# Better Training using Weight-Constrained Stochastic Dynamics

**Benedict Leimkuhler**[1] **Tiffany Vlaar**[1] **Timothée Pouchon**[1] **Amos Storkey**[2]

## Abstract

We employ constraints to control the parameter space of deep neural networks throughout training. The use of customized, appropriately designed constraints can reduce the vanishing/exploding gradients problem, improve smoothness of classification boundaries, control weight magnitudes and stabilize deep neural networks, and thus enhance the robustness of training algorithms and the generalization capabilities of neural networks. We provide a general approach to efficiently incorporate constraints into a stochastic gradient Langevin framework, allowing enhanced exploration of the loss landscape. We also present specific examples of constrained training methods motivated by orthogonality preservation for weight matrices and explicit weight normalizations. Discretization schemes are provided both for the overdamped formulation of Langevin dynamics and the underdamped form, in which momenta further improve sampling efficiency. These optimization schemes can be used directly, without needing to adapt neural network architecture design choices or to modify the objective with regularization terms, and see performance improvements in classification tasks.

## 1. Introduction

We study stochastic training methods based on Langevin dynamics combined with algebraic constraints. Our general framework allows for incorporating constraints into standard training schemes and sampling methods for neural networks. Constraints provide direct control of the parameter space of a model and hence afford a means to improve its generalization performance. As applications, we consider magnitude control and orthogonality of neural network weights.

Current approaches to enhance the generalization performance of overparameterized neural networks consist of both explicit and implicit regularization techniques (Neyshabur et al., 2015). Examples of the former are L1 (Williams, 1995; Tibshirani, 1996) and L2 (Hoerl & Kennard, 1970) regularization, which modify the loss by adding a parameter norm penalty term. Batch normalization (BatchNorm) (Ioffe & Szegedy, 2015) is a technique that causes an implicit regularization effect. BatchNorm can be viewed as tantamount to a constraint imposed on the network's parameters during training. Although BatchNorm is widely used, explanations for the method's success remain elusive (Santurkar et al., 2018; Yao et al., 2019). The reliance on increasingly complex strategies does little to enhance the explainability of neural networks, so robust simplification of all aspects of training is desirable. The constrained approach proposed in this paper provides a conceptually straightforward and interpretable framework that offers direct control of parameter spaces, without requiring modifications to the neural network architecture or objective. The transparency of this approach allows for drawing a direct connection between the use of weight constraints and the generalisation performance of the resulting neural network.

In neural network (NN) training one aims to minimize the loss $L_X(\theta)$ for parameters $\theta \in \mathbb{R}^n$ and data $X$. Constraints can be seen as limiting cases of penalty-based regularization which replaces minimization of the loss $L_X(\theta)$ by that of the augmented loss $L_X^c(\theta) = L_X(\theta) + \frac{1}{\varepsilon^2} g(\theta)^2$, where $g(\cdot)$ is a suitable smooth function of the parameters. In the limit $\varepsilon \to 0$, these penalty terms introduce an undesirable stiffness and consequent stability restriction in gradient-based training, which limits the choice of step size (see Figure 5 for an illustration). It is therefore natural to relate the above system to a constrained optimization task subject to $g(\theta) = 0$ (see Section 3).

A popular NN training scheme is stochastic gradient descent (SGD). SGD may be improved by incorporating momenta (Sutskever et al., 2013) and additive noise (Welling & Teh, 2011; Wenzel et al., 2020), or more generally by embedding the loss gradient in a Langevin dynamics (LD) framework (Cheng et al., 2017). We will combine the resulting discretized stochastic differential equation (SDE) approach with constraints (Sec. 4). The benefit of using constrained SDEs for NN training is illustrated in Figure 1, where the

[1]Department of Mathematics, University of Edinburgh, United Kingdom [2]Department of Informatics, University of Edinburgh, United Kingdom. Correspondence to: Tiffany Vlaar <Tiffany.Vlaar@ed.ac.uk>.

*Proceedings of the 38$^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

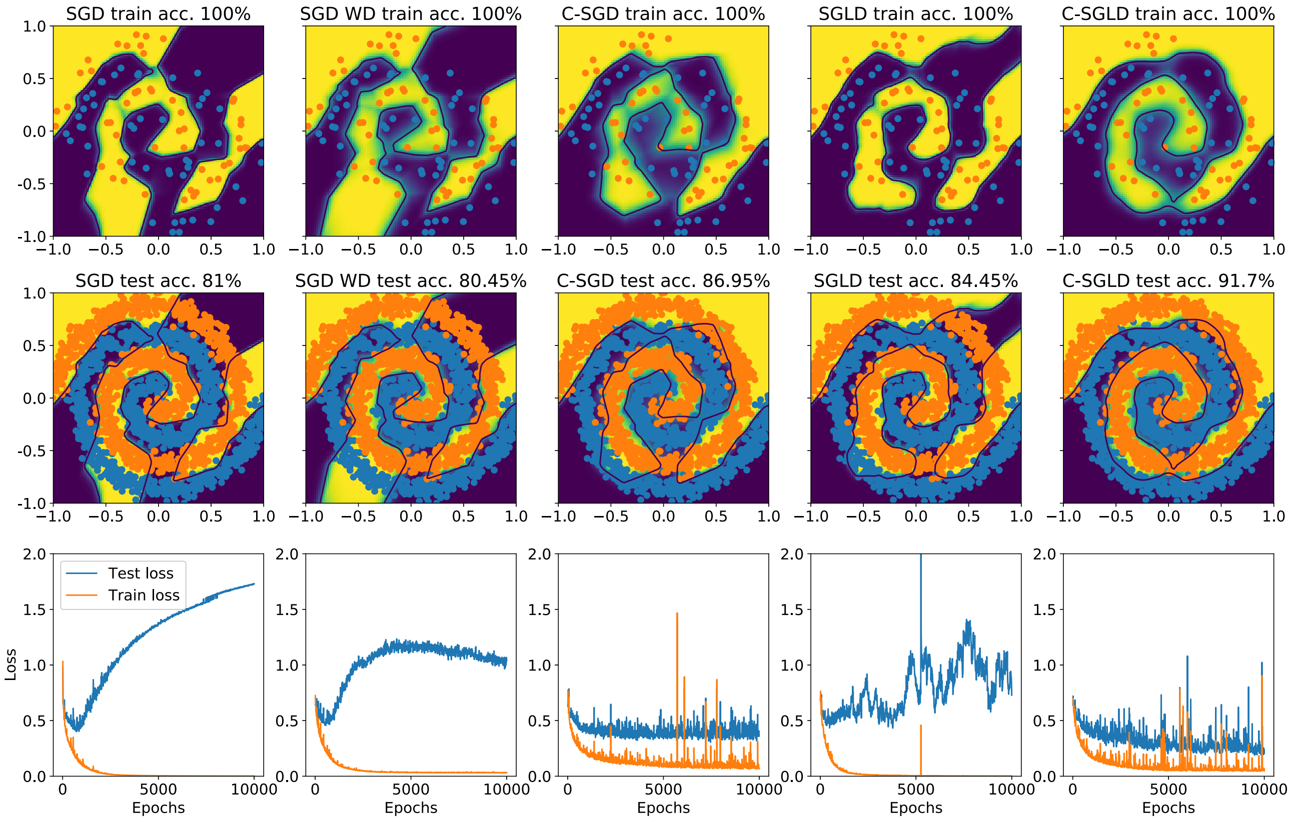

*Figure 1.* Classifiers obtained using different optimizers: SGD (1st column), SGD with weight decay (WD) (2nd col.), constrained SGD (C-SGD) (3rd col.), SGLD (Welling & Teh, 2011) (4th col.), constrained SGLD (5th col.) using a 500-node single hidden layer perceptron for a spiral binary classification problem. Top and middle row show training and test data points, respectively, and decision boundaries of the trained classifier. Bottom row shows loss curves. Hyperpar. settings: all: $h = 0.05$, 2% subsampling; SGD with WD = 1e-4; C-SGD: $r_0 = 1, r_1 = 5$ (see Eq. (2)); SGLD and C-SGLD: $\tau$ = 5e-5 (see Eq. (7)). We observe that although the use of WD can stabilize the test loss, it does not improve test accuracy (2nd col.). In contrast, our constrained approach (3rd col.) maintains a stable test loss throughout training and improved generalization performance. The use of additive noise (or low temperature) in combination with the constraints (C-SGLD, 5th col.) strongly outperforms standard SGD: 91.7% vs. 81% test acc., resp., and obtains smoothened classification boundaries.

combination of using additive noise and magnitude constraints (as defined by Eq. (2)) leads to smoother classification boundaries and significantly enhanced generalization performance (compare the 5th column, the constrained SDE approach, with column 1, standard SGD). These observations are maintained over 100 runs (see Fig. 2, Fig. 3, and Table 1). We distinguish between two different types of smoothness of the resulting classifiers: first, the curvature of the classification boundary and second, the sharpness of the transition between prediction regions belonging to different classes. As shown in Table 1 and Fig. 3 the use of magnitude constraints throughout training generates classifiers which exhibit both types of smoothness. The use of additive noise throughout training further reduces the curvature of the classification boundary. In contrast, the use of weight decay is not sufficient for SGD to obtain the same levels of smoothness. See Appendix D for further numerical details.

*Table 1.* Accompanies Fig. 1 and 2, with same hyperparameter settings. We present estimates of the mean, standard deviation (std), and maximum (max) curvature of classifier boundaries obtained using different optimizers evaluated over 100 runs after training for a fixed number of 10,000 epochs. We computed our curvature estimates using the method described in Appx. D, which we suggest is indicative of the curvature of the locally smoothed classification boundary and allows us to compare the relative curvature estimates of classifiers trained using different optimizers. The combined use of constraints and additive noise (C-SGLD) obtains much lower curvatures compared to SGD with weight decay (WD).

| | *Curvature Approximation* | | |
|---|---|---|---|
| *Optimizer* | Mean | Std | Max |
| SGD | 519 | $4.33 \cdot 10^4$ | $3.26 \cdot 10^8$ |
| SGD with WD | 51.1 | $3.80 \cdot 10^3$ | $1.14 \cdot 10^7$ |
| C-SGD | 9.38 | 317 | $5.58 \cdot 10^5$ |
| SGLD | 8.73 | 189 | $6.27 \cdot 10^5$ |
| **C-SGLD** | 6.08 | 40.8 | $1.43 \cdot 10^5$ |

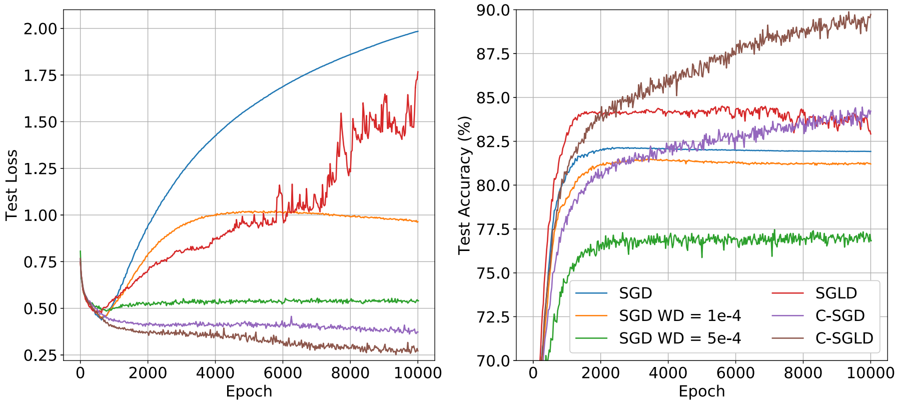

*Figure 2.* Same data and hyperparameter settings as for Fig. 1, but these results are averaged over 100 runs. Constrained approaches, C-SGD and C-SGLD (with additive noise), clearly outperform standard SGD and SGD with WD in terms of test loss and test accuracy.

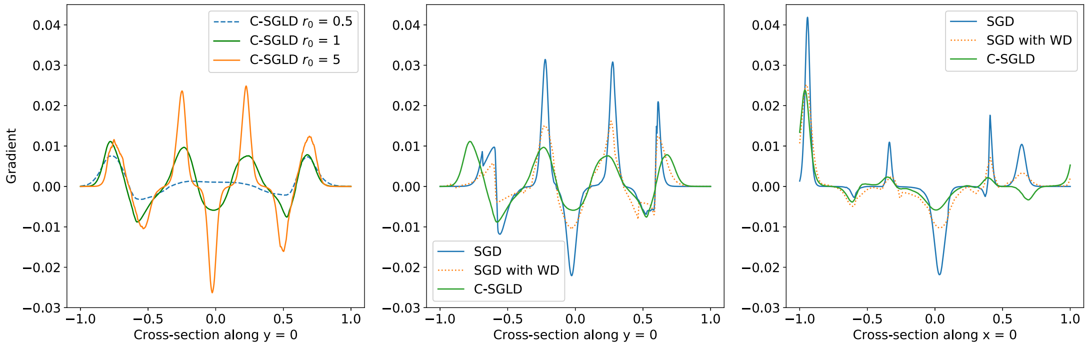

*Figure 3.* Same data and hyperparameter settings as in Fig. 1 and 2. We show gradients of the prediction along horizontal ($y = 0$) and vertical cross-sections ($x = 0$). The results are averaged over 100 runs and evaluated at 10,000 epochs. Our constrained approach C-SGLD exhibits less sharp transitions between classes than SGD (middle/right plot). The size of the constraint directly controls this property (this is illustrated in the left plot for the input layer with constraint size $r_0$ in Eq. (2) for all input layer weights).

Apart from such magnitude constraints, the general framework provided in this paper allows for straightforward incorporation of other constraints. Another specific example we consider is orthogonality of the weight matrix. We provide detailed algorithms for both of these purposes and for a general constraint in a Langevin dynamics setting (Sec. 4 and Appendix B) and show improved generalization performance on classification tasks (Sec. 5).

Concretely, our contributions are:

- We introduce the use of constrained stochastic differential equations for neural network training.
- We provide a general mathematical framework that allows for implementation of new constraints.
- We propose specific constraints, magnitude control and orthogonality of neural network weights, and provide algorithms to accompany these. The benefit of using these is illustrated for several networks and datasets and is shown to outperform soft constraints (such as weight decay or orthogonal regularization).
- We provide PyTorch code to support our algorithms, which can be found on https://github.com/TiffanyVlaar/ConstrainedNNtraining

## 2. Background and Related Work

Neural network loss landscapes are notoriously difficult to characterize rigorously due to their high-dimensionality and non-convexity. Although they appear to contain multiple, roughly equivalent local minima which exhibit nearly zero training loss (Choromanska et al., 2015; Kawaguchi, 2016; Keskar et al., 2017), not all these minima obtain the same generalization performance (Chaudhari et al., 2017; Wu et al., 2017). The training of deep neural networks is hypersensitive to e.g., the choice of initialization (Sutskever et al., 2013), optimizer (Wilson et al., 2017), and hyperparameter settings (Jastrzębski et al., 2018), including learning rate scheduling (Loshchilov & Hutter, 2017; Smith, 2017). Without careful hyperparameter tuning, the loss landscape may not be explored sufficiently by the optimization scheme, thus resulting in a reduced generalization performance of the trained network (Zhang et al., 2015; Keskar et al., 2017).

Sampling methods, which use small amounts of additive noise (Leimkuhler et al., 2019; Wenzel et al., 2020), have been found to enhance exploration and speed the approach to 'good' minima, which enhance their generalization to nearby data sets. Hence, we incorporate the flexibility to use additive noise to enhance exploration in our optimization schemes by taking a constrained SDE approach to neural network training. We propose a general mathematical framework for this purpose and consider the ergodic properties of the idealized SDEs associated with gradient schemes, which may help these methods to ensure robust exploration of a useful range of parameters (Sec. 4). We further propose specific constraints (Sec. 3) and show that the use of these leads to enhanced performance compared to soft constraints, such as weight decay or orthogonal regularization (Sec. 5).

**Magnitude control of neural network weights.** In this work we consider a circle constraint, which limits the magnitude of the size of the weights (we typically leave the biases unconstrained). A corresponding soft constraint, which adds a penalty term to the loss, is weight decay or L2 regularization (Hoerl & Kennard, 1970). We also propose a sphere constraint, which is analogous to max-norm (Srebro & Shraibman, 2005; Srivastava et al., 2014) as used in some regularization procedures. However, applying this constraint in combination with additive noise does yield a distinctive training method.

**Orthogonality of the weight matrix.** The concept of orthogonality has surfaced several times in the recent neural network literature. Orthogonal matrices have properties (norm preservation, unit singular values) which are thought to provide enhanced numerical stability (Zhou et al., 2006; Rodríguez et al., 2017). An orthogonal matrix $Q \in \mathbb{R}^{r \times s}$ (i.e., $Q^T Q = I_s$) is an isometry: $\|Qz\| = \|z\| \; \forall z \in \mathbb{R}^s$. Orthogonal weight matrices were shown to mitigate the vanishing/exploding gradient problem in RNNs (Pascanu et al., 2013; Arjovsky et al., 2016; Vorontsov et al., 2017) and are developing a growing following in the CNN literature as well (Rodríguez et al., 2017; Bansal et al., 2018; Huang et al., 2018; Li et al., 2019). Orthogonal initialization is linked to achieving dynamical isometry (Saxe et al., 2013; Pennington et al., 2017; 2018), which can accelerate training. Xiao et al. (2018) were able to train 10,000 layer vanilla CNNs, without learning rate decay, BatchNorm or residual connections, by using initial orthogonal convolution kernels.

Methods for enforcing orthogonality during training include the use of 'soft' constraints which add a restraint term to the loss (Brock et al., 2017; Xie et al., 2017; Bansal et al., 2018) and hard constraints based on optimization over Stiefel manifolds (Huang et al., 2018; Jia et al., 2019). The latter requires repeated singular value decomposition of high-dimensional matrices during training, which is costly. Int his work we propose a straightforward algorithm to incorporate orthonormality constraints for rectangular matrices within our NN training framework, with manageable additional cost. We make no empirical claims over other manifold optimization methods, but rather provide a framework for network optimization that is theoretically sound, flexible enough to incorporate new constraints, and demonstrates good properties relative to standard SGD training or simple soft constraint approaches.

**Constrained SDEs.** In this work we focus on optimization schemes for neural networks using constrained Langevin dynamics in both its overdamped and underdamped (with momentum) form. A discussion of the properties of unconstrained Langevin dynamics in its overdamped and underdamped forms was studied in Pavliotis (2014). We consider the specific issues associated to the extension of the standard framework to constrained SDEs. The ergodic properties of constrained Langevin (in the absence of gradient noise) were previously studied in Lelièvre et al. (2010) (overdamped) and Lelièvre et al. (2012) (underdamped). Exponential convergence to equilibrium for constrained overdamped Langevin is a consequence of a Poincaré inequality. Poincaré inequalities on manifolds and their use in the analysis of diffusion processes are presented in Bakry et al. (2013), Chapter 4. Finally, Langevin dynamics discretizations are studied in Faou & Lelièvre (2009); Lelièvre et al. (2010) (overdamped) and Lelièvre et al. (2012); Leimkuhler & Matthews (2016) (underdamped).

An alternative to the use of constrained SDEs are constrained Hamiltonian Monte Carlo (HMC) methods (Graham & Storkey, 2017; Zappa et al., 2018; Lelièvre et al., 2020). Although HMC schemes have nil sampling bias if fully converged, their acceptance rates depend on stepsize and system size (Beskos et al., 2013; Bou-Rabee & Sanz-Serna, 2018). In practice SDE-based methods are often preferred in many high-dimensional sampling calculations compared to HMC schemes as they are found to offer greater overall efficiency for a fixed computational budget.

## 3. Neural Networks with Constraints

Imposing good priors on neural networks is known to improve performance, e.g. convolutional neural networks (CNNs) suit image datasets better than overparameterized fully connected NNs, despite being a subset of the latter (d'Ascoli et al., 2019). Using constraints also arises naturally in the control of vanishing/exploding gradients. In Appendix C we illustrate this and also provide a connection between the magnitude of the weights and the smoothness of the interpolant. These observations suggest the use of constraints to control the magnitudes of individual weights and/or to limit the growth of gradients in deep networks. We present various approaches in this section.

We consider a $L$-layer neural network, which has parameters $\theta \in \mathbb{R}^n$, with a weight matrix $W^\ell \in \mathbb{R}^{d^\ell \times d^{\ell-1}}$ and bias vector $b^\ell \in \mathbb{R}^{d^\ell}$ for each layer $\ell$. To allow for inequality constraints, we define slack variables vector $\xi \in \mathbb{R}^{n^\xi}$ and consider variable $q = (\theta, \xi) \in \mathbb{R}^d$, where $d = n + n^\xi$. The constraint manifold is

$$\Sigma = \{q \in \mathbb{R}^d \mid g(q) = 0\}, \;\; g : \mathbb{R}^d \to \mathbb{R}^m. \qquad (1)$$

We partition $\theta = (\theta^u, \theta^c)$ into unconstrained $\theta^u \in \mathbb{R}^{n^u}$ and constrained $\theta^c \in \mathbb{R}^{n^c}$ parameters. We typically only constrain the neural network weights, not the biases.

**Circle constraints:** In a *circle constraint*, we restrict each parameter in $\theta^c$ as $|\theta^c_i| \le r_i$, where $r_i > 0$ is given. We thus introduce $m = n^c = n^\xi$ slack variables $\xi_i$ and define

$$g_i(q) = |\theta^c_i|^2 + |\xi_i|^2 - r_i^2 \qquad 1 \le i \le m. \qquad (2)$$

If $q \in \Sigma$, then the parameters in $\theta^c$ are bounded as desired.

**Sphere constraints:** In a similar way, we could opt to restrict the sums of squares of weights associated to the input channels of any node. For layer $\ell$, we denote the $i$-th row of the weight matrix $W^\ell$ as $\theta^{c,i}$, set $\theta^u = b^\ell$, introduce $m = d^\ell$ slack variables $\xi_i$, and define as *sphere constraint*:

$$g_i(q) = \|\theta^{c,i}\|^2 + |\xi_i|^2 - r_i^2, \; 1 \le i \le m, \qquad (3)$$

where $\|\cdot\|$ denotes the Euclidean norm. Sphere constraints are analogous to max-norm (Srebro & Shraibman, 2005; Srivastava et al., 2014), but have been unexplored in combination with additive noise. We leave this for future work.

**Orthogonality constraints:** We set $\theta^u = b^\ell$, and define as *orthogonality constraint* for layer $\ell$ with $n^\ell$ parameters

$$g(q) = \begin{cases} (W^\ell)^T W^\ell - I_{n^{\ell-1}} & \text{if } n^{\ell-1} \le n^\ell, \\ W^\ell (W^\ell)^T - I_{n^\ell} & \text{otherwise.} \end{cases} \qquad (4)$$

As the matrix equality $g(q) = 0$ is symmetric, it corresponds to $m = s(s+1)/2$ constraints, where $s = \min\{n^{l-1}, n^l\}$.

# 4. Constrained SDEs and their Discretization

In this chapter we describe SDE-based methods for constrained neural network training. We first introduce standard (unconstrained) Langevin dynamics in Section 4.1. Then in Section 4.2 we discuss properties of constrained Langevin dynamics (LD), such as ergodicity and exponential convergence to equilibrium, which ensures the effectiveness of our schemes as training methods. In Section 4.3 we discuss the discretization of constrained Langevin dynamics in both the overdamped and the underdamped case, where the use of momenta allow us to accelerate the training process. The choice of discretization scheme will strongly affect the efficiency and robustness of the resulting training method. Hence, to allow for ease and efficacy of implementation of our methods, we describe the most appropriate discretization schemes in detail in Appendix B for both the general setting and for the specific constraints we consider in this paper, i.e., circle and orthogonality constraints.

## 4.1. Langevin Dynamics

Consider the unconstrained Langevin system of SDEs

$$\begin{aligned} \mathrm{d}\theta_t &= p_t \, \mathrm{d}t, \\ \mathrm{d}p_t &= (-\nabla L(\theta_t) - \gamma p_t)\, \mathrm{d}t + \sqrt{2\gamma\tau}\, \mathrm{d}\mathcal{W}_t, \end{aligned} \qquad (5)$$

with momenta $p$, parameters $\theta$, loss $L(\theta)$, temperature hyperparameter $\tau \ge 0$, friction hyperpar. $\gamma$, and $d$-dim. Wiener process $\mathcal{W}$ (Leimkuhler & Matthews, 2015). Under some mild assumptions, Langevin dynamics is provably ergodic, which means that its solutions sample the distribution:

$$\rho \propto \; \exp[-(L(\theta) + \|p\|^2/2)/\tau]. \qquad (6)$$

The temperature hyperparameter $\tau$, which controls the additive noise level, provides a direct connection between a pure optimization and sampling approach. The standard Bayes posterior is recovered for $\tau = 1$, whereas setting $\tau = 0$ will provide maximum a posteriori (MAP) point estimates. The range of values in between corresponds to an artificially sharpened posterior, where as $\tau \to 0$, the posterior probability mass is confined closer and closer to the modes of the distribution.[1] Using low temperatures (Leimkuhler et al., 2019; Wenzel et al., 2020), sampling methods have been found to enhance exploration and speed the approach to 'good' minima, which enhance their generalization to nearby data sets. In this work we therefore consider a constrained SDE approach to neural network training to allow for the incorporation of both constraints and additive noise.

## 4.2. Constrained Langevin: Ergodicity and Central Limit Theorem.

The neural network loss function naturally extends to the variable $q = (\theta, \xi) \in \mathbb{R}^d$ taking the form $V(q) = L(\theta)$ (note that in particular $\nabla_\xi V = 0$). The first continuous training method we consider is the constrained overdamped Langevin[2] system

$$\begin{aligned} \mathrm{d}q_t &= -\nabla V(q_t)\, \mathrm{d}t + \sqrt{2\tau}\, \mathrm{d}\mathcal{W}_t - \nabla_q g(q_t)\, \mathrm{d}\lambda_t, \\ 0 &= g(q_t), \end{aligned} \qquad (7)$$

[1] Techniques such as annealing or simulated tempering vary $\tau$ throughout training to enhance the optimization process (Kirkpatrick et al., 1983; Marinari & Parisi, 1992).

[2] Unconstrained stochastic gradient overdamped Langevin dynamics is analogous to the algorithm known as SGLD (Welling & Teh, 2011) in the machine learning literature. In SGLD one adds an additional additive noise term (typically with constant variance) to the dynamics. For a decaying sequence of stepsizes $h_n \to 0$ one expects to eventually sample from a known stationary distribution.

where $\mathcal{W}$ is a $d$-dim. Wiener process, $\tau \geq 0$ is the temperature hyperparameter, and $\lambda_t$ is an $\mathbb{R}^m$-valued vector of Lagrange multipliers. Provided the initial configuration $q_0$ satisfies the constraint, any trajectory $q_t$ of (7) remains on the constraint manifold $\Sigma$ defined in Eq. (1). For $\beta^{-1} = \tau > 0$, (7) is equivalent to an underlying ergodic (unconstrained) SDE (see Appx. A.1) with unique invariant measure

$$\mathrm{d}\nu_\Sigma = Z^{-1} e^{-\beta V(q)}\,\mathrm{d}\sigma_\Sigma,\; Z = \int_\Sigma e^{-\beta V(q)}\,\mathrm{d}\sigma_\Sigma, \qquad (8)$$

where $\sigma_\Sigma$ is the surface measure on $\Sigma$.

Ergodicity ensures that averages of observables with respect to $\nu_\Sigma$ can be approximated by time averages of trajectories of (7). To ensure the practical use of (7) as a training method, we need the convergence to occur in a reasonable time. Thanks to the reversibility of the underlying SDE (see Appx. A.1), exponential convergence to equilibrium occurs as a consequence of a Poincaré inequality for $\nu_\Sigma$ (see Appx. A.2, A.3 and Bakry et al. (2013)). We provide a summary of the results here and refer to Appx. A for more details.

A Poincaré inequality holds under a curvature-dimension assumption: there exists $\rho > 0$ such that

$$CD(\rho, \infty): \qquad \mathrm{Ric}_{\mathfrak{g}} + \beta \nabla^2_{\mathfrak{g}} V \geq \rho \mathfrak{g}, \qquad (9)$$

in the sense of symmetric matrices. The terms in (9) rely on the structure of $\Sigma$ as a Riemannian manifold: $\mathfrak{g}$ is the Riemannian metric, $\mathrm{Ric}_{\mathfrak{g}}$ is the Ricci curvature tensor and $\nabla^2_{\mathfrak{g}} V$ is the Hessian of $V$ on the manifold. Under (9) we have the following result ((Bakry et al., 2013), Appx. A.2).

**Theorem 4.1** *Assume that there exists $\rho > 0$ and $N > n$ such that $CD(\rho, N)$ holds. Then $\nu_\Sigma$ satisfies a Poincaré inequality: there exists a constant $L > 0$ such that*

$$\int_\Sigma \left|\phi(q) - \langle\phi\rangle_{\nu_\Sigma}\right|^2 d\nu_\Sigma(q) \leq \frac{1}{2L} \int_\Sigma \left|\Pi(q)\nabla\phi(q)\right|^2 d\nu_\Sigma$$
$$\forall \phi \in H^1(\nu_\Sigma), \qquad (10)$$

*where $\Pi(q)$ is the projection onto the cotangent space $T^*_q\Sigma$ Eq. (A-8) and $H^1(\nu_\Sigma)$ is the space of functions with square $\nu_\Sigma$-integrable gradients Eq. (A-7).*

Consequences of Theorem 4.1 are the exponential convergence and a central limit theorem (CLT) for the convergence in Eq. (A-2) (see Appx. A.3).

**Corollary 4.2** *If (9) holds then*

$$\int_\Sigma \left|\mathbb{E}(\phi(q_t) \mid q_0) - \langle\phi\rangle_{\nu_\Sigma}\right|^2 d\nu_\Sigma(q_0) \leq C(\phi) e^{-2L/\beta t}$$
$$\forall \phi \in H^1(\nu_\Sigma), \qquad (11)$$

*where $C(\phi)$ depends only on $\phi$. Furthermore we have the following convergence in law:*

$$\sqrt{T}\big(\langle\phi\rangle_T - \langle\phi\rangle_{\nu_\Sigma}\big) \to \mathcal{N}(0, \sigma^2_\phi) \quad \text{as } T \to \infty,$$

*where the asymptotic variance $\sigma^2_\phi$ is bounded as $\sigma^2_\phi \leq \frac{\beta}{L}\int_\Sigma \left|\phi - \langle\phi\rangle_{\nu_\Sigma}\right|^2 d\nu_\Sigma$.*

In $\mathbb{R}^n$ assumption (9) is equivalent to convexity of $V$, which is known to be too strong a requirement (a confining assumption is sufficient, see e.g. Lelièvre & Stoltz (2016)). Although (9) can certainly be weakened, the above results ensure that provided the curvature of the manifold is well behaved, sampling on $\Sigma$ has similar properties as on a flat space.

Introducing momenta $p$ leads to constrained underdamped Langevin dynamics, the 2nd order counterpart of Eq. (7)

$$\begin{aligned} &\mathrm{d}q_t = p_t\,\mathrm{d}t,\;\; 0 = g(q_t), &(12)\\ &\mathrm{d}p_t = (-\nabla V(q_t) - \gamma p_t)\,\mathrm{d}t + \sqrt{2\gamma\tau}\,\mathrm{d}\mathcal{W}_t - \nabla g(q_t)\mathrm{d}\lambda_t, \end{aligned}$$

where $\gamma$ is the friction hyperparameter. The constraint induces a cotangency condition: $p \in T^*_q\Sigma$, where $T^*_q\Sigma = \{p \in \mathbb{R}^d \mid \nabla^T g(q) p = 0\}$ is the cotangent space of the manifold $\Sigma$. The corresponding phase space is the cotangent bundle $T^*\Sigma = \{(q,p) \mid q \in \Sigma, p \in T^*_q\Sigma\}$. Given an initial pair $(q,p) \in T^*\Sigma$, any trajectory $(q_t, p_t)$ of (12) stays on $T^*\Sigma$ for all time.

(12) is equivalent to an underlying ergodic SDE, whose invariant measure is $\mathrm{d}\mu = e^{-\beta H(q,p)}\mathrm{d}\sigma_{T^*\Sigma}$, with Hamiltonian $H(q,p) = V(q) + \frac{1}{2}p^T p$ and Liouville measure of the cotangent bundle $\sigma_{T^*\Sigma}$ (Lelièvre et al., 2012). Based on the result for the unconstrained case, we expect exponential convergence to equilibrium also to hold here, but will leave this technical proof (e.g. based on hypocoercivity (Villani, 2009; Lelièvre & Stoltz, 2016)) for future work.

### 4.3. Discretization of Constrained Langevin Dynamics.

The simplest iteration scheme $q_n \in \Sigma \mapsto q_{n+1} \in \Sigma$ for constrained overdamped Langevin dynamics (7) consists of an Euler–Maruyama step followed by projection onto the constraint manifold $\Sigma$. The best choice for the projection is constraint-specific.

For circle constraints we suggest orthogonal projection, which is both explicit and robust (we describe this in detail in Appx. B.3). For orthogonality constraints, we derive an efficient quasi-Newton scheme to solve the non-linear system for the projection step (Appx. B.5). We present the resulting training scheme in Algorithm 1, where we denote $Q = W^\ell$ if $n^\ell \leq n^{\ell-1}$ and $Q = (W^\ell)^T$ otherwise, and present one training iteration $Q_n \in \Sigma \mapsto Q_{n+1} \in \Sigma$. Further, we denote $h$ as the stepsize, $G(Q) = \nabla_Q V(Q)$ and $\tilde{G}$ the gradient of the loss evaluated on a randomly subsampled partial data set. $R_n$ is an independent standard random normal matrix of the same size as $Q$. The initialization must be done with care: the constrained parameters and the potential slack variable must satisfy the constraint initially.

**Algorithm 1** Orthog. constraint overdamped Langevin

```
Every step:
  $Q^{(0)} = Q_n - h\tilde{G}(Q_n) + \sqrt{2\tau h}R_n,$
  for $k = 0$ to $K-1$ do
    $Q^{(k+1)} = Q^{(k)} - \frac{1}{2}Q_n\big((Q^{(k)})^T Q^{(k)} - I_s\big),$
  end for
  $Q_{n+1} = Q^{(K)}.$
```

**Algorithm 2** Orthog. constraint underdamped Langevin

```
Every step:
  $Q^{(0)} = Q_n + hP_n,$
  for $k = 0$ to $K-1$ do
    $Q^{(k+1)} = Q^{(k)} - \frac{1}{2}Q_n\big((Q^{(k)})^T Q^{(k)} - I_s\big),$
  end for
  $Q_{n+1} = Q^{(K)},\ \bar{P}_{n+1} = P_n + \frac{1}{h}\big(Q_{n+1} - Q^{(0)}\big),$
  $P_{n+1} = \Pi_{Q_{n+1}}\bar{P}_{n+1}$
                                                              (A)

  $\bar{P}_{n+1} = P_n - h\tilde{G}(Q_n),$
  $P_{n+1} = \Pi_{Q_n}\bar{P}_{n+1},$
                                                              (B)

  $P_{n+1} = e^{-\gamma h}P_n + \sqrt{\tau(1-e^{-2\gamma h})}R_n,$
  $P_{n+1} = \Pi_{Q_n}\bar{P}_{n+1}$
                                                              (C)
```

For underdamped Langevin dynamics a common way of building discretization schemes is via the use of splitting methods (Leimkuhler & Matthews, 2016). For the constrained underdamped Langevin system (12) an ABO splitting strategy under $0 = g(q_t),\ \ 0 = \nabla_q g(q_t)p_t$ gives:

$$\begin{aligned}
&\text{A: } \mathrm{d}q_t = p_t\,\mathrm{d}t, \quad \mathrm{d}p_t = -\nabla_q g(q_t)\,\mathrm{d}\lambda_t,\\
&\text{B: } \mathrm{d}q_t = 0,\ \mathrm{d}p_t = -\nabla_q V(q_t)\,\mathrm{d}t - \nabla_q g(q_t)\,\mathrm{d}\mu_t, \qquad (13)\\
&\text{O: } \mathrm{d}q_t = 0,\ \mathrm{d}p_t = -\gamma p_t\,\mathrm{d}t + \sqrt{2\gamma\tau}\,\mathrm{d}\mathcal{W}_t - \nabla g(q_t)\,\mathrm{d}\nu_t,
\end{aligned}$$

In the specific case $\tau = 0$ and by re-scaling $\mu = e^{-\gamma h}/h$ and $\delta t = h^2$, an OBA sequence is equivalent to the standard PyTorch form of SGD with momentum $\mu$ and stepsize $\delta t$ (Paszke et al., 2017; Leimkuhler et al., 2019). As alternative one could use a symmetric splitting method, e.g. BAOAB method (Leimkuhler et al., 2016), but this would lose its accuracy order advantage in the presence of gradient noise.

In (13) the B and O components can be solved exactly (in law) while the A component can be approximated using a standard scheme for constrained ODEs (e.g. SHAKE or RATTLE (Leimkuhler & Reich, 2004)[Chap. 7]). Importantly, the A component does not involve the evaluation of the gradient. For circle constraints the A step can be solved explicitly and the corresponding algorithm is provided in detail in Appendix B.4. For orthogonality constraints all details are provided in Appendix B.6, but we will provide the algorithm here. For $Q \in \Sigma$, the projection onto the cotangent space $T^*_Q\Sigma$ is defined as $\Pi_Q : \mathbb{R}^{r\times s} \to \mathbb{R}^{r\times s}$,

$$\bar{P} \mapsto \Pi_Q\bar{P} = \bar{P} - \frac{1}{2}Q(\bar{P}^T Q + Q^T\bar{P}). \qquad (14)$$

We initialize the parameters and momenta (using projection (14)) to obey the constraint. Then the ABO steps $(Q_n, P_n) \in T^*\Sigma \mapsto (Q_{n+1}, P_{n+1}) \in T^*\Sigma$ are given by Algorithm 2, where $\tilde{G}(Q)$ is the gradient of the loss evaluated on a subset of the data. More details in Appx. B.

## 5. Numerical Experiments

The use of constraints can enhance generalization performance. We support this claim by comparing the performance of neural network architectures trained using the constrained approaches described in this paper to nets trained using unconstrained SGD. We typically set $\tau = 0$ and use equivalent learning rates to present a fair comparison between constrained and unconstrained approaches. We denote our circle and orthogonal Constrained *overdamped* Langevin Algorithms as c-CoLA-*od* and o-CoLA-*od*, respectively. We compare *underdamped* variants (CoLA-*ud*) with SGD with momentum (SGD-m).

### 5.1. Orthogonality Constraints

In Fig. 4 we want to train a multi-layer perceptron (MLP) with $p$ hidden layers on a tightly wound spiral binary classification problem (Fig. D1) and compare the performance of SGD with our orthogonality-preserving overdamped Langevin method o-CoLA-od. For SGD we show results for i) standard PyTorch initialization, ii) orthogonal initialization, and iii) orthogonal regularization ('soft constraint'), where a penalty term is added to the loss to encourage orthogonality of the NN weight matrices. Our o-CoLA-od method clearly outperforms all of these variants in terms of test accuracy for MLPs with more than 3 hidden layers. In Appx. D (Fig. D2) we show that the use of a small temperature perturbation can speed up training even further and slightly increase the test accuracy. The performance of the soft constraint approach can be somewhat improved by lowering the stepsize, yet cannot match the performance of o-CoLA-od (see Figure 5). This illustrates the undesirable stiffness introduced into the system by using penalty-based regularization. The use of o-CoLA-od also removes the need to tune an additional parameter (the penalty strength).

For a ResNet-34 architecture with BatchNorm and learning rate (LR) decay on CIFAR-10 (Krizhevsky & Hinton, 2009) data our underdamped orthogonal constrained method, without weight decay (WD) significantly outperforms SGD-m without WD (Fig. 6). The overdamped case is presented in the supplement, Fig. D3. In future work we will explore the nuances of combining orthogonality constraints with BatchNorm, residual connections and LR decay.

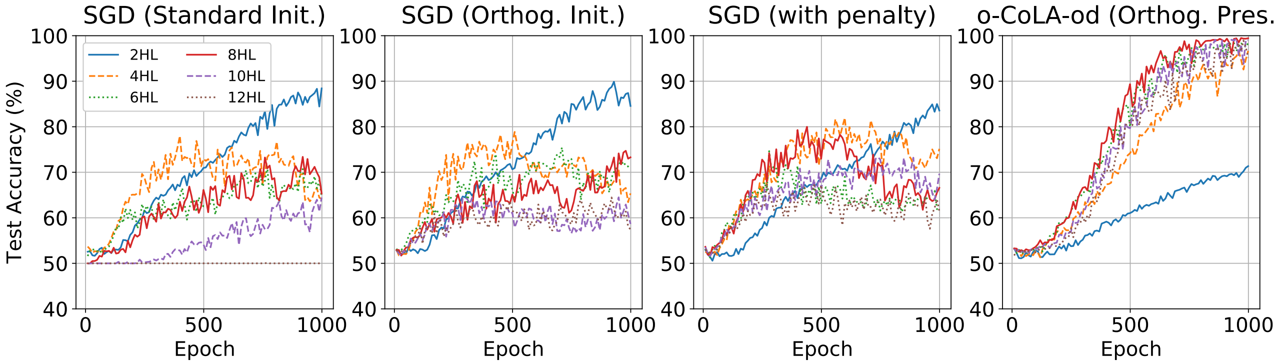

*Figure 4.* Test acc. of MLPs with $p$-number of 100-node hidden layers (HL), ReLU activation. The MLPs are trained on a 4-turn spiral dataset (Fig. D1) using SGD with standard initialization (1st column), SGD with orthogonal initialization (2nd col.), SGD with orthogonal regularization ('soft constraint') by adding a penalty term with strength $\lambda$ to the loss (3rd col.), and o-CoLA-od with $\tau = 0$ (4th col.). For the orthogonal regularization approach and o-CoLA-od we constrain weights in all layers, apart from input and output layers. We set stepsize $h = 0.1$ for all methods and use 5% subsampling. We found the optimal penalty strength $\lambda = 0.05$ for the orthogonal regularization method through line search. Results are averaged over 10 runs. We observe that our o-CoLA-od method significantly outperforms unconstrained SGD and SGD with a soft constraint for MLPs with more than 3 hidden layers.

### 5.2. Circle Constraints

We evaluate our circle constrained c-CoLA-ud method on the Fashion-MNIST data set (Xiao et al., 2017). We reduce the amount of training data to 10K samples and use the remaining 60K samples as test data. c-CoLA-ud clearly outperforms SGD-m in terms of both test accuracy and test loss for a 1000-node single hidden layer perceptron (see Fig. 7). The lower test loss of c-CoLA-ud is maintained during training and the method shows no signs of overfitting, thus eliminating the need for early stopping. Even with weight decay, SGD-m is outperformed by its constrained counterpart (for more detailed hyperparameter studies see Appx. D). We also show that a small transformer (Vaswani et al., 2017) with 2 encoder layers (each with 2-head self-attention and 200-node feed-forward network) trained using c-CoLA-ud achieves a lower validation loss on NLP datasets than its unconstrained counterpart, SGD-m (Table 2).

## 6. Conclusion

We provide a general framework that can be used to directly influence the parameter space of deep neural networks. The constrained SDE-based algorithms described in this paper allow for the use of additive noise to enhance exploration but can also be used directly in combination with standard SGD approaches. We provide a mathematical framework to study these regularized training methods as discretizations of constrained Langevin dynamics and provide detailed discretization schemes (see Appendix B). As specific examples of constraints we consider circle and orthogonality constraints, which obtain improved generalization performance on classification tasks compared to unconstrained SGD and soft constraint approaches. Further uses of our general framework are left for future work.

*Table 2.* Minimum val. loss on Penn Treebank data (batchsize 1024) (Marcus et al., 1993) and Wikitext-2 (batchsize 128) (Merity et al., 2017) using a transformer trained using c-CoLA-ud or SGD-m. Hyperpar. c-CoLA-ud: $h = 0.4, r = 0.5, r_L = 0.1, r_N = 1, r_A = 1, \tau = 0, \gamma = 0.5$ (Treebank) and $\gamma = 1$ (Wikitext-2), where the subscripts $L, N, A$ represent the radii belonging to the linear, norm and self- attention layers respectively. The transformer trained using c-CoLA-ud obtains lower validation losses. Studies with weight decay are provided in the supplement.

| *Optimizer* | Penn Treebank | Wikitext-2 |
|---|---|---|
| c-CoLA-ud | **4.81** | **5.09** |
| SGD $h = 0.1$ | | |
| $mom = 0.7$ | 4.87 | 5.13 |
| $mom = 0.8$ | 4.83 | 5.13 |
| $mom = 0.9$ | 4.84 | 5.13 |
| SGD $h = 0.2$ | | |
| $mom = 0.7$ | 4.83 | 5.13 |
| $mom = 0.8$ | 4.83 | 5.14 |

## Acknowledgements

The authors wish to thank Gabriel Stoltz and Tony Lelièvre for helpful discussions on constrained SDEs. Benedict Leimkuhler is a fellow of the Alan Turing Institute which is supported by EPSRC grant EP/N510129/1. During the creation of this paper Timothée Pouchon was supported by the Swiss National Science Foundation, project P2ELP2_188037. Tiffany Vlaar is supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016508/01), the Scottish Funding Council, Heriot-Watt University and the University of Edinburgh.

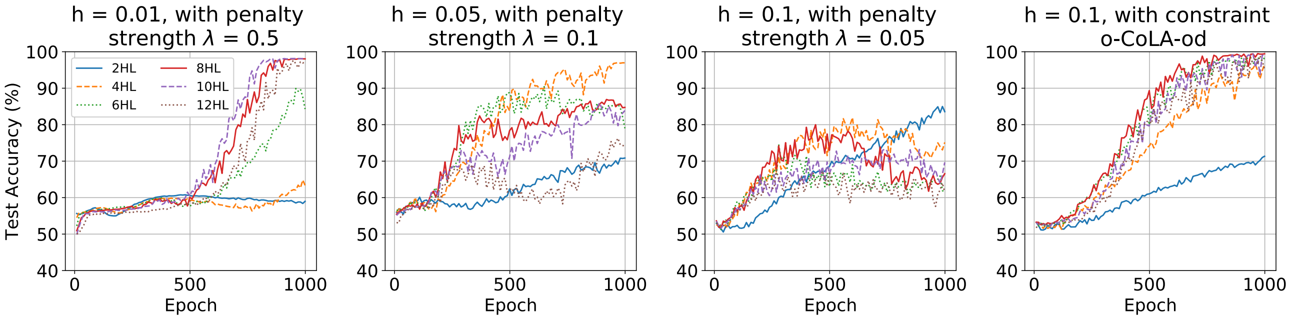

*Figure 5.* Same set-up as for Figure 4. MLPs with varying numbers of hidden layers (HL) were trained using o-CoLA-od with $h = 0.1$ (right-most) and using SGD with a penalty term added to the loss (results are presented in the 1st three columns with varying stepsizes $h$ and penalty strengths $\lambda$). Results are averaged over 10 runs. We illustrate that the use of a penalty-based soft constraint introduces an undesirable stiffness into the system, needing the stepsize to be lowered to improve performance and to allow for the use of larger penalty strengths. The soft constraint approach is unable to reach the same performance as our o-CoLA-od method (right-most) and its performance is heavily dependent on the choice of penalty strength and step size.

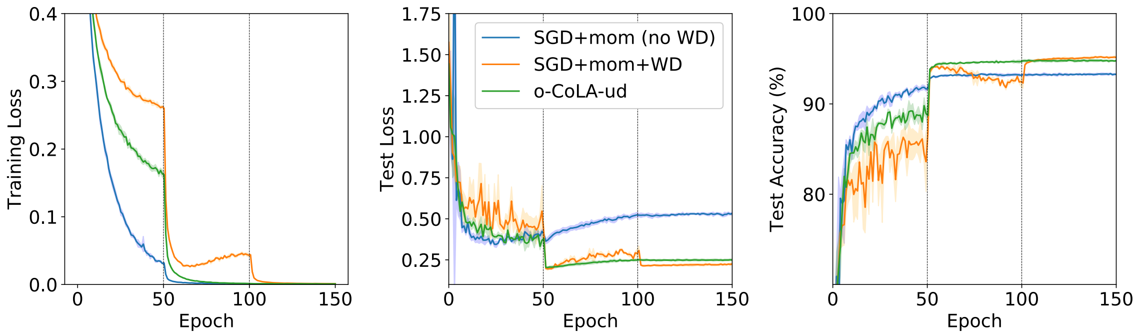

*Figure 6.* Train (left) & test (middle) loss and test accuracy (right) averaged over 5 runs of a ResNet-34 with BatchNorm trained using SGD-m vs. o-CoLA-ud with $\tau = 0$ on CIFAR-10. For SGD we initially use $h = 0.1$ and decay by a factor 10 every 50 epochs (indicated by the vertical black dotted lines). We set momentum = 0.9 and present results with and without WD. o-CoLA-ud (with $\gamma = 0.5$) did not use WD. Its learning rate was re-scaled to match the parameters of SGD-m and used the same LR schedule. The o-CoLA-ud method without weight decay strongly outperforms SGD-m without weight decay.

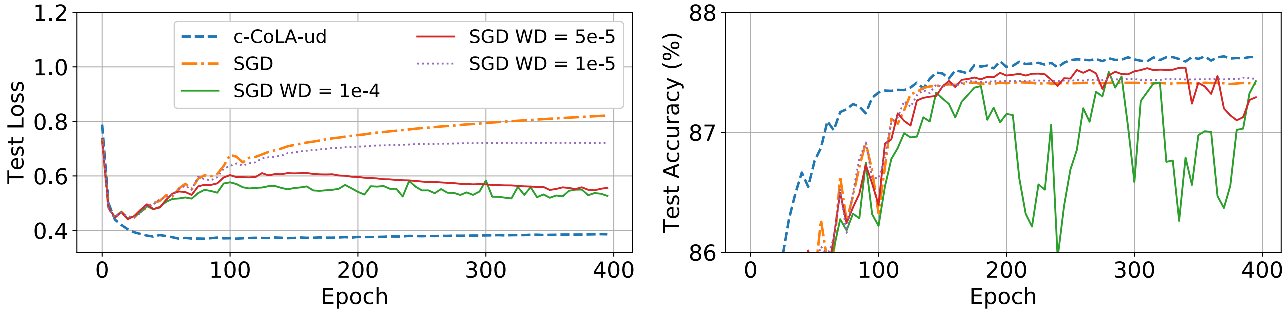

*Figure 7.* Test loss (left) and test accuracy (right) averaged over 5 runs of a 1000-node SHLP trained using SGD-m vs. c-CoLA-ud on Fashion-MNIST (batchsize 128, number of training data samples reduced to 10K). After a line search we chose the best performing hyperparameter setting for SGD, namely $h = 0.1, mom = 0.8$, and varied the amount of weight decay (WD). Standard deviations are provided in the supplement. Hyperparameters c-CoLA-ud: $h = 0.3, \gamma = 1, r_0 = 0.05, r_1 = 0.1, \tau = 0$. Due to the small training dataset size both methods quickly reached 100% training accuracy, but c-CoLA-ud is superior in its test loss and test accuracy.

## References

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pp. 1120–1128, 2016.

Bakry, D. and Émery, M. Diffusions hypercontractives. In Azéma, J. and Yor, M. (eds.), *Séminaire de Probabilités XIX 1983/84*, pp. 177–206, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. ISBN 978-3-540-39397-9.

Bakry, D., Gentil, I., and Ledoux, M. *Analysis and geometry of Markov diffusion operators*, volume 348. Springer Science & Business Media, 2013.

Bansal, N., Chen, X., and Wang, Z. Can we gain more from orthogonality regularizations in training deep CNNs? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4266–4276, 2018.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 2013.

Bhattacharya, R. N. On the functional central limit theorem and the law of the iterated logarithm for Markov processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 60(2):185–201, 1982.

Bou-Rabee, N. and Sanz-Serna, J. Geometric integrators and the Hamiltonian Monte Carlo method. *Acta Numerica*, 27:113–206, 2018.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. J. Neural photo editing with introspective adversarial networks. *ICLR*, 2017.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-SGD: Biasing gradient descent into wide valleys. *ICLR*, 2017.

Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. Underdamped Langevin MCMC: A non-asymptotic analysis. *arXiv:1707.03663*, 2017.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. *AISTATS*, 2015.

d'Ascoli, S., Sagun, L., Bruna, J., and Biroli, G. Finding the needle in the haystack with convolutions: on the benefits of architectural bias. *NeurIPS*, 2019.

Faou, E. and Lelièvre, T. Conservative stochastic differential equations: Mathematical and numerical analysis. *Mathematics of computation*, 78(268):2047–2074, 2009.

Graham, M. and Storkey, A. Asymptotically exact inference in differentiable generative models. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 499–508, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hoerl, A. and Kennard, R. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12: 55–67, 1970.

Huang, L., Liu, X., Lang, B., Wei Yu, A., and Li, B. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in SGD. *ICANN*, 2018.

Jia, K., Li, S., Wen, Y., Liu, T., and Tao, D. Orthogonal deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Kawaguchi, K. Deep learning without poor local minima. *NeurIPS*, 2016.

Keskar, N., Mudigere, D., Nocedal, J., and M. Smelyanskiy, P. T. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.

Kipnis, C. and Varadhan, S. R. S. Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions. *Communications in Mathematical Physics*, 104(1):1–19, 1986.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Leimkuhler, B. and Matthews, C. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Interdisciplinary Applied Mathematics. Springer, 2015.

Leimkuhler, B. and Matthews, C. Efficient molecular dynamics using geodesic integration and solvent–solute splitting. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2189): 20160138, 2016.

Leimkuhler, B. and Reich, S. *Simulating Hamiltonian dynamics*, volume 14. Cambridge university press, 2004.

Leimkuhler, B., Matthews, C., and Stoltz, G. The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. *IMA Journal of Numerical Analysis*, 36(1):13–79, 2016.

Leimkuhler, B., Matthews, C., and Vlaar, T. Partitioned integrators for thermodynamic parameterization of neural networks. *Foundations of Data Science*, 1(4):457–489, 2019.

Lelièvre, T. and Stoltz, G. Partial differential equations and stochastic methods in molecular dynamics. *Acta Numerica*, 25:681–880, 2016.

Lelièvre, T., Stoltz, G., and Rousset, M. *Free energy computations: A mathematical perspective*. Imperial College Press, 2010. ISBN 9781848162488.

Lelièvre, T., Rousset, M., and Stoltz, G. Langevin dynamics with constraints and computation of free energy differences. *Mathematics of computation*, 81(280):2071–2125, 2012.

Lelièvre, T., Stoltz, G., and Zhang, W. Multiple projection MCMC algorithms on submanifolds. *arXiv:2003.09402*, 2020.

Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R., and Jacobsen, J. Preventing gradient attenuation in Lipschitz constrained convolutional networks. *NeurIPS*, 2019.

Loshchilov, I. and Hutter, F. Stochastic gradient descent with warm restarts. *ICLR*, 2017.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Marinari, E. and Parisi, G. Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, 1992.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *ICLR*, 2017.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318, 2013.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. 2017.

Pavliotis, G. A. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014.

Pennington, J., Schoenholz, S., and Ganguli, S. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems*, pp. 4785–4795, 2017.

Pennington, J., Schoenholz, S., and Ganguli, S. The emergence of spectral universality in deep networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1924–1932, 2018.

Persson, P. The level set method. Lecture notes MIT 16.920J / 2.097J / 6.339J, Numerical Methods for Partial Differential Equations, October 2006.

Rodríguez, P., Gonzàlez, J., Cucurull, G., Gonfaus, J. M., and Roca, X. Regularizing CNNs with locally constrained decorrelations. *ICLR*, 2017.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120*, 2013.

Smith, L. N. Cyclical learning rates for training neural networks. *Worshop on Application of Computer Vision*, 2017.

Srebro, N. and Shraibman, A. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, pp. 545–560. Springer, 2005.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. *ICML*, 2013.

Tibshirani, R. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Villani, C. Hypocoercivity. *Memoirs of the American Mathematical Society*, 202(950), 2009.

Vorontsov, E., Trabelsi, C., Kadoury, S., and Pal, C. On orthogonality and learning recurrent networks with long term dependencies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3570–3578. JMLR. org, 2017.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011.

Wenzel, F., Roth, K., Veeling, B. S., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. How good is the Bayes posterior in deep neural networks really? *arXiv:2002.02405*, 2020.

Williams, P. Bayesian regularization and pruning using a Laplace prior. *Neural computation*, 7(1):117–143, 1995.

Wilson, A., Roelofs, R., Stern, M., Srebro†, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. *NeurIPS*, 2017.

Wu, L., Zhu, Z., and E, W. Towards understanding generalization of deep learning: Perspective of loss landscapes. *ICML*, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pp. 5393–5402, 2018.

Xie, D., Xiong, J., and Pu, S. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6176–6185, 2017.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. PyHessian: Neural networks through the lens of the Hessian. *arXiv:1912.07145*, 2019.

Zappa, E., Holmes-Cerfon, M., and Goodman, J. Monte Carlo on manifolds: Sampling densities and integrating functions. *Communications on Pure and Applied Mathematics*, 71(12):2609–2647, 2018.

Zhang, S., Choromanska, A., and LeCun, Y. Deep learning with elastic averaging SGD. *NeurIPS*, 2015.

Zhou, J., Do, M., and Kovacevic, J. Special paraunitary matrices, Cayley transform, and multidimensional orthogonal filter banks. *IEEE Transactions on Image Processing*, 15(2):511–519, 2006.