Revisiting the noise Model of SGD

Barak Battash
Faculty of Engineering, Bar Ilan University
barakbattash@gmail.com

Lior Wolf
School of Computer Science, Tel Aviv University
wolf@cs.tau.ac.il

Ofir Lindenbaum
Faculty of Engineering, Bar Ilan University
ofir.lindenbaum@biu.ac.il

Abstract

The effectiveness of stochastic gradient descent (SGD) is significantly influenced by stochastic gradient noise (SGN). Following the central limit theorem, stochastic gradient noise (SGN) was initially described as Gaussian, but recently, Simsekli et al. demonstrated that $S\alpha S$ Lévy better characterizes the stochastic gradient noise. Here, we revisit the noise model of SGD and provide robust, comprehensive empirical evidence that SGN is heavy-tailed and is better represented by the $S\alpha S$ distribution. Furthermore, we argue that different deep neural network (DNN) parameters preserve distinct SGN properties throughout training. We develop a novel framework based on Lévy-driven stochastic differential equation (SDE), where one-dimensional Lévy processes describe each DNN parameter. This leads to a more accurate characterization of the dynamics of SGD around local minima.

1 Introduction

The tremendous success of deep learning [3, 16, 23] can be partly attributed to implicit properties of the optimization tools, in particular, the popular SGD [37, 4] scheme. Despite its simplicity, i.e., being a noisy first-order optimization method, SGD empirically outperforms gradient descent (GD) and second-order methods. By evading sharp basins and settling in wide minima, the stochastic gradient noise of SGD can improve the generalization of the model [52, 41]. The DNN architecture and data distribution impact the formation and amplitude of SGD noise, which results from stochasticity in the mini-batch sampling operation. Therefore, understanding the properties of SGD is of high priority.

Analyzing the behavior of SGD optimization for non-convex cost functions is ongoing research [6, 49, 9, 34, 14, 24, 42, 52, 47]. Studies mainly examine the distribution and nature of the noise, with its ability to escape local minima and generalize better [17, 13, 44, 12, 49, 21].

SGD is based on an iterative update rule, where the k-th step of the iterative update is formulated as:

$$w_k = w_{k-1} - \frac{\eta}{B} \sum_{\ell \in \Omega_k} \nabla U^{(\ell)}(w_{k-1}) = w_{k-1} - \eta_k \nabla U(w_{k-1}) + \eta_k \zeta_k, \tag{1}$$

where w denotes the weights (parameters) of the DNN, $\nabla U(w)$ is the gradient of the objective function, B is the batch size, $\Omega_k \subset \{1, .., D\}$, is the randomly selected mini-batch. Thus $|\Omega_k| = B$, D is the number of data points in the dataset, ζ_k is the SGD noise, which is formulated as $\zeta_k = \nabla U(w_k) - \frac{1}{B} \sum_{\ell \in \Omega_k} \nabla U^{(\ell)}(w_k)$, i.e., the difference between the gradient produced by GD and SGD, finally η depicts the learning rate, and η_k indicates the learning rate at step k.

By modeling SGD using a continuous-time SDE, we can examine the evolution of the dynamic system in the continuous time domain [51, 30, 46, 6, 17, 38].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Model	Data	Gauss	$S\alpha S$	$S\alpha S$ Wins
Clip-base [B=32]	Laion400M	$0.0038 \pm 3.83e^{-06}$	$0.0028 \pm 2.76e^{-06}$	96.60%
Clip-base [B=64]	Laion400M	$0.0034 \pm 3.00e^{-06}$	$0.0029 \pm 2.44e^{-06}$	96.80%
Clip-base [B=256]	Laion400M	$0.0040 \pm 2.64e^{-06}$	$0.0036 \pm 2.08e^{-06}$	96.88%
Clip-large [B=32]	Laion400M	$0.0033 \pm 3.03e^{-06}$	$0.0028 \pm 2.41e^{-06}$	96.67%
EfficientNet-b2	ImageNet	0.0180 ± 0.0049	0.00092 ± 0.0001	99.74%
EfficientNet-b3	ImageNet	0.02410 ± 0.0058	0.00096 ± 0.0001	99.63%
EfficientNet-b4	ImageNet	0.03439 ± 0.0089	0.00213 ± 0.0006	99.68%
FlexVit	ImageNet	0.03399 ± 0.0156	0.00211 ± 0.0003	99.62%
Vit base	ImageNet	0.06495 ± 0.0264	0.00656 ± 0.00126	99.74%
Vit small	ImageNet	0.02870 ± 0.0131	0.0030 ± 0.0009	99.56%

Table 1: Evaluating the fitting error of the empirical distribution of the SGN. We evaluate several architectures and datasets. 10,000 parameters were sampled from each network. $S\alpha S$ Wins- indicates the portion of parameters better fitted by $S\alpha S$ distribution.

Many previous works [51, 28, 45, 52] use an SDE and argue that the noise is Gaussian, i.e., $u_t \sim \mathcal{N}(0, \lambda(w_k))$, where $\lambda(w_k)$ is the noise covariance matrix and formulated as follows [51] :

$$\lambda(w_k) = \frac{1}{B}\left[\frac{1}{D}\sum_{d=1}^{D}\nabla U^{(d)}(w_k)\nabla U^{(d)}(w_k)^T - \nabla U(w_k)\nabla U(w_k)^T\right].$$

Recently, [51] showed the importance of modeling the SGN as an anisotropic noise. Although SGN is mostly considered to be Gaussian [39, 36, 48, 29, 51, 33], recently [40], demonstrated that the SGN obeys $\mathcal{S}\alpha\mathcal{S}$ Lévy distribution due to its heavy-tailed nature.

In this study, we show empirically that the SGN of DNN parameters distributes differently. We further demonstrate that SGN has heavy-tail properties, making $S\alpha S$ distribution more accurately characterize it visually and numerically using multiple datasets. Based on the empirical evidence, we propose a novel dynamical system in $\mathbb{R}^N$ consisting of N one-dimensional $S\alpha S$ processes, a more accurate and closer to a real-world scenario. Finally, we use our framework to approximate the mean escape time and the likelihood of escaping the local minima via a particular parameter.

2 Related Work

Modeling SGD using differential equations is a deep-rooted method. Li et al. [26] used an SDE to approximate SGD and focused on momentum and adaptive parameter tuning schemes to study the dynamical properties of stochastic optimization. Mandt and Blei [27] employed a similar procedure to derive an SDE approximation for the SGD to study the influence of the value of the learning rate. Li et al. [25] showed that an SDE could approximate SGD in a first-order weak approximation. The early works in the field have approximated SGD by Langevin dynamic with isotropic diffusion coefficients [39, 36, 48]. Later, more accurate modeling suggested [29, 51, 33] using an anisotropic noise covariance matrix. Lately, it has been argued [40] that SGN is better characterized by $S\alpha S$ noise, presenting experimental and theoretical justifications. This model was allegedly refuted by [46], claiming that the experiments performed by [40] are inaccurate since the noise calculation was done across parameters and not across mini-batches. Lévy driven SDEs Euler approximation literature is sparser than for the Brownian motion SDEs; however, it is still intensely investigated; for more details about the convergence of Euler approximation for Lévy discretization, see [32, 35, 5].

3 Framework

We consider a DNN with $\bar{\mathbf{L}}$ layers and a total of N weights (parameters), the domain $\mathcal{G}$ is the local environment of a minimum, $\mathcal{G} \subseteq \mathbb{R}^N$ is a bounded and relatively compact subspace, please see Sec. D for more rigours and detailed definition of $\mathcal{G}$. Our framework considers an N-dimensional dynamic system, representing the update rule of SGD as a Lévy-driven stochastic differential equation. In contrast to previous works [50, 40], our framework does not assume that SGN distributes the same for every parameter l in the DNN. Thus, the SGN of each parameter is characterized by a different α. The governing SDE that depicts the SGDs dynamic inside the domain $\mathcal{G}$ at time t is as follows:

$$W_t = w - \int_0^t \nabla U(w^p)\, dp + \sum_{l=1}^{N} s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon_l (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} r_l L_t^l, \tag{2}$$

where W_t is the process that depicts the evolution of DNN weights at time t. $L_t^l \in \mathbb{R}$ is a mean-zero $S\alpha S$ Lévy processes with a stable parameter α_l. $\lambda_l(t) \in \mathbb{R}^N$ is the l-th row of the noise covariance matrix, $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones, and its purpose is to sum the l-th row of the noise covariance matrix. $r_l \in \mathbb{R}^N$ is a unit vector and we demand $|\langle r_i, r_j \rangle| \neq 1$, for $i \neq j$, we will use r_i as a one-hot vector. s_t represents the learning rate scheduler, and w are the initial weights, $\epsilon = \eta^{\frac{\alpha-1}{\alpha}}$, and η is the learning rate. Let us remind that $alpha$ is the stability parameter of the $S\alpha S$ distribution and in this work $\alpha \in (0.5, 2)$. The SDE construction is detailed in Sec. E.

Remark L_t^l can be decomposed into a small jump part ξ_t^l, and an independent part with large jumps ψ_t^l, i.e. $L_l = \xi_t^l + \psi_t^l$, more information on $S\alpha S$ process appears in A.3.

Let $\sigma_{\mathcal{G}} = \inf\{t \geq 0 : W_t \notin \mathcal{G}\}$ depict the first exit time from $\mathcal{G}$. τ_k^l denotes the time of the k-th largest jump of parameter l, which is driven by the process ψ^l , where we define $\tau_0 = 0$. The interval between large jumps is denoted as: $S_k^l = \tau_k^l - \tau_{k-1}^l$ and is exponentially distributed with mean $\beta_l(t)^{-1}$, while τ_k^l is gamma distributed $Gamma(k, \beta_l(t))$; where $\beta_l(t)$ is the intensity of the jump and will be defined in Sec 3.2. We define the arrival time of the k-th jump of all parameters combined as τ_k^*, for $k \geq 1$ we can write $\tau_k^* \triangleq \bigwedge_{\tau_j^l > \tau_{k-1}^*} \tau_j^l$, following that $S_k^* = \tau_k^* - \tau_{k-1}^*$. Jump heights are notated as: $J_k^l = \psi_{\tau_k}^l - \psi_{\tau_{k^-}}^l$. We will define α_ν as the average α over the entire DNN; this will help us describe the global properties of our network.
Let us define a measure of horizontal distance from the domain boundary using d_l^+ and d_l^-; we present a rigorous formulation of our assumptions in Sec. D. We define two additional processes to better understand the dynamics inside the basin (between the large jumps).
The deterministic process denoted as Y_t is affected by the drift alone, without any perturbations. This process starts within the domain and does not escape it as time proceeds. The drift forces this process towards the stable point W^* as $t \to \infty$, i.e., the local minimum of the basin; furthermore, the process converges to the stable point exponentially fast and is defined for $t > 0$, and $w \in \mathcal{G}$ by:

$$Y_t(w) = w - \int_0^t \nabla U(Y_s)\, ds. \tag{3}$$

The following Lemma shows how fast Y_t converges to the local minima from any starting point w inside the domain.

Lemma 3.1. *$\forall w \in \mathcal{G}$, $\tilde{U} = U(w) - U(W^*)$, the process Y_t converges to a minimizer W^* exponentially fast: $\|Y_t - W^*\|^2 \leq \frac{2\tilde{U}}{\mu} e^{-2\mu t}$. The complete proof appears in Appendix B.6.*

The small jumps process Z_t is composed of the deterministic process Y_t and a stochastic process with infinite small jumps denoted as ξ_t (see more details in A.3). Z_t describes the system's dynamic in the intervals between the large jumps; hence we add an index k that represents the index of the jump, for instance, $Z_{t,k}$ represent the time t between the jump k and $k+1$. Due to strong Markov property, $\xi_{t+\tau}^l - \xi_\tau^l, t \geq 0$ is also a Lévy process with the same law as ξ^l. Hence, for $t \geq 0$ and $k \geq 0$: $\xi_{t,k}^l = \xi_{t+\tau_{k-1}}^l - \xi_{\tau_{k-1}}^l$. The full small jumps process for $\forall t \in [0, S_k]$ is defined as:

$$Z_{t,k} = w - \int_0^t \nabla U(Z_s) ds + \sum_{l=1}^N s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon_l (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} r_l \xi_{t,k}^l. \tag{4}$$

In the following proposition, we estimate the deviation in the l-th parameter between the SDE solution driven by the process of the small jumps $Z_{t,k}^l$, and the deterministic trajectory.

Proposition 3.2. *Let $T_\epsilon > 0$ exponentially distributed with parameter β_l and $\rho \in (0,1)$, $\forall w \in \mathcal{G}$, and $\bar{\theta}_l \triangleq -\rho(1-\alpha_l) + 2 - 2\theta_l$, s.t. $\theta_l \in (0, \frac{2-\alpha_l}{4})$, the following holds:*

$$P\left(\sup_{t \in [0, T_\epsilon]} |Z_{t,k}^l - Y_{t,k}^l| \geq c\bar{\epsilon}^{\theta_l} \right) \leq C_{\theta_l} \bar{\epsilon}^{\bar{\theta}_l}. \tag{5}$$

Where $C_{\theta_l} > 0$ and $c > 0$ are constants, let us remind the reader that: $\bar{\epsilon}_l = s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon_l$. Precisely, proposition 3.2 describes the distance between the deterministic process $Y_{t,k}$ and the process of small jumps $Z_{t,k}$ at time t that occurs in the interval after the jump k and before jump $k+1$. It indicates

that between large jumps, the processes are close to each other with high probability. The complete proof appears in Appendix B.3.

Let us present additional notations: $H()$ and ∇U are the Hessian and the gradient of the objective function. To denote different mini-batches, we use subscript d. That is, $H_d()$ and $\nabla U_d(W^*)$ are the Hessian and gradient of the d mini-batch. To represent different parameters, as before we will use subscript l, for example, $\nabla u_{d,l}$, is the gradient of the l-th parameter after a forward pass over mini-batch d. Furthermore, $h_{l,j}$ represents the l-th row and j-th column of $H(W^*)$, which is the Hessian after a forward pass over the entire dataset D, i.e., the Hessian when performing standard gradient descent. Next, we turn our attention to another property of the process of the small jumps $Z^l_{t,k}$. This will help us understand the noise covariance matrix. Using stochastic asymptotic expansion, we can approximate $Z^l_{t,k}$ using the deterministic process and a first-order approximation of $Z^l_{t,k}$.

Lemma 3.3. *For a general scheduler s_t , $\rho \in (0,1)$, $\forall w^l, w^j \in \mathcal{G}$, starting point after a big jump at time $\tau^*_k + p$ where $p \to 0$, and $A_{lj}(t) \triangleq \bar{\epsilon}_l w^j e^{-h_{jj}t} \mu^l_\xi (2t + \frac{1}{h_{ll}}(1 - e^{-h_{ll}t}))$, for $t \in [0, S^*_k)$ the following fulfills:*

$$\mathbb{E}[Z^l_{t,k} Z^j_{t,k}] = w^l w^j e^{-(h_{ll}+h_{jj})t} + A_{jl}(t) + A_{lj}(t) + \mathcal{O}(\epsilon^2). \tag{6}$$

Where $\mu^i_\xi = 2t\left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)}-1}{1-\alpha_l}\right]$, $\bar{\epsilon}_l = s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon_l$. w^j, w^l are the weight value of parameters j and l respectively at time t. Lemma 3.3 depicts the dynamics between two parameters in the intervals between the large jumps; this helps us to express the covariance matrix of the noise accurately; the complete derivation of this result appears in Appendix B.4.

3.1 Noise covariance matrix

The covariance of the noise matrix holds a vital role in modeling the training process; in this subsection, we aim to achieve an expression of the noise covariance matrix based on the stochastic processes we presented in the previous subsection. We can achieve the following approximation using stochastic Taylor expansion near the basin W^*.

Proposition 3.4. *Let us define $\tilde{u}_{lj} = \frac{1}{D}\sum_{d=1}^{D} \nabla u_{d,l} \nabla u_{d,j}$, $\tilde{h}_{l,m,p,j} := \frac{1}{B}\sum_{b=1}^{B} h_{b,l,m} h_{b,p,j}$, $h_{l,m,p,j} := h_{l,m} h_{p,j}$ and $\bar{h}_{l,m,p,j} := \tilde{h}_{l,m,p,j} - h_{l,m,p,j}$, then for any $t \in [0, S^*_k)$, the sum of the l-th row of the covariance matrix:*

$$\mathbf{1}^T \lambda^k_l(W_t) = \frac{1}{B}\sum_{j=1}^{N} \tilde{u}_{lj} + \frac{1}{B}\sum_{j,m,p=1}^{N} \bar{h}_{l,m,p,j}(w^m w^p e^{-(h_{mm}+h_{pp})t} + A_{mp}(t) + A_{pm}(t)) + \mathcal{O}(\bar{\epsilon}^2), \tag{7}$$

where $A_{mp}(t)$ and $A_{pm}(t)$ are defined in lemma 3.3. We note that $h_{l,m,p,j}$ and $\tilde{h}_{l,m,p,j}$ represent the interaction of two terms in the Hessian matrix when performing GD and SGD respectively, and $\bar{h}_{l,m,p,j}$ is the difference between them. The proof of the proposition appears in Appendix B.5. Note that the influence of the batch size B on the noise mainly appears in Eq. 7. Suggesting that larger values of B will smooth and decrease the absolute values in the covariance matrix, as expected.

3.2 Jump Intensity

Let us denote $\beta_l(t)$ as the jump intensity of the compound Poisson process Ψ_l. $\beta_l(t)$ simultaneously responsible for scaling the jump frequency and size. Jumps are distributed according to the law $\beta_l(t)^{-1}\nu_\eta$, and the jump intensity is formulated as:

$$\beta_l(t) = \nu_{\Psi_l}(\mathbb{R}) = \int_{\mathbb{R}/[-O,O]} \nu_l(dy) = \frac{2}{\alpha_i} s_t^{\rho(\alpha_l - 1)} \epsilon_l^{\rho\alpha_l}, \tag{8}$$

where the integration boundary is $O \triangleq \epsilon^{-\rho} s_t^{-\rho\frac{\alpha_l - 1}{\alpha_l}}$, which is time-dependent, due to the learning rate scheduler, which decreases the size and frequency of the large jumps, thus the jump intensity is not stationary. Hence, changing the learning rate during training enables us to increase and decrease the frequency and amplitude of the jumps. The entire DNN jump intensity as $\beta_S(t) \triangleq \sum_{l=1}^{N} \beta_l(t)$.

The probability of escaping the local minima in the first jump, in a single parameter perspective, is expressed by: $P(s_t\epsilon(\mathbf{1}^T\lambda_l(t))^{\frac{1}{\alpha_l}}J_1^l \notin [d_l^-, d_l^+]) = \frac{m_l(t)\Phi_l s_t^{\alpha_l-1}}{\beta_l(t)}$, where $m_l(t) = \frac{(\mathbf{1}^T\lambda_l(t))^{\frac{1}{\alpha_l}}\epsilon_l^{\alpha_l}}{\alpha_l}$, and $\Phi_l = (-d_l^-)^{-\alpha_l} + (d_l^+)^{-\alpha_l}$.

4 Theorems

Our work assumes that the training process can exit from the domain only at times that coincide with large jumps; please see Sec. A.6 for the mathematical evidence. Using the information described above, we analyze the escaping time for the exponential and multi-step schedulers; expanding our framework for more LRdecay schemes is straightforward. Let us define a constant that will be used for the remainder of the paper: $A_{l,\nu} \triangleq (1-\bar{m}_\nu\bar{\beta}_\nu^{-1}\Phi_\nu)(1-\bar{\beta}_l\bar{\beta}_S^{-1})$, for the next theorem we denote: $C_{l,\nu,p} \triangleq \frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)(\alpha_l-1)}$, where $C_{l,\nu,p}$ depends on α_l, γ, and on the difference $\alpha_l - \alpha_\nu$. The following theorem describes the approximated mean escape time for the exponential scheduler:

Theorem 4.1. *Given $C_{l,\nu,p}$ and $A_{l,\nu}$, let s_t be an exponential scheduler $s_t = t^{\gamma-1}$, the mean transition time from the domain $\mathcal{G}$:*

$$\mathbb{E}[\sigma_{\mathcal{G}}] \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\beta_l(\bar{m}_l\Phi_l)^{1-C_{l,\nu,p}}}{\beta_S(1+(\gamma-1)(\alpha_l-1))}\Gamma\left(C_{l,\nu,p}\right).$$

Where Γ is the gamma function, $\bar{m}_l = \frac{\bar{\lambda}_l^{\alpha_l}\epsilon_l^{\alpha_l}}{\alpha_l}$ and $\bar{\beta}_l = \frac{2\epsilon_l^{\rho\alpha_l}}{\alpha_l}$ is the time independent jump intensity. See Appendix B.1 for the full proof.

It can be observed from Thm. 4.1 that as γ decreases, i.e., faster learning rate decay, the mean transition time increases. Interestingly, when $\alpha_l \to 2$ (nearly Gaussian) and $\gamma \to 0$, the mean escape time goes to infinity, which means the training process is trapped inside the basin.

Corollary 4.2. *Using Thm. 4.1, if the cooling rate is negligible, i.e $\gamma \to 1$, the mean transition time:*

$$\mathbb{E}[\sigma_{\mathcal{G}}] \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{1}{\beta_S 1^T\bar{\lambda}_l\epsilon^{\alpha_l(1-\rho)}\Phi_l}. \tag{9}$$

Col. 4.2 clearly shows that the mean escape time is not affected by the basin height but the basin width which is represented by Φ_l. Further, since local minima in DNNs are mostly asymmetric, in $1d$ perspective, we can note that the distant edge ($max(-d_i^-, d_i^+)$) of the domain $\mathcal{G}$ does not affect on the mean escape time. Furthermore, one can note that the escape time dependency on the learning rate is polynomial.

The framework presented in this work enables us to understand in which direction r_i the training process is more probable to exit the basin $\mathcal{G}$, i.e., which parameter is more liable to help the process escape; this is a crucial feature for understanding the training process. The following theorems will be presented for the exponential scheduler but can be expanded for any scheduler.

Theorem 4.3. *Let s_t be an exponential scheduler $s_t = t^{\gamma-1}$, $C_l \triangleq \frac{(\gamma-1)(\alpha_l-1+\rho(2\alpha_l-\alpha_\nu-\alpha_l))+2}{(\gamma-1)(\alpha_l-1)+1}$, for $\delta \in (0, \delta_0)$, the probability of the training process to exit the basin through the l-th parameter is as follows:*

$$P(W_\sigma \in \Omega_i^+(\delta)) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i} \frac{\beta_l^2(\bar{m}_l\Phi_l)^{-C_l}}{\beta_S((\gamma-1)(\alpha_l-1)+1)}\Gamma\left(C_l\right). \tag{10}$$

Let us focus on the term that describes the i-th parameter: $P(W_\sigma \in \Omega_i^+(\delta)) \leq \frac{\bar{m}_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\sum_{l=0}^{N}\tilde{C}_l$, where $\tilde{C}_l$ encapsulate all the terms that do not depend on i. When considering SGN as Lévy noise, we can see that the training process needs only polynomial time to escape a basin. The following result helps us to assess the escaping ratio of two parameters.

Corollary 4.4. *The ratio of probabilities for exiting the local minima from two different DNN parameters is:*

$$\frac{P(W_\sigma \in \Omega_l^+(\delta))}{P(W_\sigma \in \Omega_j^+(\delta))} \leq \frac{1^T\lambda_l^{\alpha_l}}{1^T\lambda_j^{\alpha_j}}\eta^{(\alpha_l-\alpha_j)(1-\rho)}\frac{(d_l^+)^{-\alpha_l}}{(d_j^+)^{-\alpha_j}}. \tag{11}$$

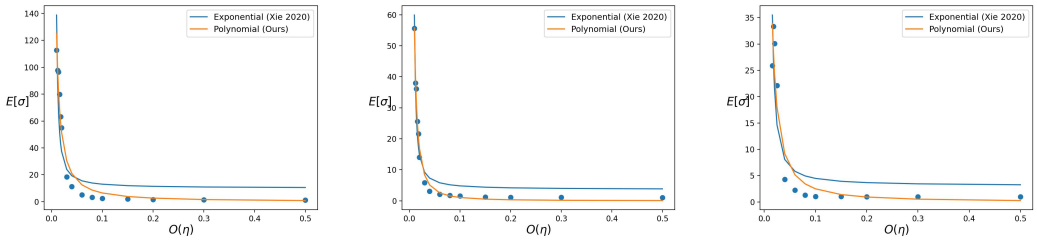

Figure 1: The mean escape time of SGD on Breastw (left), Cardio (middle), and Satellite (right) datasets. The plots show the fitting based on two methods: ours and [46] using a batch size of 32. The dots represent the mean escape time. A dot is an average of over 100 random seeds and several learning rates. Our theory better explains the empirical results for all three datasets examined.

Let us remind the reader that (d_i^+) is a function of the horizontal distance from the domain's edge. Therefore, the first conclusion is that the higher (d_l^+) is, the lower the probability of exiting from the l-th direction. However, the dominant term is $\eta^{(\alpha_l-\alpha_j)(1-\rho)}$, combining both factors, parameters with lower α will have more chance of being in the escape path. It can also be seen from the definition of β_l that parameters with lower α jump earlier and contribute more significant jump intensities. We can conclude by writing:

$$\frac{P(W_\sigma \in \Omega_l^+(\delta))}{P(W_\sigma \in \Omega_j^+(\delta))} \propto \eta^{\Delta_{l,j}}, \tag{12}$$

where $\Delta_{l,j} = \alpha_l - \alpha_j$.

5 Experiments

This section presents the core experimental results supporting our analysis; additional experiments can be found in the Appendix. All the experiments were conducted using SGD without momentum and weight decay.

Stochastic gradient noise distribution We empirically show that SGN is better characterized using the $S\alpha S$ Lévy distribution. Our evaluation follows [46], calculating the noise of each parameter separately using multiple mini-batches; as opposed to [40] that calculated the noise of multiple parameters on one mini-batch and averages over all parameters and batches to characterize the distribution of SGN. In [46], the authors estimate SGN on a DNN with randomly initialized weights; we, on the other hand, estimate the properties of SGN based on a pre-trained DNN. We use several datasets and architectures to empirically evaluate the properties of the SGN. Our quantitative results presented in Table 1 depict the fitting error of the empirical distribution of SGN with three distributions: (1) Gaussian [46], (2) $S\alpha S$ with constant α [40], and (3) $S\alpha S$ with multiple α_i values (ours). Our results show strong evidence that SGN is best explained by $S\alpha S$ distribution. In the Appendix, we provide additional experiments demonstrating that distinct parameters hold different noise distributions.

Mean escape time To corroborate Theorem 4.1, we train a three-layer network with Relu activation on "BreastW," "Satellite," and "Cardio" datasets [10]. We first train the model using SGD and a batch size of 256 until reaching a local minimum (see discussion Appendix A.4). After reaching the critical point, we decrease the mini-batch size to 32, and try to escape the critical minimum, Fig 1 shows the escape time using different learning rates. The number of iterations measures the escape time, averaged over 100 seeds. We fit empirical results to two theories, ours and [46], with the same amount of free parameters. The results in Fig 1 show the mean escape time using a batch size of 32; we observe that our theory better explains the empirical results on all three datasets.

6 Conclusions and Limirations

This work corroborates that the $S\alpha S$ better characterized SGN qualitatively and quantitatively. Furthermore, we show that distinct parameters are better characterized by different distribution parameters, α_i. Based on the mentioned experiments, we constructed a framework in $\mathbb{R}^N$ consisting of N one-dimensional Lévy processes with α_i-stable components. This framework enables us to characterize better the nature of DNN training with SGD, such as the escaping properties from different local minima, a learning rate scheduler, and other parameters' effects in the DNN.

The presented framework is valid once the training process is near a local minimum; our work does not address the dynamics and noise characteristics of SGD at an early training stage. Furthermore, the evolution of α in time is still unclear and demands future research.

References

[1] H. Amann. *Gewöhnliche differentialgleichungen*. Walter de Gruyter, 2011.

[2] V. Bally and D. Talay. The law of the euler scheme for stochastic differential equations: Ii. convergence rate of the density. 1996.

[3] Y. Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[4] L. Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8): 12, 1991.

[5] T. Burghoff and I. Pavlyukevich. Spectral analysis for a discrete metastable system driven by lévy flights. *Journal of Statistical Physics*, 161(1):171–196, 2015.

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

[7] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv e-prints*, art. arXiv:1810.03505, Oct. 2018.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, art. arXiv:1810.04805, Oct. 2018.

[9] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

[10] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[11] T. H. Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, pages 292–296, 1919.

[12] J. Z. HaoChen, C. Wei, J. D. Lee, and T. Ma. Shape matters: Understanding the implicit bias of the noise covariance. *arXiv preprint arXiv:2006.08680*, 2020.

[13] F. He, T. Liu, and D. Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. 2019.

[14] H. He, G. Huang, and Y. Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *arXiv preprint arXiv:1902.00744*, 2019.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, art. arXiv:1512.03385, Dec. 2015.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[17] W. Hu, C. Junchi Li, L. Li, and J.-G. Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv e-prints*, art. arXiv:1705.07562, May 2017.

[18] P. Imkeller and I. Pavlyukevich. First exit times of sdes driven by stable lévy processes. *Stochastic Processes and their Applications*, 116(4):611–642, 2006.

[19] P. Imkeller and I. Pavlyukevich. Lévy flights: transitions and meta-stability. *Journal of Physics A: Mathematical and General*, 39(15):L237, 2006.

[20] J. Jacod, T. G. Kurtz, S. Méléard, and P. Protter. The approximate euler method for lévy driven stochastic differential equations. In *Annales de l'IHP Probabilités et statistiques*, volume 41, pages 523–558, 2005.

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

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[23] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[24] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

[25] Q. Li, C. Tai, and W. E. Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv e-prints*, art. arXiv:1511.06251, Nov. 2015.

[26] Q. Li, C. Tai, and E. Weinan. Dynamics of stochastic gradient algorithms. *ArXiv*, abs/1511.06251, 2015.

[27] S. Mandt and D. M. Blei. Continuous-time limit of stochastic gradient descent revisited. 2015.

[28] S. Mandt, M. D. Hoffman, and D. M. Blei. A Variational Analysis of Stochastic Gradient Algorithms. *arXiv e-prints*, art. arXiv:1602.02666, Feb. 2016.

[29] S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

[30] Q. Meng, S. Gong, W. Chen, Z.-M. Ma, and T.-Y. Liu. Dynamic of Stochastic Gradient Descent with State-Dependent Noise. *arXiv e-prints*, art. arXiv:2006.13719, June 2020.

[31] Q. Meng, S. Gong, W. Chen, Z.-M. Ma, and T.-Y. Liu. Dynamic of stochastic gradient descent with state-dependent noise. *arXiv preprint arXiv:2006.13719*, 2020.

[32] R. Mikulevicius and C. Zhang. On the rate of convergence of weak Euler approximation for non-degenerate SDEs. *arXiv e-prints*, art. arXiv:1009.4728, Sept. 2010.

[33] T. Mori, L. Ziyin, K. Liu, and M. Ueda. Power-law escape rate of SGD. *arXiv e-prints*, art. arXiv:2105.09557, May 2021.

[34] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In *International conference on machine learning*, pages 2603–2612. PMLR, 2017.

[35] P. Protter, D. Talay, et al. The euler scheme for lévy driven stochastic differential equations. *The Annals of Probability*, 25(1):393–423, 1997.

[36] M. Raginsky, A. Rakhlin, and M. Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703. PMLR, 2017.

[37] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[38] I. Sato and H. Nakagawa. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 982–990, Bejing, China, 22–24 Jun 2014. PMLR. URL `http://proceedings.mlr.press/v32/satoa14.html`.

[39] I. Sato and H. Nakagawa. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In *International Conference on Machine Learning*, pages 982–990. PMLR, 2014.

[40] U. Simsekli, L. Sagun, and M. Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

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

[42] S. L. Smith, B. Dherin, D. G. Barrett, and S. De. On the origin of implicit regularization in stochastic gradient descent. *arXiv preprint arXiv:2101.12176*, 2021.

[43] A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

[44] J. Wu, W. Hu, H. Xiong, J. Huan, and Z. Zhu. The multiplicative noise in stochastic gradient descent: Data-dependent regularization, continuous and discrete approximation. *CoRR*, 2019.

[45] J. Wu, W. Hu, H. Xiong, J. Huan, V. Braverman, and Z. Zhu. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, pages 10367–10376. PMLR, 2020.

[46] Z. Xie, I. Sato, and M. Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. *arXiv e-prints*, pages arXiv–2002, 2020.

[47] K. You, M. Long, J. Wang, and M. I. Jordan. How Does Learning Rate Decay Help Modern Neural Networks? *arXiv e-prints*, art. arXiv:1908.01878, Aug. 2019.

[48] Y. Zhang, P. Liang, and M. Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR, 2017.

[49] M. Zhou, T. Liu, Y. Li, D. Lin, E. Zhou, and T. Zhao. Toward understanding the importance of noise in training neural networks. In *International Conference on Machine Learning*, pages 7594–7602. PMLR, 2019.

[50] P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *arXiv preprint arXiv:2010.05627*, 2020.

[51] Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma. The Anisotropic Noise in Stochastic Gradient Descent: Its Behavior of Escaping from Sharp Minima and Regularization Effects. *arXiv e-prints*, art. arXiv:1803.00195, Feb. 2018.

[52] L. Ziyin, K. Liu, T. Mori, and M. Ueda. On minibatch noise: Discrete-time sgd, overparametrization, and bayes. *arXiv preprint arXiv:2102.05375*, 2021.

Appendix

A Additional details and results

Here we provide additional information required to reproduce our results and for the completeness of our exposition.

A.1 Notations

Symbol	Description
t	Train iteration
$S\alpha S$	Symmetric α stable
U	Potential/ loss function
W_t	The process that depicts DNN weights time evolution.
Y_t	The deterministic process.
Z_t	The small jumps process
L_t^l	Mean-zero $S\alpha S$ Lévy processes in 1d- represent the SGN of the l-th parameter
ψ_t	Large jump part of L_t
ξ_t	Small jump part of L_t
η	Learning rate
B	Batch size
Ω	Batch sample ($\|\Omega\| = B$)
D	Number of samples in training datasets
s_t	LR scheduler at time t
γ	Cooling rate
α	Stability parameter of $S\alpha S$ dist.
λ	Noise covariance matrix
τ_k^l	The time of the k-th large jump of parameter l
S_k^l	The difference between the $(k-1)$-th large jump and k-th large jump of parameter l
β_l	The jump intensity of the compound Poisson process ξ_l
J	Large jumps height

A.2 Technical details

We trained several CNNs on the CINIC dataset [7] and the BERT base model on CoLA [43] dataset. All models are trained until reaching convergence. Using the pre-trained weights, we sample 100 random parameters; for each parameter, we estimate the noise by computing the gradients of all of the mini-batches in the dataset without updating the weights. Then, we fitted the empiric stochastic gradient noise to multiple distributions; Sum of square error (SSE) is used to evaluate the quality of our fit. We trained four ResNet variants Resnet18/34/50, those models were trained using SGD optimizer, learning rate of 0.01, and a batch size of 128 (A.4). We used a multistep learning rate scheduler on epochs 200 and 400 to accelerate the convergence. We examine the SGD noise of BERT model, which was fine-tuned on CoLA [43] dataset using Adam optimizer with a learning rate of 2e-05 and batch size of 32 for 20 epochs. This is the standard Bert fine-tuning procedure. The results are shown in Tab. F.1. Visual examples for the heavy-tailed nature of SGN can be seen in Fig. 5, and additional results are presented in Sec. F.3. These results corroborates our claim of the heavy-tailed

nature of SGN, even for different DNN architectures (CNN and Transformer based models) and input domains (text and images).

A.3 $S\alpha S$ background

A Lévy process is random with independent and stationary increments, continuous in probability, and possesses right-continuous paths with left limits. Except for special cases, its probability density does not generally have a closed-form formula. Hence the process is characterized by the Lévy–Hincin formula. In this paper, the noise is assumed to be best fitted by symmetric α stable Lévy distribution, also known as Lévy flights (LF), and mainly parameterized using a stability parameter α, hence the characteristic function:

$$\mathbb{E}[e^{iwL_t^l}] = exp\{-t\int_{\mathbb{R}/\{0\}}[e^{iwy} - 1 - iwy\mathbb{I}\{|y| \leq 1\}]\frac{dy}{|y|^{1+\alpha_l}}\}, \tag{13}$$

where $\mathbb{I}\{B\}$ denotes the indicator function of a set with the corresponding generating triplet $(0, \nu_l, 0)$ and the Lévy measure $\nu_l(dy) = |y|^{-1-\alpha_l}, y \neq 0, \alpha_l \in (0, 2)$.In this work we assume $\alpha_l \in (0.5, 2)$. Unlike Brownian motion which almost surely holds continuous path, Lévy motion might obtain large discontinuous jumps. Using Lévy- Itô-decomposition of L^l can be decomposed into a small jump part ξ_t^l, and an independent part with large jumps ψ_t^l, i.e., $L_i = \xi_t^l + \psi_t^i$.
The process ξ_t^l has an infinite Lévy measure with support:$\{y|0 < \|y\| \leq \epsilon_l^{-\rho}\}$,$\forall\rho \in (0, 1)$, and makes infinitely many jumps on any time interval. The absolute value of ξ_t^l jumps is bounded by $\epsilon^{-\rho}$. ψ_t^i is a compound Poisson process with finite Lévy measure, and is responsible on the big jumps, more details about ψ_t^i in Sec. 3.2

A.4 Selecting minimum point

In order to find local minimum, we measure the loss of the entire data, i.e. loss when running GD; if the loss does not change more then ϵ for more then 100 iterations, we exit the training process and select the checkpoint as a minimum point. Since we do not know the domain boundary of the current minimum, we measure the number of iterations until the training process passes a predefined loss delta (ΔL) from the current local minimum.

A.5 Assumption on the Potential near critical points

We assume that the potential $U(W_t)$ is μ-strongly convex and can be approximated by a second order Taylor approximation near critical points that will be noted as W^*:

$$U(W) = U(W^*) + \nabla U(W^*)(W - W^*) + \frac{1}{2}(W - W^*)^T H(W^*)(W - W^*) \tag{14}$$

This does not mean that $U(W)$ fulfills any of the assumptions above in general.

A.6 Exiting the potential using large jumps

We assume that the process is able to exit only when large jump occurs, this assumption is based on a few realizations; first, the deterministic process Y_t initialized in any point $w \in \mathcal{G}_\delta$, will converge to the local minima of the domain by the positive invariance of the process, see assumptions in Appendix D. Second, Y_t converges to the minimum much faster than the average temporal gap between the large jumps; third, using lemma 3.1, we conclude that the small jumps are less likely to help the process escape from the local minimum. Next, we will show evidence for the second realization mentioned above, the relaxation time T_R^l is the time for the deterministic process Y_t^l, starting from any arbitrary $w \in \mathcal{G}$, to reach an $\bar{\epsilon}_l^\zeta$-neighbourhood of the attractor. For some $C_1 > 0$, the relaxation time is

$$T_R^l = \max\left\{\int_{d_l^-}^{-\bar{\epsilon}_l^\zeta} \frac{dy}{-U'(y)_l}, \int_{\bar{\epsilon}_l^\zeta}^{d_l^+} \frac{dy}{U'(y)_l}\right\} \leq C_1|ln\bar{\epsilon}_l|. \tag{15}$$

Now, let us calculate the expectation of $S_k^* = \tau_k^* - \tau_{k-1}^*$, i.e. the interval between the large jumps:

$$\mathbb{E}[S_k^l] = \mathbb{E}[\tau_k^l - \tau_{k-1}^l] = \beta_l^{-1} = \frac{\alpha_l}{2}\bar{\epsilon_l}^{-\rho\alpha_l}. \tag{16}$$

Since $\bar{\epsilon} \in (0,1)$, usually even $\bar{\epsilon} \ll 1$, it is easy to notice that $\mathbb{E}[S_k^l] \gg T_R$; thus we can approximate that the process W_t is near the neighborhood of the basin, right before the large jumps. This means that it is highly improbable that two large jumps will occur before the training process returns to a neighborhood of the local minima.

B Proofs

B.1 Proof of Theorem 4.1

The first equality is true under the assumption that the process can exit the basin only when large jumps occur.

$$\begin{aligned}
\mathbb{E}[\sigma_{\mathcal{G}}] &= \sum_{k=1}^{\infty} \mathbb{E}[\tau_k^*]\mathbb{I}\{\sigma_{\mathcal{G}} = \tau_k^*\} \qquad (17)\\
&= \sum_{k=1}^{\infty} \mathbb{E}[\tau_k^* \mathbb{I}\{\sum_{l=0}^{N} s_t \epsilon (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} J_1^l \mathbb{I}\{\tau_1^l = \tau_1^*\} \in \mathcal{G}, \sum_{l=0}^{N} s_t \epsilon (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} J_2^l \mathbb{I}\{\tau_2^l = \tau_2^*\} \in \mathcal{G}\\
&, .., \sum_{l=0}^{N} s_t \epsilon (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} J_k^l \mathbb{I}\{\tau_k^l = \tau_k^*\} \notin \mathcal{G}\}]\\
&= \sum_{k=1}^{\infty} \mathbb{E}[\tau_k^* \mathbb{I}\{J_1^* \in \mathcal{G}, J_2^* \in \mathcal{G}, .., J_k^* \notin \mathcal{G}\}] \leq \sum_{k=1}^{\infty} \mathbb{E}[\tau_k^* (1 - \mathbb{I}\{J_k^* \notin \mathcal{G}\})^{k-1} \mathbb{I}\{J_k^* \notin \mathcal{G}\}]\\
&= \sum_{k=1}^{\infty} \sum_{l=1}^{N} \mathbb{E}[\tau_k^l (1 - \mathbb{I}\{J_k^l \notin \mathcal{G}\})^{k-1} \mathbb{I}\{J_k^l \notin \mathcal{G}\} \mathbb{I}\{\tau_k^l = \tau_k^*\}]\\
&\leq \sum_{k=1}^{\infty} \sum_{l=1}^{N} \sum_{w=1}^{k} \mathbb{E}[\tau_w^l (1 - \mathbb{I}\{s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon (\mathbf{1}^T \lambda_l(t))^{\frac{1}{\alpha_l}} J_w^l \notin \mathcal{G}\})^{w-1} (1 - \mathbb{I}\{s_t^{\frac{\alpha_\nu - 1}{\alpha_\nu}} \epsilon \mathbf{1}^T \lambda_\nu(t) J_w^m \notin \mathcal{G}\})^{k-w}\\
&\mathbb{I}\{s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon \mathbf{1}^T \lambda_i(t) J_w^l \notin \mathcal{G}\} \mathbb{I}\{\tau_w^l = \tau_k^*\}] \quad .
\end{aligned}$$

$\mathbb{I}\{\tau_w^l = \tau_k^*\}$ incorporates the probability that the k-th jump occurred by the l-th parameter, and the chance that within a total of k jumps the parameter l, will respect the w-th jump:

$$\begin{aligned}
\mathbb{I}\{\tau_w^l = \tau_k^*\} &= \frac{\beta_l(t)}{\beta_S(t)} \binom{k-1}{w-1} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w} \qquad (18)\\
&\frac{\beta_l(t)}{\beta_S(t)} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}
\end{aligned}$$

We will estimate the average probability of the DNN to escape the basin i.e. the general expression: $[1 - \frac{s_t^{\alpha_\nu - 1} m_\nu(t)}{\beta_\nu(t)} \Phi_\nu]^{k-w}$, by using α_ν as the average α value of the network.

$$\begin{aligned}
&\sum_{k=1}^{\infty} \sum_{l=0}^{N} \sum_{w=1}^{k} \int_0^{\infty} \frac{\beta_l(t)}{\beta_S(t)} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w} \beta_l(t) t \qquad (19)\\
&e^{-\beta_l(t)t} \frac{(\beta_l(t)t)^{w-1}}{(w-1)!} [1 - \frac{s_t^{\alpha_l - 1} m_l(t)}{\beta_l(t)} \Phi_l]^{w-1} [1 - \frac{s_t^{\alpha_\nu - 1} m_\nu(t)}{\beta_\nu(t)} \Phi_\nu]^{k-w} \frac{s_t^{\alpha_l - 1} m_l(t)}{\beta_l(t)} \Phi_l dt\\
&= \sum_{k=1}^{\infty} \sum_{l=0}^{N} \int_0^{\infty} \frac{\beta_l(t)}{\beta_S(t)} t e^{-\beta_l(t)t} s_t^{\alpha_l - 1} m_l(t) \Phi_l\\
&\sum_{w=1}^{k} \frac{[\beta_l(t)t - s_t^{\alpha_l - 1} m_l(t) \Phi_l t]^{w-1}}{(w-1)!} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left[\left(1 - \frac{s_t^{\alpha_\nu - 1} m_\nu(t)}{\beta_\nu(t)} \Phi_\nu\right) \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{k-w} dt\\
&= \sum_{k=1}^{\infty} \sum_{l=0}^{N} \int_0^{\infty} \frac{\beta_l(t)}{\beta_S(t)} t e^{-\beta_l(t)t} s_t^{\alpha_l - 1} m_l(t) \Phi_l
\end{aligned}$$

$$\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{k-1} L_{k-1}\left(\frac{\frac{\beta_l(t)}{\beta_S(t)}(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}{\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]}\right)$$

$$=\sum_{l=0}^{N}\int_0^\infty \frac{\beta_l(t)}{\beta_S(t)}te^{-\beta_l(t)t}s_t^{\alpha_l-1}m_l(t)\Phi_l$$

$$\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1} e^{-\frac{\frac{\beta_l(t)}{\beta_S(t)}(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}{\left[1-\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]}}dt$$

$$=\sum_{l=0}^{N}\int_0^\infty \frac{\beta_l(t)}{\beta_S(t)}te^{-\beta_l(t)t}s_t^{\alpha_l-1}m_l(t)\Phi_l$$

$$\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1} e^{-\frac{\frac{\beta_l(t)}{\beta_S(t)}(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}{\left[\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu+\frac{\beta_l(t)}{\beta_S(t)}-\frac{s_t^{\alpha_\nu-1}m_\nu(t)\Phi_\nu}{\beta_\nu(t)}\frac{\beta_l(t)}{\beta_S(t)}\right]}}dt$$

$$\leq\sum_{l=0}^{N}\int_0^\infty \frac{\beta_l(t)}{\beta_S(t)}te^{-\beta_l(t)t}s_t^{\alpha_l-1}m_l(t)\Phi_l$$

$$\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1} e^{-(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}dt$$

$$\leq\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\int_0^\infty \frac{\beta_l}{\beta_S}ts_t^{\alpha_l-1+\rho(\alpha_l-\alpha_\nu)}\bar{m}_l\Phi_l e^{-s_t^{\alpha_l-1}\bar{m}_l\Phi_l t}dt$$

$$=\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\int_0^\infty \frac{\beta_l}{\beta_S}t^{1+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}\bar{m}_l\Phi_l e^{-t^{1+(\gamma-1)\rho(\alpha_l-1)}\bar{m}_l\Phi_l}dt$$

$$=\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\beta_l\bar{m}_l\Phi_l}{\beta_S(1+(\gamma-1)\rho(\alpha_l-1))}$$

$$\left[(\bar{m}_l\Phi_l)^{-\frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)\rho(\alpha_l-1)}}\Gamma\left(\frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)\rho(\alpha_l-1)}\right)\right]dt$$

$$=\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{\beta_l\bar{m}_l\Phi_l}{\beta_S(1+(\gamma-1)\rho(\alpha_l-1))}(\bar{m}_l\Phi_l)^{-C_{l,\nu,p}}\Gamma\left(C_{l,\nu,p}\right)dt \quad .$$

Where $A_{l,\nu}\triangleq\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]$,$C_{l,\nu,p}\triangleq\frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)\rho(\alpha_l-1)}$. Further to ease the calculation assumed that the time dependency: $\frac{\beta_l(t)}{\beta_S(t)}=\frac{\bar{\beta}_l}{\bar{\beta}_S}s^{\rho(\alpha_l-\alpha_\nu)}$. If the cooling rate is negligible, i.e. $\gamma\to 1$, the mean transition time:

$$\mathbb{E}[\sigma_{\mathcal{G}}]\leq\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{1}{\beta_S(1^T\lambda_l)^{\frac{1}{\alpha_l}}\epsilon^{\alpha_l(1-\rho)}\Phi_l}. \tag{20}$$

B.2 Proof of Theorem 4.3

$$P(W_\sigma\in\Omega_i^+(\delta))=\sum_{k=1}^{\infty}\prod_{j=1}^{k-1}P(J_j^*\in\mathcal{G})P(J_k^*\in\Omega_i^+) \tag{21}$$

$$=\sum_{k=1}^{\infty}\prod_{j=1}^{k-1}P(J_j^*\in\mathcal{G})P(J_k^*>d_i^+)$$

$$
\begin{aligned}
&= \sum_{k=1}^{\infty}(1-P(J_k^* \notin \mathcal{G}))^{k-1}P(J_k^* \geq d_i^+) \\
&\leq \sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k-1}(1-P(s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\lambda_l(t))^{\frac{1}{\alpha_l}}J_w^l \notin \mathcal{G}))^{w-1}(1-P(s_t^{\frac{\alpha_\nu-1}{\alpha_\nu}}\epsilon\mathbf{1}^T\lambda_\nu(t)J_w^v \notin \mathcal{G}))^{k-w}P(J_w^l \geq d_i^+)P(\tau_w^l=\tau_k^*) \\
&= \sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k-1}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\frac{(k-1)!}{(w-1)!(k-w)!}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}\beta_l(t) \\
&e^{-\beta_l(t)t}\frac{(\beta_l(t)t)^{w-1}}{(w-1)!}[1-\frac{s_t^{\alpha_l-1}m_l(t)}{\beta_l(t)}\Phi_l]^{w-1}[1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu]^{k-w}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i} \\
&= \sum_{k=1}^{\infty}\sum_{l=1}^{N}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\beta_l(t)e^{-\beta_l(t)t}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i} \\
&\sum_{w=1}^{k-1}[1-\frac{s_t^{\alpha_l-1}m_l(t)}{\beta_l(t)}\Phi_l]^{w-1}\frac{(k-1)!}{(w-1)!(k-w)!}\frac{(\beta_l(t)t)^{w-1}}{(w-1)!} \\
&\left((1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right)^{k-w}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}dt \\
&= \sum_{k=1}^{\infty}\sum_{l=0}^{N}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\beta_l(t)e^{-\beta_l(t)t}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i} \\
&\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{k-1}L_{k-1}\left(\frac{\frac{\beta_l(t)}{\beta_S(t)}(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}{\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]}\right)dt \\
&= \sum_{l=0}^{N}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\beta_l(t)e^{-\beta_l(t)t}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i} \\
&\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1}e^{-\frac{\frac{\beta_l(t)}{\beta_S(t)}(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}{\left[\frac{s_t^{\alpha_\nu-1}m_\nu(t)\Phi_\nu}{\beta_\nu(t)}+\frac{\beta_l(t)}{\beta_S(t)}-\frac{s_t^{\alpha_\nu-1}m_\nu(t)\Phi_\nu}{\beta_\nu(t)}\frac{\beta_l(t)}{\beta_S(t)}\right]}}dt \\
&\leq \sum_{l=0}^{N}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\beta_l(t)e^{-\beta_l(t)t}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i}\left[\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1} \\
&e^{-(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}dt \\
&= \sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2}{\bar{\beta}_S}\int_0^{\infty}t^{(\gamma-1)(\alpha_i-1+\rho(2\alpha_l-\alpha_\nu-\alpha_i))+1}e^{-t^{(\gamma-1)\rho(\alpha_l-1)+1}\bar{m}_l\Phi_l}dt \\
&= \sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2(\bar{m}_l\Phi_l)^{-\frac{(\gamma-1)(\alpha_i-1+\rho(2\alpha_l-\alpha_\nu-\alpha_i))+2}{(\gamma-1)\rho(\alpha_l-1)+1}}}{\beta_S((\gamma-1)\rho(\alpha_l-1)+1)} \\
&\Gamma\left(\frac{(\gamma-1)(\alpha_i-1+\rho(2\alpha_l-\alpha_\nu-\alpha_i))+2}{(\gamma-1)\rho(\alpha_l-1)+1}\right) \quad .
\end{aligned}
$$

Notating: $C_l \triangleq \frac{(\gamma-1)(\alpha_i-1+\rho(2\alpha_l-\alpha_\nu-\alpha_i))+2}{(\gamma-1)\rho(\alpha_l-1)+1}$,$A_{l,\nu} \triangleq \left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]$

$$
\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2(\bar{m}_l\Phi_l)^{-C_l}}{\beta_S((\gamma-1)\rho(\alpha_l-1)+1)}\Gamma(C_l) \tag{22}
$$

When $\gamma \to 1$:

$$
\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2}{\beta_S(\bar{m}_l\Phi_l)^2} \quad . \tag{23}
$$

B.3 Proof of Proposition 3.2

$\forall k \in \mathbb{N}$, let $S_k \geq 0$, $w \in \mathcal{G}$, $C_E < 1$, the following event can be defined:

$$\mathbf{E}^i_{t,k} = \left\{ \sup_{t\in[0,S_k]} |\epsilon \xi^i_{t,k}| < C_E \right\} \quad . \tag{24}$$

There exist $\bar{\epsilon}_0$, s.t $\forall \bar{\epsilon} \leq \bar{\epsilon}_0$, the following is true:

$$\begin{aligned}
\left\{ \sup_{t\in[0,S_k]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right\} &= \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon} X^i_{t,k}(w) + R^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right\} \qquad (25)\\
&\subseteq \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon} X^i_{t,k}(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta \right\} \cup \left\{ |R^i_{t,k}(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta \right\} \\
&\subseteq \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right\} \cup \left\{ \left\{ |R^i_{t,k}(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta \right\} \cap \mathbf{E}^i_{t,k} \right\} \cup \left\{ \left\{ |R^i_{t,k}(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta \right\} \cap \mathbf{E}^c_{t,k^i} \right\} \\
&\subseteq \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right\} \cup \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z\sqrt{C_R}}\bar{\epsilon}^{0.5\theta} \right\} \cup \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq C_E \right\} \\
&\subseteq \left\{ \sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right\} \quad .
\end{aligned}$$

Using Kolmogorov's inequality, for $C_\theta > 0$:

$$\begin{aligned}
P\left(\sup_{t\in[0,S_k]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right) &\leq P\left(\sup_{t\in[0,S_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right) \qquad (26)\\
&\leq \frac{4C_Z^2}{c^2\bar{\epsilon}^{2\theta}}\,\mathbb{E}[\bar{\epsilon}\xi^i_{t,k}]^2 = \frac{8C_Z^2}{c^2}\bar{\epsilon}^{2-2\theta}\left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)}-1}{1-\alpha_l}\right]T \leq \frac{8C_Z^2}{c^2}\left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}}{1-\alpha_l}\right]T \\
&= \bar{C}_\theta \bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}T \quad .
\end{aligned}$$

Final step:

$$P\left(\sup_{t\in[0,T]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right) = \int_0^\infty P\left(\sup_{t\in[0,\tau]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right) \beta_i e^{-\beta_i\tau} d\tau \tag{27}$$

$$\begin{aligned}
&= \bar{C}_\theta \bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta} \int_0^\infty \tau^{1-\rho(1-\alpha_l)+2-2\theta} \beta_i e^{-\beta_i\tau} d\tau \\
&= \bar{C}_\theta \bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta} \frac{\Gamma(2-\rho(1-\alpha_l)+2-2\theta)}{\beta_i^{2-\rho(1-\alpha_l)+2-2\theta}} = C_\theta \bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}
\end{aligned}$$

.

B.4 Proof of Lemma 3.3

In this subsection we will show the full derivation of the approximation of $Z^l_{t,k}$ using stochastic asymptotic expansion, the representation of Z_t in powers of $\bar{\epsilon} = s_t^{\frac{\alpha-1}{\alpha}}\epsilon$:

$$Z^i_{t,k} = Y^i_{t,k} + \bar{\epsilon}X^i_{t,k} + R^i_{t,k} \quad . \tag{28}$$

Where $R^i_{t,k}$ is the error term, we will not discuss this term, for more details see [18]. $X^i_{t,k}$ is the first approximation of $Z^i_{t,k}$ in powers of $\bar{\epsilon}$ and $Y^i_{t,k}$ is the deterministic process. As we show in 4, the relaxation time is much smaller than the interval between the large jumps, hence it's effect on Z_t is negligible, thus we will assume: $Z_{t,k} \approx \bar{\epsilon}X_{t,k}$. $X^i_{t,k}$ satisfying the following stochastic differential equation:

$$X^i_{t,k} = \int_0^t H(Y_p(w))_{ii} Z^i_{p,k} dp + \xi^i_{p,k} \quad . \tag{29}$$

The solution to this equation:

$$X^i_{t,k} = \int_0^t e^{-\int_p^t H(Y_u(w))_{ii} du} d\xi^i_{p,k} \quad . \tag{30}$$

Using integration by parts:

$$X^i_{t,k} = \xi^i_{t,k} - \int_0^t \xi^i_{p,k} H(Y_p(w))_{ii}) e^{-\int_p^t H(Y_u(w))_{ii}) du} dp \tag{31}$$

$$\begin{aligned}
\mathbb{E}[X^l_{t,k}] &=^* \mu^l_\xi t - \int_0^t \mu^l_\xi t H(Y_p(w)))_{ll} e^{-\int_p^t H(Y_u(w))_{ll}) du} dp \qquad (32)\\
&= \mu^l_\xi t - \int_0^t \mu^l_\xi p h_{ll} e^{-\int_p^t h_{ll} du} dp \\
&= \mu^l_\xi t - \int_0^t \mu^l_\xi p h_{ll} e^{-h_{ll}(t-p)} dp \\
&= \mu^l_\xi t - [\mu^l_\xi h_{ll} [(-\frac{(h_{ll}p+1)}{h^2_{ll}}) e^{-h_{ll}(t-p)}]^t_0 \\
&= \mu^l_\xi t - [\mu^l_\xi h_{ll} [(-\frac{(h_{ll}t+1)}{h^2_{ll}}) + (\frac{1}{h^2_{ll}}) e^{-h_{ll}(t)}] \\
&= \mu^l_\xi t + \mu^l_\xi \frac{(h_{ll}t+1)}{h_{ll}} - \mu^l_\xi \frac{1}{h_{ll}} e^{-h_{ll}t} \\
&= \mu^l_\xi (2t + \frac{1}{h_{ll}} - \frac{1}{h_{ll}} e^{-h_{ll}t}) \quad .
\end{aligned}$$

* using Fubini.
Where $\mu^l_\xi \triangleq \mu^l_\xi(t)$ is the first moment of $\xi^l_{t,k}$:

$$\mu^l_\xi(t) \triangleq \mathbb{E}[\xi^l_{t,k}] = 2t \int_1^{\bar\epsilon^{-\rho}} \frac{dy}{y^{\alpha_l}} = 2t [\frac{1}{1-\alpha_l} y^{1-\alpha_l}]^{\bar\epsilon^{-\rho}}_1 = 2t \left[\frac{\bar\epsilon^{-\rho(1-\alpha_l)} - 1}{1-\alpha_l}\right] \quad . \tag{33}$$

We will keep the previous assumptions [19, 18] on the geometry of the potential, that near the basin:$U(w) = h_{ll}\frac{w^2}{2} + o(w^2)$. Hence we can estimate the expected value of a product of the two processes:

$$\begin{aligned}
\mathbb{E}[Z^i_{t,k} Z^j_{t,k}] &= \mathbb{E}[Y^i_t Y^j_t + \bar\epsilon_j Y^i_t X^j_{t,k} + \bar\epsilon_i Y^j_t X^i_{t,k} + \bar\epsilon_j \bar\epsilon_i X^j_{t,k} X^i_{t,k}] \qquad (34)\\
&\approx^* \mathbb{E}[Y^i_t Y^j_t] + \bar\epsilon_j Y^i_t \mathbb{E}[X^j_{t,k}] + \bar\epsilon_i Y^j_t \mathbb{E}[X^i_{t,k}] \\
&= Y^i_t Y^j_t + \bar\epsilon_j Y^i_t \mathbb{E}[X^j_{t,k}] + \bar\epsilon_i Y^j_t \mathbb{E}[X^i_{t,k}] \\
&= Y^i_t Y^j_t + \bar\epsilon_j Y^i_t \mu^j_\xi (2t + \frac{1}{h_{jj}} - \frac{1}{h_{jj}} e^{-h_{jj}t}) + \bar\epsilon_i Y^j_t \mu^l_\xi (2t + \frac{1}{h_{ii}} - \frac{1}{h_{ii}} e^{-h_{ii}t}) \\
&\approx w_i w_j e^{-(h_{ii}+h_{jj})t} + \bar\epsilon_j w_i e^{-h_{ii}t} 2t \left[\frac{\bar\epsilon^{-\rho(1-\alpha_j)} - 1}{1-\alpha_j}\right] (2t + \frac{1}{h_{jj}} - \frac{1}{h_{jj}} e^{-h_{jj}t}) \\
&+ \bar\epsilon_i w_j e^{-h_{jj}t} 2t \left[\frac{\bar\epsilon^{-\rho(1-\alpha_i)} - 1}{1-\alpha_i}\right] (2t + \frac{1}{h_{ii}} - \frac{1}{h_{ii}} e^{-h_{ii}t}) \quad .
\end{aligned}$$

*Neglecting terms with order $\bar\epsilon^2$.

B.5 Proof of Proposition 3.4

SGD's covariance:

$$\Sigma_t = \frac{1}{D} \left[\frac{1}{B} \sum_{i=1}^{Q} \nabla U(W_t)_i \nabla U(W_t)^T_i - \nabla U(W_t) \nabla U(W_t)^T\right] \quad . \tag{35}$$

We can approximate the loss landscape near the basin using Taylor expansion:

$$U(W_t) = U(W^*) + \nabla U(W^*)(W - W^*) + \frac{1}{2}(W_t - W^*)^T \nabla^2 U(W^*)(W_t - W^*) \quad . \tag{36}$$

Examining SGD's gradient on the b-th data point, using the approximation in 36:

$$\nabla U(W_t)_i \approx \nabla U_d(W^*) + \nabla^2 U_d(W^*)(W_t - W^*) \quad . \tag{37}$$

The exact gradient (of GD) is:

$$\nabla U(W_t) \approx \nabla^2 U(W^*)(W_t - W^*) \quad . \tag{38}$$

As a result of empirical evidence in [31] on the minimum of the covariance curve of SGD, we will drop the first order from the approximation of $\nabla U_d(W) \nabla U_d(W)^T$. Hence Eq. 35 can be written as:

$$\lambda(W_t) = \frac{1}{B}\left[\frac{1}{D}\sum_{d=1}^{D} \nabla U_d(W^*)\nabla U_d(W^*)^T + H_d(W^*)W_t W_t^T H_d(W^*) - H(W^*)W_t W_t^T H(W^*)\right] \tag{39}$$

$$\sum_{d=1}^{D} H_d(W^*)W_t W_t^T H_d(W^*) = \frac{1}{D}\sum_{k=1}^{N}\sum_{p=1}^{N}\sum_{d=1}^{D} h_{d,i,k}\tilde{w}_{k,p}h_{d,p,j}$$

Where $\tilde{w}_{ij} = w_i w_j$

$$\begin{aligned}\lambda_{i,j}(t) &= \frac{1}{B}\left[\sum_{k=1}^{N}\sum_{p=1}^{N}(\frac{1}{D}\sum_{d=1}^{D} h_{d,i,k}h_{d,p,j} - h_{i,k}h_{p,j})\tilde{w}_{k,p} + \frac{1}{D}\sum_{d=1}^{D}\nabla u_{d,i}\nabla u_{d,j}\right] \\ &= \frac{1}{B}\left[\sum_{k=1}^{N}\sum_{p=1}^{N}(\frac{1}{D}\sum_{d=1}^{D} h_{d,i,k}h_{d,p,j} - h_{i,k}h_{p,j})\tilde{w}_{k,p} + \tilde{u}_{d,i,j}\right]\end{aligned} \tag{40}$$

$\tilde{u}_{i,j} \triangleq \frac{1}{D}\sum_{d=1}^{D}\nabla u_{d,i}\nabla u_{d,j}$, the gradient of all samples in the dataset. Let us denote: $\bar{h}_{i,k,p,j} \triangleq \frac{1}{D}\sum_{d=1}^{D} h_{d,i,k}h_{d,p,j} - h_{i,k}h_{p,j}+$

$$\begin{aligned}\lambda_{i,j}(t) &= \frac{1}{B}\left[\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}W_{t,k}W_{t,p}\right] \\ &= \frac{1}{B}\left[\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}Z_{t,k}Z_{t,p}\right] \\ &= \frac{1}{B}\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}\,\mathbb{E}[Z_{t,k}Z_{t,p}]\end{aligned} \tag{41}$$

$$\begin{aligned}&\lambda_{i,j}(t) = \frac{1}{BD}\tilde{u}_{ij}+ \\ &\frac{1}{B}[\sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}(w_k w_p e^{-(h_{kk}+h_{pp})t} + \bar{\epsilon}_p w_k e^{-h_{kk}t}\mu_\xi^p(2t + \frac{1}{h_{pp}}(1 - e^{-h_{pp}t})) \\ &+ \bar{\epsilon}_k w_p e^{-h_{pp}t}\mu_\xi^k(2t + \frac{1}{h_{kk}}(1 - e^{-h_{kk}t})))] + \mathcal{O}(\bar{\epsilon}^2)\end{aligned} \tag{42}$$

B.6 Proof of Lemma 3.1

We will denote W^* as the optimal point in the basin, using the differential form, it is known that:

$$\frac{dY_t}{dt} = -\nabla U(Y_t) \quad . \tag{43}$$

Let us denote:$\zeta(t) = U(Y_t) - U(W^*)$, directly from that notation:

$$d\zeta(t) = \langle \nabla U(Y_t), dY_t \rangle = - \left\| \nabla U(Y_t) \right\|^2 \quad . \tag{44}$$

Since $U(Y_t)$ is $\mu-$strongly convex near the basin W^*:

$$\begin{aligned} U(Y_t) - U(W^*) &\leq \frac{1}{2\mu} \left\| \nabla U(Y_t) \right\|^2 \\ - 2\mu\zeta(t) &\geq d\zeta(t) \quad . \end{aligned} \tag{45}$$

Using Gronwall's lemma [11]::

$$U(Y_t) - U(W^*) \leq (U(w) - U(W^*))e^{-2\mu t} \quad . \tag{46}$$

Directly from strong convex propriety $U(Y_t) - U(W^*) \geq \frac{\mu}{2} \left\| Y_t - W^* \right\|^2$, we can achieve:

$$\left\| Y_t - W^* \right\|^2 \leq \frac{2(U(w) - U(W^*))}{\mu} e^{-2\mu t} = \frac{2\zeta(t)}{\mu} e^{-2\mu t} \quad . \tag{47}$$

C Extras

Lemma C.1. *$\forall T \in [S_j, S_{j+1}]$, $\forall j \in \mathbb{N}$,and $\forall w \in [d_i^-, d_i^+]$ there exist a finite C_Z s.t:*

$$\sup_T |X_t^i(w)| \leq C_z^I \sup_T |\xi_t^i| \quad . \tag{48}$$

Using stochastic asymptotic expansion:

$$|X_t^i(w)| \leq \sup_{t\in[0,T]} |\xi_{t,l}| \left(1 + \sup_{t\in[0,T]} \int_0^t H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp \right) \quad . \tag{49}$$

For some $\delta > 0$, the inequality :$m_1^i \leq \sup_{|w|\leq\delta} H(Y_p(w)) \leq \inf_{|w|\leq\delta} H(Y_p(w)) \leq m_2^i$.
Let us denote:

$$C_1 = \max_{w\in\mathcal{G}} \int_0^{\hat{T}} H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp \quad . \tag{50}$$

For arbitrary $\hat{T} \leq t$:

$$\begin{aligned} &\int_0^t H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp = \\ &\qquad \int_0^{\hat{T}} H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp + \int_{\hat{T}}^t H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp \end{aligned} \tag{51}$$

The estimate for the first term:

$$\begin{aligned} &\int_0^{\hat{T}} H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp = e^{-\int_{\hat{T}}^t H(Y_u(w))_{ii} du} \int_0^{\hat{T}} H(Y_p(w))_{ii} e^{-\int_p^{\hat{T}} H(Y_u(w))_{ii} du} dp \\ &\leq e^{-m_1^i (t-\hat{T})} C_1 \leq C_1 \quad . \end{aligned} \tag{52}$$

The second sum:

$$\int_{\hat{T}}^t H(Y_p(w))_{ii} e^{-\int_p^t H(Y_u(w))_{ii} du} dp \leq \int_{\hat{T}}^t m_2^i e^{-m_1^i (t-p)} dp \leq \frac{m_2^i}{m_1^i} \quad . \tag{53}$$

And: $C_Z^l = C_1 + \frac{m_2^l}{m_1^l}$.

D Framework properties and notations

Let us first make few assumptions on the geometry of $\mathcal{G}$ and notations:

1. *Near the basin W^*, $\nabla U : \bar{\mathcal{G}} \to \mathbb{R}^d$.*
2. *U is $\mu-$strongly convex .*
3. *The boundary of our domain is denoted as $\partial\mathcal{G}$, which is a C^1 manifold, so that the vector field of the outer normals on the boundary exists. This means that ∇U "points into $\mathcal{G}$", hence:*
$$\langle \nabla U(w), n(w) \rangle < -\frac{1}{C} \quad , \tag{54}$$
for any $w \in \partial\mathcal{G}$
4. *Zero is an attractor of the domain (i.e. $\nabla U(0) = 0$, and for every starting value $w \in \mathcal{G}$,the deterministic solution vanishes asymptotically:*
$$lim_{t\to\infty} Y_t(w) \to 0 \quad . \tag{55}$$
5. *Let us define the inner part of $\mathcal{G}$ as $\mathcal{G}_\delta = \{y \in \mathcal{G} : dict(w, \partial\mathcal{G}) \geq \delta\}$,*

where $C > 1$.
Let us define $\delta_0 > 0$ as the point which if $\|w\| < \delta_0$ then $w \in \mathcal{G}$ and $\forall \delta \in (0, \delta)$. The following is valid:

- *From the exponential stability of 0, $\|Y_t\| < Ce^{-\frac{1}{C}t} \|w\|$.*
- *For $\|w\| < \delta_0$, and $g^i_{w,+} = w + tr_i$, $g^i_{w,-} = w - tr_i$, we shall define the distance to the boundary as:*
$$d^+_i(w) \triangleq inf\{t > 0 : g^i_{w,+}(t) \in \partial\mathcal{G}\} \quad . \tag{56}$$
- *We will define δ-tubes as $\Omega^+_i(\delta) \triangleq \{w \in \mathbb{R}^d : \|\langle w, r_i\rangle r_i\| < \delta, \langle w, r_i\rangle > 0\} \cap \mathcal{G}^c$ and $\Omega^-_i(\delta) \triangleq \{w \in \mathbb{R}^d : \|\langle w, r_i\rangle r_i\| < \delta, \langle w, r_i\rangle < 0\} \cap \mathcal{G}^c$.*
- *$\mathcal{G}_\delta$ with the dynamic process Y_t and the initial point $w \in \mathcal{G}_\delta$ is a Positively invariant set [1] .*

E Constructing the SDE

Let us first define our SGD iterative update rule:

$$w_k = w_{k-1} - \bar{\eta}_k \nabla U(w_k) + \bar{\eta}_k \zeta_k \quad . \tag{57}$$

$\zeta_k \in \mathbb{R}^N, w_k \in \mathbb{R}^N, \nabla U(w_k) \in \mathbb{R}^N$. Let us remind that $\lambda^k \in R^{NxN}$ approximates the noise covariance matrix :

$$\lambda_k = \frac{1}{D}\left[\frac{1}{B}\sum_{i=1}^{Q} \nabla U(w_k)_i \nabla U(w_k)_i^T - \nabla U(w_k)\nabla U(w_k)^T\right] \quad . \tag{58}$$

The SGN is assumed to be modeled by a Levy-stable random variable, $\zeta^l_k \sim S\alpha S(1^T\lambda^k_l)$, note that $1^T\lambda^k_l$ is a scalar, and it represents the sum of interactions of parameter's l with the rest of the parameters in the DNN. Let us start with the following SDE:

$$W_t = \int_0^t \nabla U(W_p)\, dp + \int_0^t \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T\lambda_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_t) r_l dL^l_t \quad . \tag{59}$$

We aim to use the Euler-Maruyama method and Levy process properties to achieve Eq. 57. Let us define the time discretization constant as $\eta_k > 0$, we split $(0, t)$ to M splits: $0 = \tau_0 < \tau_1 < ... < \tau_k < .. < \tau_{M-1} = t$,where $\tau_i - \tau_{i-1} = \eta$ thus for $\tau_i \in (0, t)$ using Euler-Maruyama method:

$$w_{\tau_k} = w_{\tau_{k-1}} - \nabla U(w_{\tau_k}, \tau_k)(\tau_k - \tau_{k-1}) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha}} ((1^T\lambda_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_{\tau_k}) r_l (L^l_{\tau_k} - L^l_{\tau_{k-1}}) \quad . \tag{60}$$

Model	Data	Gauss	$S\alpha S$ Const α	$S\alpha S$
ResNet18	CINIC10	0.138 ± 0.040	0.156 ± 0.072	$\mathbf{0.066} \pm 0.026$
ResNet34	CINIC10	0.157 ± 0.077	0.233 ± 0.115	$\mathbf{0.114} \pm 0.073$
ResNet50	CINIC10	0.141 ± 0.072	0.147 ± 0.088	$\mathbf{0.096} \pm 0.061$
Bert $[B = 8]$	Cola	0.214 ± 0.064	0.197 ± 0.087	$\mathbf{0.071} \pm 0.032$
Bert $[B = 32]$	Cola	0.032 ± 0.027	0.036 ± 0.019	$\mathbf{0.017} \pm 0.013$

Table 2: The fitting error between SGN and $S\alpha S$/Gaussian distribution. Averaged over $10,000$ randomly sampled parameters. Top three tows, three different CNNs trained on the CINIC10 data with a batch size of 400. Bottom two rows, BERT [8] base model trained on the Cola dataset with different batch sizes B. Sum of Squares Error (SSE) is used to evaluate the fitting error of each distribution. "Gauss" represents the Gaussian distribution. Our results demonstrate that $S\alpha S$ better depicts SGN.

Model	Data	Gauss	$S\alpha S$	$S\alpha S$ Wins
EfficientNet-b2	ImageNet	0.01083 ± 0.0032	0.00101 ± 0.0002	99.58%
EfficientNet-b3	ImageNet	0.01385 ± 0.0034	0.00130 ± 0.0004	99.46%
EfficientNet-b4	ImageNet	0.02062 ± 0.0059	0.001936 ± 0.0006	99.56%
FlexVit	ImageNet	0.02497 ± 0.0102	0.00391 ± 0.0018	99.13%
Vit base	ImageNet	0.04576 ± 0.0191	0.00419 ± 0.0009	99.59%
Vit small	ImageNet	0.0208 ± 0.0095	0.00210 ± 0.0004	99.30%

Table 3: Subset of ImageNet with 200k images. The batch size is 64, and 10,000 parameters were sampled.

Using Levy stationary increments property: The difference $L_m - L_n$, for $m > n$ distributes $L_m - L_n \sim S\alpha S((m-n)^{\frac{1}{\alpha}})$, further for clarity we will mark w_{τ_k} as w_k.

$$w_k = w_{k-1} + \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T \lambda_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_k) r_l S_k^l \quad . \tag{61}$$

Where $S_k^l \sim S\alpha S(\eta^{\frac{1}{\alpha_l}})$. Using $S\alpha S$ characteristic function and the fact L_t is a real value process: $S_k^l = \zeta_k^l \eta^{\frac{1}{\alpha_l}} ((1^T \lambda_l)^{\frac{1}{\alpha_l}})^{-\frac{1}{\alpha_l}}$, let us use this identity:

$$w_k = w_{k-1} + \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T \lambda_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_k) r_l \zeta_k^l \eta^{\frac{1}{\alpha_l}} ((1^T \lambda_l)^{\frac{1}{\alpha_l}})^{-\frac{1}{\alpha_l}} \quad . \tag{62}$$

$$w_k = w_{k-1} - \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta_k r_l \zeta_k^l \quad . \tag{63}$$

Since we defined (for simplicity) r_l as one hot vector we can deduce:

$$w_k = w_{k-1} - \eta_k \nabla U(w_k) + \eta \zeta_k \quad . \tag{64}$$

For the convergence of the Euler-Maruyama discritization please see [20, 35, 2].

F Experimental Section

F.1 Additional results

F.2 LRdecay plot

In figure 2 we present the results of the learning rate decay experiment described in the main text. Specifically, our result suggests that reducing the noise magnitude plays an important role in the dynamics of learning rate decay.

F.3 Empirical evidence of the heavy tail nature of SGN

In figure 3 and 4 we present histograms demonstrating the heavy tail nature of SGN.

F.4 Additional escape time experiments

F.5 α_i Variability

Figure 7 shows how different SGNs attribute different parameters in the same DNN.

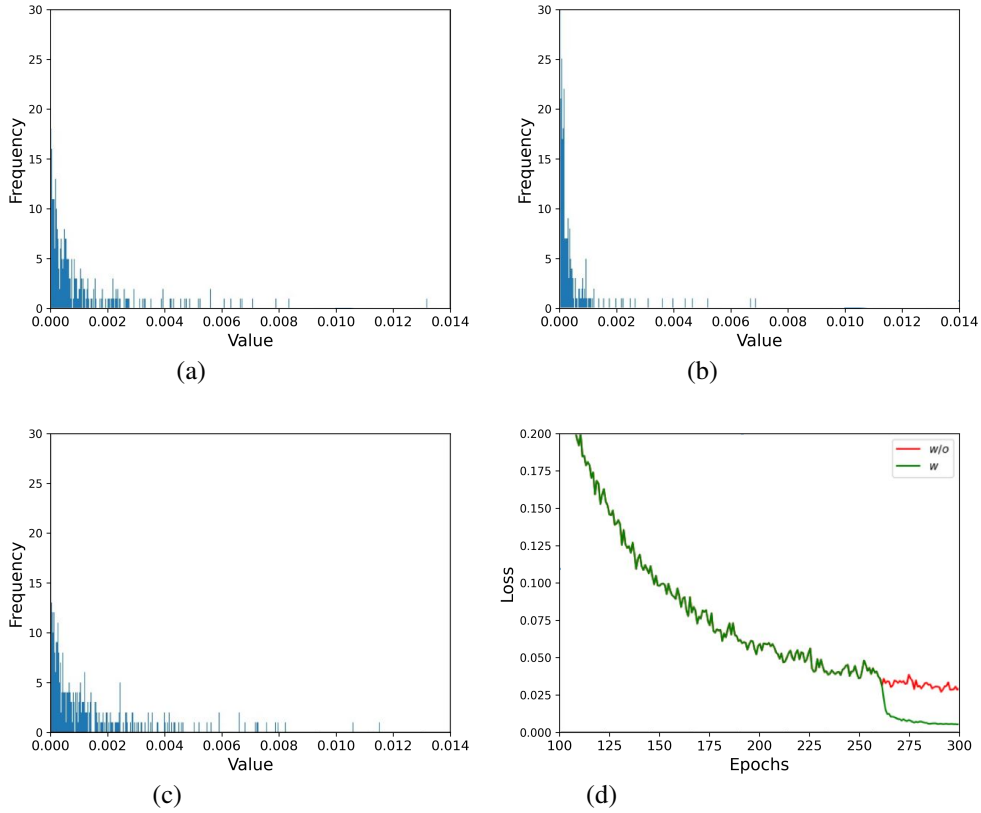

Figure 2: The stochastic gradient noise of a single parameter in ResNet110 [15]. (a) Before applying learning rate decay, at epoch 279. (b) After applying learning rate decay, at epoch 281. (c) Without learning rate decay, at epoch 280. (d) The training loss with and without learning rate decay applied at epoch 280.

F.6 α_i as a function of the layer in the DNN

Fig. 8 The caption explores the heavy-tail level of the SGN for each layer. The left figure depicts ResNet18 [15] on CINIC10 [7] and CIFAR10 [22] , revealing that layers closer to the prediction layer exhibit a higher SGN, suggesting their propensity to escape local minima. The right figure shows Mobilenet on CIFAR100, with multiple layers displaying high α_i values. These layers employ a distinct activation function, HardSigmoid, which involves clipping and contributes to the heavier tails observed.

F.7 Escape axis plot

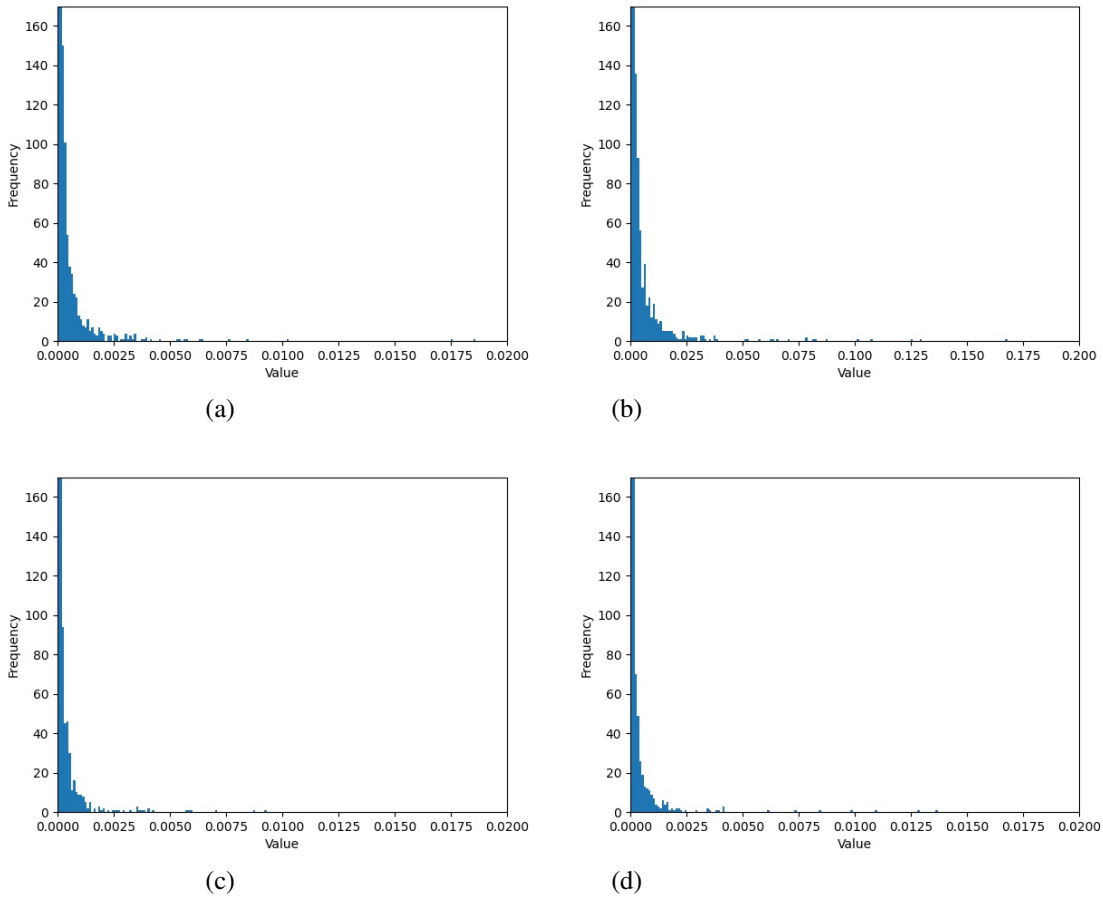

Figure 3: The stochastic gradient noise of a ResNet50 trained on CIFAR100 for four randomly sampled parameters, please zoom in in order to see the long tail behavior.

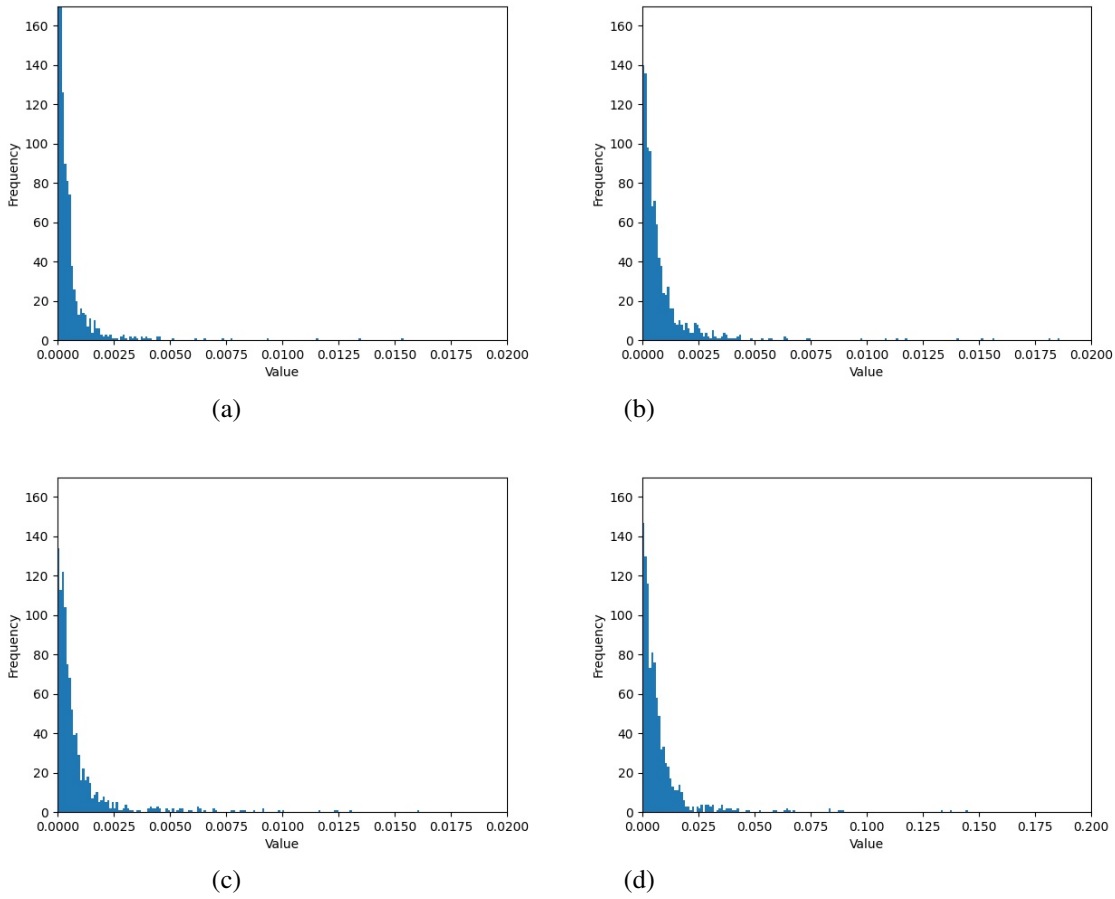

Figure 4: The stochastic gradient noise of a ResNet18 trained on CIFAR100 for four randomly sampled parameters, please zoom in in order to see the long tail behavior.

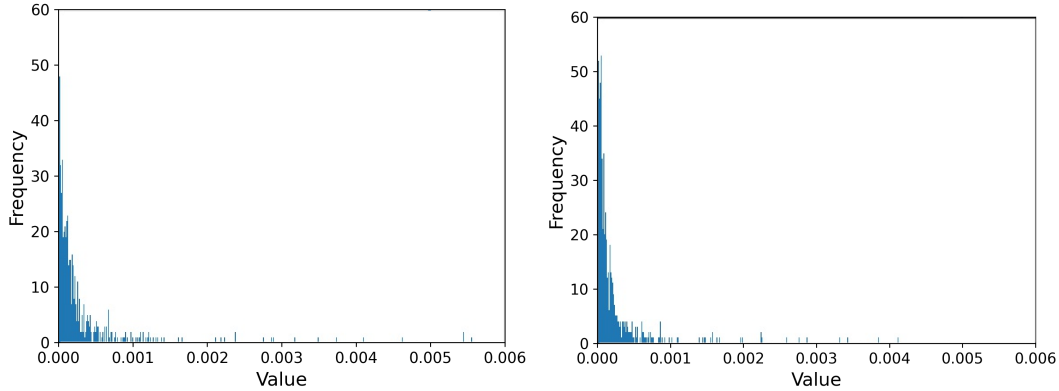

Figure 5: Histograms of the stochastic gradient noise for a single parameter in ResNet34 for: (left) layer number 1, (right) layer number 2. The plots qualitatively show that SGN is far from Normal distribution and presents heavy tail nature.

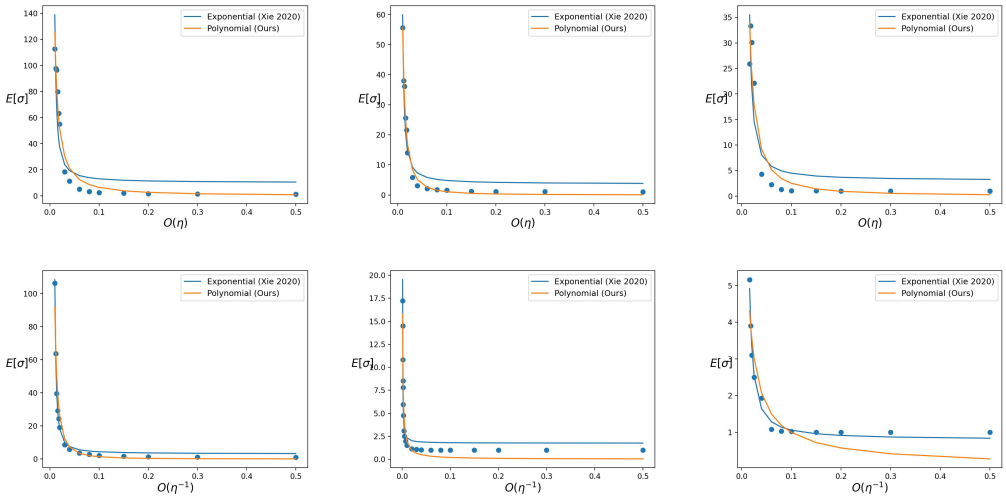

Figure 6: The mean escape time of SGD on Breastw (left), Cardio (middle), and Satellite (right) datasets. The plots show the fitting base on two methods: ours and [46], on the upper row shows escaping with batch size 32, while the bottom row with batch size 8. Each dot represents the mean escape time for a sweep of learning rates. The dot is an average of over 100 random seeds for each learning rate. One can observe that the empiric results are better explained by our theory for a batch size of 32 in all three datasets examined. On the contrary, using batch size 8, our theory overshoot when predicting escape time for the Satellite dataset, which is competitive on Cardio and better on the BreastW dataset.

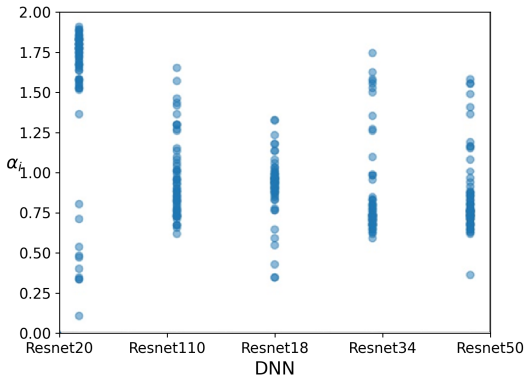

Figure 7: Each dot represents the distribution parameter α_i of a single weight in the DNN. Values on the x-axis represent five different DNNs, left to right: ResNet20/110/18/34/50 [15]; this plot confirms that distinct weights in a DNN lead to different noise distributions during training.

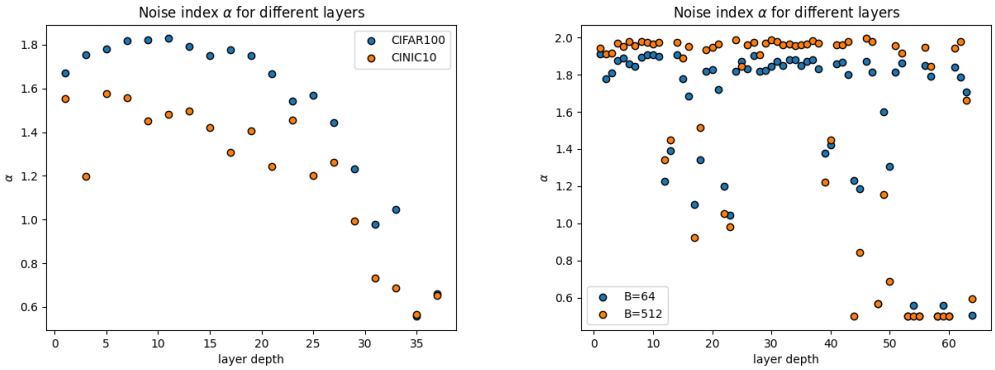

Figure 8: All plots shows the heavy-tail level of the SGN per layer, where low index are layers closer to the input. The left image shows ResNet18 on both CINIC10 and CIFAR10 datasets; a clear pattern is that layers closer to the prediction layer hold heavier SGN, which suggests that those layers are more probable to escape local minima. The right image shows Mobilenet trained on CIFAR100; unlike ResNet18, there are a few layers with high α_i, interestingly those layers contain a unique activation function HardSigmoid, which performs clipping, thus could explain the larger value of α.

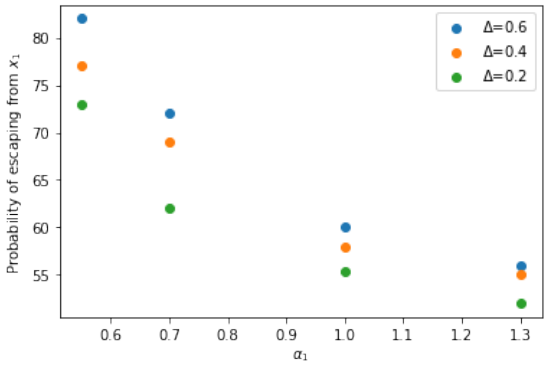

Figure 9: Four different values of α_1 and three values of Δ are selected, and the y-axis shows the probability of escaping from x_1, which is the axis with lower α. For example, the top-left most dot (blue) shows that when $\alpha_1 = 0.55$ and $\alpha_2 = 1.05$ the probability of the process to escape from axis x_1 is $\sim 82\%$.