ON THE EFFECT OF STOCHASTIC NOISE ON KERNEL – TARGET ALIGNMENT: AN NTK ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural Tangent Kernel (NTK) has emerged as an important framework to understand the behavior of neural networks. An important notion of the kernel is *kernel–target alignment*, which measures how much the top eigenfunctions of NTK are aligned to the target function. In a rich feature learning regime, it is observed that NTK evolves to increase the kernel–target alignment. Recent work has shown that this particular evolution is prominent during the edge of stability (EoS) from a large learning rate of gradient descent (GD). In this work, we examine the evolution of NTK under stochastic gradient descent (SGD). Across networks of various architectures, we observe that stochastic noise (e.g. from a small batch size of SGD) can induce larger kernel–target alignment. Theoretically, in a toy example of a diagonal neural network, we show that stochastic noise can selectively suppress weakly target-aligned modes and thereby increase the kernel–target alignment. We also connect such mode suppression to low-rank bias of SGD. Empirically, we observe that low-rank bias is one of the elements that are correlated with the kernel–target alignment. However, we also find that rank reduction alone does not determine kernel–target alignment; rather, the effect depends critically on whether the remaining spectral directions align to the target.

1 INTRODUCTION

Understanding the theoretical foundations underlying deep learning has long been a central challenge for the research community. Among the various theoretical frameworks, one framework that has been in the spotlight is the framework of *Neural Tangent Kernel* (Jacot et al., 2018; Arora et al., 2019c). This framework focuses on infinitely wide neural networks with a specific initialization. Under gradient flow, an infinitely wide network falls into the so-called *lazy-training* regime (Chizat et al., 2019) and evolves as a linearized model in parameter space (Lee et al., 2019; Liu et al., 2020). Here, the linearized model is equivalent to a kernel regression solution with a kernel called Neural Tangent Kernel (NTK). In the lazy-training regime, NTK is deterministic at initialization and remains unchanged during training, which makes the convergence analysis in function space much easier and explainable (Cao et al., 2021; Tancik et al., 2020).

Not only convergence, but also generalization can be understood in a kernel sense. An important notion of the kernel connected to generalization is the *kernel–target alignment*, which measures how much the top eigenfunctions of the kernel are aligned to the target function. Bordelon et al. (2020); Canatar et al. (2021) show that kernel methods can generalize better when the top eigenfunctions of the kernel compose the target function. This result can be extended to neural networks via NTK in regimes where the trained network is well described as a kernel method (Arora et al., 2019b). Arora et al. (2019b) propose a generalization bound for networks in a lazy regime, with a dominant term as the alignment between inverse NTK and the ground-truth labels. The proposed bound can be improved when the top eigenfunctions of NTK are aligned to the target function; in other words, the kernel–target alignment is higher.

While these works mainly studied neural networks in a lazy-training regime with constant NTK, recent works have pointed out the limitations of the lazy-training regime and the assumption on constant NTK (Allen-Zhu & Li, 2019; Vyas et al., 2022). In the rich feature learning regime, into which most finite-width networks fall, NTK evolves significantly during training. Still, it is important to track the evolution of NTK since a broad class of neural networks in the small-initialization

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

regime can be approximately viewed as kernel regression solutions of the evolved data-dependent NTK (Atanasov et al., 2022). In previous works, an interesting phenomenon has been observed: the evolution of NTK and its alignment with the target are closely related (Wang et al., 2023). It is observed that NTK evolves to increase the kernel–target alignment (Ortiz-Jiménez et al., 2021; Baratin et al., 2021), and the top eigenvectors of NTK of the training set align with the ground-truth labels (Kopitkov & Indelman, 2020).

The aforementioned prior works study the evolution of NTK under gradient flow or gradient descent (GD) with a small learning rate. On the other hand, a recent work (Jiang et al., 2026) focuses on the setting of GD with a large learning rate, studying the network in an edge-of-stability (EoS) regime (Cohen et al., 2021). This work observes that a larger learning rate induces the evolved NTK to have a higher kernel–target alignment.

While Jiang et al. (2026) analyze the setting of GD, we examine the evolution of NTK under stochastic gradient descent (SGD), which is more commonly used in practice. We study how the stochastic noise affects the evolution of NTK and the kernel–target alignment. While SGD and GD in EoS are similar in a sense that they both induce oscillatory parameter dynamics, we observe that a different mechanism impacts the evolution of NTK under SGD. Our main findings are as follows:

- Across neural networks of various architectures (ResNet-18 and tiny ViT), we observe that sufficiently strong stochasticity, induced by smaller batch sizes or larger learning rates, can increase the kernel–target alignment of NTK.
- We theoretically illustrate how stochastic noise can induce a higher kernel–target alignment of NTK. Using a label-noise diffusion model, we analyze how stochastic noise can selectively suppress weakly target-aligned modes and thereby increase kernel–target alignment in a toy example of a diagonal neural network.
- One important inductive bias of SGD is the low-rank bias (Wang & Jacot, 2024). We first recall that penalizing the trace of the Hessian can induce both a low-rank bias and larger kernel–target alignment in a two-layer linear neural network. Empirically, we conduct experiments on a two-layer ReLU neural network to inspect the relationship between the low-rank bias and the kernel–target alignment. We observe that the low-rank bias of SGD is one factor correlated with the kernel–target alignment. However, we also reveal that the effect is governed not simply by rank reduction, but by which spectral directions are suppressed relative to the target.

2 Background

2.1 Neural Tangent Kernel and Kernel – Target Alignment

Here, we briefly introduce notations and notions of Neural Tangent Kernel (Jacot et al., 2018) and kernel–target alignment. We compose the training set $\mathcal{D}$ with input data $\mathcal{X} = \{x \in \mathbb{R}^d | (x, y) \in \mathcal{D}\}$ and ground-truth labels $\mathcal{Y} = \{y \in \mathbb{R} | (x, y) \in \mathcal{D}\}$. A neural network $f : \mathcal{X} \to \mathcal{Y}$ with parameters θ is trained on the training set. We consider supervised learning where the network is trained under the loss function $\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} l(f(x, \theta), y)$. A common theoretical choice to minimize loss would be gradient descent, which can be viewed as a first-order Euler method for solving an ODE called the *gradient flow*. Based on the gradient flow, one can describe the evolution of the parameters and the output of the neural network as:

$$\dot{\theta} = -\eta \nabla_\theta \mathcal{L} = -\eta \nabla_\theta f(\mathcal{X}) \nabla_f \mathcal{L} \tag{1}$$

$$\dot{f}(\mathcal{X}) = -\eta \nabla_\theta f(\mathcal{X})^\top \nabla_\theta f(\mathcal{X}) \nabla_f \mathcal{L} := -\eta K_t(\mathcal{X}, \mathcal{X}) \nabla_f \mathcal{L} \tag{2}$$

where $f(\mathcal{X}) \in \mathbb{R}^{|\mathcal{D}|\times 1}$ denotes a concatenation of outputs from all training samples. The kernel function $K_t(x, x') := \nabla_\theta f(x)^\top \nabla_\theta f(x')$ is called *Neural Tangent Kernel* (NTK).

We first study the case of neural networks with a large width or a large initialization scale of parameters. In this case, a network falls into a regime called the *lazy-training* regime (Chizat et al., 2019), where the parameters hardly move from the initialized point. Also, NTK hardly changes and remains deterministic at initialization, $K_0(\mathcal{X}, \mathcal{X})$. Under MSE loss, $l(f, y) = \frac{1}{2}\|f - y\|_2^2$, the training dynamics become linearized. With such a linearized dynamics, the network can be expressed as

a kernel method when the initial network output is zero (Lee et al., 2019):

$$f(x) = K_0(x, \mathcal{X})K_0^{-1}(\mathcal{X}, \mathcal{X})\mathcal{Y}. \tag{3}$$

Since the solution depends on the kernel, NTK is an important element for generalization of a network. It is known that kernel methods tend to generalize better when the target function is well aligned with top eigenfunctions of the kernel (Bordelon et al., 2020; Canatar et al., 2021). This perspective extends to neural networks through the NTK framework, which provides a kernel-based description of their learning dynamics (Arora et al., 2019b). Arora et al. (2019b) propose a generalization bound for networks in a lazy regime, with a dominant term as the alignment between inverse NTK and the ground-truth labels. The generalization bound improves when the top eigenfunctions of NTK are aligned to the target function.

To measure the alignment of the kernel to the target function, we use the following notion:

Definition 2.1. (Kernel–target alignment) Given NTK as $K(\mathcal{X}, \mathcal{X}) \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{D}|}$ and the ground-truth labels as $\mathcal{Y} \in \mathbb{R}^{|\mathcal{D}|\times 1}$, the *kernel–target alignment* is defined as (Cortes et al., 2012):

$$\mathcal{A}(K, \mathcal{Y}\mathcal{Y}^\top) := \frac{\langle K, \mathcal{Y}\mathcal{Y}^\top\rangle}{\|K\|_F\|\mathcal{Y}\|_2^2}. \tag{4}$$

In a discrete sense, the kernel–target alignment of the training set measures how much top eigenfunctions of NTK are aligned to the target function.

On the other hand, in a usual case where the network has a finite width or small initialization, the network falls into a *rich feature learning* regime. Here, NTK $K_t(\mathcal{X}, \mathcal{X})$ significantly evolves. Still, it is important to track the evolution of NTK since a large class of networks can be approximated as a kernel method up to an error dependent on initialization scale (Atanasov et al., 2022). Under small initialization and a proper assumption on the NTK evolution from Atanasov et al. (2022), a network with a homogeneous activation function approximately evolves as a kernel method:

$$f(x) \approx K_\infty(x, \mathcal{X})K_\infty^{-1}(\mathcal{X}, \mathcal{X})\mathcal{Y}. \tag{5}$$

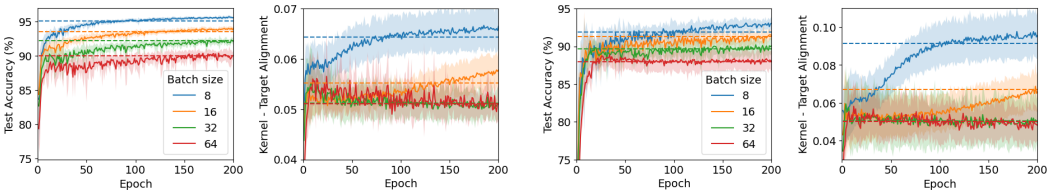
2.2 Low-rank bias of SGD

While it is easy to conceive the idea that the stochastic noise would hamper the training of neural networks, it is empirically observed that SGD often improves generalization of the network (LeCun et al., 2002; Keskar et al., 2016; Smith & Le, 2018). Based on this observation, many works have sought to analyze the implicit bias of SGD (Smith et al., 2021). One line of work studies the implicit bias of SGD toward low-rank weight matrices. Low-rank bias has also been observed without stochastic noise, where L_2 regularization (Dai et al., 2021) and small initialization (Arora et al., 2018; 2019a; Jacot et al., 2021; Li et al., 2021) have been identified as possible mechanisms.

Meanwhile, Wang & Jacot (2024) pointed out that stochastic noise of SGD can strengthen the low-rank bias of the network. This work studies L_2-regularized deep linear networks for matrix completion and shows that SGD strengthens the low-rank bias through one-way transitions: it can move from rank-overestimating critical points to lower-rank regions, while these regions are absorbing. This bias can improve prediction of unobserved matrix entries when the target is low rank. In this work, we study the relationship between the low-rank bias of SGD and the kernel–target alignment.

2.3 Stochastic Collapse

The generalization properties of SGD are often attributed to the simplicity bias of SGD, including the aforementioned works on the low-rank bias. Meanwhile, there is another line of work that studies how SGD makes the network compressible and simple: *stochastic collapse* (Chen et al., 2023). This work shows how stochastic noise of SGD attracts training dynamics toward *invariant sets*, parameter subsets that are impossible to escape once reached under SGD. Sign-induced invariant sets arise when a neuron's incoming parameters and outgoing parameters vanish, effectively yielding a sparser subnetwork. Permutation-induced invariant sets arise when two neurons in the same layer acquire identical incoming and outgoing parameters, effectively yielding a low-rank subnetwork. This work derives conditions for these sets to become *stochastically attractive* and drag nearby parameters toward themselves. Our theoretical analysis builds on this phenomenon, but focuses on how noise-induced suppression of modes reshapes the NTK spectrum and the kernel–target alignment.



(a) CIFAR-10 (b) Imagenette

Figure 1: Experiments of ResNet-18 trained on CIFAR-10 (left column) and Imagenette (right column), varying the batch size as $B \in \{8, 16, 32, 64\}$ with a fixed learning rate of 0.001. (i) Left: test accuracy (%), (iii) Right: kernel–target alignment of NTK. The dotted lines denote the test accuracy and the kernel–target alignment when the training loss is identical across batch sizes. The kernel–target alignment of NTK increases when the batch size falls below a certain threshold, even when models were compared at the same training loss. The test accuracy increased with a smaller batch size and a larger kernel–target alignment.

3 Empirical observations

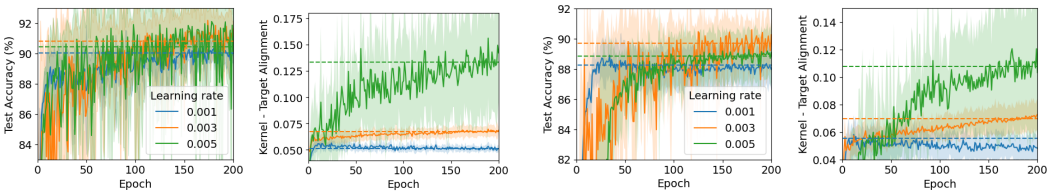
In this section, we investigate whether stochasticity affects the kernel–target alignment of NTK. As an ablation study, we report test accuracy to examine whether the resulting alignment is associated with generalization, although this relationship need not hold across architectures. We train non-linear networks that are relatively complex but small enough to feasibly compute empirical NTKs: a ResNet-18 (He et al., 2016) and a tiny ViT (Dosovitskiy et al., 2021) of depth-6. We train networks on two datasets: CIFAR-10 (Krizhevsky et al., 2009) and the 160px-version of Imagenette (Howard, 2019; Deng et al., 2009). Since we want to analyze networks of scalar outputs, we extract the samples of two ground-truth labels (0 – 1) from the datasets and train the networks on the extracted sample sets. The ground-truth labels are modified to symmetric binary labels of either -1 or $+1$.

To inspect the effect of stochastic noise, we conduct experiments varying two factors: batch size and learning rate. For all datasets and architectures, the networks were trained with either (i) batch sizes of $B \in \{8, 16, 32, 64\}$ with a fixed learning rate of 0.001 or (ii) learning rates of $\eta \in \{0.001, 0.003, 0.005\}$ with a fixed batch size of 64. Since the kernel–target alignment is mostly studied for MSE loss, we also use MSE loss for the training of the networks. To remove the influence of other factors, the networks are trained using SGD with no momentum and no random data augmentation. We use a small weight decay of $1e-4$. Empirical NTKs are measured from 200 randomly-sampled data points from the training set. The networks are trained until the training accuracy hits 100%, and the plots describe the means and standard deviations of total 10 runs.

Batch size. Figure 1 shows the result for ResNet-18 when the batch size of SGD was varied. As a result, the kernel–target alignment of NTK increased for batch sizes below a certain threshold ($B \leq 16$) at later epochs, with the smallest batch size exhibiting the highest alignment. This trend remained consistent even when models were compared at the same training loss (denoted by dotted lines): sufficiently small batch sizes yielded higher kernel–target alignment.

Learning rate. Figure 2 shows the results for ResNet-18 obtained by varying the learning rate of SGD. As a result, the kernel–target alignment of NTK increased once the learning rate increased above a certain threshold ($\eta \geq 0.003$). The same pattern held at matched training loss, with kernel–target alignment increasing when the learning rate entered a sufficiently large regime. Thus, sufficiently strong mini-batch noise was associated with higher kernel–target alignment.

Ablation study on generalization. In previous works (Arora et al., 2019b; Atanasov et al., 2022), the kernel–target alignment has been connected to generalization of a network. Here – though this lies tangential to our central argument – we observe the correlation between the kernel–target alignment and generalization. For ResNet-18, the test accuracy increased with a smaller batch size, and sufficiently small batch sizes yielded both higher test accuracy and larger kernel–target alignment. Even when models were compared at the same training-loss level (denoted by dotted lines), smaller batch sizes yielded higher test accuracy. For large learning rates, the optimization of the network



(a) CIFAR-10 (b) Imagenette

Figure 2: Experiments of ResNet-18 trained on CIFAR-10 (left column) and Imagenette (right column), varying the learning rate as $\eta \in \{0.001, 0.003, 0.005\}$ with a fixed batch size of 64. (i) Left: test accuracy (%), (iii) Right: kernel–target alignment of NTK. The dotted lines denote the test accuracy and the kernel–target alignment when the training loss is identical across learning rates. The kernel–target alignment of NTK increases when the learning rate increases above a certain threshold, even when models were compared at the same training loss.

was unstable, and the test accuracy exhibited oscillations throughout training. Still, for ResNet-18, the average test accuracy showed a modest improvement as the learning rate entered a sufficiently large regime, although the improvement was not monotonic with the learning rate. The trend was the same at matched training loss: thus, sufficiently large learning rates were associated with both higher kernel–target alignment and improved test performance, although their relationship was not strictly monotonic. Meanwhile, ViT exhibited the same trend on kernel–target alignment but a different relationship between alignment and generalization, as discussed in Appendix A.2.

4 Toy example on a two-layer diagonal neural network

We next study a simple mechanism by which stochastic noise can induce higher kernel–target alignment. In a two-layer diagonal linear network, we show that stochasticity can selectively suppress modes according to their target signal strength. This mechanism builds on the framework of stochastic collapse (Chen et al., 2023), where noise attracts dynamics toward sign-induced invariant sets. On the training set $\mathcal{D}$ with input data $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d | (\boldsymbol{x}, y) \in \mathcal{D}\}$ and ground-truth labels $\mathcal{Y} = \{y \in \mathbb{R} | (\boldsymbol{x}, y) \in \mathcal{D}\}$, a neural network $f_{\boldsymbol{a},\boldsymbol{b}} : \mathcal{X} \to \mathcal{Y}$ with parameters $\boldsymbol{a}, \boldsymbol{b}$ is trained on the training set. The assumed diagonal network can be expressed as below when feature vectors v_i are fixed and orthonormal ($\boldsymbol{v}_i \perp \boldsymbol{v}_j$ for $i \neq j$, $\|\boldsymbol{v}_i\|_2 = 1$), which simulates linear neural networks with little rotation of singular weight vectors:

$$f_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x}) = \sum_{i=1}^{d} a_i b_i \phi_i(\boldsymbol{x}), \quad \phi_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{v}_i \tag{6}$$

We assume a supervised learning case where the network is trained under MSE loss, $l(f, y) = \frac{1}{2}\|f - y\|_2^2$. For simplicity, we assume that the target is an output of a linear teacher β: $y = \beta^\top \boldsymbol{x}$.

We now introduce stochastic perturbations into the optimization dynamics. Here, following Chen et al. (2023), we model stochasticity by perturbing gradient flow through *label noise*. We first analyze label noise as a tractable proxy for mini-batch noise, then next extend to the case of mini-batch noise for the early dynamics of training. For ease of analysis, we assume the following conditions:

Assumption 4.1. (Whitened input) The inputs are whitened as $\Sigma_{xx} := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right] = I$.

Assumption 4.2. (Structural label noise) At each update, the target is perturbed by fresh Gaussian noise as $y \mapsto y + \xi$ when $\xi \sim \mathcal{N}(0, \zeta^2)$.

Assumption 4.3. (Balanced initialization) For every $i \in [d]$, the initial weights of a_i and b_i are balanced as $a_i^2 = b_i^2$ at $t = 0$.

Assumption 4.4. (Positive sign) For every $i \in [d]$, the signs of target signal strength $\mu_i := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[y\phi_i(\boldsymbol{x})\right]$ and initial neuron variable $s_i := a_i b_i$ are positive as $\frac{s_i}{|s_i|} = \frac{\mu_i}{|\mu_i|} = 1$ at $t = 0$.

Under these assumptions, training can be viewed as the process of adjusting each neuron variable $s_i := a_i b_i$ to match the corresponding target signal strength $\mu_i := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[y\phi_i(\boldsymbol{x})\right]$. Now the training loss can be expressed as below with an additional constant $\mathcal{C}$:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(y - f_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x})\right)^2 = \frac{1}{2}\sum_{i=1}^{d}(s_i - \mu_i)^2 + \mathcal{C}. \tag{7}$$

The target signal strength μ_i measures how much the feature vector $\boldsymbol{v}_i$ is aligned to the target vector β as $\mu_i = \beta^\top \boldsymbol{v}_i$. We first analyze the label diffusion.

Label diffusion. We study the network trained under gradient descent perturbed with Gaussian label noise. It is equivalent to minimizing $l(f, y) = \frac{1}{2}\|f - y - \xi\|_2^2$ and the gradient update with a learning rate η becomes $\theta^{(t+1)} = \theta^{(t)} - \eta\nabla\mathcal{L}(\theta^{(t)}) - \eta\mathbb{E}\left[\xi\nabla f(\boldsymbol{x})\right]$. Taking the limit as $\eta \to 0$, we can approximate gradient updates with an SDE.

Stochastic dynamics. Now we obtain the gradient of the parameters a_i and b_i with respect to the loss in Equation 7 to compute the deterministic gradient flow. The label diffusion perturbs the gradient flow and the stochastic dynamics of the parameters become as below when $\tau := \eta\zeta^2$:

$$da_i = -b_i(s_i - \mu_i)dt + \sqrt{\tau}b_i dB_{i,t} \tag{8}$$

$$db_i = -a_i(s_i - \mu_i)dt + \sqrt{\tau}a_i dB_{i,t}. \tag{9}$$

Under these dynamics, we find that the balance between a_i and b_i is preserved throughout training. It can be confirmed by checking the evolution of $R_i = a_i^2 - b_i^2$. By Itô's lemma, $dR_i = -\tau R_i dt$ and $R_i = a_i^2 - b_i^2 = 0$ remains during training.

By the balanced norms and positiveness of neuron variables, $s_i = (a_i^2 + b_i^2)/2$ remains, which is justified later. Then, one can obtain the dynamics of the neuron variables s_i by Itô's lemma:

$$ds_i = 2s_i(\mu_i + \frac{\tau}{2} - s_i)dt + 2\sqrt{\tau}s_i dB_{i,t} \tag{10}$$

$$d\log s_i = (2\mu_i - \tau - 2s_i)dt + 2\sqrt{\tau}dB_{i,t} \tag{11}$$

Here, it is safe to keep s_i positive as $(a_i^2 + b_i^2)/2$ since the solution of the SDE remains positive:

$$s_i(t) = \frac{\exp((2\mu_i - \tau)t + 2\sqrt{\tau}B_{i,t})}{s_i(0)^{-1} + 2\int_0^t \exp((2\mu_i - \tau)u + 2\sqrt{\tau}B_{i,u})}. \tag{12}$$

Here, the assumptions on the positive signs of s_i and μ_i can be easily relaxed to an assumption on the identical signs of s_i and μ_i since their signs do not change throughout training.

Crucially, by Equation 11, if s_i is sufficiently small, s_i is also *attracted to zero* when $\mu_i < \tau/2 = \eta\zeta^2/2$. This shows that the neuron vanishes if the target signal strength of the neuron, μ_i, is weaker than a certain threshold, $\eta\zeta^2/2$, which is dependent on the noise scale and the learning rate.

NTK dynamics. We now analyze NTK of the diagonal linear network. By aggregating the derivatives of the parameters, NTK of the network can be expressed as

$$K_t(\boldsymbol{x}, \boldsymbol{x}') = \sum_i \left[\frac{\partial f}{\partial a_i}(\boldsymbol{x})\frac{\partial f}{\partial a_i}(\boldsymbol{x}') + \frac{\partial f}{\partial b_i}(\boldsymbol{x})\frac{\partial f}{\partial b_i}(\boldsymbol{x}')\right] = \sum_i 2s_i(t)\phi_i(\boldsymbol{x})\phi_i(\boldsymbol{x}'). \tag{13}$$

Under the whitened-input assumption, $2s_i(t)$ is the NTK eigenvalue associated with the eigenfunction $\phi_i(x)$ because $\boldsymbol{v}_i$'s are orthonormal and $\phi_i(x)$ becomes an eigenfunction under whitened data.

Since s_i is suppressed if $\boldsymbol{v}_i$ is not aligned to β with weak target signal strength μ_i, the eigenvalue $2s_i$ is also suppressed if the eigenfunction $\phi_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{v}_i$ is not aligned to the target $y = \beta^\top \boldsymbol{x}$. Thus, as the noise scale increases, eigenfunctions with weak target alignment are suppressed from the NTK spectrum. This can make the kernel–target alignment of NTK larger under certain conditions:

Proposition 4.5. *The kernel–target alignment of NTK for the diagonal network in Equation 6 is*

$$\mathcal{A}(s, \mu) = \frac{\sum_i s_i \mu_i^2}{\sqrt{\sum_i s_i^2}\sum_i \mu_i^2}. \tag{14}$$

Now let $\mathbf{C}$ *denote the indices of neuron variables that are suppressed to zero as* $s_j = 0$ *for* $j \in \mathbf{C}$*. If* μ_j*'s for* $j \in \mathbf{C}$ *are sufficiently small, the following holds and the kernel–target alignment increases:*

$$\sum_{j\in\mathbf{C}} s_j\mu_j^2 < \left(\sqrt{1 + \frac{\sum_{j\in\mathbf{C}} s_j^2}{\sum_{i\notin\mathbf{C}} s_i^2}} - 1\right)\sum_{i\notin\mathbf{C}} s_i\mu_i^2. \tag{15}$$

Thus, if stochastic noise is moderate and only suppresses feature vectors with weak target alignment, it can increase the kernel–target alignment of NTK. One thing to be warned is that, if stochastic noise is too large and it suppresses the directions with large target alignment, the kernel–target alignment of NTK can decrease. Hence, the relationship between stochastic noise and the kernel–target alignment is not strictly monotone.

Extension to mini-batch noise. While the derivation above uses label noise, the local stochastic dynamics can be connected to mini-batch noise when the parameters are initialized sufficiently small as $f_{\boldsymbol{a},\boldsymbol{b}}(x) \approx 0$. Consider an SGD update with a mini-batch $\mathcal{B}$ of size B. The initial local SGD dynamics become $a_i^{(t+1)} = a_i^{(t)} + \eta b_i^{(t)}(\mu_i + \epsilon_{i,\mathcal{B}})$, $b_i^{(t+1)} = b_i^{(t)} + \eta a_i^{(t)}(\mu_i + \epsilon_{i,\mathcal{B}})$ with $\mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}] = 0$. For independent sampling, the variance of the effective mini-batch noise is

$$\mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}^2] = \frac{1}{B}\operatorname{Var}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y\phi_i(\boldsymbol{x})] = \frac{1}{B}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})^2(\boldsymbol{x}^\top \boldsymbol{v}_i)^2\right] - \mu_i^2\right), \tag{16}$$

We further assume isotropic Gaussian inputs as $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$. Aligning the mini-batch noise variance with label-noise diffusion yields the effective diffusion scale for mode i as

$$\tau_i := \eta\mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}^2] = \frac{\eta}{B}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})^2(\boldsymbol{x}^\top \boldsymbol{v}_i)^2\right] - \mu_i^2\right) = \frac{\eta}{B}\left(\|\beta\|_2^2 + \mu_i^2\right). \tag{17}$$

For modes with weak target alignment as $\mu_i^2 \ll \|\beta\|_2^2$, the effective noise scale remains approximately $\tau_i \approx \eta\|\beta\|_2^2/B$ as the target signal strength μ_i becomes small. Therefore, under a covariance-matched local diffusion approximation, such modes with weak target alignment are relatively more susceptible to suppression if the batch size is small and noise scale increases. Unlike the label-noise model, usual mini-batch noise may additionally contain correlations across different modes, thus, this correspondence should be understood as a local mode-wise approximation.

Implications. In this example, stochastic noise drives SGD closer to a sign-induced invariant set. Increasing the effective noise scale preferentially suppresses feature directions with weak target alignment, thereby reducing their contribution to the network output. Such a suppression will regularize the complexity of the network and prevent overfitting in certain directions (Chen et al., 2023). We also empirically analyze the case of a two-layer ReLU network and observe how suppression and amplification of modes appear for non-linear networks in Appendix A.4.

One thing to note is that the aforementioned feature directions are identical to the eigenfunctions of NTK. Hence, stochastic noise removes the eigenfunctions with weak target alignment. Such pruning can increase the kernel–target alignment when the suppressed modes have sufficiently weak target alignment. In other words, the kernel–target alignment is expected to exhibit *threshold-like behavior*, increasing once the noise scale exceeds a certain threshold, which is consistent with the results on ResNet-18. Though a diagonal linear network is not identical to the complex networks used in previous experiments, this simplified example illustrates a mechanism by which stochastic noise can increase the kernel–target alignment. This differs from the previous works such as Chen et al. (2023), where, under the restricted spectral initialization, a scalar-output linear network is rank-one and collapses entirely, rather than selectively suppressing weakly target-aligned modes in a way that can increase kernel–target alignment.

5 Low-rank bias and kernel – target alignment

The previous section showed that suppressing selected modes can increase the kernel–target alignment. This observation naturally connects to the known low-rank bias of neural network optimization, in the sense that low-rank bias also suppresses certain spectral directions of weights (Dai et al., 2021; Arora et al., 2018; 2019a; Jacot et al., 2021; Li et al., 2021). This connection could be interesting for SGD since, with weight decay, SGD strengthens low-rank bias (Wang & Jacot, 2024).

We illustrate an extensive perspective on how suppression of spectral directions due to low-rank bias can be connected to kernel–target alignment. We assume a two-layer linear neural network:

$$f_{\boldsymbol{a},\boldsymbol{W}}(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{W}\boldsymbol{x}, \quad \boldsymbol{a} \in \mathbb{R}^N, \quad \boldsymbol{W} \in \mathbb{R}^{N\times d} \tag{18}$$

trained for the linear target function $y = \beta^\top \boldsymbol{x}$ under MSE loss. As done in the previous section, we assume that the inputs are whitened as $\Sigma_{\boldsymbol{x}\boldsymbol{x}} := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right] = I$.

Now we consider an important element of SGD: leading the parameters to flat minima (Keskar et al., 2016; Jastrzkebski et al., 2017; Zhu et al., 2019). As an illustrative proxy for flatness, we fix the effective predictor as the target vector ($\boldsymbol{W}^\top \boldsymbol{a} = \beta$) and obtain the solution that minimizes the sharpness of the loss surface. Here, we use the *trace of Hessian* as the measurement of sharpness since it is the average curvature under isotropic parameter perturbations.

$$\min_{\boldsymbol{a},\boldsymbol{W}} \operatorname{tr}\boldsymbol{H} = \|\boldsymbol{W}\|_F^2 + d\|\boldsymbol{a}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{W}^\top \boldsymbol{a} = \beta \tag{19}$$

This can be a useful problem setting, since networks under label noise SGD are known to first fall into a manifold of interpolating solutions and then progressively decrease the trace of the Hessian (Li et al., 2022), a decrease similarly observed in mini-batch SGD (Li et al., 2020). Any weight $\boldsymbol{W}$ that matches the effective predictor to the target vector can be expressed as

$$\boldsymbol{W} = \frac{\boldsymbol{a}\beta^\top}{\|\boldsymbol{a}\|_2^2} + \boldsymbol{Q}, \quad \boldsymbol{Q}^\top \boldsymbol{a} = 0. \tag{20}$$

Since $\|\boldsymbol{W}\|_F^2 = \frac{\|\beta\|_2^2}{\|\boldsymbol{a}\|_2^2} + \|\boldsymbol{Q}\|_F^2$, $\boldsymbol{W}$ that satisfies the trace minimization will have $\boldsymbol{Q} = 0$. Hence, the weight of the network becomes rank-one, with the singular vector aligned to the target vector β. The corresponding NTK of the two-layer linear neural network is as follows and the first term is aligned to the target when $\boldsymbol{W}$ is aligned to the target vector β.

$$K_t(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \left[\boldsymbol{W}^\top \boldsymbol{W} + \|\boldsymbol{a}\|_2^2 \boldsymbol{I}\right] \boldsymbol{x}' \xrightarrow{\boldsymbol{W} = \frac{\boldsymbol{a}\beta^\top}{\|\boldsymbol{a}\|_2^2}} \frac{y(\boldsymbol{x}) y(\boldsymbol{x}')}{\|\boldsymbol{a}\|_2^2} + \|\boldsymbol{a}\|_2^2 \boldsymbol{x}^\top \boldsymbol{x}' \tag{21}$$

Minimizing the trace of Hessian given the fixed effective predictor can induce the low-rank bias of the network and favor larger kernel–target alignment. Thus, with low-rank bias, certain spectral directions with weak target alignment can be suppressed, which is naturally connected to the kernel–target alignment. Such a viewpoint on rank reduction can describe the evolution of NTK especially when the initialization of parameters is not small. Previous works on the kernel–target alignment (Atanasov et al., 2022; Jiang et al., 2026) rely on the assumption that the initialization scale is very small then justify the reason $\boldsymbol{W}$ becomes rank-1 under deterministic gradient updates. In contrast, this viewpoint on suppression of spectral directions shows how the low-rank evolution becomes viable even when the initialization is moderate, which can be more practical in usual settings.

5.1 Empirical observations

In this section, we empirically investigate the relationship between the kernel–target alignment and suppression of singular directions; in other words, the low-rank bias of SGD. To facilitate direct measurement of weight-matrix rank, we train simple two-layer ReLU neural networks with width of 256. We train networks on CIFAR-10 with extracted samples of binarized labels. To inspect the effect of stochastic noise, we conduct experiments varying the batch size as $B \in \{16, 32, 64, 128\}$ with a fixed learning rate of 0.001.

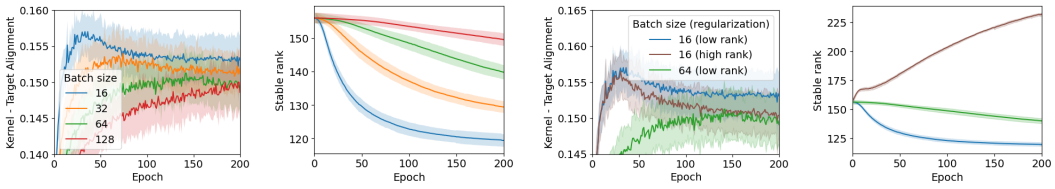
We train the network with two kinds of losses. The baseline objective is the standard MSE loss. The second augments the MSE objective with an orthogonality regularizer on the first-layer weights when the network can be expressed as $f(x) = \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x})$:

$$\mathcal{L}_{orth}(\boldsymbol{W}) = \left\| \frac{\boldsymbol{W}^\top \boldsymbol{W}}{\mathrm{tr}\boldsymbol{W}^\top \boldsymbol{W}} - \frac{1}{d}\boldsymbol{I} \right\|_F^2 \tag{22}$$

while $\boldsymbol{W} \in \mathbb{R}^{N \times d}, \boldsymbol{a} \in \mathbb{R}^N, \sigma(z) = \max(z, 0)$. This regularizer explicitly discourages low-rank solutions by orthogonalizing the first layer weight of the network. This allow us to isolate the role of rank reduction in kernel–target alignment. Since $\mathcal{L}_{orth}$ was small initially, the regularization strength was set as 100.

Low-rank bias of SGD. Figure 3a shows the result on the correlation between the rank and the kernel–target alignment. To measure the effective rank of the first-layer weight, we measured the stable rank, $\|\boldsymbol{W}\|_F^2 / \|\boldsymbol{W}\|_2^2$. As a result, the stable rank of the weight decreased as the batch size was set smaller. Meanwhile, the kernel–target alignment generally increased as the batch size became sufficiently small. Together, these results are consistent with an observation that target-aligned singular directions are preferentially preserved. It also shows that the rank and the kernel–target alignment are moderately correlated when a network is trained without explicit regularization.

Is low-rank bias equivalent to the increase of kernel–target alignment? Now we train the network with an explicit regularization on the rank. We trained the network for batch size of 16 but with the aforementioned regularization. Figure 3b shows the result: when comparing it to the results with the same batch size of 16 but without regularization, the kernel–target alignment decreased. This



(a) Experiments without regularization on the rank.

(b) Experiments with regularization on the rank.

Figure 3: Experiments of a two-layer ReLU neural network trained on CIFAR-10, varying the batch size as $B \in \{16, 32, 64, 128\}$ with a fixed learning rate of 0.001. In each subfigure, (i) Left: kernel–target alignment of NTK, (ii) Right: stable rank of the first-layer weight. The left subfigure shows that the rank of the weight matrices and the kernel–target alignment are correlated when the network was trained without explicit regularization. However, with explicit regularization, the kernel–target alignment of batch size 16 is identical to the one of batch size 64 but with no regularization, while the stable rank of the formal one is much higher. This shows that the low-rank bias is a phenomenon distinct from the increase in the kernel–target alignment.

again shows the correlation between the suppression of singular directions and the kernel–target alignment. One thing to note was that, at later epochs, the resulting kernel–target alignment from explicit regularization was nearly identical to that obtained with batch size of 64 without regularization. In contrast, the stable rank of batch size 16 with regularization was much higher than the stable rank of batch size 64 without regularization. This shows that the rank is not the only factor of the kernel–target alignment. The alignment is governed not simply by rank reduction, but by the target alignment of remaining spectral directions.

This is plausible since the low rank of weights is not a necessary condition for a higher kernel–target alignment. Recall Equation 20: while $\boldsymbol{Q} = 0$ might increase the kernel–target alignment, the kernel–target alignment can be large if Gram component $\boldsymbol{Q}^\top \boldsymbol{Q}$ has a sufficiently large Rayleigh quotient along the target vector β, relative to its total spectral magnitude. Thus, the increase of kernel–target alignment is a phenomenon that takes one step further from the low-rank bias. While the low-rank bias relates solely to the suppression of specific spectral directions in the network, kernel–target alignment assesses whether the directions surviving this suppression are properly aligned with the target function.

6 Conclusion

In this work, we studied how stochastic noise in SGD affects the kernel–target alignment of the NTK. Across multiple architectures, we empirically found that stronger stochasticity can induce higher kernel–target alignment. To provide a mechanistic explanation, we analyzed a two-layer diagonal linear network and showed that stochastic noise can preferentially suppress modes with weak target alignment, thereby increasing kernel–target alignment. We further investigated the connection to the low-rank bias of SGD. Both our theoretical illustration and empirical results indicate that rank reduction alone does not determine kernel–target alignment; rather, the effect depends critically on whether the remaining directions align to the target. This makes a distinction from previous works on implicit bias of SGD. Since kernel–target alignment is connected to generalization in regimes where neural networks are well described by kernel methods, our results provide a potential mechanism through which the implicit bias of SGD may influence generalization in such regimes.

Limitations. First, we only theoretically analyzed diagonal linear networks as a main example. While the analysis can give a hint on how noise-induced mode suppression affects the kernel–target alignment, the dynamics cannot be exactly applied to more complex networks. Second, while addressing this limitation lies beyond the scope of our central contributions; conventional analyses linking kernel–target alignment to generalization may not directly extend to Transformer-based architectures such as ViTs. Our ViT experiments exhibit a different relationship between kernel–target alignment and test performance, suggesting that the connection between evolved-NTK alignment and network generalization may depend substantially on the architecture and training regime. These are limitations of our analysis, and we expect future works could solve these problems.

FURTHER STATEMENTS

AI DISCLOSURE

In this work, we used generative AI tools to (i) assist in the writing of proofs, (ii) assist with translation, and (iii) design or provide feedback on research methodology or experiments for low-rank bias experiments. We have not used generative AI tools to (i) generate synthetic data sets, (ii) provide critical ingredients for proving mathematical claims, (iii) propose or refine hypotheses, (iv) implement methods, (v) clean and reformat dataset, (vi) support qualitative and thematic data analysis, (vii) interpret results, (viii) help develop theoretical models, and (ix) formulate mathematical claims. Additionally, we used generative AI tools to (i) create or modify scientific figures or images, (ii) summarize or analyze existing literature, (iii) discover research topics or identify gaps, brainstorming, and (iv) edit a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work contributes to the theoretical understanding of stochastic gradient descent by analyzing how stochastic noise influences the evolution of the Neural Tangent Kernel and kernel–target alignment. Such insights may guide the design of more effective and data-efficient optimization algorithms, potentially improving the reliability and generalization of machine learning systems across scientific and engineering applications. From a societal perspective, better generalization may reduce failure rates and improve the robustness of AI models deployed in real-world settings. On the other hand, our work is theoretical in nature and does not introduce new datasets or application-specific technologies. As with most advances in optimization, the resulting methods could also be used to improve models for applications with harmful societal consequences. We therefore encourage responsible deployment consistent with existing ethical guidelines and regulations.

REPRODUCIBILITY STATEMENT

The datasets used in our experiments are fully public, and the architectures of the networks are described in our paper. Please refer to Appendix A.1 for more details on our experimental settings: details on the architecture (e.g. width, depth, dimension, etc.) and the optimization hyperparameters (e.g. optimizer, learning rate, batch size, etc.) are described in the aforementioned section. For theoretical results, the detailed proofs are described in Appendix B along with the required definitions and assumptions; so check the section if one needs a detailed explanation on the theoretical results.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5857d68cd9280bc98d079fa912fd6740-Paper.pdf.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pp. 244–253. PMLR, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019a.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 09–15 Jun 2019b.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019c.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, pp. 2269–2277. PMLR, 2021.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pp. 1024–1034. PMLR, 2020.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1):2914, 2021.

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.

Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. Stochastic collapse: How gradient noise attracts sgd dynamics towards simpler subnetworks. *Advances in Neural Information Processing Systems*, 36:35027–35063, 2023.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

Zhen Dai, Mina Karzand, and Nathan Srebro. Representation costs of linear neural networks: Analysis and design. *Advances in Neural Information Processing Systems*, 34:26884–26896, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet. https://github.com/fastai/imagenette, 2019. fast.ai.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Stanislaw Jastrzkebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Kaiqi Jiang, Jeremy Cohen, and Yuanzhi Li. Understanding the evolution of the neural tangent kernel at the edge of stability. *Advances in Neural Information Processing Systems*, 38:31996–32036, 2026.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Dmitry Kopitkov and Vadim Indelman. Neural spectrum alignment: Empirical study. In *International Conference on Artificial Neural Networks*, pp. 168–179. Springer, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0d1a9651497a38d8b1c3871c84528bd4-Paper.pdf.

Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 190–198. SIAM, 2020.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=AHOs7Sm5H7R.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=siCt4xZn5Ve.

Chaoyue Liu, Libin Zhu, and Misha Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33: 15954–15964, 2020.

TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 34:8998–9010, 2021.

Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Loucas Pillaud Vivien, Julien Reygner, and Nicolas Flammarion. Label noise (stochastic) gradient descent implicitly solves the lasso for quadratic parametrisation. In *Conference on Learning Theory*, pp. 2127–2159. PMLR, 2022.

Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.

Zhichao Wang, Andrew Engel, Anand D Sarwate, Ioana Dumitriu, and Tony Chiang. Spectral evolution and invariance in linear-width neural networks. *Advances in neural information processing systems*, 36:20695–20728, 2023.

Zihan Wang and Arthur Jacot. Implicit bias of sgd in l_2-regularized linear dnns: One-way jumps from high to low rank. In *International Conference on Learning Representations*, volume 2024, pp. 11141–11167, 2024.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *ICML*, pp. 7654–7663, 2019. URL http://proceedings.mlr.press/v97/zhu19e.html.

A Additional Experiments

A.1 Experimental details

Datasets. We trained networks on two datasets: CIFAR-10 (Krizhevsky et al., 2009) and the 160px-version of Imagenette (Howard, 2019; Deng et al., 2009). We extracted the samples of only two ground-truth labels (0 – 1) from the datasets and trained the networks on the extracted sample sets. The ground-truth labels were modified to binary labels of either -1 or $+1$.

Architectures We trained non-linear networks that are relatively complex but small enough to feasibly compute empirical NTKs: a ResNet-18 (He et al., 2016) and a tiny ViT (Dosovitskiy et al., 2021). We used ResNet-18 from the version of torchvision (maintainers & contributors, 2016) while we used our custom ViT. Since empirical NTKs are hard to compute, we reduced the size of ViT: the depth was 6, the dimension of patch embeddings was 192, the number of heads was 3, the head dimension was 64, and the dimension of the feed-forward layer (MLP) was 768. For the experiments on a ReLU neural network, the width was 256 and the depth was 2.

Implementation details. We conducted experiments varying two factors: batch size and learning rate. For all datasets and architectures, the networks were trained with either (i) batch sizes of $B \in \{8, 16, 32, 64\}$ with a fixed learning rate of 0.001 or (ii) learning rates of $\eta \in \{0.001, 0.003, 0.005\}$ with a fixed batch size of 64. We also use MSE loss for the training of the networks. To remove the influence of other factors, the networks were trained on SGD with no momentum and no random data augmentation. The weight decay was set small: $1e-4$. We run training 10 times for 200 epochs for ResNet-18 and 400 epochs for tiny ViT. Tiny ViTs were trained using bfloat16 (BF16) precision. Empirical NTKs were measured from 200 randomly-sampled data points from the training set, and we computed a part of NTK at one time with batch size of 5.

Regularization. In Section 5, to increase the rank of the weights by force, we used a regularization on the orthogonality of the first-layer weight. If the network can be expressed as $f(x) = \boldsymbol{a}^\top \sigma(\boldsymbol{W}\boldsymbol{x})$, the regularization has the form of:

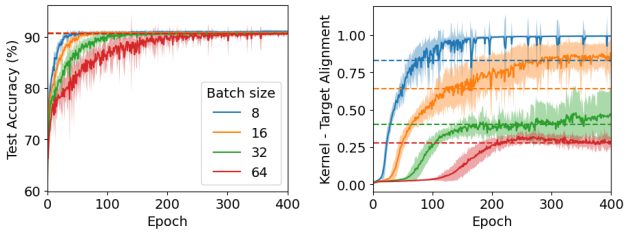
$$\mathcal{L}_{orth}(\boldsymbol{W}) = \left\| \frac{\boldsymbol{W}^\top \boldsymbol{W}}{\mathrm{tr}\boldsymbol{W}^\top \boldsymbol{W}} - \frac{1}{d}\boldsymbol{I} \right\|_F^2 \tag{23}$$

while $\boldsymbol{W} \in \mathbb{R}^{N\times d}, \boldsymbol{a} \in \mathbb{R}^N, \sigma(z) = \max(z, 0)$. Since $\mathcal{L}_{orth}$ was initially small, the regularization strength was set as 100.

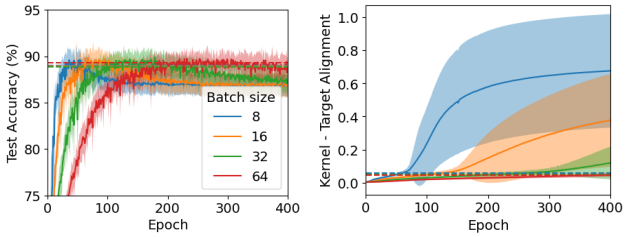
Algorithm 1 Normalized Orthogonality Regularization

Require: Weight tensor W, $\varepsilon > 0$
1: $W \leftarrow \mathrm{Flatten}(W, 1)$ if $\dim(W) > 2$
2: $G \leftarrow WW^\top$ if $\mathrm{rows}(W) \le \mathrm{cols}(W)$, otherwise $G \leftarrow W^\top W$
3: $r \leftarrow \mathrm{rows}(G)$
4: $\widetilde{G} \leftarrow G/(\mathrm{tr}(G) + \varepsilon)$
5: **return** $\mathcal{L}_{\mathrm{orth}} \leftarrow \left\| \widetilde{G} - I_r/r \right\|_F^2$

Training resources. The experiments were run on 2 RTX 3090 GPUs and AMD Ryzen Threadripper PRO 5955WX 16-Cores, for 2 hours for each experiment with ResNet-18, 4 hours for ViT.

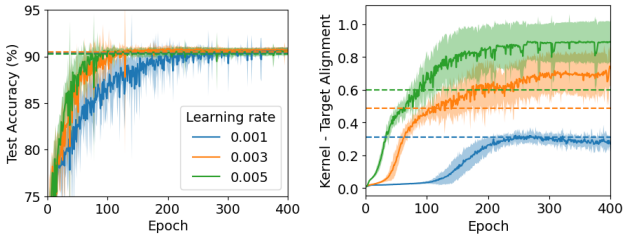


(a) CIFAR-10

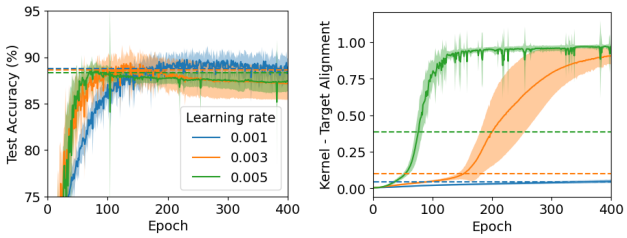


(b) Imagenette

Figure 4: Experiments of a tiny ViT trained on CIFAR-10 (first row) and Imagenette (second row), varying the batch size as $B \in \{8, 16, 32, 64\}$ with a fixed learning rate of 0.001. (i) Left: test accuracy (%), (iii) Right: kernel–target alignment of NTK. The dotted lines denote the test accuracy and the kernel–target alignment when the training loss is identical across batch sizes. The kernel–target alignment of NTK increases when the batch size falls below a certain threshold, even when models were compared at the same training loss.



(a) CIFAR-10



(b) Imagenette

Figure 5: Experiments of a tiny ViT trained on CIFAR-10 (first row) and Imagenette (second row), varying the learning rate as $\eta \in \{0.001, 0.003, 0.005\}$ with a fixed batch size of 64. (i) Left: test accuracy (%), (iii) Right: kernel–target alignment of NTK. The dotted lines denote the test accuracy and the kernel–target alignment when the training loss is identical across learning rates. The kernel–target alignment of NTK increases when the learning rate increases above a certain threshold, even when models were compared at the same training loss.

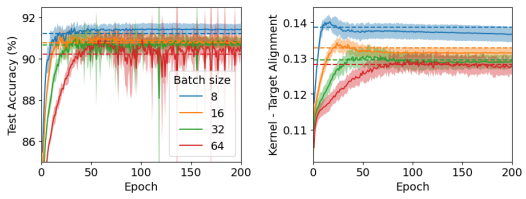
A.2 Experiments on a tiny ViT

Batch size. Figure 4 shows the result when the batch size of SGD was varied. As a result, the kernel–target alignment of NTK increased once the batch size fell below a certain threshold. This trend remained consistent in the case of CIFAR-10 even when models were compared at the same training loss: batch sizes below a certain threshold yielded higher kernel–target alignment.

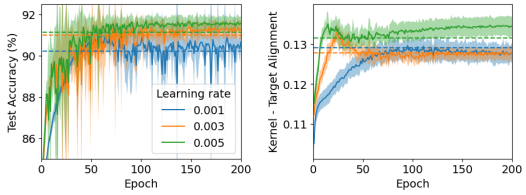
Learning rate. Figure 5 shows the result with variations on the learning rate of SGD. As a result, the kernel–target alignment of NTK increased once the learning rate increased above a certain threshold. The same pattern held after controlling the training loss as identical across learning rates, with kernel–target alignment increasing when the learning rate entered a sufficiently large regime.

Generalization. However, a tiny ViT did not show a correlation between the kernel–target alignment and the test accuracy. In the experiments of Imagenette, the test accuracy at convergence was higher for a larger batch size, even though the kernel–target alignment was lower. When the accuracy was compared at the same training loss, the accuracy was similar across different batch sizes. Also, the test accuracy at convergence was higher for a smaller learning rate, even though the kernel–target alignment was lower. Conventional analyses linking kernel–target alignment to network generalization may not directly extend to Transformer-based architectures such as ViTs. Consistent with this limitation, our ViT experiments exhibit a different relationship between batch size, kernel–target alignment, and test performance. We conjecture that ViTs deviate substantially from homogeneous architectures and may therefore be less accurately described as kernel machines governed by their evolved NTKs.

A.3 Experiments on a multi-layer perceptron



(a) Batch size

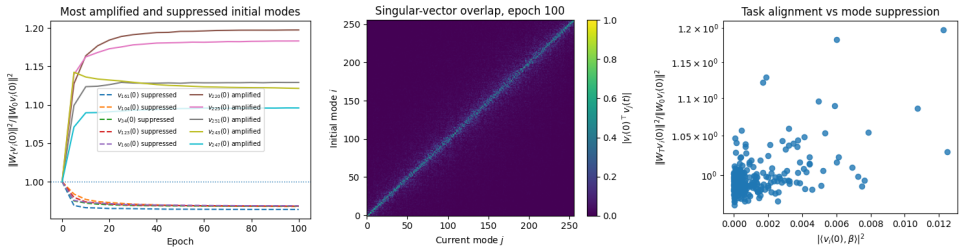


(b) Learning rate

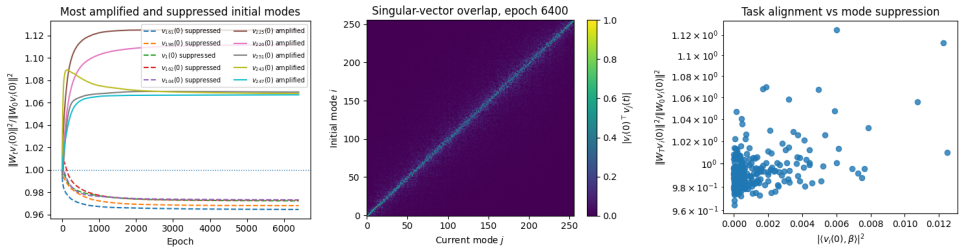
Figure 6: Experiments of a three-layer MLP trained on CIFAR-10, varying the batch size as $B \in \{8, 16, 32, 64\}$ with a fixed learning rate of 0.001 (upper row) and varying the learning rate as $\eta \in \{0.001, 0.003, 0.005\}$ with a fixed batch size of 64 (lower row). (i) Left: test accuracy (%), (iii) Right: kernel–target alignment of NTK. The kernel–target alignment of NTK increases with a stochastic noise scale above a certain threshold.

We also conducted experiments on a 3-layer MLP of width 1024. The batch size and the learning rate was set identical to previous experiments. Figure 6 shows the result. As a result, the kernel–target alignment of NTK increased once the batch size decreased below a certain threshold. For learning rate, the kernel–target alignment of NTK increased once the learning rate increased above a certain threshold of 0.005. The test accuracy increased with kernel–target alignment, showing an empirical correlation between the two quantities. However, the effect was not as dramatic as in the experiments of ResNet-18 or tiny ViT since the ReLU network is relatively simple and has a shallow depth.

A.4 SUPPRESSION AND AMPLIFICATION OF MODES IN A TWO-LAYER RELU NETWORK



(a) Batch size = 8



(b) Batch size = 512

Figure 7: Experiments of a two-layer MLP trained on 5000 randomly-sampled MNIST under a random linear target function. The upper row shows the result for batch size of 8, and the lower row shows the result for batch size of 512. (i) Left: suppression and amplification of initial singular directions, (ii) Middle: rotation of singular directions, (iii) Right: target alignment vs suppression of modes. It is possible to observe that the modes aligned with the target tend to be amplified while those not aligned tend to be suppressed. With a smaller batch size, the relative amplification is larger.

In this section, we simulate how features are suppressed and amplified by SGD for networks that are not diagonal. We employ a two-layer ReLU neural network with a normal initialization scale since (i) the network has non-linearity and (ii) the rank and singular directions of the weights can be easily obtained. The network was trained on 5000 randomly-sampled MNIST under a random linear target function. We varied the batch size as 8 and 512, and matched the number of training steps by controlling the number of epochs respectively as 100 and 6400. We dissect the first layer weight as $W(t) = \sum_i s_i(t)u_i(t)v_i(t)^\top$ with SVD decomposition and analyze it with three metrics:

Amplification of initial singular directions. In the left figures of Figure 7, we measure $R_i(t) = \|W(t)v_i(0)\|^2/\|W(0)v_i(0)\|^2$. We show the top-5 amplified and suppressed directions in the plots. While the maximum amplification was 1.20 for batch size of 8, it was 1.12 for batch size of 512 which is smaller. It shows that the relative amplification of modes is larger in smaller batches. Furthermore, the stable rank for batch size of 8 was about 40.37 while the one for 512 was about 59.28, which implies that certain singular directions are suppressed under a small batch size.

Rotation of singular directions. In the middle figures of Figure 7, we measure the pairwise similarity between the initial singular directions and current directions: $|v_i(0)^\top v_j(t)|$. If the plots focus on diagonal elements, it indicates little rotation of singular directions. While the plots are not completely diagonal, much of the elements are concentrated in diagonal regions. This limited rotation suggests that tracking approximately persistent singular directions may provide a useful qualitative approximation to the mode-wise dynamics assumed in Section 4.

Target alignment vs suppression of modes In the right figures of Figure 7, we seek the relationship between the suppression of directions, $R_i(t)$, and the target alignment of directions, $|v_i(0)^\top \beta|$. The plots show a weak correlation: the modes with weak target alignment tend to be the suppressed modes. It shows that the collapse shown in Section 4 might hold for complex non-linear networks.

B THEORETICAL DETAILS ON THE TOY EXAMPLE

Here, we pack the necessary notations and assumptions again. On the training set $\mathcal{D}$ with input data $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d | (\boldsymbol{x}, y) \in \mathcal{D}\}$ and ground-truth labels $\mathcal{Y} = \{y \in \mathbb{R} | (\boldsymbol{x}, y) \in \mathcal{D}\}$, a neural network $f_{\boldsymbol{a},\boldsymbol{b}} : \mathcal{X} \to \mathcal{Y}$ with parameters $\boldsymbol{a}, \boldsymbol{b}$ is trained on the training set. The network can be expressed as follows when feature vectors v_i are fixed and orthonormal ($\boldsymbol{v}_i \perp \boldsymbol{v}_j$ for $i \neq j$, $\|\boldsymbol{v}_i\|_2 = 1$):

$$f_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x}) = \sum_{i=1}^{d} a_i b_i \phi_i(\boldsymbol{x}), \quad \phi_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{v}_i \tag{24}$$

We assume that the network is trained under MSE loss, $l(f, y) = \frac{1}{2}\|f - y\|_2^2$. For simplicity, we assume that the target is an output of a linear teacher β: $y = \beta^\top \boldsymbol{x}$.

We now assume a case where the *label noise* interrupts the deterministic gradient flow. For ease of analysis, we assume the following conditions:

Assumption B.1. (Whitened input) The inputs are whitened as $\Sigma_{xx} := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right] = I$.

Assumption B.2. (Structural label noise) At each update, the target is perturbed by fresh Gaussian noise as $y \mapsto y + \xi$ when $\xi \sim \mathcal{N}(0, \zeta^2)$.

Assumption B.3. (Balanced initialization) For every $i \in [d]$, the initial weights of a_i and b_i are balanced as $a_i^2 = b_i^2$ at $t = 0$.

Assumption B.4. (Positive sign) For every $i \in [d]$, the signs of target signal strength $\mu_i := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[y\phi_i(\boldsymbol{x})\right]$ and initial neuron variable $s_i := a_i b_i$ are positive as $\frac{s_i}{|s_i|} = \frac{\mu_i}{|\mu_i|} = 1$ at $t = 0$.

Under these assumptions, each neuron variable $s_i := a_i b_i$ is adjusted to match the corresponding target signal strength $\mu_i := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y\phi_i(x)\right]$. Now the training loss can be expressed as follows with an additional constant $\mathcal{C}$:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left(y - f_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{x})\right)^2 = \frac{1}{2}\sum_{i=1}^{d}(s_i - \mu_i)^2 + \mathcal{C}. \tag{25}$$

The target signal strength μ_i measures how much the feature vector $\boldsymbol{v}_i$ is aligned to the target vector β as $\mu_i = \beta^\top \boldsymbol{v}_i$. We first analyze the label diffusion.

Label diffusion. Training the network under gradient descent perturbed with Gaussian label noise is equivalent to minimizing $l(f, y) = \frac{1}{2}\|f - y - \xi\|_2^2$. The gradient update with a learning rate η becomes $\theta^{(t+1)} = \theta^{(t)} - \eta\nabla\mathcal{L}(\theta^{(t)}) - \eta\mathbb{E}\left[\xi\nabla f(\boldsymbol{x})\right]$. Taking the limit as $\eta \to 0$, we can approximate gradient updates with an SDE of *label-noise gradient flow*.

$$d\theta(t) = -\nabla\mathcal{L}(\theta(t)) + \sqrt{2D_{LN}(\theta(t))}dB_t \tag{26}$$

where the noise covariance term becomes

$$D_{LN}(\theta(t)) = \frac{\eta\zeta^2}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[\nabla f_{\theta(t)}(\boldsymbol{x})\nabla f_{\theta(t)}(\boldsymbol{x})^\top\right] \tag{27}$$

Here, the derivatives of the loss with respect to scalar parameters are

$$\frac{\partial f(\boldsymbol{x})}{\partial a_i} = b_i\phi_i(\boldsymbol{x}), \quad \frac{\partial f(\boldsymbol{x})}{\partial b_i} = a_i\phi_i(\boldsymbol{x}) \tag{28}$$

Therefore, the SDE noise covariance term for a concatenation of a_i and b_i becomes

$$\begin{pmatrix} da_{i,noise} \\ db_{i,noise} \end{pmatrix}\begin{pmatrix} da_{i,noise} & db_{i,noise} \end{pmatrix} = \eta\zeta^2\mathbb{E}_{\boldsymbol{x}}\left[\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})^\top\right]\begin{pmatrix} b_i \\ a_i \end{pmatrix}\begin{pmatrix} b_i & a_i \end{pmatrix}dt = \eta\zeta^2\delta_{ij}\begin{pmatrix} b_i \\ a_i \end{pmatrix}\begin{pmatrix} b_i & a_i \end{pmatrix}dt \tag{29}$$

Thus, we can obtain the diffusion term of the concatenation as

$$\begin{pmatrix} da_{i,noise} \\ db_{i,noise} \end{pmatrix} = \sqrt{\eta\zeta^2}\begin{pmatrix} b_i \\ a_i \end{pmatrix}dB_{i,t} \tag{30}$$

Stochastic dynamics. Now we obtain the gradient of the parameters a_i and b_i with respect to the loss in Equation 7 to compute the deterministic gradient flow.

$$\frac{\partial \mathcal{L}}{\partial a_i} = b_i(s_i - \mu_i), \quad \frac{\partial \mathcal{L}}{\partial b_i} = a_i(s_i - \mu_i) \tag{31}$$

Now, the label diffusion perturbs the gradient flow. Based on the previous analysis, the stochastic dynamics of the parameters become as follows when $\tau := \eta\zeta^2$:

$$da_i = -b_i(s_i - \mu_i)dt + \sqrt{\tau} b_i dB_{i,t} \tag{32}$$

$$db_i = -a_i(s_i - \mu_i)dt + \sqrt{\tau} a_i dB_{i,t}. \tag{33}$$

Under these dynamics, we find that the balance between a_i and b_i is preserved througout training. It can be confirmed by checking the evolution of $R_i = a_i^2 - b_i^2$. By Itô's lemma,

$$d(a_i^2) = 2a_i da_i + (da_i)^2 = 2a_i da_i + \tau b_i^2 \tag{34}$$

$$d(b_i^2) = 2b_i db_i + (db_i)^2 = 2b_i db_i + \tau a_i^2 \tag{35}$$

Since the deterministic terms cancel in difference, $dR_i = -\tau R_i dt$ and $R_i = a_i^2 - b_i^2 = 0$ remains during training. Thus, $a_i^2 = b_i^2$ remains during training.

Then, one can obtain the dynamics of the neuron variables s_i by Itô's lemma:

$$ds_i = a_i db_i + b_i da_i + d\left[a_i, b_i\right] \tag{36}$$

Compute the necessary dynamics as

$$a_i db_i + b_i da_i = -(a_i^2 + b_i^2)(s_i - \mu_i)dt + \sqrt{\tau}(a_i^2 + b_i^2)dB_{i,t} \tag{37}$$

$$d\left[a_i, b_i\right] = (\sqrt{\tau} b_i)(\sqrt{\tau} a_i)dt = \tau s_i dt. \tag{38}$$

By the balanced norms and positiveness of neuron variables, $s_i = (a_i^2 + b_i^2)/2$ remains. Aggregate them and we obtain the following dynamics.

$$ds_i = (a_i^2 + b_i^2)(\mu_i - s_i)dt + \tau s_i dt + \sqrt{\tau}(a_i^2 + b_i^2)dB_{i,t} \tag{39}$$

$$= 2s_i(\mu_i + \frac{\tau}{2} - s_i)dt + 2\sqrt{\tau} s_i dB_{i,t} \tag{40}$$

Here, it is safe to keep s_i positive as $(a_i^2 + b_i^2)/2$ since the solution of the SDE remains positive:

$$s_i(t) = \frac{\exp((2\mu_i - \tau)t + 2\sqrt{\tau} B_{i,t})}{s_i(0)^{-1} + 2\int_0^t \exp((2\mu_i - \tau)u + 2\sqrt{\tau} B_{i,u})}. \tag{41}$$

and we obtain an equation of the same form that often appears in a stochastic environment (Oksendal, 2013).

The solution above can be obtained from computing the dynamics of s_i^{-1} and taking an ansatz; regarding s_i^{-1} as multiplication of two factors. Though the setting is different, we refer to the proofs in Chen et al. (2023) for the last solution since the derivation of the last solution is identical to ours. Here, the assumptions on the positive signs of s_i and μ_i can be easily relaxed to an assumption on the identical signs of s_i and μ_i since their signs also do not change during training.

By Itô's rule, the dynamics of the logarithm from the neuron variable becomes

$$d\log s_i = (2\mu_i - \tau - 2s_i)dt + 2\sqrt{\tau} dB_{i,t} \tag{42}$$

Another important thing to obtain from the equation above is that if $s_i \to 0$, s_i is *attracted to zero* when $\mu_i < \frac{\tau}{2} = \frac{\eta\zeta^2}{2}$. Thus, stochastic collapse suppresses certain neurons if the target signal strength of the neuron, μ_i, is weaker than a certain threshold, $\frac{\eta\zeta^2}{2}$, which is dependent on the noise scale and the learning rate.

NTK dynamics. We now obtain NTK of our network. By aggregating the derivatives of the parameters, NTK of the network can be expressed as

$$K_t(x,x') = \sum_i \left[\frac{\partial f}{\partial a_i}(x)\frac{\partial f}{\partial a_i}(x') + \frac{\partial f}{\partial b_i}(x)\frac{\partial f}{\partial b_i}(x')\right] \tag{43}$$

$$= \sum_i (a_i^2 + b_i^2)\phi_i(x)\phi_i(x') = \sum_i 2s_i(t)\phi_i(x)\phi_i(x'). \tag{44}$$

It is possible to notice that a doubled neuron variable, $2s_i$, becomes an eigenvalue of NTK because $\boldsymbol{v}_i$'s are orthonormal and $\phi_i(x)$ becomes an orthonormal eigenfunction under whitened data.

$$\mathbb{E}_{\boldsymbol{x}}\left[\phi_i(x)\phi_j(x)\right] = \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{v}_i^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{v}_j\right] = \boldsymbol{v}_i^\top \boldsymbol{v}_j = \delta_{ij} \tag{45}$$

Since s_i collapses if $\boldsymbol{v}_i$ is not aligned to β, which corresponds to a weak target signal strength μ_i, the eigenvalue $2s_i$ also collapses if the eigenfunction $\phi_i(x) = x^\top \boldsymbol{v}_i$ is not aligned to the target $y = \beta^\top x$. Thus, if the scale of stochastic noise is large, only eigenfunctions with large target alignment μ_i will remain in NTK. This can make the kernel–target alignment of NTK larger under certain conditions. First, we obtain the kernel–target alignment of NTK as

$$\mathcal{A}(s,\mu) = \frac{\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}\left[\sum_i 2s_i(t)\phi_i(x)\phi_i(x')\right]}{\sqrt{\sum_i (2s_i)^2 \sum_i (\beta^\top \boldsymbol{v}_i)^2}} = \frac{\sum_i s_i\mu_i^2}{\sqrt{\sum_i s_i^2 \sum_i \mu_i^2}}. \tag{46}$$

The kernel–target alignment increases if the following holds for $\mathbf{C}$, which is the set of indices of the collapsed neuron variables s_i:

$$\frac{\sum_i s_i\mu_i^2 - \sum_{j\in\mathbf{C}} s_j\mu_j^2}{\sqrt{\sum_i s_i^2 - \sum_{j\in\mathbf{C}} s_j^2}} > \frac{\sum_i s_i\mu_i^2}{\sqrt{\sum_i s_i^2}} \tag{47}$$

which is satisfied if μ_i's for $i \in \mathbf{C}$ are small as

$$\sum_{j\in\mathbf{C}} s_j\mu_j^2 < \left(\sqrt{1 + \frac{\sum_{j\in\mathbf{C}} s_j^2}{\sum_{i\notin\mathbf{C}} s_i^2}} - 1\right)\sum_{i\notin\mathbf{C}} s_i\mu_i^2. \tag{48}$$

Thus, if stochastic collapse is moderate and only suppresses feature vectors with weak target alignment, it can increase the kernel–target alignment of NTK. One thing to be warned is that, if stochastic noise is too large and it suppresses the directions with large target alignment, the kernel–target alignment of NTK can decrease. Hence, the relationship between stochastic noise and the kernel–target alignment is certainly not monotone.

Extension to mini-batch noise. The local stochastic dynamics can be connected to mini-batch noise when the parameters are initialized sufficiently close to zero such that $f_{\boldsymbol{a},\boldsymbol{b}}(x) \approx 0$. Near initialization, the derivative of the loss for one sample becomes

$$\nabla_{a_i} l \approx -b_i(\beta^\top \boldsymbol{x})(\boldsymbol{v}_i^\top \boldsymbol{x}), \quad \nabla_{b_i} l \approx -a_i(\beta^\top \boldsymbol{x})(\boldsymbol{v}_i^\top \boldsymbol{x}) \tag{49}$$

Consider an SGD update with a mini-batch $\mathcal{B}$ of size B. The initial local SGD dynamics become

$$a_i(t+1) = a_i(t) + \frac{\eta}{B}\sum_{n\in\mathcal{B}} b_i(t)(\beta^\top \boldsymbol{x}_n)(\boldsymbol{v}_i^\top \boldsymbol{x}_n) = a_i(t) + \eta b_i(t)(\mu_i + \epsilon_{i,\mathcal{B}}) \tag{50}$$

$$b_i(t+1) = b_i(t) + \frac{\eta}{B}\sum_{n\in\mathcal{B}} a_i(t)(\beta^\top \boldsymbol{x}_n)(\boldsymbol{v}_i^\top \boldsymbol{x}_n) = b_i(t) + \eta a_i(t)(\mu_i + \epsilon_{i,\mathcal{B}}) \tag{51}$$

with $\mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}] = 0$. For independent sampling, the variance of the effective mini-batch noise is

$$\mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}^2] = \frac{1}{B}\operatorname{Var}_{(\boldsymbol{x},y)\sim\mathcal{D}}[y\phi_i(\boldsymbol{x})] = \frac{1}{B}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})^2(\boldsymbol{x}^\top \boldsymbol{v}_i)^2\right] - \mu_i^2\right), \tag{52}$$

We further assume isotropic Gaussian inputs $\boldsymbol{x} \sim \mathcal{N}(0, I)$. By matching the second moment of the mini-batch noise to the label-noise diffusion, the mode-dependent effective diffusion scale is

$$\begin{aligned}
\tau_i &:= \eta \mathbb{E}_{\mathcal{B}}[\epsilon_{i,\mathcal{B}}^2] && (53)\\
&= \frac{\eta}{B}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})^2 (\boldsymbol{x}^\top \boldsymbol{v}_i)^2\right] - \mu_i^2\right) && (54)\\
&= \frac{\eta}{B}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})^2\right]\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\boldsymbol{x}^\top \boldsymbol{v}_i)^2\right] + 2(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}\left[(\beta^\top \boldsymbol{x})(\boldsymbol{x}^\top \boldsymbol{v}_i)\right])^2 - \mu_i^2\right) && (55)\\
&= \frac{\eta}{B}\left(\left(\|\beta\|_2^2 + 2\mu_i^2\right) - \mu_i^2\right) && (56)\\
&= \frac{\eta}{B}\left(\|\beta\|_2^2 + \mu_i^2\right), && (57)
\end{aligned}$$

which can be confirmed by Isserlis's theorem.

In particular, for weakly target-aligned modes with $\mu_i^2 \ll \|\beta\|_2^2$, the effective noise scale remains approximately $\tau_i \approx \eta\|\beta\|_2^2/B$ even as the target signal strength μ_i becomes small. Therefore, under a covariance-matched local diffusion approximation, such weakly target-aligned modes are relatively more susceptible to suppression if the batch size is small and noise scale increases. Unlike the label-noise model, usual mini-batch noise may additionally contain correlations across different modes, thus, this correspondence should be understood as a local mode-wise approximation.