Stochastic Restarting to Overcome Overfitting in Neural Networks with Noisy Labels

Youngkyoung Bae[1]* **Yeongwoo Song**[2]* **Hawoong Jeong**[2, 3]†
[1]Department of Physics and Astronomy, Seoul National University
[2]Department of Physics, KAIST [3]Center for Complex Systems, KAIST
tardis_95@snu.ac.kr, ywsong1025@kaist.ac.kr, hjeong@kaist.edu

Abstract

Despite its prevalence, giving up and starting over may seem wasteful in many situations such as searching for a target or training deep neural networks (DNNs). Our study, though, demonstrates that restarting from a checkpoint can significantly improve generalization performance when training DNNs with noisy labels. In the presence of noisy labels, DNNs initially learn the general patterns of the data but then gradually overfit to the noisy labels. To combat this overfitting phenomenon, we developed a method based on stochastic restarting, which has been actively explored in the statistical physics field for finding targets efficiently. By approximating the dynamics of stochastic gradient descent into Langevin dynamics, we theoretically show that restarting can provide great improvements as the batch size and the proportion of corrupted data increase. We then empirically validate our theory, confirming the significant improvements achieved by restarting. An important aspect of our method is its ease of implementation and compatibility with other methods, while still yielding notably improved performance. We envision it as a valuable tool that can complement existing methods for handling noisy labels.

1 Introduction

When we explore a search space having complex choices of training schemes or search for appropriate hyper-parameters of deep neural networks (DNNs), we often meet circumstances that cause us to just give up and train the network all over again. This is akin to our experiences in daily life, where we face various tasks that require solving problems through hit-and-miss. For example, when trying to find a beloved one's face in a crowd, our eyes typically flick back to a chosen starting point after scanning the surrounding area. Similarly, when searching for a misplaced wallet after a big night out, we often fail to locate it and restart our search from some original location. These patterns are also frequently observed in animal behavior, such as foraging for food and returning to familiar locations such as nests or dens. In these situations, one might think that revisiting previously visited places is a waste of time and resources, potentially diminishing search performance. However, recent developments in statistical physics have proved that restarting from the starting or mid-point can improve the performance of the search process, meaning it is not so haphazard after all.

This effect of restarting from a particular configuration has been extensively investigated in the field of statistical physics in recent years [1, 2]. These investigations typically involve a blind searcher who evolves their current state stochastically over time without knowledge of the target's location. Surprisingly, it has been found that restarting does not hinder the search process but rather can make the searcher more efficient across diverse conditions, including scenarios with high dimensions or the

*Equal Contribution.
†Correspondence to Hawoong Jeong.

Preprint. Under review.

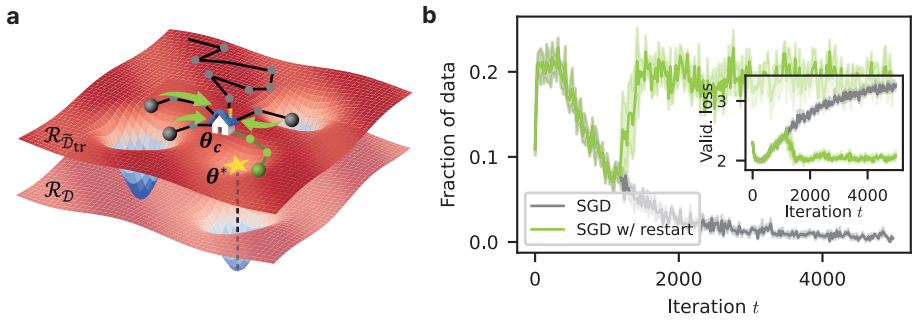

Figure 1: (a) Schematic of stochastic gradient descent (SGD) dynamics with stochastic restarting. The network parameters vector $\boldsymbol{\theta}$ evolves via SGD to find an optimal value $\boldsymbol{\theta}^*$ on the training risk landscape $\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}$ (upper colormap), which differs from the true risk landscape $\mathcal{R}_{\mathcal{D}}$ (lower colormap) due to corrupted data. Here, $\boldsymbol{\theta}$ resets to the checkpoint $\boldsymbol{\theta}_c$ (home icon) with the restart probability r and restarts from $\boldsymbol{\theta}_c$. (b) Fraction of correctly predicted data with wrong labels during training with SGD (gray) and SGD with restart (green). The inset shows the validation losses during training.

presence of external forces [3–11]. Capitalizing on the success of the restarting strategy, numerous algorithms incorporating this approach have begun to emerge in diverse fields such as molecular dynamics simulations [12, 13] and queuing systems [14].

In this work, we apply the stochastic restarting strategy to supervised learning with noisy labels and show that it can prevent overfitting to corrupted data (also called the memorization effect). During training, our method resets the model parameters to a checkpoint with a certain probability and restarts the training process [Fig. 1(a)]. We make an in-depth exploration to understand the conditions and mechanisms by which restarting can help stochastic gradient descent (SGD) find the optimal parameters by mapping the SGD dynamics to the corresponding Langevin dynamics. Our main contributions are summarized as follows.

- We develop a new DNN training method incorporating stochastic restarting, and we provide both theoretical and empirical evidence that restarting can mitigate overfitting to corrupted data. We also discuss the means to select preferable checkpoints (Sec. 4.1).
- Drawing from insights into the advantageous mechanism of restarting in search processes, we find that the improvement gained through restarting increases as the stochasticity of the SGD dynamics and the proportion of corrupted training data increase (Sec. 4.2).
- We show that the restarting method can be seamlessly incorporated into existing methods and consistently improves generalization performance across several standard benchmark datasets (Sec. 4.4). We emphasize that our method is easy and compatible with other methods, but yields significantly improved performance.

2 Related Works

2.1 Deep learning from noisy labels

While the accessibility of large datasets has propelled remarkable advancements in DNNs, the presence of noisy labels within these datasets often leads to erroneous model prediction [15]. Specifically, DNNs tend to overfit the entire corrupted training dataset by memorizing the wrong labels, which degenerates their generalization performance on a test dataset. Numerous studies have been conducted to address this overfitting phenomenon [16–21], and it has been revealed that DNNs initially learn the clean data (general patterns) during an early learning stage and then gradually memorize the corrupted data (task-specific patterns) [22, 23]. The overfitting issue stemming from the memorization effect can be seen in Fig. 1(b), where the model's accuracy in predicting the true labels of the data with noisy labels exhibits an inverted U-shaped curve as the model progressively memorizes the noise. Based on this understanding, the surprising effectiveness of the early-stopping method [24] in alleviating the memorization effect becomes evident; as such, various methods have been proposed to leverage this insight, including Co-teaching [25], SELFIE [26], early learning regularization (ELR) [27], and

robust early learning [28]. Our proposed algorithm also capitalizes on this insight by enabling the DNN to restart from a checkpoint, i.e., previously visited parameters during early learning stages.

2.2 Search processes and stochastic restarting in statistical physics

Search processes are ubiquitous across various domains, spanning from systems in nature to applications in engineering. For instance, ligands exhibit search processes as they navigate toward target binding sites within proteins [29–31], and similarly, predators employ search strategies to locate their prey in the wild [32, 33]. In engineering, search processes are relevant to finding primary research studies [34], ranking web pages [35], and determining optimal hyper-parameters for training algorithms [36]. Although diverse search strategies are employed depending on the problem at hand, they share a common goal: to identify an efficient search protocol. Efficiency is typically assessed by the time required to reach a target, referred to as the first passage time (FPT) in the context of random walk literature [37]. Numerous search strategies have been investigated to achieve this goal, including the Lévy strategies [38, 39], self-avoiding walks [40, 41], intermittent strategies [42], persistent random walks [43], and more [44]. One recent strategy that has garnered attention is stochastic restarting, with studies showcasing its ability to enhance the search performance [2, 7, 45, 10]. In particular, these studies have demonstrated that stochastic restarts prevent a random searcher from wandering too far, thereby ensuring a finite mean time to find a target, whereas the mean time is infinite for a diffusive particle without restarting. Drawing from this concept, we introduce a restarting method for training DNNs and illustrate its effectiveness in addressing the noisy label problem.

3 Methodology

In this section, we introduce our simple Algorithm 1 utilizing stochastic restarting to mitigate the memorization effect on training data with noisy labels. We also demonstrate how restarting can lead to improved generalization performance by approximating SGD dynamics into Langevin dynamics.

Problem setup. Consider a c-class classification problem, which is a supervised learning task aimed at training a function to map input features to labels through a DNN. Let $\mathcal{X} \subset \mathbb{R}^p$ be the feature space, $\mathcal{Y} = \{0,1\}^c$ be the label space in one-hot vector form, and $\boldsymbol{f}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ be a DNN model where $\boldsymbol{\theta} \in \mathbb{R}^d$ encompasses all trainable parameters in the DNN. The goal is to find an optimal $\boldsymbol{\theta}^*$ such that $\boldsymbol{f}_{\boldsymbol{\theta}^*}$ accurately assigns labels to corresponding input features, given an unknown joint probability distribution $P_{\mathcal{D}}$ over $\mathcal{X} \times \mathcal{Y}$ [Fig. 1(a)]. To obtain this, a training algorithm is applied to minimize the risk $\mathcal{R}_{\mathcal{D}}(\boldsymbol{\theta}) \equiv \langle \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) \rangle_{\mathcal{D}}$ during training, where $\mathcal{L}$ denotes a loss function (e.g., cross-entropy loss). Here, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ denotes the loss for a sample $(\boldsymbol{x}, \boldsymbol{y})$ from $P_{\mathcal{D}}$ with a given model $\boldsymbol{f}_{\boldsymbol{\theta}}$, and $\langle \cdot \rangle_{\mathcal{D}}$ denotes the average over $P_{\mathcal{D}}$. In a typical classification problem, the DNN is trained by minimizing the risk on the training dataset $\mathcal{D}_{\mathrm{tr}}$ via SGD and $\boldsymbol{\theta}^*$ is selected at the minimum risk on the validation dataset $\mathcal{D}_{\mathrm{val}}$ to mitigate overfitting on $\mathcal{D}_{\mathrm{tr}}$, where $\mathcal{D}_{\mathrm{tr(val)}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_{\mathrm{tr(val)}}}$ and each $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is sampled from $P_{\mathcal{D}}$. Empirically, the risk on the training (validation) dataset is computed as $\mathcal{R}_{\mathcal{D}_{\mathrm{tr(val)}}}(\boldsymbol{\theta}) = \left(1/N_{\mathrm{tr(val)}}\right) \sum_{i=1}^{N_{\mathrm{tr(val)}}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta})$. In the presence of noisy labels, suppose we have a corrupted training dataset $\tilde{\mathcal{D}}_{\mathrm{tr}} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{N_{\mathrm{tr}}}$, where $\tilde{\boldsymbol{y}}$ is a noisy label that may

Algorithm 1 Stochastic Restarting

Require: Corrupted training set $\tilde{\mathcal{D}}_{\mathrm{tr}}$, validation set $\mathcal{D}_{\mathrm{val}}$, restart probability r, threshold $\mathcal{T}$.

Initialize $\boldsymbol{\theta}_0$ and set $t = 0$, $\boldsymbol{\theta}_c =$ None, and $\boldsymbol{\theta}_{\mathrm{best}} =$ None
for $t = 0$ to T **do**
  Update $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_t - \frac{\eta}{B} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{B}_t} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}_t)$ where $\mathcal{B}_t$ is a randomly sampled batch from $\tilde{\mathcal{D}}_{\mathrm{tr}}$
  if $\boldsymbol{\theta}_c \neq$ None and $rand(0,1) < r$ **then**
    Restart $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_c$
  end if
  $\boldsymbol{\theta}_{\mathrm{best}} \leftarrow \mathrm{Valid}(\boldsymbol{\theta}_t, \mathcal{D}_{\mathrm{val}})$ where $\mathrm{Valid}(\boldsymbol{\theta}_t, \mathcal{D}_{\mathrm{val}})$ checks whether $\mathcal{R}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}_t)$ is the minimum.
  if $\boldsymbol{\theta}_{\mathrm{best}}$ remains unchanged for $\mathcal{T}$ iterations **or** $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{\mathrm{best}}$ **then**
    Set the checkpoint $\boldsymbol{\theta}_c \leftarrow \boldsymbol{\theta}_{\mathrm{best}}$
  end if
end for

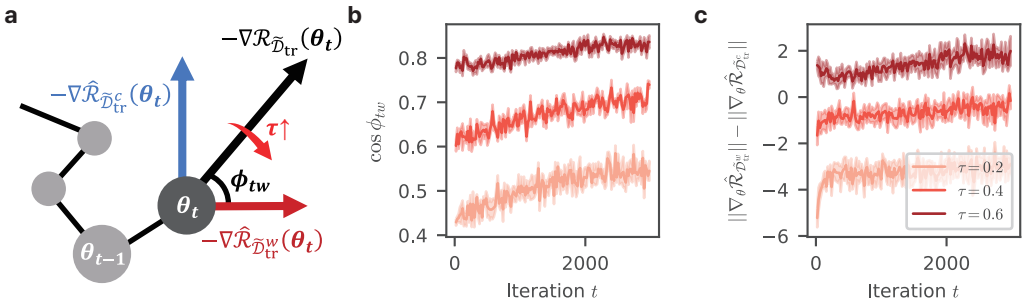

Figure 2: (a) Schematic of $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta})$, decomposed by two orthogonal terms $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta})$ and $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta})$. (b) Cosine similarity between $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta})$ and $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta})$, denoted by $\cos\phi_{tw}$, throughout all training iterations for varying noise rate τ. (c) Magnitude difference between the two vectors $\|\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta})\| - \|\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta})\|$ throughout all training iterations for varying τ. Here, we set the batch size to $B = 8$ in setting 1 described in Sec. 4.

be corrupted from a ground truth label $\boldsymbol{y}_i$, and $(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)$ is sampled from the corrupted distribution $P_{\tilde{\mathcal{D}}}$. This corrupted dataset can be partitioned into two subsets, i.e., $\tilde{\mathcal{D}}_{\rm tr} \equiv [\tilde{\mathcal{D}}^c_{\rm tr}, \tilde{\mathcal{D}}^w_{\rm tr}]$, where $\tilde{\mathcal{D}}^c_{\rm tr}$ ($\tilde{\mathcal{D}}^w_{\rm tr}$) consists of $N^c_{\rm tr}$ ($N^w_{\rm tr}$) samples with correct (wrong) labels. Note that $N^c_{\rm tr} = (1-\tau)N_{\rm tr}$ and $N^w_{\rm tr} = \tau N_{\rm tr}$ for an unknown noise rate $\tau \in [0, 1]$.

3.1 SGD dynamics with noisy labels

When we apply the minibatch SGD to minimize the empirical risk $\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, the update rules of $\boldsymbol{\theta}$ at each training iteration t can be represented by

$$\begin{aligned}\Delta\boldsymbol{\theta}_t &= -\frac{\eta}{B}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{B}_t}\nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\\ &= -\frac{\eta}{N_{\rm tr}}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{D}_{\rm tr}}\nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t) + \left(\frac{\eta}{N_{\rm tr}}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{D}_{\rm tr}}\nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t) - \frac{\eta}{B}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{B}_t}\nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\right),\end{aligned} \tag{1}$$

where $\boldsymbol{\theta}_t$ is $\boldsymbol{\theta}$ at the t-th iteration, $\Delta\boldsymbol{\theta}_t \equiv \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$, and $\mathcal{L}_i(\boldsymbol{\theta}_t) \equiv \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}_t)$ for simplicity. Here, $\eta > 0$ is the learning rate and $\mathcal{B}_t$ is a minibatch of size B consisting of independent and identically distributed (i.i.d.) samples from $\mathcal{D}_{\rm tr}$. While the first term on the right-hand-side (RHS) is deterministic for a given $\mathcal{D}_{\rm tr}$, the second term on the RHS is stochastic due to the randomly sampled batch at each iteration. Thus, Eq. (1) can be rewritten as

$$\Delta\boldsymbol{\theta}_t = -\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\eta + \boldsymbol{\xi}_t\sqrt{\eta}, \tag{2}$$

where a random noise vector $\boldsymbol{\xi}_t \equiv \sqrt{\eta}\left(\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t) - \nabla_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\right) \in \mathbb{R}^d$ satisfies $\langle\boldsymbol{\xi}_t\rangle_{\mathcal{D}_{\rm tr}} = \mathbf{0}$ and $\langle\boldsymbol{\xi}_t\boldsymbol{\xi}_s^{\rm T}\rangle_{\mathcal{D}_{\rm tr}} = 2\mathsf{D}(\boldsymbol{\theta}_t)\delta_{ts}$ with $\mathsf{D}(\boldsymbol{\theta}_t) \equiv \eta\Sigma(\boldsymbol{\theta}_t)/(2B)$, where δ_{ij} denotes the Kronecker delta (see details in the Supplementary Materials and Refs. [46–48]). In terms of Langevin dynamics, $\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta})$ and $\mathsf{D}(\boldsymbol{\theta})$ correspond to the potential and diffusion matrix, respectively, where the former generates the deterministic long-term trend called drift and the latter determines the level of stochasticity of the system [49]. As a result, the SGD dynamics of $\boldsymbol{\theta}_t$ can be understood by the Langevin dynamics of a d-dimensional particle diffusing with drift $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)$ and diffusion matrix $\mathsf{D}(\boldsymbol{\theta}_t)$.

For a corrupted dataset $\tilde{\mathcal{D}}_{\rm tr}$, the equation of SGD dynamics remains analogous to Eq. (2) when we substitute $\mathcal{D}_{\rm tr}$ with $\tilde{\mathcal{D}}_{\rm tr}$. Then, we can divide the drift vector $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta}_t)$ into two components, one originating from $\tilde{\mathcal{D}}^c_{\rm tr}$ and the other from $\tilde{\mathcal{D}}^w_{\rm tr}$ as follows [Fig. 2(a)]:

$$\Delta\boldsymbol{\theta}_t = -\left[\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t) + \nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)\right]\eta + \boldsymbol{\xi}_t\sqrt{\eta}, \tag{3}$$

with the gradients from the correct part $\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t) \equiv (1-\tau)\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)$ and the gradients from the wrong part $\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t) \equiv \tau\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$. Note that $\langle\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)\rangle_{\mathcal{D}} = \langle\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\rangle_{\mathcal{D}}$,

implying that $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)$ reflects the gradients toward the true optimum, while $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ reflects the gradients toward the false optimum by memorizing the noisy labels. Additionally, $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)$ and $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ are generally orthogonal to each other due to their high-dimensionality [Fig. S.2 in the Supplementary Materials], leading to $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta}_t)$ being represented by the sum of two orthogonal vectors. Thus, $-\nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta}_t)$ becomes more correlated with $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ as the noise rate τ increases. Fig. 2(b) and (c) illustrate that $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ becomes increasingly dominant so that the drift gradually tilts toward wrong directions as τ increases, where $\cos\phi_{tw}$ denotes the cosine similarity between $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)$ and $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ and $\|\cdot\|$ denotes the Euclidean norm of a vector. This gradually tilting trend toward a wrong direction can also be observed with respect to iteration t, implying that $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ becomes increasingly dominant as the learning process progresses beyond an early learning phase [27]. Therefore, in the presence of noisy labels, we can see that $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\rm tr}}(\boldsymbol{\theta}_t)$ emerges and hinders the search for the optimal parameters $\boldsymbol{\theta}^*$.

3.2 Stochastic restarting method

We now describe how the stochastic restarting process can be integrated into SGD. Based on this, we establish the specific premises of this work. Let $\boldsymbol{\theta}_c$ be a checkpoint to restart from and r be the restart probability at each iteration t, where a checkpoint refers to previously visited model parameters during training. By incorporating the restarting process, Eq. (2) for $\tilde{\mathcal{D}}_{\rm tr}$ can be expressed as

$$\boldsymbol{\theta}_{t+1} = \begin{cases} \boldsymbol{\theta}_c, & \text{with probability } r, \\ \boldsymbol{\theta}_t - \nabla_{\boldsymbol{\theta}}\mathcal{R}_{\tilde{\mathcal{D}}_{\rm tr}}(\boldsymbol{\theta}_t)\eta + \boldsymbol{\xi}_t\sqrt{\eta}, & \text{otherwise.} \end{cases} \tag{4}$$

The SGD dynamics with stochastic restarting involves two processes: restarting from a checkpoint $\boldsymbol{\theta}_c$ with probability r [top of Eq. (4)], and maintaining the SGD dynamics with probability $1-r$ [bottom of Eq. (4)]. Note that Eq. (4) shares the same form as the (overdamped) Langevin equation with Poissonian reset [3], and also that training DNNs to find optimal parameters can be likened to a search process for an unknown target. These parallels imply that similar advantages may arise in the training process of DNNs as in the search process of Langevin dynamics. In this search process, stochastic restarting has been proven to aid random searches for an unknown target by suppressing trajectories away from the target and increasing the chances of finding it. Here, the search efficiency is usually quantified by the mean first passage time (MFPT) [37], the average time to find a target. As mentioned in Sec. 2, the MFPT can be reduced by restarting in various situations. The beneficial properties of restarting in Langevin dynamics can be summarized as follows (see the Supplementary Materials for a brief review and formal statement of stochastic restarting with MFPT):

> When the stochasticity is sufficiently larger than the drift component toward a target, there is an optimal restart probability, and restarting can be beneficial for random searches.

Importantly, in supervised learning with noisy labels, the stochasticity of the SGD dynamics increases as batch size B decreases, and the drift component toward a target $-\nabla_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\rm tr}}(\boldsymbol{\theta}_t)$ weakens as the noise rate τ increases. Therefore, the above statement and the observation about stochasticity and drift suggest that the restarting strategy would be beneficial to search for optimal parameters in the presence of noisy labels. In Sec. 4, we perform several experiments and empirically show that stochastic restarting enhances generalization performance.

4 Experiments

In our experiments, we perform image classification tasks with noisy labels in the following settings.

Setting 1 (Sec. 4.1, 4.2, and 4.3). To examine the impact of stochastic restarting on the noisy label problem, we first utilize a small dataset called ciFAIR-10 [50], a variant of CIFAR-10 [51]. We employ a vanilla convolutional neural network (VCNN, see the Supplementary Materials) to facilitate a straightforward testing of our claims. Training is performed using cross-entropy loss and the SGD optimizer with a fixed learning rate of 10^{-2}. A clean validation set, $\mathcal{D}_{\rm val}$, is used to select the best model and monitor the validation loss during training.

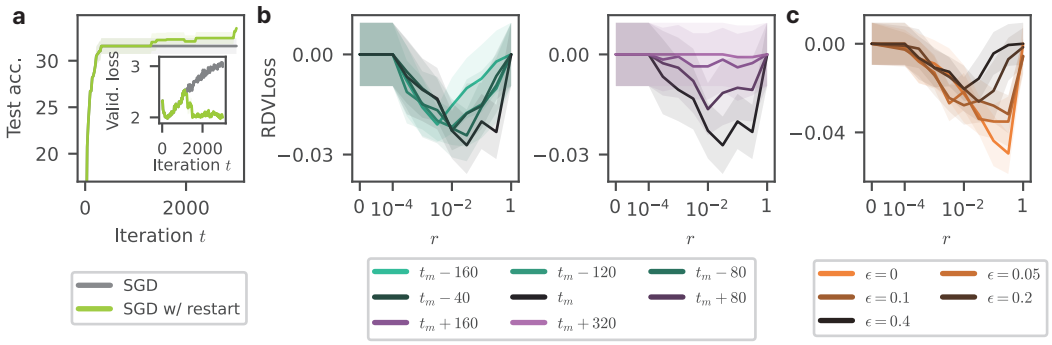

Figure 3: (a) Test accuracies of the SGD (gray) and the SGD with our restarting method (green) during training. The inset shows the validation losses. (b,c) Relative difference of validation loss (RDVLoss) with varying the checkpoint to restart with respect to the restart probability r. In (b), based on the checkpoint at the overfitting iteration t_m, RDVLoss is obtained in earlier iterations (left) and later iterations than t_m (right). $t_m + \delta t$ denotes the iteration where the checkpoint is selected. In (c), RDVLoss is plotted with the perturbed checkpoint parameters $\boldsymbol{\theta}_{c,\epsilon} \equiv \boldsymbol{\theta}_c + \epsilon\hat{\boldsymbol{n}}$, where $\boldsymbol{\theta}_c$ denotes the checkpoint and $\hat{\boldsymbol{n}}$ denotes a random unit vector. The shaded areas denote the standard error.

Setting 2 (Sec. 4.4). We assess the generalization performance of our method on two benchmark datasets, CIFAR-10 and CIFAR-100 [51], utilizing a ResNet-34 [52] and training it with SGD with a momentum of 0.9. Additional details for the choice of hyper-parameters are provided in the Supplementary Materials. To demonstrate the efficacy of our method, we compare test accuracy with and without restarting. Note that the optimizer and learning rate scheduler do not restart throughout this experiment. To consider practical situations, a corrupted validation set, $\tilde{\mathcal{D}}_{\text{val}}$, is used for model selection and validation loss monitoring during training.

Across both settings, we apply symmetric noise with a noise rate τ, where each label in c classes is randomly flipped to an incorrect label in other classes with equal probability $\tau/(c-1)$. Note that all results are obtained from the model at the optimal iteration based on minimum validation loss as default, and also that the resulting test accuracy is evaluated on the clean validation set, i.e., the test dataset $\mathcal{D}_{\text{te}}$ is set to $\mathcal{D}_{\text{val}}$. To check the effectiveness of restarting compared to the original training, we use the relative difference in validation loss (RDVLoss) and the relative difference in test accuracy (RDTAcc.) as metrics to indicate the relative improvement compared to the baseline. These metrics are calculated by $[v(r) - v_{\text{base}}]/v_{\text{base}}$, where $v(r)$ is the resulting value with the restart probability r and v_{base} is the baseline value obtained from the original training ($v_{\text{base}} = v(0)$). The unnormalized results can be found in the Supplementary Materials. We repeated our experiments five times across both settings to report the average and standard error values.

4.1 Which checkpoint would be preferable to restart from?

To introduce the restarting strategy in DNN training, we first explore which checkpoint is suitable to restart from to find optimal parameters. A straightforward choice is to select the parameters at the overfitting iteration t_m. Here, the overfitting iteration t_m refers to an iteration where the validation loss ceases to decrease and begins to increase due to the memorization effect [inset of Fig. 1(b)]. The checkpoint at t_m, denoted by $\boldsymbol{\theta}_m$, has the minimum validation loss during training when the double descent phenomenon does not occur [53], and is typically employed as an early-stopping point. Instead of early stopping and considering $\boldsymbol{\theta}_m$ as the final model, we utilize $\boldsymbol{\theta}_m$ as the checkpoint $\boldsymbol{\theta}_c$ to restart from [Algorithm 1], leading to significantly improved results [Fig. 3(a)]. Here, $\boldsymbol{\theta}_c$ is initially set to $\boldsymbol{\theta}_m$ and adaptively changes to the parameter at a newly found minimum validation loss during training. As can be seen in Fig. 3(a), restarting suppresses the trajectory of $\boldsymbol{\theta}$ to be near $\boldsymbol{\theta}_c$, which successfully prevents memorizing the noisy labels and increases the chance to find more appropriate parameters. It is important to note that the restart probability r controls the degree of suppression and the results at $r = 0$ and $r = 1$ are almost the same due to the overfitting phenomenon (the case of $r = 1$ corresponds to the early-stopping method). Therefore, the resulting RDVLoss curve for r should be U-shaped, indicating that an optimal r exists to optimize the performance.

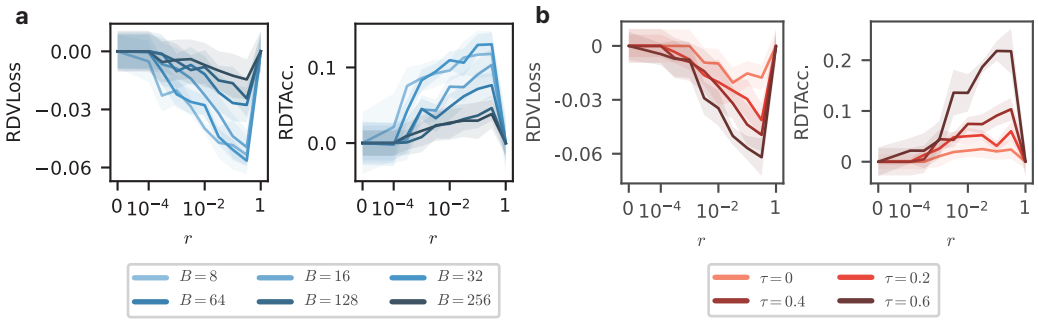

Figure 4: Relative difference of validation loss (RDVLoss, left) and relative difference of test accuracy (RDTAcc., right) results with (a) varying the batch size B, and (b) varying the noise rate τ with respect to the restart probability r. We set $\tau = 0.4$ in (a) and $B = 16$ in (b). The shaded areas denote the standard error.

While we simply select $\boldsymbol{\theta}_m$ as the initial $\boldsymbol{\theta}_c$ and adaptively update it, one may ask what effect the choice of $\boldsymbol{\theta}_c$ has. To check this, we experiment with a fixed checkpoint both earlier and later than $\boldsymbol{\theta}_m$. For earlier checkpoints [left panel in Fig. 3(b)], the improvement over restarting from $\boldsymbol{\theta}_m$ slightly decreases and the value of the optimal r gets smaller as the checkpoint gets earlier. In contrast, for later checkpoints [right panel in Fig. 3(b)], the improvement over restarting from $\boldsymbol{\theta}_m$ significantly decreases as the checkpoint gets later. These results support our understanding of the beneficial mechanism of restarting in increasing the chance of finding better parameters, because the chance would decrease as the model memorizes more noise. Therefore, we can conclude that restarting in early learning stages is a good choice; the more memorization occurs, the smaller the improvement.

We additionally experiment to verify how the effect of restarting changes with the distance between the (adaptive) checkpoint $\boldsymbol{\theta}_c$ at the minimum validation loss and a perturbed checkpoint $\boldsymbol{\theta}_{c,\epsilon}$. Here, we set the perturbed checkpoint by adding the perturbation $\epsilon\hat{\boldsymbol{n}}$ into $\boldsymbol{\theta}_c$ with varying the perturbation magnitude ϵ, where $\hat{\boldsymbol{n}} \equiv \boldsymbol{n}/\|\boldsymbol{n}\| \in \mathbb{R}^d$ is a random unit vector with a standard normal random vector $\boldsymbol{n}$. As shown in Fig. 3(c), it is observed that the benefits of restarting decrease as the distance between the checkpoint to restart from and $\boldsymbol{\theta}_m$ increases. This result also supports that while restarting can improve the generalization performance, the choice of checkpoint to restart from can affect the performance, and that the minimum validation loss point is a good choice.

4.2 Impact of stochasticity and drift on stochastic restarting

As the statement highlighted in Sec. 3.2 clarifies, it has been proven in the statistical physics field that the restarting strategy can improve search efficiency as the stochasticity becomes larger than the drift component toward a target. In this section, we verify whether this statement is also valid in the training of DNNs and show under what circumstances restarting is more effective than not restarting.

It is first important to note that the stochasticity and the drift toward a target, i.e., $-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\mathrm{tr}}}$, can be controlled by the batch size B and the noise rate τ, respectively, as illustrated in Sec. 3.1. Particularly, $\mathrm{D}(\boldsymbol{\theta}_t) \propto 1/B$ and $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\mathrm{tr}}}(\boldsymbol{\theta}_t) \propto 1-\tau$, meaning that the stochasticity increases and the drift toward a target decreases as B decreases and τ increases, respectively. We quantitatively examine the RDVLoss and the RDTAcc. values with varying B and τ with respect to the restart probability r. Remarkably, the improvements of RDVLoss and RDTAcc. with restarting become more significant as B decreases [Fig. 4(a)] and τ increases [Fig. 4(b)]. These observations strongly support our claim that stochastic restarting offers more benefits as the stochasticity increases and the drift toward a target decreases. Moreover, we expect that the optimal restart probability r^* decreases as B decreases and τ increases, but this can only be verified qualitatively because the fluctuation of the results makes it difficult to identify r^*.

4.3 Ablation study on partial restarting

Until now, we have leveraged the memorization effect in our algorithm by utilizing the parameters at the minimum validation loss as the checkpoint for restarting the entire network, a process referred to as full restarting. However, several studies have highlighted that different layers within a DNN exhibit

varied learning behaviors, leading to distinct levels of overfitting across these layers [54, 55]. A prevailing explanation for this phenomenon suggests that gradients tend to weaken as they propagate from the latter layers (closer to the output layer) to the former layers. Here, we experiment on which layers, former or latter, play a more dominant role in improving performance with the restarting method.

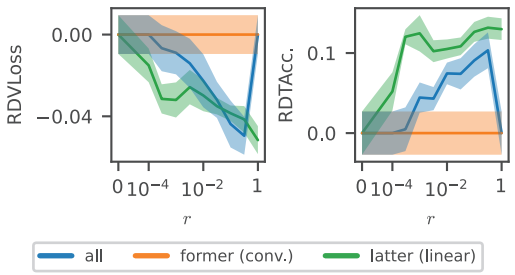

Figure 5: Relative difference of validation loss (RDVLoss) and relative difference of test accuracy (RDTAcc.) results with varying one section of the network to restart with respect to the restart probability r. We set $\tau = 0.4$ and $B = 16$. The shaded areas denote the standard error.

For this, we introduce partial restarting, which involves restarting only one section of the network layers rather than the entire network, while the remaining section of layers continues to follow the standard SGD update rule without restarting. We divide the VCNN structure into former and latter sections, comprising convolutional and linear layers, respectively, and apply partial restarting to one section. Interestingly, our experiments reveal that partial restarting of the latter section can further enhance generalization performance compared to full restarting, whereas partial restarting of the former section does not yield improvements over the case with no restarting ($r = 0$) [Fig. 5]. Moreover, even when we freeze the latter section by setting $r = 1$, partial restarting of the latter section still achieves significant improvement. We attribute these findings to a well-established observation: the former layers of CNNs tend to learn general features, while the latter layers tend to specialize in capturing specific features [56–59, 54]. In other words, the latter section composed of linear layers exhibits strong memorization of the corrupted data, leading to improved performances even when we freeze the latter section ($r = 1$), whereas the former section composed of convolutional layers focuses on learning general features, leading to no improvements even with restarting.

It is essential to note that although our ablation study suggests the effectiveness of partial restarting, our findings do not imply that partial restarting of the latter section always enhances generalization performance compared to full restarting. The extent to which each layer overfits the corrupted data depends on multiple factors, such as the network structure and the choice of loss function. Thus, determining the most effective section of the network to restart also hinges on the specific context. Future research investigating these points would be intriguing and valuable.

4.4 Results on corrupted benchmark datasets

Finally, we investigate the impact of the stochastic restarting strategy on the performance of benchmark datasets, CIFAR-10 and CIFAR-100, with varying methods (setting 2). Table 1 compares the best results without restarting (No) and with restarting (Restart) at the restart probability $r = 0.001$. In the table, CE denotes cross-entropy loss, PartRestart denotes cross-entropy loss with partial restarting (Sec. 4.3), MAE denotes robust mean absolute error [60], GCE denotes generalized cross-entropy [61], SCE denotes symmetric cross-entropy loss [62], and ELR denotes early-learning regularization [27]. The PartRestart method stochastically restarts only the last linear layer and the last two block layers of ResNet-34, not the entire network. We describe the additional details about the hyper-parameters for each method in the Supplementary Materials.

Remarkably, in all cases examined, our restarting method consistently achieves either at least equivalent or higher test accuracies compared to the baseline approach involving no restarting [Table 1]. Results show that the extent of improvement becomes more pronounced as the noise rate increases, which supports our claim that restarting becomes more advantageous with higher noise rates. Furthermore, while the PartRestart method also obtains improved performance, it does not surpass the benefits of full restarting. We conjecture that the network structure may influence the extent of additional improvements obtained, as the memorization effect in different layers can vary depending on the network architecture [63]. Unfortunately, minimal improvements are observed in the MAE results on both datasets. This is primarily because the MAE convergence is too slow to identify a suitable checkpoint for restarting, consequently resulting in few instances of restarting in many experiments. We also verify the improvement of the validation loss when using our restarting method [Table S.2 in the Supplementary Materials].

Table 1: Test accuracies (%) on test datasets with different methods. We compare the performance without restarting (No) and with restarting (Restart) at $r = 0.001$. Results are presented as the average and the standard deviation. The best results are indicated in **bold** with statistical significance.

Dataset	Method	Noise rate 0.2		Noise rate 0.4		Noise rate 0.6	
		No	Restart	No	Restart	No	Restart
CIFAR-10	CE	84.8 ± 0.4	$\mathbf{90.0 \pm 0.5}^{***}$	80.8 ± 0.7	$\mathbf{86.3 \pm 0.7}^{***}$	72.9 ± 0.7	$\mathbf{79.5 \pm 0.9}^{***}$
	PartRestart	—	$\mathbf{88.5 \pm 0.4}^{***}$	—	$\mathbf{83.7 \pm 0.9}^{***}$	—	$\mathbf{75.1 \pm 0.4}^{***}$
	MAE	91.0 ± 0.2	91.0 ± 0.2	85.6 ± 3.3	$\mathbf{85.9 \pm 3.3}$	67.1 ± 6.8	$\mathbf{67.3 \pm 6.7}$
	GCE	90.6 ± 0.1	$\mathbf{91.0 \pm 0.2}^{*}$	85.3 ± 0.4	$\mathbf{87.1 \pm 0.3}^{***}$	76.2 ± 0.5	$\mathbf{79.1 \pm 0.8}^{***}$
	SCE	91.3 ± 0.3	$\mathbf{91.4 \pm 0.1}$	86.6 ± 0.2	$\mathbf{87.9 \pm 0.2}^{***}$	80.1 ± 0.5	$\mathbf{81.6 \pm 0.6}^{**}$
	ELR	$\mathbf{91.5 \pm 0.2}$	91.4 ± 0.2	87.8 ± 0.4	87.8 ± 0.4	81.1 ± 0.3	81.1 ± 0.3
CIFAR-100	CE	62.7 ± 5.6	$\mathbf{64.5 \pm 1.5}$	45.6 ± 2.3	$\mathbf{56.9 \pm 3.0}^{***}$	32.9 ± 1.6	$\mathbf{44.1 \pm 3.1}^{***}$
	PartRestart	—	$\mathbf{64.0 \pm 0.9}$	—	$\mathbf{55.5 \pm 2.1}^{***}$	—	$\mathbf{43.5 \pm 1.5}^{***}$
	MAE	19.9 ± 2.8	19.9 ± 2.8	$\mathbf{11.0 \pm 3.8}$	10.8 ± 3.7	6.7 ± 1.2	$\mathbf{6.8 \pm 1.3}$
	GCE	68.3 ± 0.4	$\mathbf{69.3 \pm 0.2}^{**}$	61.2 ± 0.6	$\mathbf{63.2 \pm 0.3}^{***}$	50.1 ± 0.6	$\mathbf{53.1 \pm 0.8}^{***}$
	SCE	54.5 ± 1.2	$\mathbf{62.0 \pm 1.0}^{***}$	44.4 ± 1.4	$\mathbf{52.8 \pm 0.7}^{***}$	29.3 ± 2.5	$\mathbf{36.3 \pm 1.8}^{***}$
	ELR	66.4 ± 0.4	$\mathbf{67.2 \pm 1.2}$	57.4 ± 1.0	$\mathbf{60.9 \pm 2.0}^{**}$	47.2 ± 1.4	$\mathbf{48.0 \pm 2.6}$

5 Discussion

In this work, we developed a stochastic restarting method to overcome overfitting in supervised learning with noisy labels, motivated by the success of the restarting strategy in statistical physics. By analyzing SGD dynamics through the lens of Langevin dynamics, we theoretically identified factors that influence the effectiveness of restarting and empirically validated our method as well as the impact of these factors. Our experiments showed that the restarting method consistently yields additional improvements compared to existing methods on benchmark datasets. And as our method can be implemented with minimal code changes and without additional computational costs, flexibility and ease of integration with other approaches are ensured. This simplicity also facilitates extensions to other variants, such as non-Poissonian restarting [64, 65] and state-dependent restart probability [3, 66], etc [2].

The main limitation is that our method may be hard to apply when the double descent phenomenon occurs or the convergence of validation loss is too late, such as the MAE case in Table 1. Moreover, it is hard to identify the optimal restart probability from a few experiments. In fact, it has been observed that the coefficient of variation of the FPT is unity at the optimal restart probability in a random search problem [67, 68]. We believe that investigating whether a similar relationship exists in DNN training will help to identify the optimal restart probability in practice, and will be interesting for future work.

We note that the restarting method shares a similar spirit with forgetting, which refers to the loss of previously acquired knowledge [69]. Similar to restarting, forgetting was initially viewed as a catastrophic phenomenon that needed to be addressed [70, 71]. However, recent studies have highlighted its benefits, leading to its use in improving network performance [72, 73]. From this perspective, restarting can be viewed as a form of forgetting memorized task-specific patterns, but unlike general forgetting that primarily erases early experiences, restarting targets the erasure of later experiences. We hope that connections between restarting and forgetting will be further explored in future discussions.

We anticipate that our work will contribute valuable insights to two different research directions. First, we hope our work will pave the way to applying various new search strategies, beyond restarting, into neural network training. It also opens up the possibility of analyzing existing training methods from a statistical physics perspective.

Acknowledgments and Disclosure of Funding

This study was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF Grant No. 2022R1A2B5B02001752). Y.B. was supported by an NRF grant funded by the Korean government (MSIT) (No. RS-2023-00278985).

References

[1] M. R. Evans and S. N. Majumdar, "Diffusion with stochastic resetting," Phys. Rev. Lett., vol. 106, p. 160601, 2011.

[2] M. R. Evans, S. N. Majumdar, and G. Schehr, "Stochastic resetting and applications," J. Phys. A: Math. Theor., vol. 53, no. 19, p. 193001, 2020.

[3] M. R. Evans and S. N. Majumdar, "Diffusion with optimal resetting," J. Phys. A: Math. Theor., vol. 44, no. 43, p. 435001, 2011.

[4] M. R. Evans and S. N. Majumdar, "Diffusion with resetting in arbitrary spatial dimension," J. Phys. A: Math. Theor., vol. 47, no. 28, p. 285001, 2014.

[5] S. Ray, D. Mondal, and S. Reuveni, "Péclet number governs transition to acceleratory restart in drift-diffusion," J. Phys. A: Math. Theor., vol. 52, no. 25, p. 255002, 2019.

[6] D. Gupta, "Stochastic resetting in underdamped brownian motion," J. Stat. Mech., vol. 2019, no. 3, p. 033212, 2019.

[7] A. Pal, L. Kuśmierz, and S. Reuveni, "Search with home returns provides advantage under high uncertainty," Phys. Rev. Res., vol. 2, p. 043174, 2020.

[8] O. Tal-Friedman, A. Pal, A. Sekhon, S. Reuveni, and Y. Roichman, "Experimental realization of diffusion with stochastic resetting," J. Phys. Chem. Lett., vol. 11, no. 17, pp. 7350–7355, 2020.

[9] S. Ray and S. Reuveni, "Diffusion with resetting in a logarithmic potential," J. Chem. Phys., vol. 152, no. 23, p. 234110, 2020.

[10] B. De Bruyne, S. N. Majumdar, and G. Schehr, "Optimal resetting brownian bridges via enhanced fluctuations," Phys. Rev. Lett., vol. 128, p. 200603, 2022.

[11] A. Nagar and S. Gupta, "Stochastic resetting in interacting particle systems: a review," J. Phys. A: Math. Theor., vol. 56, no. 28, p. 283001, 2023.

[12] O. Blumer, S. Reuveni, and B. Hirshberg, "Stochastic resetting for enhanced sampling," J. Phys. Chem. Lett., vol. 13, no. 48, pp. 11230–11236, 2022.

[13] O. Blumer, S. Reuveni, and B. Hirshberg, "Combining stochastic resetting with metadynamics to speed-up molecular dynamics simulations," Nat. Commun., vol. 15, no. 1, p. 240, 2024.

[14] O. L. Bonomo, A. Pal, and S. Reuveni, "Mitigating long queues and waiting times with service resetting," PNAS Nexus, vol. 1, no. 3, p. pgac070, 2022.

[15] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[16] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, "A survey of label-noise representation learning: Past, present and future," arXiv preprint arXiv:2011.04406, 2020.

[17] W. Hu, Z. Li, and D. Yu, "Simple and effective regularization methods for training on noisily labeled data with generalization guarantee," in International Conference on Learning Representations, 2020.

[18] S. Li, X. Xia, S. Ge, and T. Liu, "Selective-supervised contrastive learning with noisy labels," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 316–325, 2022.

[19] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 11, pp. 8135–8153, 2023.

[20] X. Xia, B. Han, Y. Zhan, J. Yu, M. Gong, C. Gong, and T. Liu, "Combating noisy labels with sample selection by mining high-discrepancy examples," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1833–1843, 2023.

[21] Z. Huang, J. Zhang, and H. Shan, "Twin contrastive learning with noisy labels," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11661–11670, 2023.

[22] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in International Conference on Learning Representations, 2017.

[23] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in Proceedings of the 34th International Conference on Machine Learning, vol. 70 of Proceedings of Machine Learning Research, pp. 233–242, PMLR, 2017.

[24] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, vol. 108 of Proceedings of Machine Learning Research, pp. 4313–4324, PMLR, 2020.

[25] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in Advances in Neural Information Processing Systems, vol. 31, Curran Associates, Inc., 2018.

[26] H. Song, M. Kim, and J.-G. Lee, "SELFIE: Refurbishing unclean samples for robust deep learning," in Proceedings of the 36th International Conference on Machine Learning, vol. 97 of Proceedings of Machine Learning Research, pp. 5907–5915, PMLR, 2019.

[27] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in Advances in Neural Information Processing Systems, vol. 33, pp. 20331–20342, Curran Associates, Inc., 2020.

[28] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, "Robust early-learning: Hindering the memorization of noisy labels," in International Conference on Learning Representations, 2021.

[29] O. G. Berg, R. B. Winter, and P. H. Von Hippel, "Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. models and theory," Biochemistry, vol. 20, no. 24, pp. 6929–6948, 1981. PMID: 7317363.

[30] M. Coppey, O. Bénichou, R. Voituriez, and M. Moreau, "Kinetics of target site localization of a protein on dna: A stochastic approach," Biophys. J., vol. 87, no. 3, pp. 1640–1649, 2004.

[31] S. Ghosh, B. Mishra, A. B. Kolomeisky, and D. Chowdhury, "First-passage processes on a filamentous track in a dense traffic: optimizing diffusive search for a target in crowding conditions," J. Stat. Mech., vol. 2018, no. 12, p. 123209, 2018.

[32] F. Bartumeus and J. Catalan, "Optimal search behavior and classic foraging theory," J. Phys. A: Math. Theor., vol. 42, no. 43, p. 434002, 2009.

[33] G. M. Viswanathan, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, The Physics of Foraging: An Introduction to Random Searches and Biological Encounters. Cambridge University Press, 2011.

[34] O. Dieste, A. Grimán, and N. Juristo, "Developing search strategies for detecting relevant experiments," Empirical Software Engineering, vol. 14, no. 5, pp. 513–539, 2009.

[35] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Comput. Netw. ISDN Syst., vol. 30, no. 1, pp. 107–117, 1998. Proceedings of the Seventh International World Wide Web Conference.

[36] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," arXiv preprint arXiv:2003.05689, 2020.

[37] S. Redner, A Guide to First-Passage Processes. Cambridge University Press, England, 2001.

[38] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," Nature, vol. 401, no. 6756, pp. 911–914, 1999.

[39] M. A. Lomholt, K. Tal, R. Metzler, and K. Joseph, "Lévy strategies in intermittent search processes are advantageous," Proc. Natl. Acad. Sci. U.S.A, vol. 105, no. 32, pp. 11055–11059, 2008.

[40] D. J. Amit, G. Parisi, and L. Peliti, "Asymptotic behavior of the "true" self-avoiding walk," Phys. Rev. B, vol. 27, pp. 1635–1645, 1983.

[41] P. Grassberger, "Self-trapping self-repelling random walks," Phys. Rev. Lett., vol. 119, p. 140601, 2017.

[42] O. Bénichou, C. Loverdo, M. Moreau, and R. Voituriez, "Intermittent search strategies," Rev. Mod. Phys., vol. 83, pp. 81–129, 2011.

[43] V. Tejedor, R. Voituriez, and O. Bénichou, "Optimizing persistent random searches," Phys. Rev. Lett., vol. 108, p. 088103, 2012.

[44] H. Meyer and H. Rieger, "Optimal non-markovian search strategies with n-step memory," Phys. Rev. Lett., vol. 127, p. 070601, 2021.

[45] Y. Bae, G. Son, and H. Jeong, "Unexpected advantages of exploitation for target searches in complex networks," Chaos, vol. 32, no. 8, 2022. 083118.

[46] Q. Li, C. Tai, and W. E, "Stochastic modified equations and adaptive stochastic gradient algorithms," in Proceedings of the 34th International Conference on Machine Learning, vol. 70 of Proceedings of Machine Learning Research, pp. 2101–2110, PMLR, 2017.

[47] S. L. Smith and Q. V. Le, "A bayesian perspective on generalization and stochastic gradient descent," in 6th International Conference on Learning Representations ICLR, 2018.

[48] L. Ziyin, K. Liu, T. Mori, and M. Ueda, "Strength of minibatch noise in SGD," in International Conference on Learning Representations, 2022.

[49] C. W. Gardiner, Handbook of Stochastic Methods. Springer Berlin, Heidelberg, 2004.

[50] B. Barz and J. Denzler, "Do we train on test data? purging cifar of near-duplicates," J. Imag., vol. 6, no. 6, 2020.

[51] A. Krizhevsky, G. Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[53] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in International Conference on Learning Representations, 2020.

[54] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu, "Understanding and improving early stopping for learning with noisy labels," in Advances in Neural Information Processing Systems, vol. 34, pp. 24392–24403, Curran Associates, Inc., 2021.

[55] Y. Chen, A. Yuille, and Z. Zhou, "Which layer is learning faster? a systematic exploration of layer-wise convergence rate for deep neural networks," in The Eleventh International Conference on Learning Representations, 2023.

[56] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems, vol. 27, Curran Associates, Inc., 2014.

[57] G. Cohen, G. Sapiro, and R. Giryes, "Dnn or k-nn: That is the generalize vs. memorize question," arXiv preprint arXiv:1805.06822, 2018.

[58] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan, "Intrinsic dimension of data representations in deep neural networks," in Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.

[59] H. Maennel, I. M. Alabdulmohsin, I. O. Tolstikhin, R. Baldock, O. Bousquet, S. Gelly, and D. Keysers, "What do neural networks learn when trained with random labels?," in Advances in Neural Information Processing Systems, vol. 33, pp. 19693–19704, Curran Associates, Inc., 2020.

[60] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, p. 1919–1925, AAAI Press, 2017.

[61] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, (Red Hook, NY, USA), p. 8792–8802, Curran Associates Inc., 2018.

[62] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[63] J. Li, M. Zhang, K. Xu, J. Dickerson, and J. Ba, "How does a neural network's architecture impact its robustness to noisy labels?," in Advances in Neural Information Processing Systems, vol. 34, pp. 9788–9803, Curran Associates, Inc., 2021.

[64] A. Pal, A. Kundu, and M. R. Evans, "Diffusion under time-dependent resetting," J. Phys. A: Math. Theor., vol. 49, no. 22, p. 225001, 2016.

[65] A. Nagar and S. Gupta, "Diffusion with stochastic resetting at power-law times," Phys. Rev. E, vol. 93, p. 060102, 2016.

[66] E. Roldán and S. Gupta, "Path-integral formalism for stochastic resetting: Exactly solved examples and shortcuts to confinement," Phys. Rev. E, vol. 96, p. 022130, 2017.

[67] S. Reuveni, "Optimal stochastic restart renders fluctuations in first passage times universal," Phys. Rev. Lett., vol. 116, p. 170601, 2016.

[68] A. Pal and S. Reuveni, "First passage under restart," Phys. Rev. Lett., vol. 118, p. 030603, 2017.

[69] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," arXiv preprint arXiv:2307.09218, 2023.

[70] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," vol. 24 of Psychology of Learning and Motivation, pp. 109–165, Academic Press, 1989.

[71] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," Neural Netw., vol. 113, pp. 54–71, 2019.

[72] H. Zhou, A. Vani, H. Larochelle, and A. Courville, "Fortuitous forgetting in connectionist networks," in International Conference on Learning Representations, 2022.

[73] E. Nikishin, M. Schwarzer, P. D'Oro, P.-L. Bacon, and A. Courville, "The primacy bias in deep reinforcement learning," in Proceedings of the 39th International Conference on Machine Learning (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of Proceedings of Machine Learning Research, pp. 16828–16847, PMLR, 2022.

[74] R. Metzler, J.-H. Jeon, A. G. Cherstvy, and E. Barkai, "Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking," Phys. Chem. Chem. Phys., vol. 16, pp. 24128–24164, 2014.

[75] M. Villén-Altamirano, J. Villén-Altamirano, et al., "Restart: a method for accelerating rare event simulations," Queueing, Performance and Control in ATM (ITC-13), pp. 71–76, 1991.

[76] M. Luby, A. Sinclair, and D. Zuckerman, "Optimal speedup of las vegas algorithms," Inf. Process. Lett., vol. 47, no. 4, pp. 173–180, 1993.

[77] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," Knowl. Inf. Syst., vol. 14, pp. 327–346, Mar 2008.

[78] A. Pal, "Diffusion in a potential landscape with stochastic resetting," Phys. Rev. E, vol. 91, p. 012113, Jan 2015.

[79] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in International Conference on Learning Representations, 2017.

A SGD dynamics and Langevin dynamics

In this section, we explain how the dynamics of SGD can be converted to the discretized Langevin equation. We follow similar procedures as in Refs. [46–48]. As written in the main text, the network parameters $\boldsymbol{\theta}$ are updated by

$$\begin{aligned}\Delta\boldsymbol{\theta}_t &= -\frac{\eta}{B}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{B}_t}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\\ &= -\frac{\eta}{N_{\rm tr}}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{D}_{\rm tr}}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)+\left(\frac{\eta}{N_{\rm tr}}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{D}_{\rm tr}}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)-\frac{\eta}{B}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in\mathcal{B}_t}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\right)\\ &= -\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\eta+\boldsymbol{\xi}_t\sqrt{\eta},\end{aligned} \tag{S.1}$$

where the random noise vector $\boldsymbol{\xi}_t \equiv \sqrt{\eta}\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\right)\in\mathbb{R}^d$, $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)$ is the gradient of risk on $\mathcal{D}_{\rm tr}$, and $\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)$ is the gradient of risk on a minibatch $\mathcal{B}_t$. Note that $\langle\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\rangle_{\mathcal{D}_{\rm tr}}=\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)$ and $\langle\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\rangle_{\mathcal{D}_{\rm tr}}=\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)$. Using these facts, we can obtain that $\langle\boldsymbol{\xi}_t\rangle_{\mathcal{D}_{\rm tr}}=\mathbf{0}$ and

$$\begin{aligned}\langle\boldsymbol{\xi}_t\boldsymbol{\xi}_s^{\rm T}\rangle_{\mathcal{D}_{\rm tr}} &= \eta\left\langle\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\right)\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_s)-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_s}(\boldsymbol{\theta}_s)\right)^{\rm T}\right\rangle_{\mathcal{D}_{\rm tr}}\\ &= \eta\left(\left\langle\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_s}(\boldsymbol{\theta}_s)^{\rm T}\right\rangle_{\mathcal{D}_{\rm tr}}-\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_s)^{\rm T}\right).\end{aligned} \tag{S.2}$$

For the $s\neq t$ case, $\left\langle\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_s}(\boldsymbol{\theta}_s)^{\rm T}\right\rangle_{\mathcal{D}_{\rm tr}}=\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_s)^{\rm T}$ because each minibatch is independently sampled from $\mathcal{D}_{\rm tr}$. Applying $\langle\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}_t)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}_j(\boldsymbol{\theta}_t)\rangle_{\mathcal{D}_{\rm tr}}=\|\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\|^2+\Sigma(\boldsymbol{\theta}_t)\delta_{ij}$ with the covariance matrix $\Sigma(\boldsymbol{\theta}_t)$, where $\|\cdot\|$ denotes the Euclidean norm of a vector, we have

$$\left\langle\|\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{B}_t}(\boldsymbol{\theta}_t)\|^2\right\rangle_{\mathcal{D}_{\rm tr}}=\|\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{R}_{\mathcal{D}_{\rm tr}}(\boldsymbol{\theta}_t)\|^2+\frac{1}{B}\Sigma(\boldsymbol{\theta}_t). \tag{S.3}$$

Therefore, we obtain $\langle\boldsymbol{\xi}_t\boldsymbol{\xi}_s^{\rm T}\rangle_{\mathcal{D}_{\rm tr}}=2\mathsf{D}(\boldsymbol{\theta}_t)\delta_{ts}$ with $\mathsf{D}(\boldsymbol{\theta}_t)=\eta\Sigma(\boldsymbol{\theta}_t)/(2B)$. We assumed $N_{\rm tr}\gg B$ in the above derivation, but the decreasing trend of $\mathsf{D}(\boldsymbol{\theta}_t)$ with B is still valid for $N_{\rm tr}\geq B$ [48].

Let us consider the overdamped Langevin equation, a first-order stochastic differential equation describing the evolution of a particle where friction dominates over inertia. Applying the Euler method, the overdamped Langevin equation can be approximated with time interval Δt by [49]

$$\Delta\boldsymbol{x}_t=-\boldsymbol{\nabla}_{\boldsymbol{x}}V(\boldsymbol{x}_t)\Delta t+\sqrt{2\mathsf{B}(\boldsymbol{x}_t)}\Delta\boldsymbol{W}_t. \tag{S.4}$$

Here, $\boldsymbol{x}_t$ is the position of the particle at time step t, $V(\boldsymbol{x}_t)$ is the underlying potential, $\mathsf{B}(\boldsymbol{x}_t)$ is the strength of the fluctuations, called the diffusion matrix, and $\Delta\boldsymbol{W}_t$ is the random noise vector that satisfies $\langle\Delta\boldsymbol{W}_t\rangle=\mathbf{0}$ and $\langle\Delta\boldsymbol{W}_t\Delta\boldsymbol{W}_s^{\rm T}\rangle=\Delta t\delta_{ts}$, where $\langle\cdot\rangle$ denotes the ensemble average. When we simulate Eq. (S.4), we randomly sample the random real number $\boldsymbol{\zeta}_t$ from a certain probability distribution with zero-mean and unit-variance at each iteration t and represent the noise vector as $\Delta\boldsymbol{W}_t\equiv\boldsymbol{\zeta}_t\sqrt{\Delta t}$. Note that $\Delta\boldsymbol{W}_t$ is commonly assumed as an increment of the Wiener process based on the central limit theorem, so that $\boldsymbol{\zeta}_t$ is generally sampled from the standard normal distribution. This assumption is often violated in various sitautions [74], and $\boldsymbol{\zeta}_t$ can be sampled from other distributions depending on the system. Comparing Eq. (S.1) with Eq. (S.4), we can easily see that the dynamics of SGD follows the overdamped Langevin equation, with $\mathcal{R}_{\mathcal{D}_{\rm tr}}$ and D serving as the potential and the diffusion matrix, respectively. Based on this correspondence, we analyze the SGD dynamics in the language of the Langevin dynamics and introduce the stochastic restarting strategy in the main text.

B Stochastic restarting in statistical physics

We briefly introduce what stochastic restarting is and the conditions under which this strategy can help random searches (see Ref. [2] for a more detailed review). In fact, the restarting method has already been exploited in some stochastic algorithms [75–77], but much attention has been attracted by the theoretical success of stochastic restarting [1]. Several approaches have been made to deal with

search processes under restart, such as calculating the survival probability [1, 5]. Here, we present an easy but general one: we clarify that we follow the same procedure as in Refs. [67, 68, 7].

Let us consider a generic searcher that starts from a restarting point at time zero in a d-dimensional space, with the assumption that the searcher restarted at a rate γ if the target is not found. In other words, the search process is completed when the searcher finds the target before restarting; otherwise, the searcher returns to the restarting point and repeats this procedure until the target is found. This procedure can be understood in terms of two random variables, T and R, which represent the time to find a target and the time to restart, respectively. If we draw T and R from their respective distributions, we check whether $T > R$. If $T > R$, the searcher restarts before finding the target, and the search process begins anew from the restarting point. Conversely, if $T < R$, the searcher finds the target before restarting, and the search process is completed. Applying this scheme, the time to find a target, known as the first passage time (FPT) and denoted by $T(\gamma)$, can be expressed by the following renewal equation:

$$T(\gamma) = \begin{cases} T, & \text{if } T < R, \\ R + T(\gamma)', & \text{if } R \leq T, \end{cases} \tag{S.5}$$

or equivalently,

$$T(\gamma) = \min(T, R) + I(R \leq T)T(\gamma)', \tag{S.6}$$

where $T(\gamma)'$ denotes an independent and identically distributed copy of $T(\gamma)$, and $I(R \leq T)$ denotes an indicator function which equals one if $R \leq T$ and zero otherwise. Note that $T(\gamma) = T$ if there is no restart ($\gamma = 0$). Taking expectations in Eq. (S.6), we obtain the mean first passage time (MFPT):

$$\langle T(\gamma) \rangle = \frac{\langle \min(T, R) \rangle}{\Pr(T < R)}, \tag{S.7}$$

with the relations $\langle I(R \leq T) \rangle = \Pr(R \leq T)$ and $\langle T(\gamma) \rangle = \langle T(\gamma)' \rangle$. It is noteworthy that we do not assume the dynamics of the search process and the function of the restart rate. Thus, Eq. (S.7) can be applied to a general search process regardless of the distributions of T and R.

As a simple restart method, we assume a constant restart rate, i.e., the restart probability within a time interval dt is γdt [1]. Then, the distribution of R is exponential with $\gamma e^{-\gamma t}$ at time t and $\langle \min(T, R) \rangle$ can be calculated by

$$\begin{aligned} \langle \min(T, R) \rangle &= \int_0^\infty dt \, [1 - \Pr(\min(T, R) \leq t)] \\ &= \int_0^\infty dt \, \Pr(R > t) \Pr(T > t) \\ &= \int_0^\infty dt \, e^{-rt} \int_t^\infty dt' f_T(t') = \frac{1}{r} - \frac{1}{r} \int_0^\infty dt \, e^{-rt} f_T(t). \end{aligned} \tag{S.8}$$

In addition, $\Pr(T < R) = \int_0^\infty dt \;\; f_T(t) \Pr(R > t) = \int_0^\infty dt \;\; e^{-rt} f_T(t)$. Substituting these equations into Eq. (S.7), we obtain

$$\langle T(\gamma) \rangle = \frac{1 - \tilde{T}(\gamma)}{\gamma \tilde{T}(\gamma)} \tag{S.9}$$

where $\tilde{T}(\gamma) \equiv \int_0^\infty dt e^{-\gamma t} f_T(t)$ denotes the Laplace transform of T evaluated at γ. Note that $f_T(t)$ is determined by the underlying dynamics of the searcher, which may include factors such as stochasticity or external drift. Therefore, once the dynamics of a searcher are determined and $f_T(t)$ is known, we can calculate the MFPT at γ and identify whether restarting is beneficial.

To obtain the MFPT with restarting, let us specify the dynamics of a searcher. Consider a searcher diffusing in one dimension with a diffusion constant D and a constant drift v and assume that the searcher starts at the origin (restart point) and that the target we want to find is located at L (> 0). Here, D represents the stochasticity and v represents the drift toward a target. The position $x(t)$ of the searcher at time t evolves during a small time interval Δt through the Langevin equation given by

$$x(t + \Delta t) = \begin{cases} x_r, & \text{with probability } \gamma \Delta t, \\ x(t) + v\Delta t + \xi(t)\sqrt{\Delta t}, & \text{otherwise} \end{cases} \tag{S.10}$$

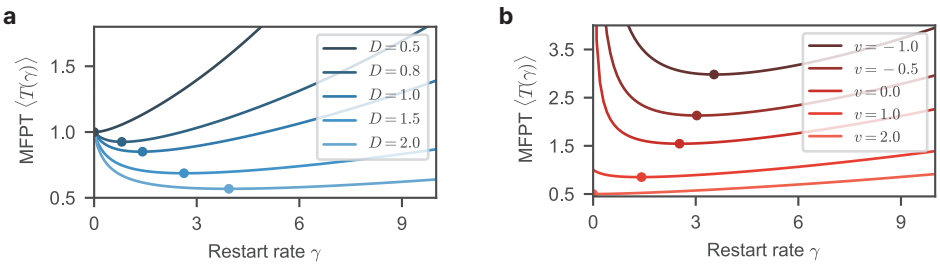

Figure S.1: The mean first passage time (MFPT) $\langle T(\gamma)\rangle$ with varying (a) the diffusion constant D and (b) the drift v with respec to the restart rate γ [see Eq. (S.13)]. Markers represent the minimum MFPT, $\langle T(\gamma^*)\rangle$. We set $v = 1$ in (a), $D = 1$ in (b), and $L = 1$ in both.

where x_r denotes the restart point set as the origin and $\xi(t)$ denotes a stochastic force, typically modeled Gaussian white noise satisfying $\langle \xi(t)\rangle = 0$ and $\langle \xi(t)\xi(s)\rangle = 2D\delta(t-s)$. In this search process, the probability distribution to find the searcher at position x at time t is known to be given by [49]

$$G_0(x,t) = \frac{1}{\sqrt{4\pi D t}}\left[e^{-\frac{(x-vt)^2}{4Dt}} - e^{\frac{Lv}{D}} e^{-\frac{(x-2L-vt)^2}{4Dt}}\right]. \tag{S.11}$$

The FPT distribution $f_T(t)$ is then derived from the probability that the searcher has not yet reached the target by time t: $f_T(t) = \frac{d}{dt}\Pr(T \geq t) = \frac{d}{dt}\int_{-\infty}^{L} dx\, G_0(x,t)$. Using this equation and the definition of $\tilde{T}(\gamma)$, we find $\tilde{T}(\gamma) = 1 - \gamma\int_{-\infty}^{L} dx\, \tilde{G}_0(x,\gamma)$ where $\tilde{G}_0(x,\gamma) \equiv \int_0^\infty dt e^{-\gamma t} G_0(x,t)$ is the Laplace transform of $G_0(x,t)$ evaluated at γ. Thus, upon calculation with these equations and Eq. (S.11), we obtain

$$\tilde{T}(\gamma) = e^{\frac{L}{2D}\left(v - \sqrt{v^2+4D\gamma}\right)}. \tag{S.12}$$

Substituting Eq. (S.12) into Eq. (S.9), we finally have

$$\langle T(\gamma)\rangle = \frac{1}{\gamma}\left[e^{\frac{L}{2D}\left(\sqrt{v^2+4D\gamma} - v\right)} - 1\right]. \tag{S.13}$$

This final expression gives the MFPT for a searcher with drift and diffusion, including the effect of restarting at a rate γ.

Examining Eq. (S.13) provides several insights into stochastic restarting. When $v \leq 0$ (i.e., the drift is zero or acts in the opposite direction to the target), $\langle T(\gamma)\rangle$ is infinite in the limit $\gamma \to 0$ but becomes finite for any finite γ. Conversely, when $v > 0$ (i.e., the drift is towards the target), $\langle T(\gamma)\rangle$ decreases with increasing γ if $\mathrm{Pe} \equiv Lv/(2D) \leq 1$. Here, Pe is the Péclet number, which represents the ratio between drift and diffusive transport rates, and the beneficial condition ($\mathrm{Pe} \leq 1$) can be identified by verifying where $[d\langle T(\gamma)\rangle/d\gamma]|_{\gamma\to 0} < 0$. Figure S.1 shows how the curve of $\langle T(\gamma)\rangle$ changes with varying Pe: the monotonically increasing curve gradually transforms into a U-shaped curve as Pe decreases and has a minimum $\langle T(\gamma)\rangle$ at the optimal restart rate $\gamma^* > 0$. These results indicate that restarting can be beneficial for target search when the stochasticity (D) is sufficiently larger than the drift toward a target (v), as we provided the statement highlighted in Sec. 3.2. Furthermore, we note that the optimal restart rate γ^* and the ratio $\langle T(0)\rangle/\langle T(\gamma^*)\rangle$ increase as Pe decreases [Fig. S.1], implying that the benefits of restarting increase as D increases and v decreases.

In our paper, we exploit the correspondence between SGD dynamics and Langevin dynamics, as described in Sec. A, and theoretically identify where stochasticity is strengthened and drift toward a target is weakened in DNN training. Although we showcased a simple one-dimensional case in this section, many studies have explored more complex scenarios including high-dimensional spaces [3], various confining potentials [78, 7], etc [2]. We believe that these studies support the possibility that restarting may provide advantages in the search process of SGD dynamics. Additionally, we acknowledge that the advantages of restarting in terms of MFPT are not directly connected to performance improvements in DNN training. We conjecture and empirically validate that similar beneficial mechanisms successfully operate in DNN training. However, it would be necessary and intriguing to find an alternative metric to represent the DNN performance and to theoretically investigate the effect of restarting.

C Description of the experiments

This section provides details on the experiments not included in the main text. For all of the experiments, we perform 5 independent runs to achieve the average and the standard error values. All runs were made independently on a single NVIDIA TITAN V GPU. All results are obtained from the model at the optimal iteration based on minimum validation loss as default. Additionally, the resulting test accuracy is evaluated on the clean validation set. The objective throughout our experiments is to compare the performance with and without restarting, not to achieve state-of-the-art performances, we did not heavily tune the hyper-parameters for each of the settings. Here, we provide a choice of the hyper-parameters for our experiments.

C.1 Network architecture

We employed a vanilla CNN (VCNN) as mentioned in Sec. 4.1, 4.2, and 4.3, to expedite a straightforward testing of our claims. It consists of simple layers, as outlined in Table S.1. In the table, the inclusion of batch normalization before the activation function and after a layer is indicated by the term "Use BatchNorm". The output dimension of a convolutional layer is represented as (C, W, H), where C denotes the number of channels, and W and H represent the width and height, respectively.

Table S.1: Network architecture of the VCNN: Layer name, output dimension of the layer, parameters of the convolutional layer (K, P, S), and activation function. Here, K, P, and S represent the size of the filter, padding, and stride, respectively.

VCNN architecture				
Layer name	Output dim	(K, P, S)	Use BatchNorm	Activation function
Input image	$(3, 32, 32)$	None	X	None
Conv2d	$(32, 32, 32)$	$(3, 1, 1)$	O	ReLU
Conv2d	$(64, 32, 32)$	$(3, 1, 1)$	O	ReLU
MaxPool2d	$(64, 16, 16)$	$(2, 0, 2)$	O	None
Conv2d	$(128, 16, 16)$	$(3, 1, 1)$	O	ReLU
Conv2d	$(128, 16, 16)$	$(3, 1, 1)$	O	ReLU
MaxPool2d	$(128, 8, 8)$	$(2, 0, 2)$	X	None
Conv2d	$(256, 8, 8)$	$(3, 1, 1)$	O	ReLU
Conv2d	$(256, 8, 8)$	$(3, 1, 1)$	O	ReLU
MaxPool2d	$(256, 4, 4)$	$(2, 0, 2)$	X	None
Dropout ($p = 0.2$)	$256 \times 4 \times 4$	None	X	None
Linear	1024	None	O	ReLU
Linear	512	None	O	ReLU
Linear	c	None	X	None

C.2 Experimental setting for results on benchmark datasets in Sec. 4.4

We set the hyper-parameters $q = 0.7$ and $k = 0$ for the GCE method [61]; we set $\alpha = 0.1$, $\beta = 1.0$ on CIFAR-10 and $\alpha = 6.0$, $\beta = 0.1$ on CIFAR-100 for the SCE method [62]; and we set $\lambda = 3$, $\beta = 0.7$ on CIFAR-10 and $\lambda = 7$, $\beta = 0.9$ on CIFAR-100 for the ELR method [27]. We set the batch size as 256, and weight decay as 5×10^{-4} for the cross-entropy loss, MAE, GCE, and the SCE method. For the ELR method, we set the batch size as 128, and weight decay as 10^{-3} as indicated in Ref. [27]. We employ a cosine annealing scheduler [79], setting the maximum number of iterations to the total iteration 5×10^4 with an initial learning rate of 0.1 for the cross-entropy loss and 0.01 for MAE [60], GCE, and the SCE method. For the ELR method, we set the initial learning rate as 0.02, and reduce it by $1/100$ after 17000 and 34000 iterations while keeping the total iteration to 5×10^4. In addition, for the ELR method, we set the restart checkpoint and select the best model based on validation accuracy instead of validation loss due to the discrepancy in the resulting test accuracies between these accuracy-based and loss-based approaches in the baseline.

D Additional results

D.1 Cosine similiaries between drifts in Sec. 3

We provide the cosine similarities between drift components, i.e., $-\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\tilde{\mathcal{D}}_{\text{tr}}}(\boldsymbol{\theta})$ (total drift), $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta})$ (drift from the correct part), and $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\text{tr}}}(\boldsymbol{\theta})$ (drift from the wrong part) with varying the noise rate in Fig. S.2. Here, we can see that the cosine similarity between $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta}_t)$ and $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\text{tr}}}(\boldsymbol{\theta}_t)$, denoted by $\cos\phi_{cw}$, is almost zero, indicating that $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta}_t)$ and $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\text{tr}}}(\boldsymbol{\theta}_t)$ are most likely to be orthogonal to each other due to the high-dimensionality of the network parameters. In addition, the cosine similarity between $-\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\tilde{\mathcal{D}}_{\text{tr}}}(\boldsymbol{\theta}_t)$ and $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta}_t)$, denoted by $\cos\phi_{tc}$, decreases with increasing τ, implying that $-\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\tilde{\mathcal{D}}_{\text{tr}}}(\boldsymbol{\theta}_t)$ becomes less correlated with $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta}_t)$. These results support our claims in Sec. 3 and the schematic in Fig. 2(a).

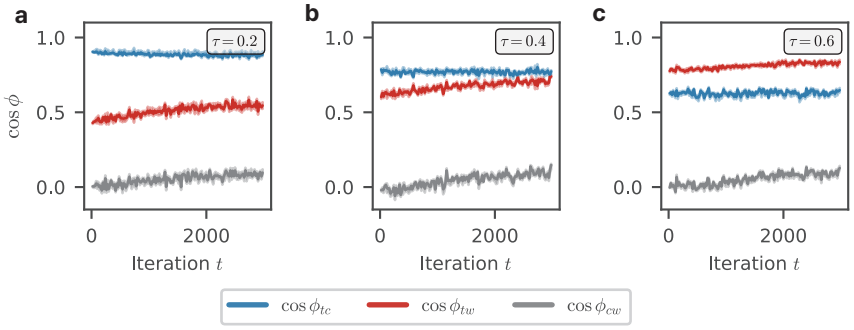

Figure S.2: Cosine similarities between $-\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\tilde{\mathcal{D}}_{\text{tr}}}(\boldsymbol{\theta})$ (total drift), $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^c_{\text{tr}}}(\boldsymbol{\theta})$ (drift from the correct part, i.e., correct drift), and $-\nabla_{\boldsymbol{\theta}} \hat{\mathcal{R}}_{\tilde{\mathcal{D}}^w_{\text{tr}}}(\boldsymbol{\theta})$ (drift from the wrong part, i.e., wrong drift) with (a) the noise rate $\tau = 0.2$, (b) $\tau = 0.4$, and (c) $\tau = 0.6$. Here, $\cos\phi_{tc}$, $\cos\phi_{tw}$, and $\cos\phi_{cw}$ denote the cosine similarity between total and correct drifts, total and correct drifts, and total and correct drifts, respectively. We set $B = 8$ in setting 1.

D.2 Unnormalized plots in Sec. 4.1, 4.2, and 4.3

In order to facilitate comparison between the restarting method and the original SGD, we normalized the validation loss and test accuracy values in the main text by calculating the relative difference in the metrics (RDVLoss and RDVAcc.). Here, we provide the unnormalized (i.e., original) validation loss and test accuracy values: Figs. S.3, S.4, and S.5 in the Supplementary Materials are the unnormalized results of Figs. 3, 4, and 5 in the main text, respectively.

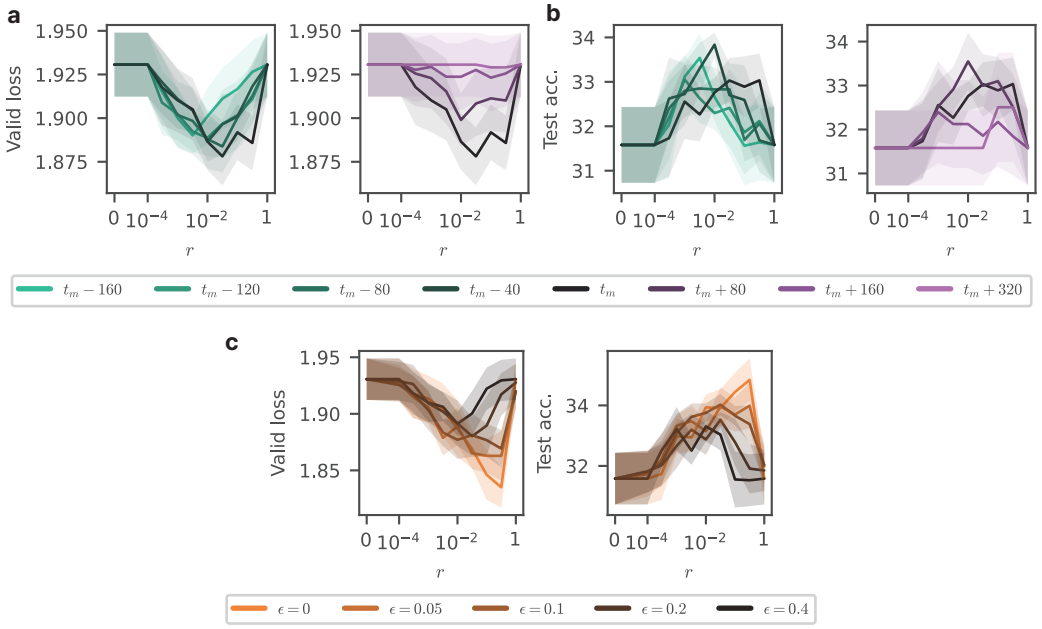

Figure S.3: (a) Validation loss and (b) test accuracy results with varying the checkpoint to restart with respect to the restart probability r. Based on the checkpoint at the overfitting iteration t_m, the results are obtained in earlier iterations (left) and later iterations than t_m (right). Here, $t_m + \delta t$ denotes the iteration where the checkpoint is selected. (c) Validation loss (left) and test accuracy (right) with the perturbed checkpoint parameters $\boldsymbol{\theta}_{c,\epsilon}$. Here, $\boldsymbol{\theta}_{c,\epsilon} \equiv \boldsymbol{\theta}_c + \epsilon \hat{\boldsymbol{n}}$ where $\boldsymbol{\theta}_c$ denotes the checkpoint and $\hat{\boldsymbol{n}}$ denotes a random unit vector. The shaded areas denote the standard error.

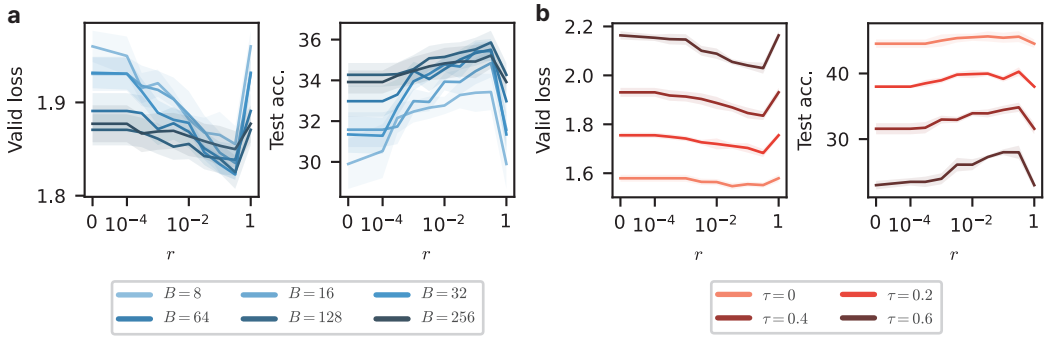

Figure S.4: Validation loss and test accuracy results with (a) varying the batch size B, and (b) varying the noise rate τ with respect to the restart probability r. We set $\tau = 0.4$ in (a) and $B = 16$ in (b). The shaded areas denote the standard error.

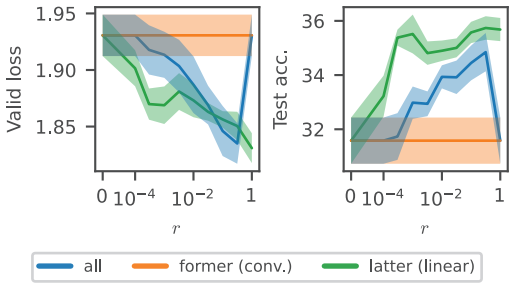

Figure S.5: Validation loss and test accuracy results with varying one section of the network to restart with respect to the restart probability r. Here, we set $\tau = 0.4$ and $B = 16$. The shaded areas denote the standard error.

D.3 Validation loss results in Sec. 4.4

Table S.2 presents the validation loss results from the corresponding models used in Table 1 in the main text. Similar to the test accuracy results in Table 1, Table S.2 shows that our restarting method consistently achieves either at least equivalent or higher validation losses compared to the baseline approach involving no restarting.

Table S.2: Validation losses with different methods. We compare the performance without restarting (No) and with restarting (Restart) at $r = 0.001$. Results are presented as the average and the standard deviation. The best results are indicated in **bold** with statistical significance.

Dataset	Method	Noise rate 0.2		Noise rate 0.4		Noise rate 0.6	
		No	Restart	No	Restart	No	Restart
CIFAR-10	CE	1.231 ± 0.018	$\mathbf{1.164 \pm 0.015}^{***}$	1.774 ± 0.010	$\mathbf{1.734 \pm 0.018}^{**}$	2.124 ± 0.009	$\mathbf{2.102 \pm 0.012}^{**}$
	PartRestart	—	$\mathbf{1.184 \pm 0.012}^{**}$	—	$\mathbf{1.748 \pm 0.014}^{*}$	—	$\mathbf{2.114 \pm 0.011}$
	MAE	0.545 ± 0.009	0.545 ± 0.009	0.968 ± 0.039	$\mathbf{0.963 \pm 0.039}$	1.422 ± 0.042	$\mathbf{1.420 \pm 0.042}$
	GCE	0.388 ± 0.008	$\mathbf{0.383 \pm 0.009}$	0.686 ± 0.007	$\mathbf{0.673 \pm 0.007}^{*}$	0.957 ± 0.009	$\mathbf{0.945 \pm 0.011}$
	SCE	2.720 ± 0.047	$\mathbf{2.707 \pm 0.041}$	4.759 ± 0.045	$\mathbf{4.702 \pm 0.055}$	6.617 ± 0.059	$\mathbf{6.558 \pm 0.069}$
	ELR	$\mathbf{1.191 \pm 0.012}$	1.194 ± 0.009	$\mathbf{1.790 \pm 0.017}$	1.791 ± 0.016	2.147 ± 0.015	$\mathbf{2.142 \pm 0.015}$
CIFAR-100	CE	2.759 ± 0.053	$\mathbf{2.583 \pm 0.051}^{***}$	3.675 ± 0.064	$\mathbf{3.540 \pm 0.074}^{*}$	4.283 ± 0.023	$\mathbf{4.201 \pm 0.026}^{***}$
	PartRestart	—	$\mathbf{2.564 \pm 0.047}^{***}$	—	$\mathbf{3.549 \pm 0.067}^{*}$	—	$\mathbf{4.206 \pm 0.019}^{***}$
	MAE	1.690 ± 0.039	1.690 ± 0.039	1.864 ± 0.044	1.864 ± 0.045	1.932 ± 0.010	$\mathbf{1.931 \pm 0.010}$
	GCE	0.644 ± 0.007	$\mathbf{0.636 \pm 0.008}$	0.896 ± 0.007	$\mathbf{0.883 \pm 0.006}^{*}$	1.129 ± 0.004	$\mathbf{1.116 \pm 0.006}^{**}$
	SCE	17.575 ± 0.341	$\mathbf{16.768 \pm 0.405}^{**}$	23.129 ± 0.108	$\mathbf{22.556 \pm 0.090}^{***}$	26.840 ± 0.106	$\mathbf{26.609 \pm 0.140}^{*}$
	ELR	2.802 ± 0.049	$\mathbf{2.709 \pm 0.159}$	3.901 ± 0.062	$\mathbf{3.580 \pm 0.160}^{**}$	4.503 ± 0.030	$\mathbf{4.450 \pm 0.141}$