, \theta)$		Loss function
$U^{tr/test}(\theta) =$	$E_{x \sim tr/test}[U(x, \theta)]$	Train/test loss at $\theta$
$\theta_k^{tr/test} =$	$\text{argmin} U^{tr/test}(\theta)$	Local minima of test/train loss
$U_k^{tr/test}$	$U^{tr/test}(\theta_k^{tr/test})$	Test/train loss at local minima
$C_k^{tr/test}$	$\partial_{\theta}^2 U^{tr/test}(\theta_k^{tr/test})$	Curvature at test/train local minima
$s_k =$	$\theta_k^{test} - \theta_k^{tr}$	Shift between train/test minima
$\rho(\theta) \approx$	$\sum_k w_k \mathcal{N}(\mu_k, \Sigma_k^2)$	Approx. parameter distribution
$b_k =$	$\theta_k^{tr} - \mu_k$	Bias of $\rho$ at $k$
$D^{tr}(\theta) =$	$\text{Cov}_{x \sim tr}[\partial_{\theta} U(x, \theta)]$	SGD diffusion matrix

## 2. Setup

In this section and the next, we present our setup and main results in the underparametrized case (section 5 extends to the overparametrized case). We assume we are given a *loss* function  $U(x, \theta)$  that depends on a single data sample  $x \in \mathbb{R}^{\iota}$  and model parameters  $\theta \in \mathbb{R}^p$ , where  $\iota$  and  $p$  are positive integers. We are also given a training set  $\{x_1^{tr}, \dots, x_{N_{tr}}^{tr}\}$  (*tr* is short for train) consisting of  $N_{tr}$  points sampled i.i.d. from an underlying train distribution, and a test set  $\{x_1^{test}, \dots, x_{N_{test}}^{test}\}$  sampled i.i.d. from an underlying test distribution. Often the train and test distributions are identical, but sometimes they differ from one another (for example, when the data distribution evolves over time and test is sampled at a later time than train). But even when train and test are both sampled from the same underlying distribution, the train and test *sets* still differ due to sampling (because each data set consists of a finite number of samples). We define the train and test losses for a given  $\theta$  as the expectation over the train and test sets, respectively:

$$U^{tr}(\theta) = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} U(x_i^{tr}, \theta), \text{ and } U^{test}(\theta) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} U(x_i^{test}, \theta). \tag{1}$$

In general,  $U^{tr}(\theta)$  and  $U^{test}(\theta)$  are similar but not identical functions, each with multiple local minima. We make three initial assumptions about the local minima: (i) the local minima of both train and test are strict, (ii) each local minimum of the train loss has a nearby local minimum of the test loss, and (iii) the train local minima are countable. We expect (i) to hold for underparametrized models (provided there are at least  $p$  linearly-independent per-sample gradients) but not for overparametrized models (which are handled in section 5, where we drop assumption (i)). Assumption (ii) is validated by our experiments, and can be satisfied, for example, in the absence of distribution shift by having large enough datasets. Both (ii) and (iii) could actually be relaxed to apply only to train local minima that receive significant weight in the SGD steady-state distribution, as (4) will make clear, but we will assume they hold for all train minima to simplify the exposition. With these assumptions, we denote the (strict) local minima of train by  $\theta_k^{tr}$  and the corresponding closest local minima of test by  $\theta_k^{test}$ , and generally reserve  $k$  as an index that runs through train local minima ( $k = 1, 2, 3, \dots$ ). We can then define the *shift* between corresponding local minima of train and test as:

$$s_k = \theta_k^{test} - \theta_k^{tr}. \tag{2}$$

The shift is small as a consequence of assumption (ii), but nonzero in general because the train and test losses are slightly different (due to dataset sampling or possibly distribution shift). He *et al* (2019b) experiments and our own demonstrate a nonzero train/test shift for several models and datasets. (We generalize definition (2) to apply to non-strict minima in section 5.) Table 1 summarizes the main notation we use for easy reference.

We further assume that a stochastic learning algorithm such as SGD processes the train dataset to produce model parameters with a distribution  $\rho(\theta)$ ; that is, different runs of SGD on the same training set yield i.i.d. samples from  $\rho(\theta)$ . The resulting models are evaluated on a test dataset to determine the test performance. One of our main goals is to understand the expected test loss over the distribution of model parameters the algorithm produces (while keeping datasets fixed).

## 3. Main results

First, consider a solution  $\hat{\theta}$  obtained by a single run of a training algorithm like SGD, i.e.  $\hat{\theta}$  is a sample from  $\rho(\theta)$ . To predict the performance of the trained model on unseen data, we would evaluate the test loss at  $\hat{\theta}$ .

We expect  $\hat{\theta}$  to be near a train local minimum  $\theta_k^{tr}$  because it was found by an algorithm that attempts to minimize the train loss. We can therefore write  $\hat{\theta} = \theta_k^{test} - s_k - \hat{b}$ , where  $s_k$  is the train/test shift introduced above, and  $\hat{b} = \theta_k^{tr} - \hat{\theta}$  denotes the *bias* of  $\hat{\theta}$  relative to the train local minimum, with both  $s_k$  and  $\hat{b}$  assumed small. We then Taylor expand  $U^{test}$  about  $\theta_k^{test}$ , noting that  $\partial_{\theta} U^{test}(\theta_k^{test}) = 0$ :

$$U^{test}(\hat{\theta}) \approx U_k^{test} + \frac{1}{2} (s_k + \hat{b})^T C_k^{test} (s_k + \hat{b}), \quad (3)$$

where we introduce the more compact notation  $U_k^{test} = U^{test}(\theta_k^{test})$  for the test loss at a local minimum, and  $C_k^{test} = \partial_{\theta}^2 U^{test}(\theta_k^{test})$  for the Hessian (or *curvature*) of the test loss at one its local minima<sup>1</sup>. Although it comes from a simple Taylor-expansion, equation (3) identifies important quantities impacting test loss that have so far been inadequately discussed in the literature. The equation says that the test loss evaluated near a train local minimum is equal to the *test loss at the test local minimum* plus a term that depends on an interaction between the *curvature*, the *shift* between the train and test local minima, and the *bias* between  $\hat{\theta}$  and the train local minimum. The *shift-curvature* term,  $\frac{1}{2} s_k C_k^{test} s_k$ , plays an especially important role in the rest of this paper; the basic intuition is that, given train and test losses with local minima slightly shifted relative to each other, the test loss evaluated near a train local minimum depends on the local curvature in the direction of the local-minimum shift (Keskar et al 2016 seem to appeal to similar intuition, but do not make it concrete). Since the shift cannot be known at training time, it is beneficial to minimize curvature in all directions, which, as we will see shortly, is what SGD does.

In order to obtain an explicit approximation for the expected test loss over a distribution of parameters, we model the parameter distribution  $\rho(\theta)$  as a countable mixture of Gaussians with one mixture component for each train loss minimum, as proposed by Jastrzkebski et al (2017) for the case of SGD. That is, we write  $\rho(\theta) = \sum_k w_k \mathcal{N}(\mu_k, \Sigma_k)$ , where  $w_k \geq 0$ ,  $\sum_k w_k = 1$ , the component means  $\mu_k$  are close to the train local minima,  $b_k = \theta_k^{tr} - \mu_k$  denote the component *biases*, and  $\Sigma_k$  denote the component covariances<sup>2</sup>. Averaging the approximate test loss in (3) over such a parameter distribution results in the approximate test performance:

$$E_{\theta \sim \rho} [U^{test}(\theta)] \approx \sum_k w_k \left\{ U_k^{test} + \frac{1}{2} \text{Tr}[\Sigma_k C_k^{test}] + \frac{1}{2} (s_k + b_k)^T C_k^{test} (s_k + b_k) \right\}, \quad (4)$$

here  $\text{Tr}[\cdot]$  denotes the trace of a matrix. Equation (4) is one of our main results, and makes explicit how curvature determines test performance: through the covariance of model parameters  $\Sigma_k$ , as well as a quadratic function of the shifts  $s_k$  and biases  $b_k$ . Buntine (1991), Jastrzkebski et al (2017) identify the same  $\text{Tr}[\Sigma_k C_k^{test}]$  contribution of curvature to test performance (under the assumption that  $C = D$ , which we remove), but the other curvature-dependent terms in (4) are new as far as we know. In particular, the new shift-curvature term (discussed above) appears as  $\frac{1}{2} w_k s_k C_k^{test} s_k$ , showing that test loss improves when  $\rho$  places more weight on minima with lower shift-curvature. The role of shift-curvature is consistent with the empirical results of Verpoort et al (2020), who show that curvature (as measured by  $\log |C|$ ) has a major impact on test performance for small sample sizes, which correspond to larger train/test shift.

We can also show that (4) is reparametrization-invariant. Although multiple authors, including Keskar et al (2016), Jiang et al (2019), and Hochreiter and Schmidhuber (1997), have suggested that generalization quality can be tied directly to curvature, Dinh et al (2017) argue against this by observing that model reparametrization can arbitrarily change the curvature of the local minima. Although the curvature alone is indeed not invariant to reparametrization, we show in appendix C that every term in (4) is reparametrization-invariant, as is the entire expression. The intuition behind the proof is that reparametrization changes terms in (4) like  $s_k$  and  $C_k^{test}$  in ways that cancel each other out, i.e. under a reparametrization  $y = r^{-1}(\theta)$ , the reparametrized shift becomes  $\partial_y r(y_k^{test})^{-1} s_k$  while the reparametrized curvature becomes  $\partial_y r(y_k^{test})^T C_k^{test} \partial_y r(y_k^{test})$  (where  $\partial_y r(y)^{-1}$  is the matrix inverse of the Jacobian of  $r$ ), so that  $s_k C_k^{test} s_k$  is left unchanged. Other authors like Smith and Le (2017) have shown reparametrization-invariant PAC-Bayes bounds for the generalization error, but we show the left-hand-side of (4) is a reparametrization-invariant quantity that directly approximates the expected test loss.

<sup>1</sup> In section 5, we find that (3) still holds in the overparametrized case after generalizing the definition of  $s_k$  to apply to non-strict minima. We further note that for asymmetric losses, only the curvature in the direction from the test toward the train local minimum matters, as explored in our experiments. Finally, in appendix E, we show how this result and others can be partially generalized to the case where the quadratic term is zero, but curvature can still be characterized in terms of a higher-order even derivative.

<sup>2</sup> Generally, such a model is reasonable when  $\rho$  has the form  $\rho(\theta) \propto e^{-c v(\theta)}$  for some constant  $c$  and function  $v$  with local minima near those of the train loss, so that  $\rho$  is multimodal with peaks near the train local minima, enabling a Laplace approximation about each local minimum of  $v$ , which yields a Gaussian mixture with small biases. We will see shortly that the SGD steady state has this form.

Our second main result is the approximate steady-state distribution of SGD, derived in section 4:

$$\rho(\theta) = \frac{1}{Z} \exp \left\{ -\frac{2}{T} v(\theta) \right\}, \quad \text{where } Z \equiv \int \exp \left\{ -\frac{2}{T} v(\theta) \right\} d\theta, \quad \text{and}$$

$$v(\theta) \equiv \int^{\theta} (D^{tr})^{-1} \left( \partial_{\theta} U^{tr} + \frac{T}{2} (\partial_{\theta} \cdot D^{tr})^T \right) \cdot d\theta. \quad (5)$$

We call  $v(\theta)$  the *effective potential*,  $D^{tr}(\theta)$  is the training set gradient covariance,  $\partial_{\theta} U^{tr}(\theta)$  is the gradient of the training loss, and  $Z$  is the partition function of this Boltzmann-like distribution. The *temperature*  $T$ , which is equal to the ratio of learning rate and minibatch size (so typically  $T \ll 1$ ), captures the strength of SGD noise (as discussed further in section 4). The effective potential is related to the training loss  $U^{tr}(\theta)$ , but in general not the same:  $v \propto U^{tr}$  only if  $D^{tr}$  is constant and isotropic<sup>3</sup>. Result (5) requires the assumption that the integrand in the line integral is a gradient, which is equivalent to having zero curl, as well as for  $D^{tr}(\theta)$  to be invertible, which is only true in underparametrized case<sup>4</sup>. Equation (5) is more general than the results in Seung et al (1992), Jastrzebski et al (2017), which assume that the noise  $D^{tr}(\theta)$  is constant and isotropic (i.e. a multiple of the identity). It is different and more explicit than results in Chaudhari and Soatto (2018) for nonconstant, anisotropic noise, enabled by the zero-curl assumption that we rely on to obtain a solution with zero probability current ( $J$  in equation (10)). Chaudhari and Soatto (2018) say that the steady state cannot have zero probability current for anisotropic and/or non-constant noise, but we offer a counterexample in appendix B.1. They also stop short of explicitly solving for the effective potential, presumably because they are rightly concerned about inverting  $D^{tr}$ ; however, our analysis of the nullspace of  $D^{tr}$  shows that this is only an issue in the overparametrized case, in which case a modified SGD still allows us to find a solution (see section 5).

Approximating  $\rho$  as a mixture of Gaussians shows that  $\rho$  assigns to basin  $k$  the weight:

$$w_k \propto \exp \left\{ -\frac{2}{T} v(\mu_k, T) \right\} \cdot |\partial_{\theta}^2 v(\mu_k, T)|^{-\frac{1}{2}}. \quad (6)$$

This shows how SGD noise mediates a trade-off between depth and curvature of the effective potential, generalizing the result given in Jastrzebski et al (2017) for the special case of constant, isotropic covariance  $D^{tr}(\theta)$ . When  $T = 0$  the weights  $w_k$  indicate that  $\rho(\theta)$  places all its probability mass on the deepest minimum of the effective potential, but as  $T$  increases,  $\rho(\theta)$  begins to transfer weight onto other minima. It then favors minima with lower curvature of the effective potential due to the inverse-square-root dependence on the determinant of the curvature of the effective potential. In particular, if two local minima have the same depth, SGD will prefer the one with lower curvature. Compared to Jastrzebski et al (2017)'s result, ours highlights the complexity arising from non-isotropic noise, in particular that the depth and curvature are those of the *effective potential* rather than those of the training loss.

Third, to approximate the expected SGD test performance, we specialize (4) to the Gaussian mixture approximation of the SGD steady-state  $\rho$  in (5) to obtain:

$$E_{\theta \sim \rho}[U^{test}(\theta)] \approx \sum_k w_k \left( U_k^{test} + \frac{1}{2} s_k^T C_k^{test} s_k \right) + O(T). \quad (7)$$

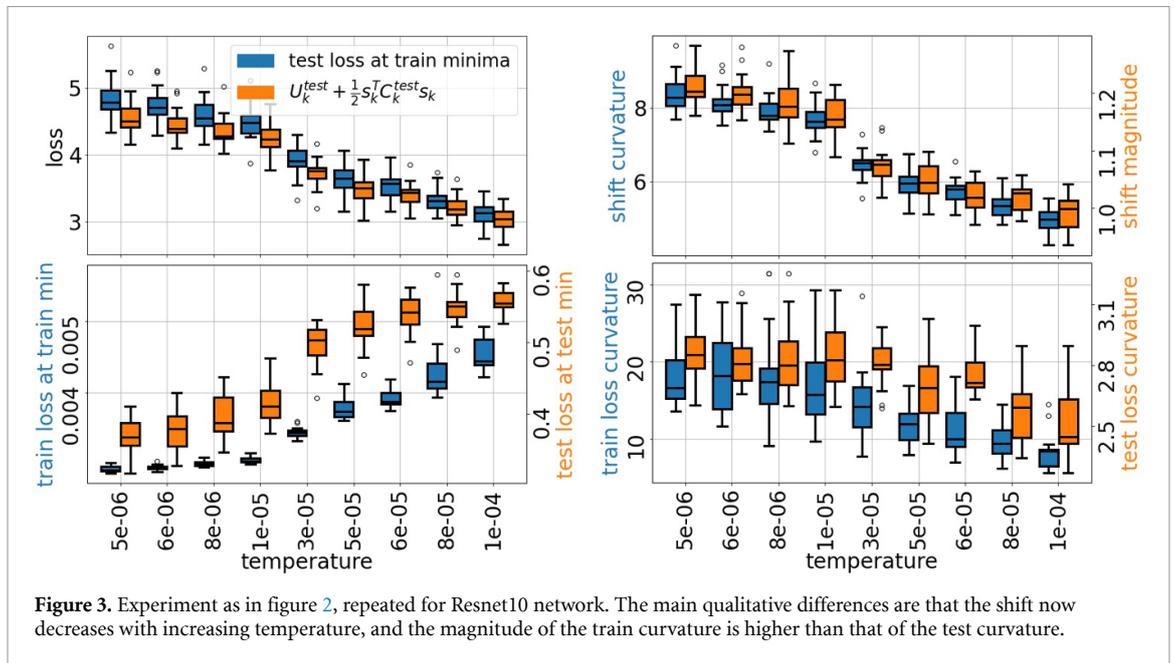
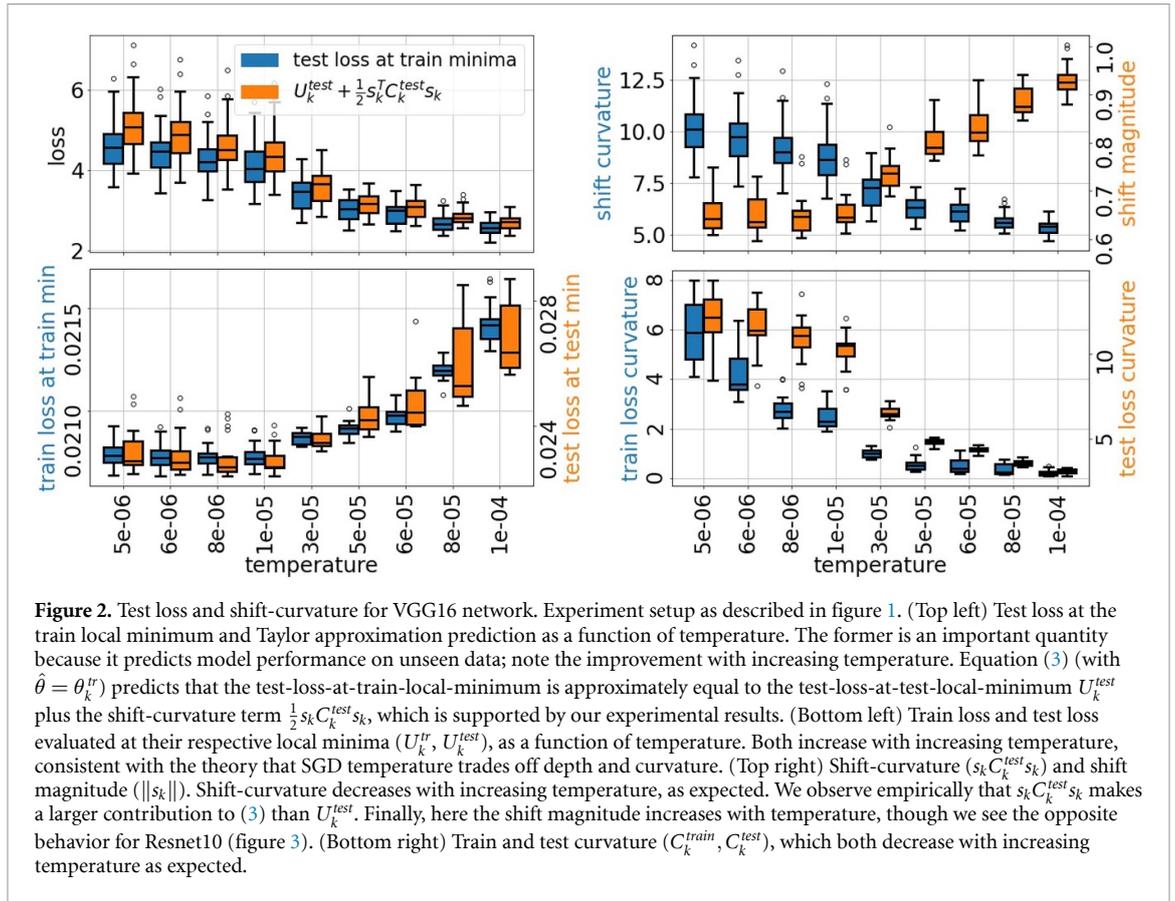
The terms in (4) involving bias  $b_k$  and covariance  $\Sigma_k$  are both  $O(T)$ , as detailed in section 4. Equation (7) shows that for SGD with small  $T$  and non-negligible shifts  $s_k$ , our newly-identified shift-curvature term may be the most significant way that curvature affects the test loss. In summary, we find that SGD temperature mediates a trade-off between depth and curvature of the effective potential, so that as temperature increases, greater weight  $w_k$  is placed on lower-curvature basins. These lower-curvature basins tend to have smaller values of shift-curvature,  $s_k^T C_k^{test} s_k$ , which can reduce the expected test loss  $E_{\theta \sim \rho}[U^{test}(\theta)]$  overall<sup>5</sup>. We illustrate this with a conceptual example in the next section.

Our experiments in figures 1–3 demonstrate the significant contribution of shift-curvature to test performance, support the inverse dependence of curvature on SGD temperature, and offer additional

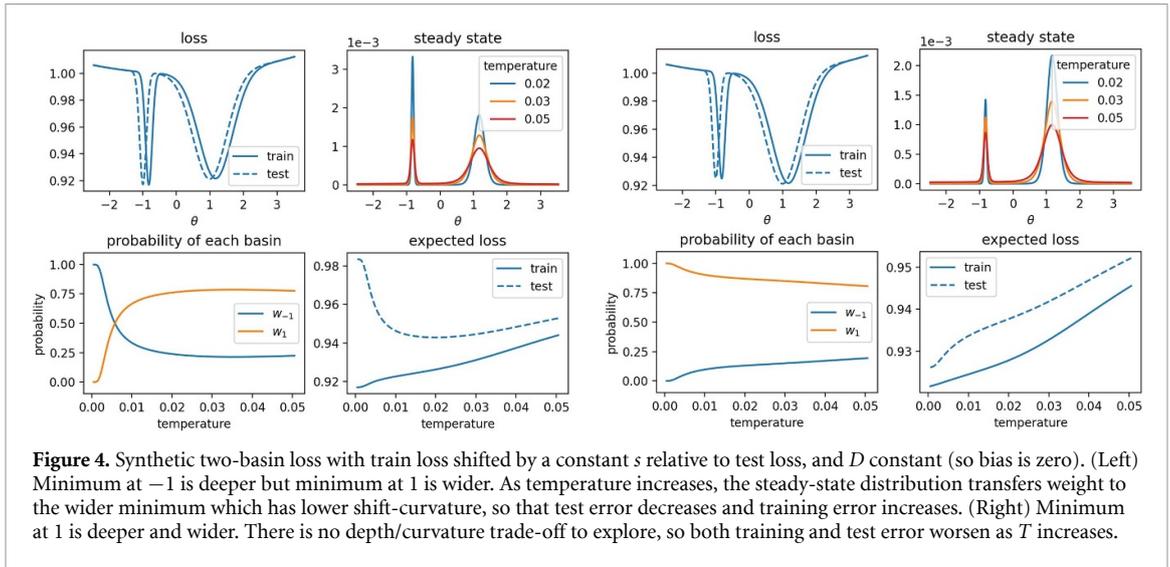
<sup>3</sup> When  $D^{tr} = d(\theta)I$  for some scalar function  $d(\theta)$ , then  $v(\theta) = \int^{\theta} \frac{1}{d(\theta)} \partial_{\theta} U^{tr} \cdot d\theta + \frac{T}{2} \log(d(\theta))$ . If in addition  $d(\theta)$  is a constant denoted by  $d$ , then  $v(\theta) = d^{-1} U^{tr}(\theta) + \text{const}$ .

<sup>4</sup> The zero-curl assumption is stated and discussed in appendix B.1 (assumption 1). The invertibility of  $D^{tr}$  is handled in section 5, where we find that an analog of (5) still holds in the overparametrized case if we add zero-mean Gaussian noise to SGD and  $\ell_2$  regularization to the training loss.

<sup>5</sup> This is in contrast to the effective potential  $v$ , since some algebra shows that  $\partial_T E_{\theta \sim \rho}[v] = \frac{1}{T^2} \text{Var}[v] > 0$ , so  $v$  can only worsen with increasing  $T$ . Interpreting  $E_{\theta \sim \rho}[v]$  as the energy of the system makes  $\partial_T E_{\theta \sim \rho}[v]$  the heat capacity (Niroomand et al 2022).



empirical insights about the test local minima, curvature, and shift. For figures 1–3, we trained VGG16 (Simonyan and Zisserman 2014) and Resnet10 (He et al 2016) networks, both without batch normalization, on CIFAR10 (Krizhevsky et al 2009). We also performed the same experiment and obtained qualitatively similar results for Resnet20 with batch normalization on CIFAR10, and Resnet18 with batch normalization a subset of ImageNet (Deng et al 2009) (figures 5–7 in appendix G.1). The experiment is designed as follows. We repeated the following procedure over a range of temperatures for the initial SGD run, and over several seeds for each temperature. For each temperature and seed, we first trained our model using SGD with the specified temperature to get close to local minima of train (note that SGD typically finds *different* train local



**Figure 4.** Synthetic two-basin loss with train loss shifted by a constant  $s$  relative to test loss, and  $D$  constant (so bias is zero). (Left) Minimum at  $-1$  is deeper but minimum at  $1$  is wider. As temperature increases, the steady-state distribution transfers weight to the wider minimum which has lower shift-curvature, so that test error decreases and training error increases. (Right) Minimum at  $1$  is deeper and wider. There is no depth/curvature trade-off to explore, so both training and test error worsen as  $T$  increases.

minima for different temperatures). Next, we initialized from the SGD solution and kept training at extremely low temperature on the train set to get even closer to the train local minimum. Finally, we again initialized from the SGD solution and trained at extremely low temperature, but this time on the *test set*, to get as close as possible to the test local minimum. Having found the train and test local minima, we linearly interpolated between them in order to study the train and test accuracy and loss along the line connecting their local minima (as shown in figure 1 for two different initial SGD temperatures; note the higher curvature at lower SGD temperature). Finally, we estimated the curvature of the loss functions along this line by reflecting them about their local minima in the direction from  $\theta_k^{test}$  toward  $\theta_k^{tr}$ , and fitting a quadratic centered at the local minimum (as shown in figure 1, right pane). (We focus on the line connecting the local minima because all quantities in (3) with  $\hat{\theta} = \theta_k^{tr}$  lie along this line. Letting  $\Theta(r)$  denote the line, with  $\Theta(0) = \theta_k^{tr}$  and  $\Theta(\|s_k\|) = \theta_k^{test}$ , the curvature of the fitted quadratic corresponds to  $s_k C_k s_k / \|s_k\|^2$ , as shown in (31).) Although we could compute the full Hessian matrix, in these experiments we are mainly interested in the curvature along the line  $\Phi(r)$  which can be computed much more cheaply via the 1D projection; further, because of the shapes of the losses we find that the symmetrized quadratic fit better captures the relevant overall curvature of the basin, as discussed in appendix F. The plots in figures 2 and 3 show trends in key quantities (losses and curvature at local minima, shift magnitude, and shift-curvature) as a function of the initial SGD temperature. The results validate (3) (with  $\hat{\theta} = \theta_k^{tr}$ ) by showing that the test loss at the train minimum is approximately equal to the test loss at the test local minimum  $U_k^{test}$  plus the shift-curvature  $\frac{1}{2} s_k^T C_k^{test} s_k$ ; empirically  $\frac{1}{2} s_k^T C_k^{test} s_k$  appears to be more significant than  $U_k^{test}$ . Further, they show that both the train and test curvatures along the line  $\Theta(r)$  decrease with increasing SGD temperature, as does shift-curvature. Interestingly, although our theory makes no predictions about the shift magnitude  $\|s_k\|$ , in our experiments it increases with temperature for VGG16, but decreases with temperature for Resnet10—but in both cases, the *shift-curvature* still decreases. Also, the test curvature is larger than the train curvature for VGG16 but the opposite is true for Resnet10.

### 3.1. A simple two-basin example

To illustrate the impact of shift and curvature on test loss and the temperature-controlled depth/curvature trade-off, we construct a simplified synthetic optimization problem where  $\theta \in \mathbb{R}$ , and the train and test losses each have two local minima (the data is implicit because we directly define the train and test losses as functions of  $\theta$ ). We construct the train and test losses to be identical except for a constant shift (i.e.  $U^{test}(\theta) = U^{tr}(\theta + s)$ ). We let the noise  $D^{tr}$  be a constant as well, so the bias is zero. Thus (5) becomes  $\rho(\theta) \propto e^{-\frac{\lambda}{2} U^{tr}(\theta)}$ , the basin weights become  $w_k \propto e^{-\frac{\lambda}{2} U_k^{tr}} (C_k^{tr})^{-\frac{1}{2}}$  and (7) becomes:

$$E_{\theta \sim \rho}[U^{test}(\theta)] \approx \sum_k w_k \left( U_k^{tr} + \frac{1}{2} s^T C_k^{tr} s \right), \tag{8}$$

since  $U_k^{tr} = U_k^{test}$  and  $C_k^{tr} = C_k^{test}$ . In figure 4, the losses each have two local minima with different curvature (hence different shift-curvature, since the shift is constant). On the left, the minimum at  $\theta = -1$  (let us assign it  $k = 0$ ) is deeper but narrower than the minimum at  $\theta = 1$  ( $k = 1$ ), so that  $U_0^{tr} < U_1^{tr}$  and  $C_0^{tr} > C_1^{tr}$ . At  $T = 0$ ,  $\rho$  is a Dirac delta at  $\text{argmin}(U^{tr})$  and  $w_0 = 1$ . As  $T$  increases,  $w_1$  increases, and we get a mixture

between the two basins. As  $w_1$  increases, the depth term worsens ( $w_0 U_0^{tr} + w_1 U_1^{tr}$  increases), while the shift-curvature terms improves ( $\frac{1}{2}s^T(w_0 C_0^{tr} + w_1 C_1^{tr})s$  decreases). For larger temperatures, the variance and/or bias terms in the loss become large enough to worsen test performance again, so there is a non-zero and finite optimal temperature that minimizes test performance. On the right, we have a similar setup, except that the deeper basin is also wider. In this case, we see no improvement in test loss with increasing temperature, since there is no trade-off to explore between depth and shift-curvature. Appendix G.3 includes similar one-basin and three-basin cases. In the one-basin case, increasing the temperature can only worsen the expected test loss, since there is no trade-off to explore, and increasing the SGD variance can only hurt both the train and test losses on average. The three basin case is qualitatively similar to the two basin case, but also shows the preference for lower curvature when the two basins are equally deep.

#### 4. Stochastic gradient descent

In this section we derive equations (5)–(7). Recall that SGD attempts to minimize the training loss  $U^{tr}(\theta)$  through a discrete-time stochastic process with state  $\theta_t \in \mathbb{R}^n$ , where  $t$  indexes the number of SGD updates:

$$\theta_{t+1} = \theta_t + \Delta_t, \text{ where } \Delta_t = -\frac{\lambda}{B} \sum_{i=1}^B \partial_{\theta} U(x_i, \theta), \quad (9)$$

where  $\lambda$  is the learning rate,  $B$  is the minibatch size, and the  $x_i$  are i.i.d. samples from the train set that comprise the minibatch. Letting  $T = \lambda/B$ , we note that  $\Delta_t$  has mean  $-\lambda \partial_{\theta} U^{tr}(\theta)$  and covariance  $\lambda T D^{tr}(\theta)$  — we refer to the latter as the SGD noise, and think of the temperature (by analogy with statistical mechanics) as a good summary of noise strength. Typical applications use  $\lambda \ll 1$ , and  $B > 1$ , so that  $T \ll 1$ , since running SGD with too high a learning rate or temperature results in unstable dynamics (especially early in a run) that often fail to converge, i.e. see Goyal *et al* (2017). So we generally assume that  $\lambda \ll 1$ , and  $T \ll 1$ . As shown by Li *et al* (2017b) and our complementary derivation in appendix A, under these conditions the probability distribution of parameters  $p(\theta, t)$  that SGD produces can be well approximated by the probability distribution  $\rho(\theta, t)$  of a continuous-time diffusion process governed by the Fokker–Planck equation:

$$\partial_t \rho(\theta, t) = \lambda \partial_{\theta} \cdot J(\theta, t), \text{ where } J(\theta, t) = \partial_{\theta} U^{tr}(\theta) \rho(\theta, t) + \frac{T}{2} \partial_{\theta} \cdot (D^{tr}(\theta) \rho(\theta, t)). \quad (10)$$

$J(\theta, t)$  is called the probability current. Redefining training time through the change of coordinates  $t = \lambda t$  has the only effect of removing  $\lambda$  from (10), i.e. yielding  $\partial_t \rho(\theta, t) = \partial_{\theta} \cdot J(\theta, t)$ , and leaving  $T$  as the only equation parameter. (Explicitly, rescaling the temporal axis of SGD runs so that  $\lambda_1 n_1 = \lambda_2 n_2$ , where  $\lambda_1, \lambda_2$  are the learning rates and  $n_1, n_2$  are the number of SGD updates for two different runs, should produce similar results as long as the diffusion approximation holds; this can be a useful rule of thumb for adjusting the number of SGD updates as the learning rate is changed.) The fact that  $T$  is the only remaining parameter after rescaling time justifies the notion that  $T$  is a good summary of SGD noise strength, and means that different SGD runs with the same value of  $T$  but different learning rates and minibatch sizes still produce approximately the same steady-state parameter distributions. The dependence of the SGD distribution on  $T$  is in agreement with theory given in Jastrzebski *et al* (2017), and the empirical results of Goyal *et al* (2017), He *et al* (2019a). We also confirm the  $T$  dependence in figure 8 (appendix G.2), which shows similar test performance for different minibatch sizes and learning rates as long as  $T$  is kept fixed. Finally, Goyal *et al* (2017) notes that the simple dependence of SGD on  $T$  breaks down when the learning rate is sufficiently large even when  $T$  is small; a situation where we expect the diffusion approximation to be inaccurate.

With the change of variables  $t = \lambda t$  just described,  $\partial_t \rho(\theta, t) = \partial_{\theta} \cdot J(\theta, t)$  is equivalent to the Langevin (stochastic differential) equation:

$$d\theta = -\partial_{\theta} U^{tr}(\theta) dt + \sqrt{TD^{tr}(\theta)} dW, \quad (11)$$

that appears in Li *et al* (2017b) and Chaudhari and Soatto (2018) as an approximation of SGD. Fokker–Planck and Langevin equations describe the same underlying continuous-time diffusion process; i.e. see Gardiner (2009). Li *et al* (2017b) showed that (10) is an order-1 weak approximation of SGD. In appendix A.1 we offer a complementary informal constructive derivation that shows how to approximate any discrete-time Markov process by a continuous-time diffusion process through the truncation of the infinite Kramers–Moyal expansion, which may be useful for analyzing other algorithms as well. Our approach essentially relies on approximating each SGD step as a sum of i.i.d. infinitesimal increments, and validates a conjecture in Bazant (2005) that the moments of the continuous-time diffusion approximation are

proportional to the cumulants of the discrete-time process<sup>6</sup>. For the rest of this paper, we assume that (10) is a good approximation to SGD, and study the resulting stationary distribution. Next, we assume that the process described by (10) is ergodic, and seek its unique steady-state solution, i.e. the distribution  $\rho(\theta)$  such that  $\partial_t \rho(\theta, t) = 0$ . We can follow Gardiner (2009) and seek a solution where  $J(\theta, t) = 0$ . Appendix B.1 shows that some algebra starting from  $J(\theta, t) = 0$ , assuming that the integrand of the line integral in (5) is a gradient, yields the steady-state distribution in (5). This approach needs an invertible  $D^{tr}(\theta)$ , which is only possible in the underparametrized case. Section 5 describes what happens in the overparametrized case.

To approximate the expected SGD test loss (7), we approximate the SGD steady-state distribution as a mixture of Gaussians (similar to Jastrzkebski et al 2017). Since (5) has the form  $\rho(\theta) \propto e^{-\frac{2}{T}v(\theta)}$ , it is reasonable to expect that when  $T \ll 1$ ,  $\rho$  is multimodal with peaks at the local minima of the effective potential  $v(\theta)$ . So we make a Laplace approximation, i.e. we approximate  $v(\theta)$  by a quadratic function in a neighborhood of any local minimum to obtain a local Gaussian approximation. Combining the local approximation at every local minimum yields a weighted Gaussian mixture approximation of  $\rho(\theta)$ , with weights given by (6), and bias and variance both  $O(T)$ . Details are in appendix B.2. (Note that the bias is linear in  $T$  but nonzero, confirming a result in Chaudhari and Soatto (2018) that the critical points of  $v$  differ linearly-in- $T$  from those of  $U^{tr}$ .) Substituting these results into the approximate test loss (4) and keeping only leading terms in  $T$  results in the approximate SGD test performance in (7). The validity of the Gaussian mixture approximation of (5) depends on the local quadratic approximation of  $v$ , which we explore in our experiments with quadratic fits to the train loss, to which  $v$  is related.

## 5. More parameters than datapoints

We now consider the overparametrized situation ( $p \geq N_{tr}$  and/or  $p \geq N_{test}$ , recalling that  $p$  denotes the number of model parameters and  $N_{tr/test}$  the number train/test model parameters). As discussed in appendix D, the overparametrized case is more complex because the per-sample gradients of the train set now do not span  $\mathbb{R}^p$ , but rather a subspace of dimension no larger than  $N_{tr}$ , denoted  $\mathcal{G}_{tr}$ , which in general depends on  $\theta$  (similarly, there is a subspace  $\mathcal{G}_{test}$  of dimension  $N_{test}$  for test). This has many implications. One is that the losses no longer necessarily have strict local minima, which means we need to generalize the definition of the shift and the Taylor approximation (3). Another is that the average training gradient  $\partial_\theta U^{tr}(\theta)$  and the range of  $D^{tr}(\theta)$  both lie in  $\mathcal{G}_{tr}$ , which implies both that the SGD updates  $\Delta_t$  in (9) lie in  $\mathcal{G}_{tr}$ , and that  $D^{tr}(\theta)$  is no longer invertible, invalidating (5). To help resolve these issues, we will need to add  $\ell_2$ -regularization to the train loss (as is common in practice), and also to modify SGD to include isotropic noise. With these changes, we can obtain analogs of our main results.

In appendix D.4, we generalize (3) to the overparametrized case. We first add  $\ell_2$ -regularization to the train loss:  $U^{tr}(\theta) = E_{x \sim tr}[U(x, \theta)] + \frac{1}{2}\alpha\theta^T\theta$ , which makes the train local minima strict (so that the bias  $\hat{b}$  in (3) makes sense); however, the test loss is unregularized so its local minima are non-strict. We find that (3) still holds if we generalize the definition of the shift to:

$$s_k = \text{Proj}_{\mathcal{G}_{test}}(\theta_k^{test} - \theta_k^{tr}), \quad (12)$$

where  $\text{Proj}_{\mathcal{G}_{test}}$  is the projector onto the subspace spanned by the per-test-sample gradients.

In order to generalize equation (5), we modify SGD to include isotropic noise in addition to the  $\ell_2$ -regularization of the train loss, as follows:

$$\theta_{t+1} = \theta_t + \Delta_t, \text{ where } \Delta_t = -\frac{\lambda}{B} \sum_{i=1}^B \partial_\theta U(x_i, \theta) - \alpha\theta + \sqrt{\lambda T} \beta w_t, \quad (13)$$

where  $\alpha, \beta$  are scalars, and where  $w_t$  is zero-mean Gaussian noise in  $\mathbb{R}^p$  with covariance equal to the identity, that is independent of everything else. The isotropic Gaussian noise ensures that  $D^{tr}(\theta)$  is invertible, while  $\ell_2$ -regularization controls the part of the drift in  $\mathbb{R}^p$  but not in  $\mathcal{G}_{tr}$ , as discussed in appendix D.3. The diffusion process approximation of the modified SGD is then exactly like the one for the underparametrized case after making the substitutions  $D^{tr}(\theta) + \beta^2 I$  in place of  $D^{tr}(\theta)$ , and  $\partial_\theta U^{tr}(\theta) + \alpha\theta$  in place of  $\partial_\theta U^{tr}(\theta)$ . With these substitutions, the steady-state distribution still has the form in equation (5), and its Gaussian mixture approximation still has the form in equation (7).

<sup>6</sup> Informally, in our derivation we need the probability distribution of  $\Delta_t$  to vary slowly with  $\theta$ , and making  $\lambda$  sufficiently small guarantees this, to ensure that infinitesimal increments comprising an SGD step are approximately i.i.d.. We also need  $T \ll 1$  to truncate the Kramers–Moyal expansion to second order and obtain (10). The same conditions also appear in Li et al (2017b):  $T \ll 1$  follows from the bound of the accuracy of the order-1 weak approximation, which is a constant times the learning rate with batch size assumed constant. That the probability of the updates varies slowly with  $\theta$  appears formally there as a Lipschitz and growth condition on the loss.

## 6. Other related work

There are many works aimed at understanding generalization in the context of SGD, and the interplay between SGD noise, curvature, and test performance. In addition to the most relevant works already discussed: Ahn *et al* (2012), Mandt *et al* (2017), Smith and Le (2017), He *et al* (2019a) take a Bayesian perspective, He *et al* (2019b) study asymmetrical valleys, Belkin *et al* (2019) focus on model capacity, Martin and Mahoney (2018) apply random matrix theory, Corneanu *et al* (2020) propose persistent topology measures, Russo and Zou (2016), Xu and Raginsky (2017) provide information-theoretic bounds, Smith *et al* (2021) perform error analysis relative to gradient flow, Lee *et al* (2017), Khan *et al* (2019) connect to Gaussian processes, Wu *et al* (2019) analyze multiplicative noise, and Zhu *et al* (2018) study ‘escaping efficiency’. He *et al* (2019b) show that SGD is biased toward the flatter side of asymmetrical loss valleys, and such a bias can improve generalization (offering additional insight into the bias term in (3) which we identify but do not focus on). Although Mandt *et al* (2017), Smith and Le (2017), He *et al* (2019a) make important connections between generalization and curvature, none provide an explicit expression for test loss in terms of curvature. Also, their analyses are focused around a single local minimum, rather than comparing different local minima as we do in this paper. Mandt *et al* (2017) show that SGD can perform approximate Bayesian inference by studying a diffusion approximation of SGD and its steady-state, but assuming a single, quadratic minimum, and constant isotropic noise; assumptions we relax in this paper. He *et al* (2019a) derive a PAC-Bayes bound on the generalization gap at a given local minimum, which shows that wider minima have a lower upper-bound on risk; in contrast, we make a direct approximation showing that wider minima have lower test loss. They also show that the bound on the risk can increase as  $\log(1/T)$  for a particular local minimum, provided the model is large relative to the  $T$  and curvature; this is quite different from our analysis, which involves analyzing the role of  $T$  in modulating the SGD steady-state distribution’s preference for different local minima (and with no assumptions on model size). Smith and Le (2017) connect Bayesian evidence to curvature at a single local minimum, and empirically suggest evidence as proxy for test loss (by contrast we connect test loss directly to curvature). They also show that the steady state of a Langevin equation with constant isotropic noise is proportional to evidence, and by analogy surmise that SGD noise drives the steady-state toward wider minima; we agree with their intuition, but we allow for non-constant non-isotropic noise, and use a Gaussian mixture approximation to directly show the preference for wider minima. Finally, there is a long history of using statistical mechanics techniques to study machine learning, as we do in this work. For example, Seung *et al* (1992) prove among many other results that the generalization gap is positive; however, like many works both before and after, they assume constant and isotropic noise. Our work motivates updating these earlier studies to the algorithms currently used in practice, e.g. by using the parameter distribution in (5).

## 7. Conclusion and discussion

This paper contributes to longstanding debates about whether curvature can predict generalization, and how and whether SGD minimizes curvature. First, we show that curvature harms test performance through three mechanisms: the shift-curvature, bias-curvature, and the covariance-curvature; we believe the first two to be new. We address the concern of Dinh *et al* (2017) by showing that all three are reparametrization-invariant although curvature is not. Our main focus is the shift-curvature, or curvature along the line connecting train and test local minima; although the shift is unknown at training time, the shift-curvature still shows that *any* directions with high curvature can potentially harm test performance if they happen to align with the shift. Second, we derive a new and explicit SGD steady-state distribution showing that SGD optimizes an effective potential related to but different from train loss, and show that SGD noise mediates a trade-off between the depth and curvature of this effective potential. Our steady-state solution removes assumptions in earlier works of constant, isotropic gradient covariance, and treats the overparametrized case with care. Third, we combine our test performance analysis with the SGD steady-state to show that for small SGD noise, shift-curvature may be the dominant mechanism. Our experiments validate our approximations of test loss, show that that shift-curvature is indeed a major contributor to test performance, and show that SGD with higher noise chooses local minima with lower curvature and lower shift-curvature.

However, many avenues for future work still remain. First of all, we have made substantial, but not complete, progress in the overparametrized case. One of the main challenges is directions of near-zero curvature (a concern raised by Draxler *et al* 2018, Sagun *et al* 2018 and others). On the positive side, overparametrization poses no problem for our test performance approximation (3) or shift-curvature concept (since zero-curvature directions simply have no effect on either local test performance or

shift-curvature). Furthermore, our SGD results eliminate constant, isotropic assumptions that appear in earlier works, clarify the challenges in the overparametrized case (such as the rank-deficient gradient covariance with varying nullspace), and still obtain the steady-state solution (5) of a slightly modified problem. However, the Gaussian mixture approximation (6), which highlights SGD's low-curvature preference, is less clear in the presence of near-zero-curvature directions. Therefore, further work is needed to clarify how near-zero-curvature directions may affect the unmodified SGD steady-state as well as its preference for low-curvature regions. (A possibly-useful intuition is that, locally, SGD should force all near-zero-curvature directions to zero, making them essentially irrelevant; however, there are difficulties in making this precise, as discussed in appendix D.) Another different limitation of our work is the assumption that SGD reaches steady state (which enables us to obtain and study an explicit parameter distribution), which may be too slow to accomplish and difficult to verify in practice. We also assume a constant learning rate, although practitioners often use learning-rate schedules. To better understand these situations, further work is needed to study the dynamics of SGD and its impact on generalization, possibly using the diffusion approximation (10). Other popular learning algorithms, like ADAM and SGD with momentum, could be studied using the techniques presented here. The diffusion approximation in (10) might yield additional insights into SGD. For example, the so-called fluctuation-dissipation relations derived in Yaida (2018) from the exact SGD process in (9) follow more directly from (10) by considering the time derivative of the average of any function of the parameters, and applying this to the parameters themselves, the parameter fluctuations, and the train loss (although the relation for train loss is quadratic in  $T$ , and deriving it via (10) only recovers the constant and linear in  $T$  terms). Studying the time derivative of other functions of the parameters may yield additional relations between key quantities. Third, our setup could be a helpful starting point for investigating model-wise double descent (Belkin *et al* 2019), by studying how the stationary distribution (5) and test performance change as model complexity increases (perhaps asymptotically via random matrix theory). Finally, our results suggest that further work on models and algorithms that promote low-curvature solutions, such as the algorithm recently proposed in Orvieto *et al* (2022), may have great practical value.

### Data availability statement

The data generated and/or analyzed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

### Acknowledgments

We would like to thank Luca Zappella and Santiago Akle for inspiring us to study this problem; Vimal Thilak for assistance with compute infrastructure, Josh Susskind, Moises Goldszmidt, and Omid Saremi for helpful discussion and suggestions on the draft; Rudolph van der Merwe and John Giannandrea for their support; and our PhD advisors Wing H Wong and George C Verghese for first introducing us to the techniques behind this work.

## Appendix A. Continuous-time approximation of SGD

### A.1. Continuous-time approximation of a discrete-time continuous-space Markov process

Our goal in this section is to construct a continuous-time Markov process that approximates a discrete-time Markov process. Later we will specialize to SGD, but for now we consider any discrete-time stochastic process with state  $\theta_j \in \mathbb{R}^p$  (where  $j$  indexes time), that evolves according to:

$$\theta_{j+1} = \theta_j + \Delta_j,$$

where  $\Delta_j \in \mathbb{R}^p$  is an arbitrary function of  $\theta_j$ . The SGD update is a special case when  $\Delta$  is given by (9). (Note the different notation from the main text:  $j$  rather than  $t$  is the discrete time index here, so that we can use  $t$  to represent continuous time.) We wish to construct a continuous-time Markov process  $\tilde{\theta}(t) \in \mathbb{R}^p$ , with probability distribution  $\rho(\theta, t) = P(\tilde{\theta}(t) = \theta)$ , that approximates  $\theta_j$  in the following sense. We assume the discrete-time updates of  $\theta_j$  occur every  $\tau > 0$  continuous-time units of  $\tilde{\theta}$ , where  $\tau$  is arbitrary. We want  $\tilde{\theta}(t)$  such that if  $\rho(\theta, (j-1)\tau) = P(\theta_{j-1} = \theta)$  then

$$\rho(\theta, j\tau) \equiv P(\tilde{\theta}(j\tau) = \theta) \approx P(\theta_j = \theta). \quad (14)$$

We will construct  $\tilde{\theta}$  via its Kramers–Moyal expansion, which is a differential equation that describes the evolution of the probability distribution of  $\tilde{\theta}$ . Truncating the Kramers–Moyal expansion to second order yields a Fokker–Planck approximation of the discrete-time stochastic process (which we will later specialize to the case of SGD). The Kramers–Moyal equation depends on the moments of the infinitesimal increments of the continuous process (i.e.  $\tilde{\theta}(t') - \tilde{\theta}(t)$ , for  $t' - t$  infinitesimally small), so our basic strategy is to construct these moments in such a way that the continuous-time update over time interval  $\tau$  matches the discrete-time update (we will find that defining the continuous moments proportional to the discrete cumulants achieves this).

We state the final results here, and justify them in the next sections. First we need to introduce some notation. We adopt multi-index notation to streamline the exposition, so for example if  $x \in \mathbb{R}^p$  then  $x^\gamma \equiv x_0^{\gamma_0} \dots x_p^{\gamma_p}$ , where  $\gamma \in \mathbb{N}^p$ . Dropping the time index  $j$  from the discrete-time update  $\Delta$  for now, we denote the moments and cumulants of  $\Delta$  (which play an important role) by  $m_\gamma(\theta) = E[\Delta^\gamma]$  and  $\kappa_\gamma(\theta) = \log m_\gamma(\theta)$ , respectively, with  $\gamma \in \mathbb{N}^p$  (using multi-index notation for  $\Delta^\gamma$ ). Next, we define the increment of the continuous-time process  $\tilde{\theta}$  for any two times  $t' \geq t$  as the random variable  $\tilde{\Delta}(dt) \equiv \tilde{\Delta}(t' - t) = \tilde{\theta}(t') - \tilde{\theta}(t)$ . (Note that the Markov property implies that the increment only depends on the time difference  $dt = t' - t$  when the value of  $\tilde{\theta}(t)$  is known.)

With this notation, we can present our main results. In order to specify the process  $\tilde{\theta}$  via the Kramers–Moyal expansion, we actually only need to define the probability distribution of  $\tilde{\Delta}(dt)$  when  $t' - t = dt$  is infinitesimally small. We will show that we can (approximately) match the continuous-time update over time interval  $\tau$  to the discrete-time update by defining the moments of  $\tilde{\Delta}(dt)$  as:

$$E[\tilde{\Delta}(dt)^\gamma] = \frac{\kappa_\gamma(\theta(t))}{\tau} dt, \tag{15}$$

with  $\gamma \in \mathbb{N}_0^p$  (where  $\mathbb{N}_0^p$  denotes  $p$ -dimensional non-negative integers). That is, the moments of the small-time increments of  $\tilde{\theta}(t)$  are directly proportional to the cumulants of the discrete time process  $\theta_j$  it seeks to approximate. (We will find that resulting process will approximate the discrete one well when  $\kappa_\gamma(\theta(t))$  does not change much in value within each  $\tau$  time increment, so that the small time increments during that time period are approximately independent and identically distributed.)

Then we will show that (setting  $\tau = 1$  for simplicity now) the following Kramers–Moyal (KM) expansion describes the evolution of the distribution of  $\tilde{\theta}$ :

$$\partial_t \rho(\theta, t) = \sum_{\gamma \in \mathbb{N}_1^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \{ \kappa_\gamma(\theta) \rho(\theta, t) \}, \tag{16}$$

where  $\partial_\theta^\gamma f(\theta) = \partial_{\theta_0}^{\gamma_0} \dots \partial_{\theta_{p-1}}^{\gamma_{p-1}} f(\theta)$  for an arbitrary function  $f(\theta)$  (as is standard in multi-index notation), and  $\mathbb{N}_1^p$  denotes the  $p$ -dimensional positive integers. The KM expansion can be truncated to second order to obtain a Fokker–Planck approximation of the discrete-time stochastic process.

*A.1.1. Moments of the continuous time process in terms of cumulants of the discrete time process*

Our goal in this section is to justify equation (15). Let  $dt = \tau/K$  for some positive integer  $K$ , and define  $t_j = jdt$  for  $j = 0, \dots, K$ , and  $dt_j = t_{j+1} - t_j = dt$ . Recall that  $\tilde{\Delta}(dt_j)$  is the small time increment of our continuous process between  $t_{j-1}$  and  $t_j$ , so that  $\tilde{\Delta}(\tau, K) = \sum_{j=1}^K \tilde{\Delta}(dt_j)$  is the total change in our process over  $\tau$  time units when breaking the temporal interval into  $K$  equal-sized increments. We want to determine the moments of  $\tilde{\Delta}(dt_j)$  so that  $E[\tilde{\Delta}(\tau)^\gamma] = \lim_{K \rightarrow \infty} E[\tilde{\Delta}(\tau, K)^\gamma]$  matches the corresponding moment of the discrete SGD step  $E[\Delta(\theta)]$ . Assuming the increments  $\tilde{\Delta}(dt_j)$  are approximately i.i.d. within each time interval of length  $\tau$ , we can approximate  $\tilde{\Delta}(\tau, K)$  as a sum of i.i.d. random variables. With simplified notation, we model the problem as follows. Consider a sum  $S_K = \sum_{i=1}^K X_i$ , where  $X_i$  are i.i.d. random variables (each in  $\mathbb{R}^p$ ). Suppose that we know the ‘desired’ limiting random variable  $S$ , along with its cumulants. We want to find the moments of the i.i.d. random variables  $X$  so that  $\lim_{K \rightarrow \infty} S_K = S$ . We will use here the following definitions. Let  $m_\gamma^X$  and  $\kappa_\gamma^X$  denote the moments and cumulants of an arbitrary random variable  $X$ . Let  $M_X(t) = E[e^{t^T X}]$  be the moment generating function of  $X$ , and  $C_X(t) = \log M_X(t)$  the cumulant generating function, so that  $m_\gamma^X = \partial^\gamma M_X(t)|_{t=0}$  and  $\kappa_\gamma^X = \partial^\gamma C_X(t)|_{t=0}$ . We will also use the identity:

$$e^x = \lim_{K \rightarrow \infty} \left( 1 + \frac{x}{K} \right)^K. \tag{17}$$

To restate the problem more precisely in this notation, we want to find  $m_\gamma^X$  such that  $\lim_{K \rightarrow \infty} C_{S_K}(t) = C_S(t)$ . Using the i.i.d. assumption and identity 17, we obtain:

$$M_{S_K}(t) = E[e^{tS_K}] = \prod_{i=1}^K E[e^{tX_i}] = (M_X)^K, \text{ so}$$

$$\lim_{K \rightarrow \infty} M_{S_K}(t) = \lim_{K \rightarrow \infty} (M_X)^K = \lim_{K \rightarrow \infty} (1 + \xi/K)^K, \text{ where } \xi \text{ satisfies } M_X \equiv 1 + \xi/K$$

$$= e^\xi,$$

so  $\lim_{K \rightarrow \infty} C_{S_K}(t) = \xi$ , and since we want  $\lim_{K \rightarrow \infty} C_{S_K}(t) = C_S(t)$ , we equate  $\xi = C_S(t)$  to find that  $M_X(t) = 1 + C_S(t)/K$ , so:

$$m_\gamma^X = \partial^\gamma M_X(t)|_{t=0} = \frac{1}{K} \partial^\gamma C_S(t)|_{t=0} = \frac{1}{K} \kappa_\gamma^S.$$

In order to achieve the desired limit, we then need the moments of the i.i.d. variables  $X$  to be equal to  $1/K$  times the cumulants of the desired limiting distribution  $S$ . To explicitly connect this result back to the continuous approximation of SGD, we associate  $X_i = \tilde{\Delta}(dt_i)$ ,  $S_j = \tilde{\Delta}(\tau, K)$ ,  $S = \Delta(\theta)$ , and  $dt = \tau/K$ . Then  $m_\gamma^X = \frac{1}{K} \kappa_\gamma^S$  translates to equation (15), as we desired. Note that the i.i.d. assumption on the increments  $\tilde{\Delta}(dt_j)$  is an approximation that introduces error into (15). In the approximation, we assume that  $P(\tilde{\Delta}(dt_j)) \approx P(\tilde{\Delta}(dt_0))$  for  $t_j = jdt, j = 1, \dots, K, dt = \tau/K$ . If  $\tau$  is small and  $\tilde{\Delta}$  varies slowly, then this condition will approximately hold.

A.1.2. The continuous Kramers–Moyal expansion

Our goal in this section is to obtain the Kramers–Moyal expansion (16). The approach is essentially to Taylor-expand the Chapman–Kolmogorov equation, take the limit as  $dt \rightarrow 0$ , and use result (15) for the moments of the infinitesimal increments. The Markov assumption for  $\tilde{\theta}(t)$  implies the continuous-time Chapman Kolmogorov equation:

$$\rho(\theta, t') = \int \rho(\theta - \Delta, t) W(\Delta, t' | \theta - \Delta, t) d\Delta, \tag{18}$$

where  $W(\Delta, t' | \theta, t) = P(\tilde{\theta}(t') = \theta + \Delta | \tilde{\theta}(t) = \theta)$  is the transition probability function, and  $t' \geq t$ . Substituting  $t' = t + dt$ , and using our definition of  $\tilde{\Delta}$ , we obtain:

$$\rho(\theta, t + dt) = \int \rho(\theta - \Delta, t) W(\Delta, t + dt | \theta - \Delta, t) d\Delta,$$

where  $W(\Delta, t + dt | \theta, t) = P(\tilde{\Delta}(dt) = \Delta | \tilde{\theta}(t) = \theta)$ . (19)

Next we Taylor-expand the left-hand-side of (19) in  $\Delta$  (recalling that in multi-index notation, the infinite Taylor expansion of  $f(\theta)$  is  $f(\theta + h) = \sum_{\gamma \in \mathbb{N}_0^p} \frac{\partial_\theta^\gamma f(\theta)}{\gamma!} h^\gamma$ ):

$$\begin{aligned} \rho(\theta, t + dt) &= \int \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-\Delta)^\gamma}{\gamma!} \partial_\theta^\gamma (\rho(\theta, t) W(\Delta, t + dt | \theta, t)) d\Delta \\ &= \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \left( \rho(\theta, t) \int \Delta^\gamma W(\Delta, t + dt | \theta, t) d\Delta \right) \\ &= \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \left( \rho(\theta, t) E[\tilde{\Delta}(dt)^\gamma] \right) \\ &= \rho(\theta, t) + \sum_{\gamma \in \mathbb{N}_1^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \left( \rho(\theta, t) E[\tilde{\Delta}(dt)^\gamma] \right) \end{aligned}$$

Finally we take the limit as  $dt \rightarrow 0$  and use (15):

$$\begin{aligned} \implies \partial_t \rho(\theta, t) &= \lim_{dt \rightarrow 0} \frac{\rho(\theta + \Delta, t + dt) - \rho(\theta + \Delta, t)}{dt} \\ &= \sum_{\gamma \in \mathbb{N}_1^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \left( \rho(\theta, t) \lim_{dt \rightarrow 0} \frac{E[\tilde{\Delta}(dt)^\gamma]}{dt} \right) \\ &= \sum_{\gamma \in \mathbb{N}_1^p} \frac{(-1)^\gamma}{\tau \gamma!} \partial_\theta^\gamma \{ \kappa_\gamma(\theta) \rho(\theta, t) \}, \end{aligned}$$

The result above is equation (16).

A.1.3. The continuous process approximates the discrete process

Now that we have defined  $\tilde{\theta}$  via its Kramers–Moyal expansion (16), we need to check that it approximates SGD in the sense of equation (14) (we need to show that  $\rho(\theta, j\tau) \approx P(\theta_j = \theta)$ ). We seek to understand the approximation error in terms of the difference between  $\tilde{m}_\gamma(\theta)$  and  $m_\gamma(\theta)$ . The error arises because  $\tilde{m}_\gamma(\theta)$  relies on the approximation that small-time increments of  $\tilde{\theta}(t)$  are i.i.d.. We assume that  $\rho(\theta, t) = P(\theta_j = \theta)$ , and then study  $\rho(\theta, t + \tau)$ , starting from the Chapman–Kolmogorov for  $\tilde{\theta}(t)$  (18):

$$\begin{aligned} \rho(\theta, t + \tau) &= \int \rho(\theta - \Delta, t) W(\Delta, t + \tau | \theta - \Delta, t) d\Delta \\ &= \int P(\theta_j = \theta - \Delta) \left( P(\Delta_j = \Delta | \theta_j = \theta - \Delta) \right. \\ &\quad \left. + \{ W(\Delta, t + \tau | \theta - \Delta, t) - P(\Delta_j = \Delta | \theta_j = \theta - \Delta) \} \right) d\Delta \\ &= P(\theta_{j+1} = \theta) + \int P(\theta_j = \theta - \Delta) \{ W(\Delta, t + \tau | \theta - \Delta, t) - P(\Delta_j = \Delta | \theta_j = \theta - \Delta) \} d\Delta \\ &= P(\theta_{j+1} = \theta) + \int \rho(\theta - \Delta, t) W(\Delta, t + \tau | \theta - \Delta, t) d\Delta \\ &\quad - \int P(\theta_j = \theta - \Delta) P(\Delta_j = \Delta | \theta_j = \theta - \Delta) d\Delta. \end{aligned} \tag{20}$$

The two integrals above can be Taylor-expanded in the same way as in our derivation of the Kramers–Moyal expansion to yield:

$$\begin{aligned} e(\theta) &= \rho(\theta, t + \tau) - P(\theta_{j+1} = \theta) = \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \left\{ \rho(\theta, t) E[\tilde{\Delta}(\tau)^\gamma] - P(\theta_j = \theta) E[\Delta_j^\gamma] \right\} \\ &= \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \{ \rho(\theta, t) (\tilde{m}_\gamma(\theta) - m_\gamma(\theta)) \} = \sum_{\gamma \in \mathbb{N}_0^p} \frac{(-1)^\gamma}{\gamma!} \partial_\theta^\gamma \{ \rho(\theta, t) e_\gamma(\theta) \}, \end{aligned} \tag{21}$$

where we define the error in the  $\gamma$ -th moment to be  $e_\gamma(\theta) = \tilde{m}_\gamma(\theta) - m_\gamma(\theta)$ . So, i.e.

$$\int |e(\theta)| d\theta \leq \sum_{\gamma \in \mathbb{N}_1^p} \frac{1}{\gamma!} |\partial_\theta^\gamma \{ \rho(\theta, t) e_\gamma(\theta) \}|. \tag{22}$$

Recall that the errors  $e_\gamma(\theta)$  are due to the i.i.d. assumption on the infinitesimal increments in the continuous approximation; that is, the assumption that the distribution of  $\tilde{\Delta}$  is approximately constant in an interval of length  $\tau$ . Therefore, if  $\tilde{\Delta}$  varies slowly relatively to the timescale  $\tau$ , the approximation will be close.

A.2. Continuous-time diffusion approximation of SGD

We now choose  $\Delta$  to be as in equation (9) to study SGD. Our goal here is to approximate SGD with a continuous-time process. Because cumulants of independent random variables are additive, and letting  $\omega_\gamma(\theta)$  be the  $\gamma$ th cumulant of  $\partial_\theta U(x_i, \theta)$  (i.e. the gradient evaluated at a single sample of  $x$ ), we find that the cumulant for the minibatch of size  $B$  is:

$$\kappa_\omega(\theta) = (-T)^{|\gamma|} B \omega_\gamma(\theta).$$

So the KM expansion (16) of the continuous-time SGD approximation then becomes:

$$\partial_t \rho(\theta, t) = B \sum_{\gamma \in \mathbb{N}_1^p} \frac{T^{|\gamma|}}{\gamma!} \partial_\theta^\gamma (\omega_\gamma(\theta) \rho(\theta, t)). \tag{23}$$

Clearly, then, as  $T$  gets small, fewer terms in the expansion matter. The first two cumulants for SGD are the mean and variance of the gradients of  $U$  over the training distribution, respectively:

$$\omega_1(\theta) = E_{x \sim \pi} [\partial_\theta U(x, \theta)], \quad \omega_2(\theta) = \text{Cov}_{x \sim \pi} [\partial_\theta U(x, \theta)].$$

When  $T$  is small enough, or when the cumulants  $\omega_\gamma(\theta)$  are small for  $|\gamma| > 2$  (i.e. when  $\partial_\theta U(x_i, \theta)$  is Gaussian, cumulants higher than 2 are zero), we can approximate the KM expansion of (23) by the

Fokker–Planck (FP) equation that retains only the first two terms in the expansion. Switching to the notation of the main text:

$$\omega_1(\theta) = \partial_\theta U^{\text{tr}}(\theta), \quad \omega_2(\theta) = D^{\text{tr}}(\theta),$$

(so  $D^{\text{tr}}(\theta) = \omega_2(\theta) \in \mathcal{R}^{p \times p}$  is now the empirical covariance matrix of the gradients in the training, or diffusion matrix, and  $U^{\text{tr}}(\theta)$  is the train loss), we obtain:

$$\partial_t \rho(\theta, t) := \lambda \sum_{i=1}^p \partial_{\theta_i} \{ \partial_{\theta_i} U^{\text{tr}}(\theta) \rho(\theta, t) \} + \frac{1}{2} \lambda T \sum_{i,j=1}^p \partial_{\theta_i, \theta_j}^2 \{ D_{ij}^{\text{tr}}(\theta) \rho(\theta, t) \}.$$

Rewriting using the divergence operator  $\partial_\theta \cdot v = \sum_{i=1}^p \partial_{\theta_i} v(\theta)$ , we obtain the Fokker–Planck SGD equation stated in the main text (10).

### A.2.1. When is the SGD diffusion approximation accurate?

The FP approximation of SGD follows from two approximations. First, we need the increments  $\tilde{\Delta}(dt_j)$  within a time interval of  $\tau$  (corresponding to a single SGD update) to be approximately i.i.d. so that the KM expansion (16) is an accurate approximation of the discrete time SGD process. This condition holds when the product of the expected change in  $\theta$  during a single update and of the derivative w.r.t.  $\theta$  of the density of any small increment is small<sup>7</sup>. The second condition required for the FP equation to hold is that terms of order  $|\gamma| > 2$  in the KM expansion must be small (relative to the  $|\gamma| \leq 2$  terms), so that the truncation that yields the FP equation is appropriate. The latter is satisfied when  $T$  is small, and/or the third and higher cumulants of the gradients (i.e.  $\omega_\gamma(\theta)$  for  $|\gamma| > 2$ ) are small. Finally, our steady-state analysis in the next section relies on the distribution having actually reached steady-state, and the number of SGD steps required to reach steady-state scales inversely with temperature, so in practice with very small temperature the steady-state could be difficult to attain. Therefore, practically speaking, we expect the SGD diffusion approximation to start to break down for large  $T$  (i.e. large ratio of LR to batch size), or for fixed small  $T$  at extreme (very small or very large) learning rates or batch sizes. A large learning rate makes the expected change in  $\theta$  in a single update large, potentially violating the i.i.d. assumption of the small time increments, while a large batch size at small fixed temperature also implies a large learning rate, and has the same effect. We similarly expect the mean change in an SGD update to be large at the beginning of an SGD run, and our FP approximation to not be valid during some initial transient period. Lastly, the i.i.d. assumption can be violated when the derivative of the mean and covariance of a single SGD update with respect to  $\theta$  is large.

## Appendix B. Steady state of the SGD diffusion approximation

### B.1. Steady state distribution

Our goal in this section is to show that (5) is the steady-state solution of our SGD diffusion approximation (10), and clarify the assumptions under which this is true. Our solution of the Fokker–Planck equation is identical to one presented in Gardiner (2009) section 6.2. We consider the underparametrized case (section 5 extends to the overparametrized case), so that  $D^{\text{tr}}(\theta)$  is invertible and the SGD drift term spans  $\mathbb{R}^p$  even without regularization. We assume the process is ergodic, and therefore has a single steady-state distribution  $\rho(\theta)$ . To find it, we first set (10) to zero and obtain the steady-state condition

<sup>7</sup>To see this, consider the joint probability of two infinitesimal increments within the same time period of length  $\tau$ , i.e.  $P(\tilde{\Delta}(dt_0) = x_0, \tilde{\Delta}(dt_j) = x_j) = P(\tilde{\Delta}(dt_0) = x_0) P(\tilde{\Delta}(dt_j) = x_j | \tilde{\Delta}(dt_0) = x_0)$ , where  $j > 0$ , and assume  $\theta = \theta_0$  at the beginning of the  $dt_0$  interval. We can write the latter probability as:

$$P(\tilde{\Delta}(dt_j) = x_j | \tilde{\Delta}(dt_0) = x_0) = E_W [P(\tilde{\Delta}(dt_j) = x_j | \theta_{j-1} = \theta_0 + W)], \text{ with } W = x_0 + \tilde{\Delta}(t_1, t_{k-1}),$$

is a random variable describing the increment since the start of the  $\tau$  time interval until the beginning of the  $k$ th infinitesimal increment  $t_{k-1}$ . Letting  $\mu_w$  denote the mean of  $W$ , we can Taylor expand to first order around  $W = 0$  to find that:

$$E_W [P(\tilde{\Delta}(dt_j) = x_j | \theta_{j-1} = \theta_0 + W)] \approx P(\tilde{\Delta}(dt_j) = x_j | \theta_{j-1} = \theta_0) + \mu_w' \partial_{\theta_{j-1}} P(\tilde{\Delta}(dt_j) = x_j | \theta_{j-1} = \theta_0),$$

plus higher order terms. When the second term above is small compared to the first, we can finally write that  $P(\tilde{\Delta}(dt_0) = x_0, \tilde{\Delta}(dt_j) = x_j) = P(\tilde{\Delta}(dt_0) = x_0) P(\tilde{\Delta}(dt_j) = x_j)$ , and since  $|dt_j| = |dt_0| = dt$ , and these increments only depend on these magnitudes and on  $\theta$  at the beginning of each increment (both equal to  $\theta_0$  in our expansion above), then the i.i.d. assumption is satisfied. Conversely, when  $\mu_w$  and/or  $\partial_\theta P(\tilde{\Delta}(dt_j) = x_j | \theta = \theta_0)$  are large we expect the increments not to be i.i.d. and the KM expansion to not approximate the discrete process well.

$\partial_\theta \cdot J(\theta) = 0$ . A distribution  $\rho(\theta)$  where  $J$  is constant, if it can be found, satisfies the above. We also now require that  $\rho(\theta) \rightarrow 0$  and  $\partial_\theta \rho(\theta) \rightarrow 0$  as  $\theta \rightarrow \pm\infty$ , so we attempt to find a solution that satisfies  $J(\theta) = 0$  everywhere, and where  $J(\theta) = \partial_\theta U^{tr}(\theta)\rho(\theta) + \frac{T}{2}\partial_\theta \cdot (D^{tr}(\theta)\rho(\theta))$ . Some algebra then yields the steady-state solution:

$$\frac{\partial_\theta \rho(\theta, t)}{\rho(\theta, t)} = -\frac{2}{T}D^{tr}(\theta)^{-1} \left( \partial_\theta U^{tr}(\theta) + \frac{T}{2}\partial_\theta \cdot D^{tr}(\theta) \right) := -\frac{2}{T}\mathcal{V}(\theta), \text{ so}$$

$$\int^\theta \partial_\theta \log \rho(\xi) \cdot d\xi = -\frac{2}{T} \int^\theta \mathcal{V}(\xi) \cdot d\xi := -\frac{2}{T}v(\theta, T), \text{ which can be re-arranged as}$$

$$\rho(\theta) \propto e^{-\frac{2}{T}v(\theta, T)}, \text{ with, } v(\theta, T) = \int^\theta D^{tr}(\xi)^{-1} \left( \partial_\xi U^{tr}(\xi) + \frac{T}{2}\partial_\xi \cdot D^{tr}(\xi) \right) \cdot d\xi. \quad (24)$$

This solution relies on a line integral to define  $v(\theta, T)$  that is path independent, and therefore needs the following assumption:

**Assumption 1.**  $\mathcal{V}(\theta) = D^{tr}(\theta)^{-1} (\partial_\theta U^{tr}(\theta) + \frac{T}{2}\partial_\theta \cdot D^{tr}(\theta))$  is a gradient; that is the curl of  $\mathcal{V}$  vanishes.

In other words, since  $\mathcal{V}$  is defined as  $-\partial_\theta \log \rho(\theta)$ , the first equation above can only be satisfied if  $\mathcal{V}$  is a gradient; a necessary and sufficient condition for this is the vanishing of the curl or so-called potential conditions  $\partial_{\theta_i} \mathcal{V}_j = \partial_{\theta_j} \mathcal{V}_i$  where  $\mathcal{V}_i$  denotes the  $i$ th entry of  $\mathcal{V}$ . When these conditions hold, the Hessian of  $\log \rho(\theta)$  is symmetric, as it should be. So we assume that  $\partial_\theta U^{tr}(\theta), D^{tr}(\theta)$  and are such that the assumption above holds.

*B.1.1. Examples where assumption 1 holds*

Assumption 1 clearly holds in the case where  $D^{tr}$  is constant and isotropic (i.e.  $D^{tr} = c^{-1}I$  for positive constant  $c$ ), since then  $\mathcal{V}(\theta) = c\partial_\theta U^{tr}(\theta)$ , which is the gradient of  $cU^{tr}$ . But it can also hold for nonconstant, anisotropic  $D^{tr}$  as well, as the following examples confirm.

- (a) Isotropic but non-constant noise: consider  $D^{tr}(\theta) = U^{tr}(\theta)I$ , so that  $\partial_\theta \cdot D^{tr}(\theta) = \partial_\theta U^{tr}(\theta)$ . Then  $\mathcal{V}(\theta) = (1 + \frac{T}{2})\partial_\theta \log U(\theta)$  is a gradient, and we have  $v(\theta, T) = (1 + \frac{T}{2})\log(U(\theta))$ , and  $\rho(\theta) \propto U(\theta)^{-(1+\frac{T}{2})}$ . This argument works more generally: let  $f: \mathbb{R} \rightarrow \mathbb{R}$  be any smooth scalar function, and let  $D^{tr}(\theta) = \frac{1}{f'(U(\theta))}I$  where  $f'(U(\theta)) = \partial_{U(\theta)} f(U(\theta))$  (the first example is a special case with  $f = \log U$ ). Then the chain rule implies that  $(D^{tr}(\theta))^{-1}\partial_\theta U^{tr}(\theta) = \partial_\theta f(U(\theta))$ . Similarly,  $(D^{tr}(\theta))^{-1}(\partial_\theta \cdot D^{tr}(\theta)) = -\partial_\theta \log f'(U(\theta))$ , so  $\mathcal{V} = \partial_\theta f(U(\theta)) - \frac{T}{2}\partial_\theta \log f'(U(\theta))$ ,  $v(\theta, T) = f(U(\theta)) - \frac{T}{2}\log f'(U(\theta))$ , and  $\rho(\theta) \propto e^{-\frac{2}{T}f(U(\theta)) + \log f'(U(\theta))}$ .
- (b) Anisotropic but constant noise: Consider  $D^{tr}(\theta) = \partial_\theta^2 U^{tr}(\theta)^{-1}$  and suppose that  $U^{tr}(\theta)$  is quadratic, so that  $D^{tr}$  is constant and  $\partial_\theta \cdot D^{tr} = 0$ . Then  $\partial_\theta (\frac{1}{2}\partial_\theta U^{tr}(\theta)^T \partial_\theta U^{tr}(\theta)) = \partial_\theta^2 U^{tr}(\theta) \partial_\theta U^{tr}(\theta) = (D^{tr})^{-1} \partial_\theta U^{tr}(\theta) = \mathcal{V}(\theta)$ , which shows that  $\mathcal{V}$  is a gradient. In this case,  $v(\theta) = \frac{1}{2}\partial_\theta U^{tr}(\theta)^T \partial_\theta U^{tr}(\theta)$ .
- (c) Non-constant and anisotropic noise: Let  $U^{tr}(\theta) = \sum_i U_i(\theta_i)$  and suppose that  $D^{tr}(\theta) = \text{diag}\{d_i(\theta_i)\}$ , where  $d_i(\theta_i) = \frac{1}{f'(U_i(\theta_i))}$  for an arbitrary smooth scalar function  $f$ . Note that both  $U_i$  and  $d_i$  are functions only of  $\theta_i$ . Then  $((D^{tr})^{-1}\partial_\theta U^{tr})_i = f'(U_i)\partial_{\theta_i} U_i = \partial_{\theta_i} f(U_i)$ . Also,  $((D^{tr})^{-1}\partial_\theta \cdot D^{tr})_i = \frac{\partial_{\theta_i} d_i}{d_i(\theta_i)} = \partial_{\theta_i} \log d_i(\theta_i) = -\partial_{\theta_i} \log f'(U_i(\theta_i))$ . So  $\mathcal{V}_i = \partial_{\theta_i} f(U_i) - \frac{T}{2}\partial_{\theta_i} \log f'(U_i(\theta_i))$ . Defining  $v(\theta) = \sum_i f(U_i(\theta_i)) - \frac{T}{2}\sum_i \log f'(U_i(\theta_i))$ , we get  $\partial_\theta v(\theta) = \mathcal{V}(\theta)$ . This example results in independent dynamics for each parameter, which is unlikely to hold for realistic models, but it does show that assumption 1 holds for more models with just those with constant isotropic noise. (As an additional example, note that in example 2 where  $D^{tr}(\theta) = \partial_\theta^2 U^{tr}(\theta)^{-1}$ , we obtain the same  $\mathcal{V}$  if we remove quadratic  $U^{tr}$ /constant  $D^{tr}$  assumption but take the limit as  $T \rightarrow 0$  to drop the  $\partial_\theta \cdot D^{tr}$  term.)

Further work is necessary to characterize all the situations for which SGD satisfies assumption 1. Chaudhari and Soatto (2018) state in effect that  $J \neq 0$  whenever  $D^{tr}$  is nonconstant and/or anisotropic, so the examples above are counterexamples since they all satisfy  $J = 0$  (a consequence of assumption 1).

**B.2. Gaussian mixture approximation**

We further approximate the steady-state distribution of SGD by a mixture of Gaussians, i.e. as a distribution of the form  $\rho(\theta) = \sum_k w_k \mathcal{N}(\mu_k, \Sigma_k)$ . So we need to find the means  $\mu_k$ , covariances  $\Sigma_k$ , and weights  $w_k$ . In our derivation, we assume the underparametrized case, where for example the gradient covariance and

Hessian are full rank. However, the arguments are similar in the overparametrized case using the modified SGD (13). From (5), the first two derivatives of  $v$  are:

$$\begin{aligned} \partial_\theta v(\theta, T) &= D^{tr}(\theta)^{-1} \left( \partial_\theta U^{tr}(\theta) + \frac{T}{2} \partial_\theta \cdot D^{tr}(\theta) \right) \\ \partial_\theta^2 v(\theta, T) &= \partial_\theta (D^{tr}(\theta)^{-1}) \left( \partial_\theta U^{tr}(\theta) + \frac{T}{2} \partial_\theta \cdot D^{tr}(\theta) \right) + D^{tr}(\theta)^{-1} \left( \partial_\theta^2 U^{tr}(\theta) + \frac{T}{2} \partial_\theta^T (\partial_\theta \cdot D^{tr}(\theta)) \right). \end{aligned}$$

If  $\mu$  is any local extremum of  $v$ , then:

$$\begin{aligned} 0 &= \partial_\theta v(\mu, T) = D^{tr}(\mu)^{-1} \left( \partial_\theta U^{tr}(\mu) + \frac{T}{2} \partial_\theta \cdot D^{tr}(\mu) \right) \\ \partial_\theta^2 v(\mu, T) &= D^{tr}(\mu)^{-1} \left( \partial_\theta^2 U^{tr}(\mu) + \frac{T}{2} \partial_\theta^T (\partial_\theta \cdot D^{tr}(\mu)) \right), \end{aligned}$$

where the first term of  $\partial_\theta^2 v(\theta, T)$  drops out since  $D^{tr}$  is positive definite, so  $\partial_\theta v(\mu, T) = 0 \implies \partial_\theta U^{tr}(\mu) + \frac{T}{2} \partial_\theta \cdot D^{tr}(\mu) = 0$ . Now, let  $\theta_k^{tr}$  be a local minimum of  $U^{tr}$ , and  $\mu_k$  a nearby local minimum of  $v$ . To find the approximate bias  $b_k = \theta_k^{tr} - \mu_k$ , we can expand the  $0 = \partial_\theta v(\mu, T)$  equation above:

$$0 = D^{tr}(\mu_k)^{-1} \left( \partial_\theta U^{tr}(\mu_k) + \frac{T}{2} \partial_\theta \cdot D^{tr}(\mu_k) \right) \approx D^{tr}(\mu_k)^{-1} \left( -C_k^{tr} b_k + \frac{T}{2} \partial_\theta \cdot D^{tr}(\mu_k) \right),$$

since  $\partial_\theta U^{tr}(\mu_k) \approx \partial_\theta U^{tr}(\theta_k^{tr}) + \partial_\theta^2 U^{tr}(\theta_k^{tr})(\mu_k - \theta_k^{tr}) = -C_k^{tr} b_k$ . So,

$$b_k \approx \frac{T}{2} (C_k^{tr})^{-1} (\partial_\theta \cdot D_\theta(\mu_k)) = O(T).$$

For the covariance, we can make a Laplace approximation to  $\rho(\theta)$  centered at  $\mu_k$ :

$$\begin{aligned} v(\theta, T) &\approx v(\mu_k, T) + \frac{1}{2} \partial_\theta^2 v(\mu_k, T), \text{ so} \\ \rho(\theta) &\propto e^{-\frac{2}{T} v(\theta, T)} \approx e^{-\frac{2}{T} v(\mu_k, T)} e^{-\frac{1}{T} \partial_\theta^2 v(\mu_k, T)} \\ &\propto \sqrt{|\Sigma_k|} \mathcal{N}(\mu_k, \Sigma_k), \text{ with } \Sigma_k = \frac{T}{2} (\partial_\theta^2 v(\mu_k, T))^{-1}. \end{aligned} \tag{25}$$

Therefore,  $\Sigma_k$  is also  $O(T)$  (and inversely proportional to the curvature of the effective potential). The Laplace approximation also allows us to find the weights. We use the eigendecomposition of  $\partial_\theta^2 v(\mu_k, T)$  to approximate the integral of  $\rho(\theta)$  over the basin  $\mathcal{B}_k$ :

$$\begin{aligned} \partial_\theta^2 v(\mu_k, T) &= M_k \Lambda_k M_k^T \quad (\partial_\theta^2 v(\mu_k, T) \text{ is positive definite so } \lambda_i > 0 \text{ for all } i) \\ q &= M_k^T (\theta - \mu_k) \implies e^{-\frac{1}{T} (\theta - \mu_k)^T \partial_\theta^2 v(\mu_k, T) (\theta - \mu_k)} = e^{-\frac{1}{T} q^T \Lambda_k q} \\ \int_{-\infty}^{\infty} e^{-\frac{1}{T} q^T \Lambda_k q} dq &= \prod_{i=1}^n \int_{-\infty}^{\infty} e^{-\frac{1}{T} \lambda_i q_i^2} dq_i = \prod_{i=1}^n \sqrt{\pi / \lambda_i} = \sqrt{\pi^n |\Lambda_k|^{-1}} = \sqrt{\pi^n |\partial_\theta^2 v(\mu_k, T)|^{-1}} \\ \implies w_k &= \int_{\mathcal{B}_k} \rho(\theta) d\theta \approx e^{-\frac{2}{T} v(\mu_k)} \int_{\mathcal{B}_k} e^{-\frac{1}{T} (\theta - \mu_k)^T \partial_\theta^2 v(\mu_k, T) (\theta - \mu_k)} d\theta \\ &= e^{-\frac{2}{T} v(\mu_k)} \int_{-\infty}^{\infty} e^{-\frac{1}{T} q^T \Lambda_k q} dq = e^{-\frac{2}{T} v(\mu_k)} \sqrt{\pi^n |\partial_\theta^2 v(\mu_k, T)|^{-1}} \\ &\propto e^{-\frac{2}{T} v(\mu_k)} \cdot |\partial_\theta^2 v(\mu_k, T)|^{-\frac{1}{2}}. \end{aligned}$$

This is equation (6) in the main text. It tells us that the basin weights depend on the depth and width of the basin in terms of the *effective potential*.

## Appendix C. Reparametrization invariance

### C.1. Expected test loss under any parameter distribution is reparametrization-invariant

Let  $U(\theta)$  denote any loss function (for example, the test loss, or the train loss with or without regularization), and let  $\rho(\theta)$  be any distribution over the model parameters. Consider a reparametrization  $y = r^{-1}(\theta)$  of  $\theta$  (with  $\theta, y \in \mathbb{R}^p$ ), where  $r: \mathbb{R}^p \rightarrow \mathbb{R}^p$  is invertible. For an arbitrary function  $f(\theta)$ , we define the reparametrized version by  $f^r(y) = f(r(y)) = f(\theta)$ . With this notation, we want to show that  $E_{\theta \sim \rho}[U(\theta)]$  is reparametrization-invariant.

Let  $\rho_Y(y)$  denote the p.d.f. of  $y = r^{-1}(\theta)$ , and note that (using a general formula for invertible functions of random variables)  $\theta = r(y) \implies \rho_Y(y) = \rho(\theta) \left| \frac{dr}{dy} \right|$ . Then:

$$\begin{aligned} E_{\theta \sim \rho}[U(\theta)] &= \int_{-\infty}^{\infty} U(\theta) \rho(\theta) d\theta \\ &= \int_{-\infty}^{\infty} U(r(y)) \rho(r(y)) \left| \frac{dr}{dy} \right| dy, \quad (\text{integral change of variable } y = r^{-1}(\theta)) \\ &= \int_{-\infty}^{\infty} U^r(y) \rho_Y(y) dy, \quad \text{using } \rho_Y(y) = \rho(\theta) \left| \frac{dr}{dy} \right| \\ &= E_{y \sim \rho_Y}[U^r(y)]. \end{aligned}$$

Therefore  $E_{\theta \sim \rho}[U(\theta)]$  is reparametrization-invariant.

In particular, this means that  $E_{\theta \sim \rho}[U^{test}(\theta)]$  is reparametrization-invariant, where  $U^{test}(\theta)$  is the test loss and  $\rho$  is the SGD steady-state distribution.

### C.2. Taylor approximation is reparametrization-invariant

Using equations (3) and (4), we obtain the following approximation for the expected test loss:

$$E_{\theta \sim \rho}[U^{test}(\theta)] \approx \sum_k w_k (U_k^{test} + \frac{1}{2} (s_k + b_k)^T C_k^{test} (s_k + b_k) + \frac{1}{2} \text{Tr}[\Sigma_k C_k^{test}]). \quad (26)$$

Our goal in this section is to show that this expression is approximately reparametrization-invariant for any distribution  $\rho(\theta) = \sum_k w_k \mathcal{N}(\mu_k, \Sigma_k)$  such that  $b_k$  and  $s_k$  are both small (which holds for SGD because we are already assuming that  $s_k$  is small, and the SGD steady-state distribution satisfies  $b_k = O(T) \ll 1$ ). As before, let  $r: \mathbb{R}^p \rightarrow \mathbb{R}^p$  be an invertible reparametrization. Defining  $y_k^{test} = r^{-1}(\theta_k^{test})$ , we need to show that the terms appearing on the right-hand-side of (26) are (approximately) reparametrization-invariant.

We can show that the weights  $w_k$  are reparametrization-invariant using a similar argument to  $E_{\theta \sim \rho}[U(\theta)]$ :

$$w_k = \int_{\mathcal{B}_k} \rho(\theta) d\theta = \int_{r^{-1}(\mathcal{B}_k)} \rho(r(y)) \left| \frac{dr}{dy} \right| dy = \int_{\mathcal{B}_k^y} \rho_Y(y) dy = w_k^r.$$

Now we want to show that the remaining terms in left hand side of (26) are approximately reparametrization-invariant. First, since the test loss is a scalar function, it is equal to its reparametrization, i.e.:

$$U_k^{test,r}(y_k^{test}) = U^{test}(r(y_k^{test})) \equiv U^{test}(\theta_k^{test}) = U_k^{test}.$$

So  $w_k U_k^{test}$  is invariant. To help with the other terms, note that the first two derivatives of any scalar function  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  are:

$$\begin{aligned} \partial_y f(r(y)) &= \partial_y r(y)^T \partial_\theta f(\theta), \\ \partial_y^2 f(r(y)) &= \partial_y r(y)^T \partial_\theta^2 f(\theta) \partial_y r(y), \text{ if } \theta \text{ is a local extremum of } f, \end{aligned}$$

where  $\partial_y r(y) = [\frac{\partial \theta_i}{\partial y_j}]_{i,j} \in \mathbb{R}^{n \times n}$ ,  $\partial_y f(r(y)) \in \mathbb{R}^n$ ,  $\partial_y^2 f(r(y)) \in \mathbb{R}^{n \times n}$ ,  $\partial_\theta f \in \mathbb{R}^n$ ,  $\partial_\theta^2 f \in \mathbb{R}^{n \times n}$ . We use these to find:

$$\begin{aligned}
C_k^{test,r} &= \partial_y^2 U^{test}(r(y_k^{test})) \\
&= \partial_y r(y_k^{test})^T \partial_\theta^2 U^{test}(\theta_k^{test}) \partial_y r(y_k^{test}) \quad \text{since } \partial_\theta U^{test}(\theta_k^{test}) = 0 \\
&= \partial_y r(y_k^{test})^T C_k^{test} \partial_y r(y_k^{test}) \\
s_k^r &= y_k^{tr} - y_k^{test} \\
&\approx \partial_y r(y_k^{test})^{-1} (\tilde{\theta}_k^{test} - \theta_k^{test}) + O((y_k^{tr} - y_k^{test})^2) \quad \text{since } \Delta\theta \approx (\partial_y r) \Delta y + O(\Delta y^2) \\
&= \partial_y r(y_k^{test})^{-1} s_k + O(s_k^2) \\
\Sigma_k^r &= \partial_y r(y_k^{test})^{-1} \Sigma_k \partial_y r(y_k^{test})^{-T} \quad \text{since } \rho_Y(y) = d_y r(y) \rho(\theta) \\
b_k^r &= r^{-1}(\mu_k) - y_k^{tr} \\
&\approx \partial_y r(y_k^{test})^{-1} b_k, \\
\text{Tr}(C_k^{test,r} \Sigma_k^r) &= \text{Tr}(\partial_y r(y_k^{test})^T C_k^{test} \partial_y r(y_k^{test}) \partial_y r(y_k^{test})^{-1} \Sigma_k^{-1} \partial_y r(y_k^{test})^{-T}) \\
&= \text{Tr}(C_k^{test} \Sigma_k^{-1}). \\
s_k^T C_k^{test,r} s_k &= s_k^T \partial_y r(y)^{-T} \partial_y r(y_k^{test})^T C_k^{test} \partial_y r(y_k^{test}) \partial_y r(y)^{-1} s_k + O(s_k^2) \\
&= s_k^T C_k^{test} s_k + O(s_k^2) \\
b_k^T C_k^{test,r} b_k &\approx b_k^T C_k^{test} b_k, \quad b_k^T C_k^{test,r} s_k^r \approx b_k^T C_k^{test} s_k \quad \text{similarly.} \tag{27}
\end{aligned}$$

Since we have seen that each of the terms appearing in (26) (i.e.  $w_k$ ,  $U_k^{test}$ , etc) are all reparametrization-invariant, we conclude the whole expression for the expected test loss is reparametrization-invariant.

## Appendix D. The overparametrized case

### D.1. Some intuition

One of the important properties of the overparametrized case (more model parameters than training samples) is that the loss landscape necessarily has zero- or near-zero-curvature directions, since in the absence of  $\ell_2$ -regularization there can be no more curved directions than there are training samples, and  $\ell_2$ -regularization adds a small amount of curvature in all directions. A question, then, is how zero-curvature directions affect the conclusions in this paper. As a rough intuition, we argue that zero-curvature directions may have little impact on test performance, and be deterministically forced to zero by SGD at steady state in the presence of any  $\ell_2$  regularization, making them essentially irrelevant in the context of our analysis. Consider a simplified case where the set of nonzero-curvature directions is constant (rather than depending on  $\theta$ , as it does in general) and the same for train and test. Suppose that the train loss has a small amount of  $\ell_2$ -regularization (as is typical in practice) but the test loss does not. In this case, the train local minima are strict (single points) but the test local minima are subspaces that include all the zero-curvature directions. The Taylor approximation (3) is identical for *any* point  $\theta_k^{test}$  in the test local minimum space spaces; that is, only the nonzero-curvature directions impact test performance or its shift-curvature approximation. Furthermore, SGD converges to a solution where the parameters in the zero-curvature directions are all equal to zero (as there is no diffusion in these directions and the drift term is  $-\alpha\theta$ , where  $\alpha > 0$  is the  $\ell_2$  parameter). Therefore, the zero-curvature directions are essentially irrelevant in the context of generalization and SGD.

Of course, the situation becomes more complex when we allow the nonzero-curvature directions to vary with  $\theta$  and differ for train and test. Since (3) is a local approximation, the argument above still holds, and we find that the local zero-curvature directions do not affect the test performance near a train local minimum. However, (5) is more complicated, because it depends on the global landscape where the set of nonzero-curvature directions can vary (also, the gradient variance only has rank  $N - 1$ , causing technical difficulties), which is why we require the problem modifications discussed in section 5. In addition to  $\ell_2$ -regularization on the train loss, we also need to add a small amount ( $\beta$ ) of noise to the diffusion matrix to make it invertible, which changes the problem so that SGD does not force the zero-curvature directions to zero as in the discussion above, but instead converges to an infinite-support distribution, with the (local) zero-curvature directions represented locally by Gaussians with mean zero and variance  $T\beta/\alpha$ .

### D.2. Nonzero-curvature subspaces

Our goal is to derive the results in section 5, but we first take a detour to describe in detail why and how overparametrization changes the problem. In the underparametrized case (more data samples  $N$  than model parameters  $p$ ), the per-sample gradients  $\partial_\theta U(x_i, \theta)$  typically span all of  $\mathbb{R}^p$ , while in overparametrized case

( $p > N$ ), the per-sample gradients span a subspace of dimension no larger than  $N$ . To work out the many consequences of this, we introduce some more compact notation. We consider a typical loss function  $U(\theta) = \frac{1}{N} \sum_{i=1}^N U(x_i, \theta)$ . The test loss and the train loss without  $\ell_2$ -regularization in (1) both have this form. Let  $g_i(\theta) = \partial_\theta U(x_i, \theta) \in \mathbb{R}^p$  be a shorthand for the  $N$  per-sample gradients, and  $\mathcal{G}(\theta) = \text{span} \{g_i(\theta)\}_{i=1, \dots, N}$ . Thus  $\mathcal{G}(\theta)$  is the nonzero-curvature subspace at  $\theta$ . Let  $\mathcal{G}^\perp = \mathbb{R}^p \setminus \mathcal{G}_r$  denote the complement subspace orthogonal to  $\mathcal{G}$ . To simplify exposition, we assume that all per-sample gradients are linearly independent, so that the dimension of  $\mathcal{G}$  is  $\min(p, N)$ . Thus, if  $N \geq p$ , then  $\mathcal{G} = \mathbb{R}^p$  and  $\mathcal{G}^\perp$  is empty; but if  $p > N$ , then  $\mathcal{G}$  has dimension  $N$  and  $\mathcal{G}^\perp$  has dimension  $p - N$ . Because the gradients are generally functions of  $\theta$ , the spaces  $\mathcal{G}$  and  $\mathcal{G}^\perp$  can change with  $\theta$ .

As shown at the end of this section, the average gradient is in  $\mathcal{G}$ , the range of the gradient covariance is contained in  $\mathcal{G}$  but has one less dimension, and the range of the Hessian is equal to  $\mathcal{G}$  provided  $\mathcal{G}$  is constant in a neighborhood around  $\theta$ . In particular, the gradient covariance and Hessian are both rank-deficient (hence non-invertible). All of these facts apply directly to the test loss, and to the part of the train loss excluding  $\ell_2$ -regularization. Adding  $\ell_2$ -regularization to the train loss changes the results as follows. The average gradient is  $\partial_\theta U^{tr}(\theta) = (1/N) \sum g_i(\theta) + \alpha\theta$ , so that  $\partial_\theta U^{tr}(\theta)$  is no longer constrained to lie in  $\mathcal{G}^r$  and can be anywhere in  $\mathbb{R}^p$ . The gradient covariance is unchanged since  $\alpha\theta$  is not random; in particular, it is still rank-deficient. The Hessian  $\partial_\theta^2 U^{tr}(\theta)$  becomes full-rank, because regularization adds  $\alpha$  to every eigenvalue of the Hessian. These facts have important implications for the Taylor approximation (3) and the SGD steady-state (5) in the overparametrized case. First, recalling that a local minimum is strict when the Hessian is full-rank, we see that test local minima are non-strict, while train local minima are strict only if we add  $\ell_2$ -regularization. Second,  $D^{tr}(\theta)$  is non-invertible, invalidating (5), but we can resolve this by adding isotropic Gaussian noise to the SGD updates to make  $D^{tr}(\theta)$  full rank, together with  $\ell_2$ -regularization to control the part of the drift term in  $\mathcal{G}_{tr}^\perp$ .

D.2.1. Average gradient, covariance, and Hessian are in  $\mathcal{G}$

The average gradient is  $g(\theta) = \frac{1}{N} \sum_{i=1}^N g_i(\theta)$ , which clearly lies in the span of the per-sample gradients:  $g(\theta) \in \mathcal{G}(\theta)$ . The gradient covariance is  $\text{Cov}(g(\theta)) = \frac{1}{N} \sum_{i=1}^N g_i g_i^T - g g^T \in \mathbb{R}^{p \times p}$ . We will show shortly that the range of  $\text{Cov}(g)$  is also contained in  $\mathcal{G}$ , but it is not equal to it, since  $\mathcal{G}$  has one more dimension. This means that  $\text{Cov}(g)$  has a nonempty nullspace containing  $\mathcal{G}^\perp$  and an extra direction, hence  $\text{Cov}(g)$  is not invertible. To see that  $\text{range}(\text{Cov}(g)) \subset \mathcal{G}$ , note that  $\text{Cov}(g)$  (like any  $p \times p$  sample covariance constructed from  $N$  data points) has rank  $\min(p, N - 1)$  (since it can be written as a sum of  $N$  rank-1 matrices,  $\sum (g_i - g)(g_i - g)^T$ , but  $\text{span} \{g_i - g\}$  only has dimension  $N - 1$  when  $p > N$ ). Therefore, when  $p > N$ , the nullspace of  $\text{Cov}(g)$  has dimension  $p - N + 1$ . It is also clear that the nullspace of  $\text{Cov}(g)$  contains  $\mathcal{G}^\perp$  (to see this, consider any vector  $w \in \mathcal{G}^\perp$ , i.e. such that  $w^T g_i = 0$  for all  $i \in \{1, \dots, N\}$ ; it then follows from the definition of  $\text{Cov}(g)$  that  $\text{Cov}(g)w = 0$ ). Therefore, the nullspace of  $\text{Cov}(g)$  contains all of  $\mathcal{G}^\perp$  (which has dimension  $p - N$ ), plus one extra direction. Equivalently, the range of  $\text{Cov}(g)$  is contained in  $\mathcal{G}$ , but  $\mathcal{G}$  has one extra direction. Finally, the range of the Hessian  $\partial_\theta^2 U(\theta)$  is equal to  $\mathcal{G}(\theta)$  if  $\mathcal{G}(\theta)$  is constant in some open neighborhood of  $\theta$ . To see this, note that for any  $w \in \mathcal{G}^\perp(\hat{\theta})$ , we have that  $\partial_\theta^2 U(\hat{\theta})w = \partial_\theta (g(\theta)^T)|_{\hat{\theta}} w = \partial_\theta (g(\theta)^T w)|_{\hat{\theta}} = 0$ , since  $w \in \mathcal{G}^\perp(\hat{\theta}) \implies g(\theta)^T w = 0$  for all  $\theta$  in a neighborhood of  $\hat{\theta}$  by assumption. So the nullspace of  $\partial_\theta^2 U(\hat{\theta})$  contains  $\mathcal{G}^\perp(\hat{\theta})$ . Also, assuming that the per-sample Hessians in the dataset are linearly independent, the range of the Hessian has dimension  $N$ , so it is equal to  $\mathcal{G}$ .

D.3. Overparametrized SGD steady-state

In the overparametrized case, we use the modified SGD (13) which yields the diffusion approximation with the same FP description (10) but with modified probability current  $J(\theta) = (\partial_\theta U^{tr}(\theta) + \alpha\theta) \rho(\theta) + \frac{T}{2} \partial_\theta \cdot ((D^{tr}(\theta) + \beta^2 I) \rho(\theta))$ . Recall that the diffusion matrix in the diffusion process approximation is the covariance of  $\Delta_t$ , which is now  $\lambda T (D^{tr}(\theta) + \beta^2 I)$ , and is invertible when  $\beta \neq 0$ . But the added isotropic noise creates a new problem: since  $\partial_\theta U(x_i, \theta)$  lies in  $\mathcal{G}_r$ , we now have diffusion noise but no drift in  $\mathcal{G}_{tr}^\perp$ , creating the possibility of a distribution of parameters that spreads out forever as  $t \rightarrow \infty$ . To prevent this, we need drift in  $\mathcal{G}_{tr}^\perp$  that keeps parameter values small, and  $\alpha > 0$  accomplishes this for us. So, with  $\alpha, \beta > 0$ , the same approach given in appendix B.1 works to obtain the steady-state distribution, resulting in the same form after the appropriate redefinitions of the diffusion matrix (adding  $\beta^2 I$  to it) and drift term (adding  $\alpha\theta$  to it). With  $\beta = 0$ , it is unclear how find the steady-state distribution: for example, one might try replacing the inverse of  $D^{tr}$ -projected-onto- $\mathcal{G}_r$  with its pseudoinverse, and including a delta function in  $\rho$  to enforce conditions implied by the ODE in the one-dimensional nullspace of  $D^{tr}$ -projected-onto- $\mathcal{G}_r$ , but this choice turns out to yield a nonzero divergence of the probability current  $\nabla \cdot J \neq 0$ , hence it is not a steady-state solution.

#### D.4. Overparametrized Taylor approximation of test loss

If the local minima of train and test are strict (i.e. single points), then the Taylor expansion approximation of the test loss (3) goes through unchanged. As we saw in appendix D.2, the local minima of train are always strict since we assume that the train loss includes nonzero  $\ell_2$ -regularization (which makes the Hessian full-rank). However, the local minima of test are non-strict in the overparametrized case since the Hessian is rank-deficient (unless we also add  $\ell_2$ -regularization to the test loss, which is not usually done in practice). However, we can still find a natural analog of the Taylor approximation in the overparametrized case. The test local minima are no longer strict. We assume that  $\mathcal{G}_{test}$  is constant in a neighborhood of  $\bar{\theta}_k^{test}$ , where  $g^{test}(\bar{\theta}_k^{test}) = 0$ . From appendix D, we know that  $\partial_\theta U^{test}(\theta) = g^{test}(\theta)$ , where  $g^{test}(\theta) \in \mathcal{G}_{test}^\perp(\theta)$ , and  $\mathcal{G}_{test}(\theta)$  has dimension  $N < p$ . Therefore (as shown at the end of this section) each test local minimum is a set:

$$\theta_k^{test} = \{ \theta_k^{test}(w) \equiv \text{Proj}_{\mathcal{G}_{test}}(\bar{\theta}_k^{test}) + w, \text{ for any } w \in \mathcal{G}_{test}^\perp \}, \text{ where } g^{test}(\bar{\theta}_k^{test}) = 0. \quad (28)$$

The local minima of the train loss are strict due to the  $\ell_2$ -regularization. (The train gradients are  $\partial_\theta U^{train}(\theta) = g^{train}(\theta) + \alpha\theta$ , which can only be zero when the element in  $\mathcal{G}_{tr}^\perp$  is equal to zero). Recall that in (3), we seek to approximate the test loss at a point  $\hat{\theta}$  found by a single trajectory of SGD. We know from the previous section that  $\hat{\theta}$  must have its projection on  $\mathcal{G}^\perp$  equal to zero, so it is also unique. Taylor-expanding the test loss about any  $\theta_k^{test}(w)$  in the test-loss-local-minimum subspace, we obtain:

$$\begin{aligned} U^{test}(\theta_k^{tr}) &\approx U^{test}(\theta_k^{test}(w)) + \frac{1}{2}(s(w) + \hat{b})^T \partial_\theta^2 U(\theta_k^{test}(w))(s(w) + \hat{b}), \quad \text{for any } w \\ &= U_k^{test} + \frac{1}{2}(s(w) + \hat{b})^T C_k^{test}(s(w) + \hat{b}), \quad \text{since } U, \partial_\theta^2 U \text{ do not depend on } w \in \mathcal{G}_{test}^\perp, \end{aligned}$$

where  $s(w) = \theta_k^{test}(w) - \theta_k^{tr}$ , and  $\hat{b} = \hat{\theta} - \theta_k^{tr}$ .

Since the range of  $C_k^{test}$  is orthogonal to  $\mathcal{G}_{test}^\perp$ , the term  $s(w)^T C_k^{test} s(w)$  only depends on the part of  $s(w)$  in  $\mathcal{G}_{test}$ , hence the Taylor approximation of  $U^{test}(\theta_k^{tr})$  is the same for any choice of  $w$ . We would get the same result for  $U^{test}(\theta_k^{tr})$  by defining  $s_k = s(w)$  for any  $w$ , but a natural choice (that eliminates the irrelevant-to-the-test-loss part of  $s$  in  $\mathcal{G}_{test}^\perp$ ) is  $s_k = \text{Proj}_{\mathcal{G}_{test}}(\theta_k^{test} - \theta_k^{tr})$ . (This choice of  $s(w)$  corresponds to  $w = \text{Proj}_{\mathcal{G}_{test}^\perp}(\theta_k^{tr})$ , which is the the solution to  $\min_w \|s(w)\|$ .) The overall result is:

$$U^{test}(\hat{\theta}) = U_k^{test} + \frac{1}{2}(s_k + \hat{b})^T C_k(s_k + \hat{b}), \text{ with } s_k = \text{Proj}_{\mathcal{G}_{test}}(\theta_k^{test} - \theta_k^{tr}) \text{ and } \hat{b} = \hat{\theta} - \theta_k^{tr}.$$

We see that the Taylor approximation of the test loss at a train local minimum has the same form as in equation (3), but the definition of the shift takes a more general form: it is now the projection of the difference between the test and train local minima onto  $\mathcal{G}_{test}$  (subspace for the test loss). This definition coincides with the underparametrized definition of  $s_k$  when  $\mathcal{G}_{test}$  is all of  $\mathbb{R}^p$ .

##### D.4.1. Test local minima are given by (28)

Let  $\mathcal{O} = [\mathcal{O}_G \quad \mathcal{O}_{G^\perp}]$  be an orthonormal matrix such that  $\mathcal{O}_G$  is a basis for  $\mathcal{G}_{test}$  and  $\mathcal{O}_{G^\perp}$  is a basis for  $\mathcal{G}_{test}^\perp$ .

So, i.e. the projector onto  $\mathcal{G}_{test}$  is  $\text{Proj}_{\mathcal{G}_{test}} = \mathcal{O}_G \mathcal{O}_G^T$ . Define  $z = \mathcal{O}^T \theta = \begin{bmatrix} \mathcal{O}_G^T \theta \\ \mathcal{O}_{G^\perp}^T \theta \end{bmatrix} \equiv \begin{bmatrix} z_G \\ z_{G^\perp} \end{bmatrix}$ . Then

$$\begin{bmatrix} \partial_{z_G} U^{test}(\theta) \\ \partial_{z_{G^\perp}} U^{test}(\theta) \end{bmatrix} \equiv \partial_z U^{test}(\theta) = \partial_z U^{test}(\mathcal{O}z) = \mathcal{O}^T \partial_\theta U^{test}(\theta) = \begin{bmatrix} \mathcal{O}_G^T g(\theta) \\ 0 \end{bmatrix}.$$

In particular, we have  $\partial_{z_{G^\perp}} U^{test}(\theta) = 0$ . Therefore the unregularized train loss must be constant w.r.t.  $z_{G^\perp}$ , so we can write (by picking  $z_{G^\perp} = 0$ ):

$$\begin{aligned} U^{test}(\theta) &= U^{test}(\mathcal{O}z) = U^{test}\left(\begin{bmatrix} \mathcal{O}_G & \mathcal{O}_{G^\perp} \end{bmatrix} \begin{bmatrix} z_G \\ 0 \end{bmatrix}\right) = U^{test}(\mathcal{O}_G z_G) = U^{test}(\text{Proj}_{\mathcal{G}_{test}}(\theta)) \\ &\implies U^{test}(\theta_k^{test}(w)) = U^{test}(\text{Proj}_{\mathcal{G}_{test}}(\bar{\theta}_k^{test})) = U^{test}(\bar{\theta}_k^{test}) \\ &\implies \partial_\theta U^{test}(\theta_k^{test}(w)) = \partial_\theta U^{test}(\bar{\theta}_k^{test}(w)) = 0. \end{aligned} \quad (29)$$

## Appendix E. Higher order curvature

The full Taylor-expansion of a function  $U(\theta)$  (for example, the train or test loss) about a local minimum  $\theta_k$  is given by:

$$U(\theta) = U(\theta_k) + \sum_{j=2}^{\infty} \frac{1}{j!} \partial_{\theta}^j U(\theta_k) (\theta - \theta_k)^j,$$

here we are using multi-index notation, so that

$$\frac{1}{j!} \partial_{\theta}^j U(\bar{\theta}) \theta^j \equiv \sum_{j: j_0 + \dots + j_{n-1} = |j|} \frac{1}{j_0! j_1! \dots j_{n-1}!} \frac{\partial^{|j|} U(\bar{\theta})}{\partial_{\theta}^{j_0} \partial_{\theta_1}^{j_1} \dots \partial_{\theta_{n-1}}^{j_{n-1}}} \bar{\theta}^{j_0} \theta_1^{j_1} \dots \theta_{n-1}^{j_{n-1}}.$$

Note that the derivatives of order 3 and higher are tensors, and  $\partial_{\theta}^j U(\theta_k) \theta^j$  represents a tensor contraction.

In one dimension, a critical point  $\theta_k$  of a function  $f: \mathbb{R} \rightarrow \mathbb{R}$  is a local minimum if:

$$J = \operatorname{argmin} \left\{ j : \partial_{\theta}^j f(\theta_k) \neq 0 \right\} \text{ is even, and } \partial_{\theta}^J f(\theta_k) > 0. \tag{30}$$

This follows from Taylor’s theorem. For example, if  $\partial_{\theta}^2 f(\theta_k) = 0$ , then we can look at the next two derivatives: if  $\partial_{\theta}^3 U(\theta_k) \neq 0$  then  $\theta_k$  is a saddle point; otherwise, it is a local minimum if  $\partial_{\theta}^4 f(\theta_k) > 0$  and a local maximum if  $\partial_{\theta}^4 f(\theta_k) < 0$ . If the 3rd and 4th derivatives are zero, we need to look at the 5th and 6th, and so on. In higher dimensions ( $U: \mathbb{R}^p \rightarrow \mathbb{R}$ ), we can generalize  $\partial_{\theta}^J f(\theta_k) > 0$  to  $\partial_{\theta}^J f(\theta_k) x^J > 0$  for all  $x$  (analogous to definition of positive definite Hessian). But the condition for a local minimum becomes more complex since, for example, the second derivative could be positive semidefinite (with both positive and zero eigenvalues); in positive-eigenvalue directions we can be confident of a local minimum, but in the nullspace we must consider higher derivatives. However, we also know that a critical point  $\theta_k$  is a local minimum of  $U$  if  $\theta_k$  is a local minimum of the 1D function obtained by evaluating  $U$  along *any line* through  $\theta_k$ . This characterization allows us to fall back to the simpler 1D characterization (30) along any line, and suffices for many of our purposes.

In particular, in the Taylor expansion (3), taking  $\hat{\theta} = \theta_k^{tr}$  to simplify the argument, we wish to approximate  $U^{test}(\theta_k^{tr})$  by expanding about  $\theta_k^{test}$ , so we only care about the values of  $U^{test}$  along the line between the train and test local minima. Defining:

$$\begin{aligned} f^{test}(r) &= U^{test}(\Theta(r)), \quad \Theta(r) = \theta_k^{tr} + \frac{r s_k}{\|s_k\|} \implies \Theta(0) = \theta_k^{tr}, \Theta(\|s_k\|) = \theta_k^{test} \\ \implies U^{test}(\theta_k^{tr}) &= f^{test}(0) = f^{test}(\|s_k\|) + \sum_{j=2}^{\infty} \frac{(-\|s_k\|)^{|j|}}{j!} \partial_r^j f^{test}(\|s_k\|), \\ \text{with } \partial_r^j f^{test}(r) &= \partial_{\theta}^j U^{test}(\Theta(r)) \left( \frac{s_k}{\|s_k\|} \right)^j, \end{aligned} \tag{31}$$

(note that this Taylor expansion looks slightly unusual because it is *centered* at  $\|s_k\|$ , the test local minimum, and *evaluated* at 0, the train local minimum, so that  $dr = -\|s_k\|$ .) Since  $\theta_k^{test}$  is a local minimum of  $U^{test}$ , we know that  $f^{test}$  has a local minimum at  $\|s_k\|$ , so condition (30) must hold there. This means that the order  $J$  of the minimum nonzero derivative along the line must be even, and we must have:

$$\partial_r^J f^{test}(\|s_k\|) = \partial_{\theta}^J U^{test}(\theta_k^{test}) \left( \frac{s_k}{\|s_k\|} \right)^J > 0.$$

This offers a way to characterize the curvature in terms of a higher-even-order derivative  $\partial_{\theta}^J U^{test}$ , even if the Hessian  $\partial_{\theta}^2 U^{test}$  is zero along the line between the local minima. Higher curvature along this line corresponds to larger values of  $\partial_{\theta}^J U^{test}(\theta_k^{test}) s_k^J$ . We can also partially generalize the SGD steady-state results to the higher-order curvature case. However, we currently only know how to evaluate the necessary integrals in the special case where the local curvature of  $v$  depends *only on a single even derivative* (for example, only a positive-definite 4th derivative), and where the even derivative tensor is *diagonalizable* (which is not the case in general for tensors of order 3 and higher). That is, we must assume that at its local minima  $v$  can be approximated as:

$$v(\theta) \approx v(\mu_k) + \frac{1}{J!} \partial_{\theta}^J v(\mu_k) (\theta - \mu_k)^J,$$

where  $J$  is even, and  $\partial_{\theta}^J v(\mu_k)$  is diagonalizable. Then,

$$\rho(\theta) \propto \exp \left\{ -\frac{2}{T} v(\theta) \right\} \approx \exp \left\{ -\frac{2}{T} v(\mu_k) \right\} \exp \left\{ -\frac{2}{TJ!} \partial_{\theta}^J v(\mu_k) (\theta - \mu_k)^J \right\}.$$

To find the basin weights, we need to evaluate the integral:

$$\begin{aligned} \int_{B_k} \rho(\theta) d\theta &\propto \exp\left\{-\frac{2}{T}v(\mu_k)\right\} \int_{B_k} \exp\left\{-\frac{2}{TJ_k!} \partial_{\theta}^k v(\mu_k)(\theta - \mu_k)^{J_k}\right\} d\theta \\ &= \exp\left\{-\frac{2}{T}v(\mu_k)\right\} \left(\frac{TJ_k!}{2}\right)^{\frac{n}{k}} \int e^{-\partial_{\theta}^k v(\mu_k)y^k} dy. \end{aligned}$$

To make more progress, we need to evaluate integrals of the form  $\int_{-\infty}^{\infty} \exp\left\{-\partial_x^j v(\mu)x^j\right\} d\theta$  (here  $j$  is even but otherwise arbitrary). This is where we need the assumption that  $\partial_x^j v(\mu)$  is diagonalizable, which always holds for symmetric tensors of order two with real entries (i.e. matrices), but does not hold in general for symmetric tensors of higher orders (Comon 1994). Specifically, we assume that  $\partial_x^j v(\mu)$  can be written as:

$$\begin{aligned} \partial_x^j v(\mu) &= \sum_{i=1}^r \lambda_i q_i^{\otimes j}, \quad \text{with } q_i\text{'s orthogonal, } \lambda_i > 0, \quad r \leq n \\ \implies \partial_x^j v(\mu)x^j &= \sum_{i=1}^r \lambda_i q_i^{\otimes j} x^j = \sum_{i=1}^r \lambda_i (q_i^T x)^j = \sum_{i=1}^r \lambda_i z_i^j, \quad \text{with } z = Q^T x, \quad Q = [v_1 \dots v_n], \end{aligned}$$

where if  $r < n$  we add  $n - r$  additional columns to  $Q$  to complete an orthonormal basis for  $\mathbb{R}^n$ . (Note that the assumption that the  $q_i$ 's are orthogonal implies that  $r \leq n$  since more than  $n$  vectors in  $\mathbb{R}^n$  cannot be linearly independent, and also implies that  $\lambda_i \geq 0$  for all  $i$  because  $\partial_x^j v(\mu)$  is positive semidefinite (Qi 2005). If we take only the first  $r$  nonzero eigenvalues we can assume  $\lambda_i > 0$  for all  $i = 1, \dots, r$ . In this case we have an analog of the determinant given by Qi (2005):  $|\partial_x^j v(\mu)| = \prod_{i=1}^r \lambda_i$ . Then:

$$\begin{aligned} \int_{-\infty}^{\infty} e^{-\partial_x^j v(\mu)x^j} dx &= \int_{-\infty}^{\infty} e^{-\sum_{i=1}^r \lambda_i z_i^j} dz, \quad z = Q^T x, \quad dx = |Q| dz = dz \\ &= \prod_{i=1}^r \int_{-\infty}^{\infty} e^{-\lambda_i z_i^j} dz_i = \prod_{i=1}^r \lambda_i^{-\frac{1}{j}} \int_{-\infty}^{\infty} e^{-y^j} dy_i, \quad y_i = \lambda_i^{1/j} z_i \\ &= |\partial_x^j v(\mu)|^{-\frac{1}{j}} \int_{-\infty}^{\infty} e^{-s^j} ds = 2 |\partial_x^j v(\mu)|^{-\frac{1}{j}} \Gamma\left(\frac{j+1}{j}\right) \\ \implies \int_{B_k} \rho(\theta) d\theta &\approx \frac{\exp\left\{-\frac{2}{T}v(\mu_k)\right\}}{|\partial_x^k v(\mu)|^{\frac{1}{k}}} \left(\frac{TJ_k!}{2}\right)^{\frac{n}{k}} 2\Gamma\left(\frac{J_k+1}{J_k}\right). \end{aligned}$$

To see that agrees with the  $J = 2$  case, note that  $\Gamma(3/2) = \frac{\sqrt{\pi}}{2}$ .

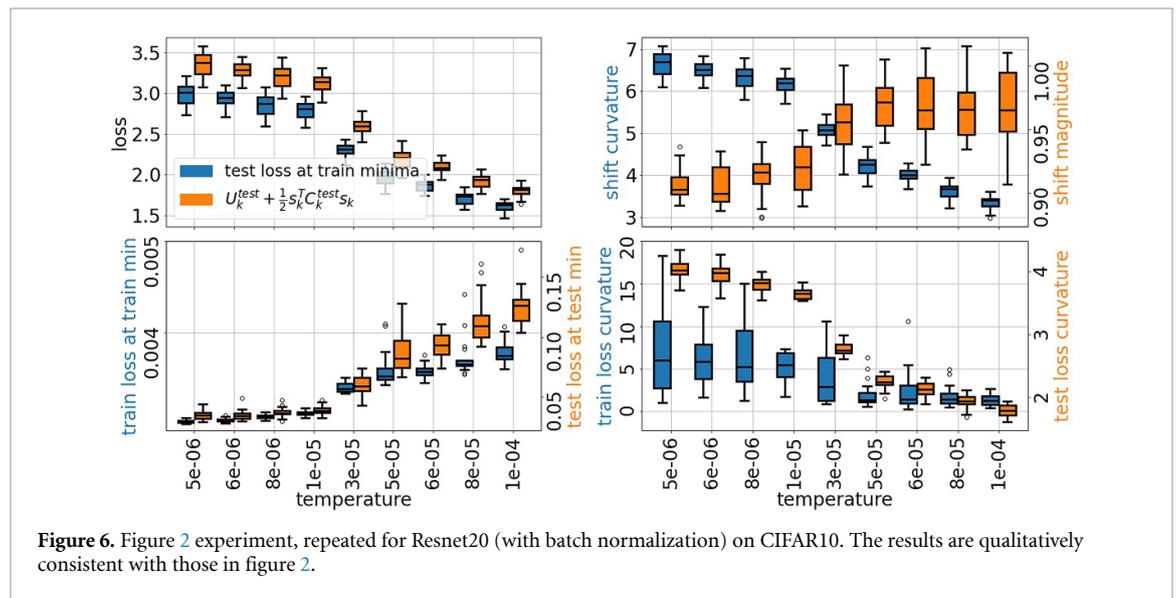
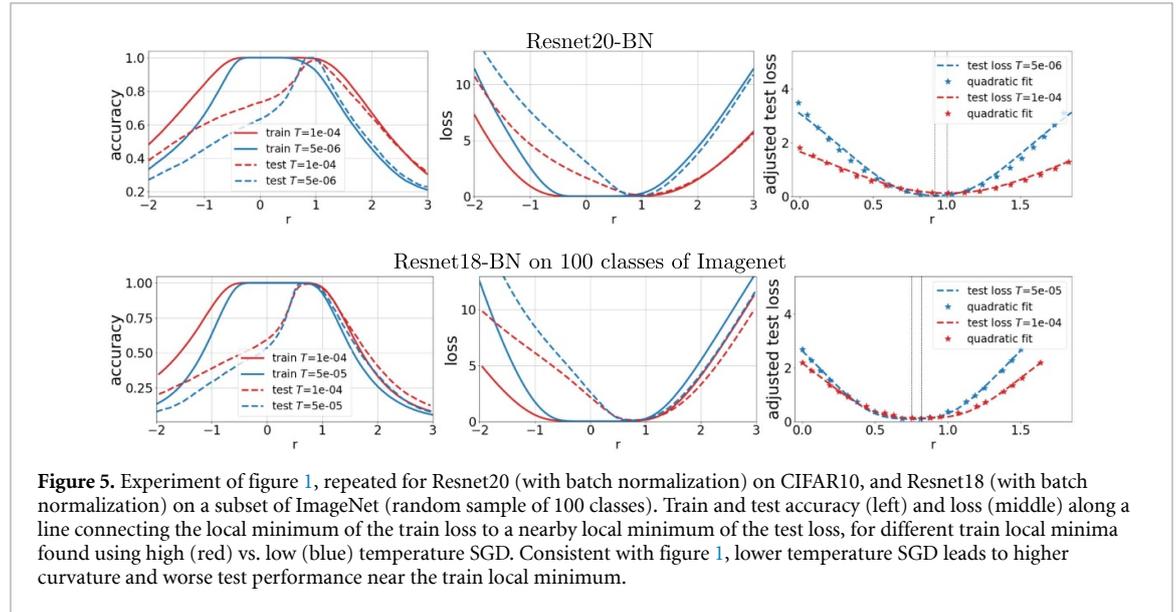
## Appendix F. Experiment notes

### F.1. Shift-curvature symmetrized quadratic fits

Due to the shapes of the losses we find that the symmetrized quadratic fit more accurately captures the relevant overall curvature of the basin. First, the loss curves tend to have a small ‘flattened’ region right around the local minima, despite looking roughly quadratic overall (we suspect the ‘flattening’ is due to sigmoid saturation), so that the curvature evaluated right at the local minima tends to underestimate the overall curvature of the basin, which is better captured by fitting a quadratic. Second, we find that the train and test losses have an asymmetrical shape (He et al 2019b) along the line between the local minima, hence we reflect about the test local minimum in the direction toward the train local minimum since this is the direction that affects the test loss at the test local minimum; similarly, we reflect the train loss about its local minimum in the same direction (i.e. away from the test local minimum) since we are interested in train curvature as a proxy for test curvature so we should study the same direction of each. We believe the observed asymmetry of the losses is due to the experimental design: to find the test minimum, we run gradient descent starting from close to the train minimum, which means we go down the steepest direction of the test basin away from the train local minimum until we reach a test local minimum; however if the test basin has a flat region at the bottom, continuing in the same direction may have zero or very small test loss until we reach the opposite wall of the basin (however, this direction has little impact on the test loss at the train local minimum).

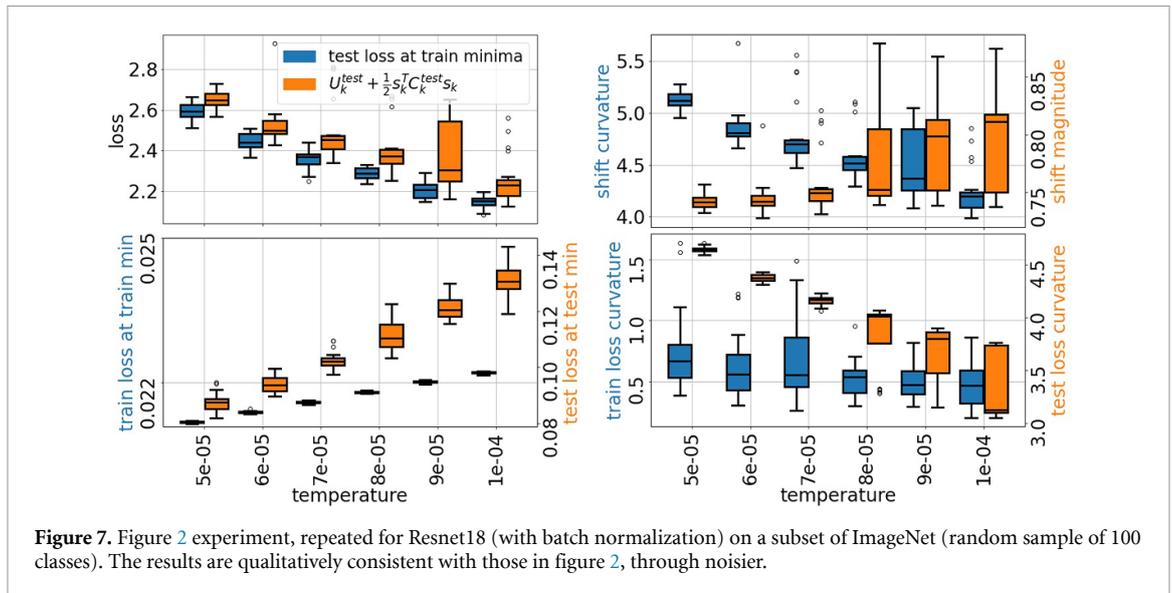
## Appendix G. Additional experiments

### G.1. More shift-curvature experiments

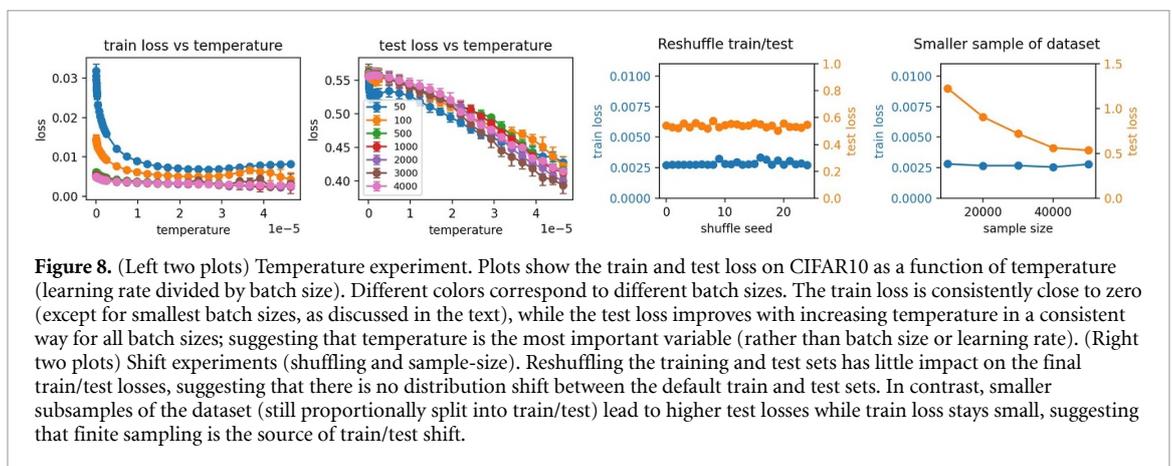


### G.2. Temperature and shift experiments

Figure 8 shows experiments on CIFAR10 (Krizhevsky *et al* 2009) to demonstrate the phenomenon of generalization improving with temperature, as well as to understand the shift between and training and test sets for CIFAR10. We used the VGG9 network Simonyan and Zisserman (2014) and the training procedure of Li *et al* (2017a) (but with no momentum). For the temperature experiment, we first trained a network using a large batch size (4096) and decreasing learning rate schedule (starting from 0.1, scaled by 0.1 at 0.5, 0.75 of total epochs, ending with LR 0.001) until the training converged. Then, we continued training from that initialization with a variety of batch sizes and learning rates (all LRs held constant in second stage), for 1000 epochs (regardless of batch size). For each second-stage run, we took the median of the last 100 steps (to eliminate spiky outliers) as the ‘final loss’. We repeated this two-stage experiment 10 times (with different initializations), and plotted the mean and standard deviation over all trials of the final train and test loss as a function of temperature (learning rate divided by batch size) (from  $2.44 \times 10^{-8}$  to  $2.44 \times 10^{-5}$ ). (At end of first stage, final temperature =  $2.44 \times 10^{-7}$ ; mean final train loss = 0.005 (var  $4 \times 10^{-07}$ ) and test loss = 0.555 (var  $1 \times 10^{-4}$ ).) Our results show that both the train loss and test loss depend primarily on the



**Figure 7.** Figure 2 experiment, repeated for Resnet18 (with batch normalization) on a subset of ImageNet (random sample of 100 classes). The results are qualitatively consistent with those in figure 2, through noisier.



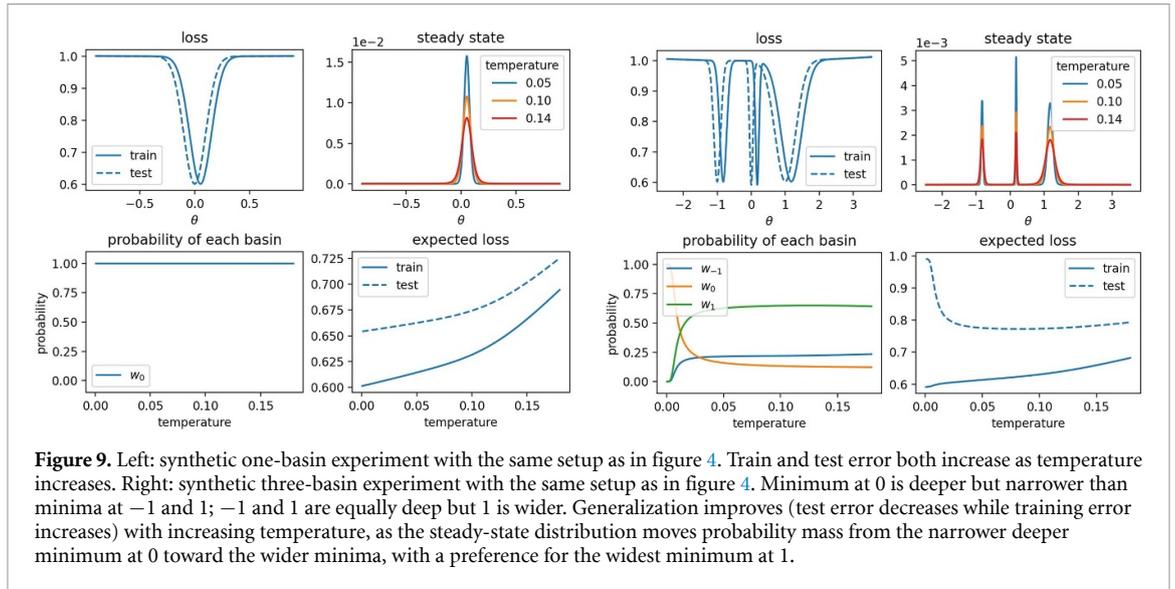
**Figure 8.** (Left two plots) Temperature experiment. Plots show the train and test loss on CIFAR10 as a function of temperature (learning rate divided by batch size). Different colors correspond to different batch sizes. The train loss is consistently close to zero (except for smallest batch sizes, as discussed in the text), while the test loss improves with increasing temperature in a consistent way for all batch sizes; suggesting that temperature is the most important variable (rather than batch size or learning rate). (Right two plots) Shift experiments (shuffling and sample-size). Reshuffling the training and test sets has little impact on the final train/test losses, suggesting that there is no distribution shift between the default train and test sets. In contrast, smaller subsamples of the dataset (still proportionally split into train/test) lead to higher test losses while train loss stays small, suggesting that finite sampling is the source of train/test shift.

temperature, and test loss decreases with increasing temperature while train loss remains roughly constant. For small batches (50 and 100), train loss was higher at the beginning of the second stage for all LR; train loss dropped significantly by the end of training with large LR but remained high with small LR. We believe this was due to mismatched batchnorm statistics and parameters (parameters learned for batch size 4096 but applied to batch size 50 or 100); with larger LR the network was able to retrain the batchnorm parameters, but smaller LR did not allow sufficient progress.

The shift experiments aimed to clarify whether the train/test shift in CIFAR10 is more likely due to distribution shift or to finite sampling. The training procedure was the same as in the initial stage of the temperature experiment. To study distribution shift, we ran a shuffling experiment, where we merged the train and test sets, reshuffled them to create a single dataset, and then randomly split the data into training and test at each trial (25 total). If there were a distribution shift between the default train and test split in the CIFAR data, this process would remove it, and we would expect the test loss to decrease. However, our experiment shows that there is no significant difference in the test loss for different shufflings of the train and test sets, suggesting that distribution shift is not present. To better understand the role of finite sampling, we ran a sample-size experiment, where for each trial we chose a subsample of the full dataset of a given size, split it proportionally into train and test, and trained on the training set. If finite sampling were causing the shift between the training and test distributions, we would expect smaller sample sizes to exacerbate the difference (since sampled distributions generally become closer to the underlying distribution as the sample size grows). Our experiment shows that the test loss increases as the sample size decreases, consistent with finite sampling as the source of distribution shift.

### G.3. One and three basin experiments

Figure 9 shows 1-basin and 3-basin experiments analogous to the 2-basin experiment in figure 4. In the 1-basin example, both train and test error increase with increasing temperature. The 3-basin example shows



qualitatively similar behavior to the 2-basin case and confirms the SGD steady-state distribution's preference for wider minima when the depths are similar.

#### G.4. Experiment details and hyperparameters

##### G.4.1. Figures 1–3 and 5–7

- Resnet10 (without batch norm): Shon (2021) haiku implementation of Resnet10 (He et al 2016). Modified by reducing blocks per group from 3 to 2, and removing batch norm.
- Resnet20 (with batch norm): Shon (2021)'s haiku implementation of Resnet20.
- Resnet18 (with batch norm) for ImageNet: haiku implementation (Hennigan et al 2020).
- VGG16: Yoshida (2021) haiku implementation of VGG16 (Simonyan and Zisserman 2014). Adapted for CIFAR10 by reducing size of dense layers to 512 as proposed by Liu and Deng (2015) and removing adaptive pooling.
- Data: CIFAR10 (Krizhevsky et al 2009), ImageNet (Deng et al 2009)
- Training: adapted from Shon (2021)'s JAX sample code.

CIFAR10 experiment hyperparameters:

Initial SGD	Find Train/Test Local Minima
BATCH_SIZE: 100	TRAIN_SAMPLE_COUNT: 50000
TRAIN_SAMPLE_COUNT: 50000	BATCH_SIZE: 10000(res) 5000(vgg)
TRAIN_SHUFFLE: True	LEARNING_RATE: 1e-4
TRAIN_DROP_LAST: True	MAX_EPOCH: 10000
MAX_EPOCH: 400	L2_REG: 1e-5
MAX_EPOCH_REF_TEMP*: 0.00005	L2_AT_EVAL: True
LEARNING_RATE: various	AUGMENT_TRAIN: False
L2_REG: 1e-5	TRAIN_SHUFFLE: False
L2_AT_EVAL: False	TRAIN_DROP_LAST: False
AUGMENT_TRAIN: False	

ImageNet experiment hyperparameters:

Initial SGD	Find Train/Test Local Minima
BATCH_SIZE: 100	TRAIN_SAMPLE_COUNT: 127149
TRAIN_SAMPLE_COUNT: 127149	BATCH_SIZE: 500
TRAIN_SHUFFLE: True	LEARNING_RATE: 1e-4
TRAIN_DROP_LAST: True	MAX_EPOCH: 300
MAX_EPOCH: 500	L2_REG: 1e-5
MAX_EPOCH_REF_TEMP*: 1e-5	L2_AT_EVAL: True
LEARNING_RATE: various	AUGMENT_TRAIN: False
L2_REG: 1e-5	TRAIN_SHUFFLE: False
L2_AT_EVAL: False	TRAIN_DROP_LAST: False
AUGMENT_TRAIN: False	

\*We scale the number of epochs (E) as  $E * T = \text{MAX\_EPOCHS} * \text{MAX\_EPOCH\_REF\_TEMP}$ .

#### G.4.2. Figures 4 and 9

The *objective* plots show the synthetic train and test objective (loss) functions  $U^{tr}(\theta)$  and  $U(\theta)$ . For the *steady-state distribution* plots, we plug the train loss  $U^{tr}(\theta)$ , the gradient variance  $D^{tr}(\theta)$ , and the current temperature into (5), and evaluate as a function of  $\theta$ . For the *probability of each basin* plots, we integrate the steady-state distribution over the basin of each minimum (i.e. between the two local maxima adjacent to the minimum, or the range endpoint—noting that  $\rho \rightarrow 0$  at the range endpoints); the probabilities are plotted as a function of temperature. Finally, the *train vs. test loss* plots show the train and test losses as a function of temperature, i.e. train loss is computed as  $\int \rho(\theta) U^{tr}(\theta) d\theta$  (where  $\rho(\theta)$  depends on the temperature per (5)).

Param	Figure 4 Left	Figure 4 Right	Figure 9 Left	Figure 9 Right
seed	0	0	0	0
minima	[-1 1]	[-1 1]	[0]	[-1 0 1]
weights	[0.021 0.1]	[0.019 0.1]	[0.1]	[0.1 0.051 0.3]
sigmas	[0.1 0.5]	[0.1 0.5]	[0.1]	[0.1 0.05 0.3]
c	0.001	0.001	0.001	0.001
stddev_shift	0.1	0.1	0.03	0.1
lscale	1.0	1.0	1.0	1.0
wscale	0.0	0.0	0.0	0.0

#### G.4.3. Figure 8

We used a slightly modified version of the code accompanying Li et al (2017a) for training a VGG9 network on CIFAR10. We changed the learning rate schedule to drop at 0.5, 0.75, and 0.9 of total epochs (rather than at fixed epochs), and also added support for reshuffling the train and test sets, and using a subsample of the full dataset.

Stage 1 hyperparameters	Stage 2 hyperparameters:	Figure 8 right
rand_seed: range(0, 10)	resume_epoch: 1200	rand_seed: -1
batch_size: 4096	rand_seed: -1	batch_size: 3000
lr: 0.1	batch_size: various	lr: 0.1
lr_decay: 0.1	lr: various	lr_decay: 0.1
epochs: 300	lr_decay: 1.0	epochs: 2000
weight_decay: 0.0005	epochs: 2200	weight_decay: 0.0005
momentum: 0.9	weight_decay: 0.0005	momentum: 0
	momentum: 0	shuffle_seed: 0:25)
		sample_size: 10000:50000

## ORCID iDs

Arwen V Bradley  <https://orcid.org/0000-0002-4086-217X>

Carlos A Gomez-Uribe  <https://orcid.org/0000-0003-0126-7180>

Manish Reddy Vuyyuru  <https://orcid.org/0000-0002-8034-5898>

## References

- Ahn S, Korattikara A and Welling M 2012 Bayesian posterior sampling via stochastic gradient Fisher scoring (arXiv:1206.6380)
- Bazant M 2005 Lecture 9: Kramers–Moyal cumulant expansion, scribed by Jacy Bird (available at: <https://math.mit.edu/classes/18.366/lec05/lec09.pdf>)
- Belkin M 2021 Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation (arXiv:2105.14368)
- Belkin M, Hsu D, Ma S and Mandal S 2019 Reconciling modern machine-learning practice and the classical bias–variance trade-off *Proc. Natl Acad. Sci.* **116** 15849–54
- Buntine W L 1991 Bayesian back-propagation *Complex Syst.* **5** 603–43
- Chaudhari P and Soatto S 2018 Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks *2018 Information Theory and Applications Workshop (ITA)* (IEEE) pp 1–10
- Comon P 1994 Tensor diagonalization, a useful tool in signal processing *IFAC Proc. Volumes* **27** 77–82
- Corneanu C A, Escalera S and Martinez A M 2020 Computing the testing error without a testing set *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 2677–85
- Deng J, Dong W, Socher R, Li L-J, Li K and Fei-Fei L 2009 ImageNet: a large-scale hierarchical image database *2009 IEEE Conf. on Computer Vision and Pattern Recognition* (<https://doi.org/10.1109/cvpr.2009.5206848>)
- Dinh L, Pascanu R, Bengio S and Bengio Y 2017 Sharp minima can generalize for deep nets *Proc. 34th Int. Conf. on Machine Learning* vol 70 (JMLR) pp 1019–28
- Draxler F, Veschgini K, Salmhofer M and Hamprecht F 2018 Essentially no barriers in neural network energy landscape *Int. Conf. on Machine Learning* (PMLR) pp 1309–18
- Gardiner C 2009 *Stochastic Methods* 4th edn (Berlin: Springer)

- Golmant N, Vemuri N, Yao Z, Feinberg V, Gholami A, Rothauge K, Mahoney M W and Gonzalez J 2018 On the computational inefficiency of large batch sizes for stochastic gradient descent (arXiv:1811.12941)
- Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y and He K 2017 Accurate, large minibatch SGD: training imagenet in 1 hour (arXiv:1706.02677)
- He F, Liu T and Tao D 2019a Control batch size and learning rate to generalize well: theoretical and empirical evidence *Advances in Neural Information Processing Systems* vol 32 pp 1143–52
- He H, Huang G and Yuan Y 2019b Asymmetric valleys: beyond sharp and flat local minima *Advances in Neural Information Processing Systems* vol 32
- He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 770–8
- Hennigan T, Cai T, Norman T and Babuschkin I 2020 Haiku: sonnet for JAX (available at: <http://github.com/deepmind/dm-haiku>)
- Hochreiter S and Schmidhuber J 1997 Flat minima *Neural Comput.* **9** 1–42
- Hoffer E, Hubara I and Soudry D 2017 Train longer, generalize better: closing the generalization gap in large batch training of neural networks *Advances in Neural Information Processing Systems* pp 1731–41
- Jastrzkebski S, Kenton Z, Arpit D, Ballas N, Fischer A, Bengio Y and Storkey A 2017 Three factors influencing minima in SGD (arXiv:1711.04623)
- Jiang Y, Neyshabur B, Mobahi H, Krishnan D and Bengio S 2019 Fantastic generalization measures and where to find them (arXiv:1912.02178)
- Keskar N S, Mudigere D, Nocedal J, Smelyanskiy M and Tang P T P 2016 On large-batch training for deep learning: generalization gap and sharp minima (arXiv:1609.04836)
- Khan M E, Immer A, Abedi E and Korzepa M 2019 Approximate inference turns deep networks into gaussian processes (arXiv:1906.01930)
- Krizhevsky A et al 2009 Learning Multiple Layers of Features from Tiny Images
- Lee J, Bahri Y, Novak R, Schoenholz S S, Pennington J and Sohl-Dickstein J 2017 Deep neural networks as gaussian processes (arXiv:1711.00165)
- Li H, Xu Z, Taylor G, Studer C and Goldstein T 2017a Visualizing the loss landscape of neural nets (arXiv:1712.09913)
- Li Q et al 2017b Stochastic modified equations and adaptive stochastic gradient algorithms *Proc. 34th Int. Conf. on Machine Learning* vol 70 (JMLR) pp 2101–10
- Li X, Gu Q, Zhou Y, Chen T and Banerjee A 2020 Hessian based analysis of SGD for deep nets: dynamics and generalization *Proc. 2020 SIAM Int. Conf. on Data Mining (SIAM)* pp 190–8
- Liu S and Deng W 2015 Very deep convolutional neural network based image classification using small training sample size *2015 3rd IAPR Asian Conf. on Pattern Recognition (ACPR) (IEEE)* pp 730–4
- Mandt S, Hoffman M D and Blei D M 2017 Stochastic gradient descent as approximate Bayesian inference *J. Mach. Learn. Res.* **18** 4873–907
- Martin C H and Mahoney M W 2018 Implicit self-regularization in deep neural networks: evidence from random matrix theory and implications for learning (arXiv:1810.01075)
- McCandlish S, Kaplan J and Amodei D (OpenAI Dota Team) 2018 An empirical model of large-batch training (arXiv:1812.06162)
- Niroomand M P, Morgan J W R, Cafolla C T and Wales D J 2022 On the capacity and superposition of minima in neural network loss function landscapes *Mach. Learn.: Sci. Technol.* **3** 025004
- Orvieto A, Kersting H, Proske F, Bach F and Lucchi A 2022 Anticorrelated noise injection for improved generalization (arXiv:2202.02831)
- Qi L 2005 Eigenvalues of a real supersymmetric tensor *J. Symb. Comput.* **40** 1302–24
- Russo D and Zou J 2016 Controlling bias in adaptive data analysis using information theory *Artificial Intelligence and Statistics (PMLR)* pp 1232–40
- Sagun L, Evci U, Güney U, Dauphin Y and Bottou L 2018 Empirical analysis of the Hessian of over-parametrized neural networks, ICLR (arXiv:1706.04454)
- Seung H S, Sompolinsky H and Tishby N 1992 Statistical mechanics of learning from examples *Phys. Rev. A* **45** 6056
- Shallue C J, Lee J, Antognini J, Sohl-Dickstein J, Frostig R and Dahl G E 2018 Measuring the effects of data parallelism on neural network training (arXiv:1811.03600)
- Shon H 2021 Simple CIFAR10 ResNet in JAX (available at: <https://github.com/hushon/JAX-ResNet-CIFAR10>)
- Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition. (arXiv:1409.1556)
- Smith S L, Dherin B, Barrett D G T and De S 2021 On the origin of implicit regularization in stochastic gradient descent (arXiv:2101.12176)
- Smith S L and Le Q V 2017 A bayesian perspective on generalization and stochastic gradient descent (arXiv:1710.06451)
- Smith S, Elsen E and De S 2020 On the generalization benefit of noise in stochastic gradient descent *Int. Conf. on Machine Learning (PMLR)* pp 9058–67
- Verpoort P C, Lee A A and Wales D J 2020 Archetypal landscapes for deep neural networks *Proc. Natl Acad. Sci.* **117** 21857–64
- Wu J, Hu W, Xiong H, Huan J and Zhu Z 2019 The multiplicative noise in stochastic gradient descent: data-dependent regularization, continuous and discrete approximation (arXiv:1906.07405)
- Xu A and Raginsky M 2017 Information-theoretic analysis of generalization capability of learning algorithms (arXiv:1705.07809)
- Yaida S 2018 Fluctuation-dissipation relations for stochastic gradient descent (arXiv:1810.00004)
- Yoshida D 2021 Vgg16 in Haiku (available at: <https://github.com/davisyoshida/vgg16-haiku>)
- You Y, Gitman I and Ginsburg B 2017 Scaling SGD batch size to 32K for imagenet training (arXiv:1708.03888) vol 6 p 6
- Zhu Z, Wu J, Yu B, Wu L and Ma J 2018 The anisotropic noise in stochastic gradient descent: its behavior of escaping from minima and regularization effects (arXiv:1803.00195)