Revisiting the Noise Model of Stochastic Gradient Descent

Barak Battash
Faculty of Engineering
Bar Ilan University

Lior Wolf
School of Computer Science
Tel Aviv University

Ofir Lindenbaum
Faculty of Engineering
Bar Ilan University

Abstract

The effectiveness of stochastic gradient descent (SGD) in neural network optimization is significantly influenced by stochastic gradient noise (SGN). Following the central limit theorem, SGN was initially described as Gaussian, but recently Simsekli et al. (2019) demonstrated that the $S\alpha S$ Lévy distribution provides a better fit for the SGN. This assertion was purportedly debunked and rebounded to the Gaussian noise model that had been previously proposed. This study provides robust, comprehensive empirical evidence that SGN is heavy-tailed and is better represented by the $S\alpha S$ distribution. Our experiments include several datasets and multiple models, both discriminative and generative. Furthermore, we argue that different network parameters preserve distinct SGN properties. We develop a novel framework based on a Lévy-driven stochastic differential equation (SDE), where one-dimensional Lévy processes describe each parameter. This leads to a more accurate characterization of the dynamics of SGD around local minima. We use our framework to study SGD properties near local minima; these include the mean escape time and preferable exit directions.

1 Introduction

The tremendous success of deep learning (Bengio, 2009; Hinton et al., 2012; LeCun et al., 2015) can be partly attributed to implicit properties of the optimization tools, in particular, the popular SGD (Robbins and Monro, 1951; Bottou, 1991) 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 lead to improved generalization (Ziyin et al., 2021; Smith et al., 2020). Based on empirical evidence, the formation and amplitude of SGD noise depend on the deep neural network (DNN) architecture and data distribution. However, a complete theoretical understanding of this interplay is still lacking. A better understanding of the properties of SGD can help break current barriers in the field.

Analyzing the behavior of SGD with non-convex cost functions is ongoing research (Draxler et al., 2018; Nguyen and Hein, 2017; He et al., 2019b; Li et al., 2017; Smith et al., 2021; Ziyin et al., 2021; You et al., 2019). The problem of analyzing SGD noise has been recently gaining attention. Studies mainly examine the distribution and nature of the noise, with its ability to escape local minima and generalize better (Hu et al., 2017; He et al., 2019a; Wu et al., 2019; HaoChen et al., 2020; Zhou et al., 2019; Keskar et al., 2016).

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

$$w_{\ell+1} = w_\ell - \frac{\eta_\ell}{B} \sum_{d \in \Omega_\ell} \nabla U^{(d)}(w_\ell) = w_\ell - \eta_k \nabla U(w_\ell) + \eta_\ell \zeta_\ell,$$

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

While gradient flow is a popular apparatus for understanding GD dynamics, continuous-time SDE is typically used to investigate the SGD optimization process. By modeling SGD using an SDE, we can examine the evolution of the dynamic system in the continuous time domain (Zhu et al., 2018; Meng et al., 2020; Xie

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

et al., 2020; Chaudhari and Soatto, 2018; Hu et al., 2017; Sato and Nakagawa, 2014a).

There is an ongoing discussion on the characteristics of SGN. Specifically, the majority of previous works (Zhu et al., 2018; Mandt et al., 2016; Wu et al., 2020; Ziyin et al., 2021) argue that SGN is better modeled using the normal distribution, i.e., $\zeta_l \sim \mathcal{N}(0, \Sigma(w_\ell))$, where $\Sigma(w_\ell)$ is the noise covariance matrix and formulated as follows (Zhu et al., 2018):

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

Recently, (Zhu et al., 2018) demonstrated that modeling the SGN as an anisotropic noise leads to an improved approximation of SGD dynamics. Although the SGN process is well modeled by a diffusion driven by a Brownian motion(Sato and Nakagawa, 2014b; Raginsky et al., 2017; Zhang et al., 2017; Mandt et al., 2017; Zhu et al., 2018; Mori et al., 2021), lately Simsekli et al. (2019) argued that SGN obeys $\mathcal{S}\alpha\mathcal{S}$ Lévy motion due to SGN's heavy-tailed nature. This model was allegedly refuted by Xie et al. (2020), claiming that the experiments in (Simsekli et al., 2019) are inaccurate.

In this study, we reclaim that SGN holds a long-tailed distribution by conducting extensive experiments using datasets of multiple types, including images, text, and tables; the evaluations were conducted using multiple architectures. Furthermore, we demonstrate that the noise associated with distinct DNN parameters is distributed differently. Following our empirical evidence, we model the training process as Lévy-driven stochastic differential equations (SDEs) in $\mathbb{R}^N$, where each parameter i has a different α_i. Within this framework, we derive a more accurate expression for mean-escape time, the probability of escaping the local minimum for axis i, and more.

Our contributions are: (1) Demonstrate that the SGN of DNN parameters distributes differently. (2) Show empirically that SGN has heavy-tail properties, making $S\alpha S$ distribution more accurately characterize it visually and numerically using both text, image, and text-image models on multiple datasets. (3) 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. (4) Approximate the mean escape time and the likelihood of escaping the local minima using a particular parameter. We also analyze additional characteristics of the training process near the local minima. (5) Demonstrate theoretically (and empirically) that parameters with low values of α_i will likely assist the training process in leaving the local minima.

Technical Remark A Symmetric α stable distribution ($S\alpha S$ or Lévy $S\alpha S$) is a heavy-tailed distribution, parameterized by the stability parameter α, where a smaller α leads to a heavier tail (i.e., extreme events are more frequent and have a greater amplitude), and vice versa. In this work, $\alpha \in (0.5, 2)$.

2 Related Work

Stochastic optimization has been demonstrated effective for several applications, including generative modeling (Li et al., 2020), support recovery (Lindenbaum and Steinerberger, 2021, 2022; Jana et al., 2023), clustering (Svirsky and Lindenbaum, 2023), and many more. Studying dynamical systems using SDEs with small random perturbations is a well-established field. Early work used Gaussian noise to model the perturbations (Kramers, 1940; Freidlin et al., 2012), which were later replaced by Lévy noise with discontinuous trajectories (Imkeller and Pavlyukevich, 2006a; Imkeller et al., 2010; Imkeller and Pavlyukevich, 2008; Burghoff and Pavlyukevich, 2015). Characterizing the noise as Lévy perturbations has attracted interest in the context of extreme events modeling, such as in climate (Ditlevsen, 1999), physics (Brockmann and Sokolov, 2002) and finance (Scalas et al., 2000).

Modeling SGD using SDEs is a deep-rooted method. Li et al. (2015) 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 (2015) employed a similar procedure to derive an SDE approximation for the SGD dynamics to study the influence of the value of the learning rate. Li et al. (2015) 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 (Sato and Nakagawa, 2014b; Raginsky et al., 2017; Zhang et al., 2017). Later, more accurate modeling suggested (Mandt et al., 2017; Zhu et al., 2018; Mori et al., 2021) using an anisotropic noise covariance matrix. Finally, Simsekli et al. (2019) demonstrated that SGN is better characterized by $S\alpha S$ noise, which was refuted by Xie et al. (2020) (see more details in Section 1).

3 Framework

Prior work that model SGN (Zhou et al., 2020; Simsekli et al., 2019) assume that the noise of each parameter in the network has the same characteristics. In contrast, we model the noise of each parameter using an $S\alpha S$ distribution with a distinct α_i value. We back this assumption with multiple experiments (see

Section 6). This allows us to construct a framework of an N-dimensional dynamic system, representing the update rule of SGD as a Lévy-driven stochastic differential equation. We consider a DNN with 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. 13 in the Appendix for more rigours and detailed definition of $\mathcal{G}$.

The governing SDE that depicts SGDs dynamics inside the domain $\mathcal{G}$ at time t is:

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

where W_t is the process that depicts the evolution of DNN weights at time t. The derivation of this SDE is presented in Sec. 14 of the Appendix. $L_t^l \in \mathbb{R}$ is a mean-zero $S\alpha S$ Lévy processes with a stable parameter α_l. $\Sigma_l(t) \in \mathbb{R}^N$ 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.

Technical 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. $L_l = \xi_t^l + \psi_t^l$, additional information on the $S\alpha S$ process appears in Appendix 9.2.

4 Analysis

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 $\Pi_k^l = \tau_k^l - \tau_{k-1}^l$ 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 4.3.

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 $\Pi_k^* = \tau_k^* - \tau_{k-1}^*$. 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_i value over all parameters, and similarly for other parameters (e.g., β_ν); this will help us describe the global properties of our network.

We denote the horizontal distance from the domain boundary using d_l^+ and d_l^-. We define two additional processes to better understand the dynamics inside the basin (between the large jumps). We present a complete formulation of our assumptions about the domain and processes in Appendix 13.

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 (by the positively invariant set assumption). 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 $t > 0$, U is μ-strongly convex in the domain $\mathcal{G}$, and $w \in \mathcal{G}$ by:

$$Y_t = w - \int_0^t \nabla U(Y_{t'})\, dt'. \tag{1}$$

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

Lemma 1. *$\forall w \in \mathcal{G}$, $\tilde{U} = U(w) - U(W^*)$, the process Y_t converges to a minimizer W^* exponentially fast:*

$$\|Y_t - W^*\|^2 \leq \frac{2\tilde{U}}{\mu} e^{-2\mu t}. \tag{2}$$

The complete proof appears in Appendix 10.6.

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 9.2). 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 $\forall t \in [0, \Pi_k]$ is defined as:

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

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 1. *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:*

$$P\left(\sup_{t \in [0, T_\epsilon]} |Z_{t,k}^l - Y_{t,k}^l| \geq c\bar{\epsilon}^{\theta_l} \right) \leq C_{\theta_l} \bar{\epsilon}^{\bar{\theta}_l}. \tag{3}$$

Where $C_{\theta_l} > 0$ and $c > 0$ are constants, and $\bar{\epsilon}_l = s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon_l$.

Proposition 1 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 jump k and before $k+1$. It indicates that between large jumps, the processes are close to each other with high probability between large jumps. The complete proof appears in Appendix 10.3.

Summary Thus far, we have presented an SDE that controls the training dynamics (Eq. 3), which consist of three components: the deterministic process (Eq. 1), the process of small jumps Eq. 4, which is mostly near the deterministic process as shown in Prop. 1, thus will not aid the training to escape the minimum, the third component is the large jump process.

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-th mini-batch. To represent different parameters, we 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. Finally, $\tilde{h}_{l,m,p,j} := \frac{1}{B}\sum_{b=1}^{B} h_{b,l,m}h_{b,p,j}$, $h_{l,m,p,j} := h_{l,m}h_{p,j}$ and $\bar{h}_{l,m,p,j} := \tilde{h}_{l,m,p,j} - h_{l,m,p,j}$ are used in the next proposition.

4.1 Small jump interactions

We turn our attention to another property of the process of the small jumps $Z^l_{t,k}$. 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 2. *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 $A_{l,j}(t) \triangleq \bar{\epsilon}_l w^j e^{-h_{j,j}t}\mu^l_\xi(2t + \frac{1}{h_{l,l}}(1-e^{-h_{l,l}t}))$, for $t \in [0, \Pi^*_k)$ the following fulfills:*

$$\mathbb{E}[Z^l_{t,k}Z^j_{t,k}] = w^l w^j e^{-(h_{l,l}+h_{j,j})t} + A_{j,l}(t) + A_{l,j}(t) + \mathcal{O}(\epsilon^2). \tag{4}$$

Where $\mu^l_\xi = 2t\left[\frac{\bar{\epsilon}_l{}^{-\rho(1-\alpha_l)}-1}{1-\alpha_l}\right]$, $\bar{\epsilon}_l = s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon_l$. w^j, w^l are the weight value of parameters j and l respectively at time t.

Lemma 2 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 10.4.

4.2 Noise covariance matrix

The covariance of the noise matrix holds a vital role in modeling the training process; in this subsection, we use the stochastic process (presented in Section 13) to derive the expression of the noise covariance matrix. Toward this goal, we use stochastic Taylor expansion near the basin W^*.

Proposition 2. *Let us define $\tilde{u}_{l,j} = \frac{1}{D}\sum_{d=1}^{D}\nabla u_{d,l}\nabla u_{d,j}$, then for any $t \in [0, \Pi^*_k)$, the sum of the l-th row of the covariance matrix:*

$$\mathbf{1}^T\Sigma^k_l(W_t) = \frac{1}{B}\sum_{j=1}^{N}\tilde{u}_{l,j} + \frac{1}{B}\sum_{j,m,p=1}^{N}\bar{h}_{l,m,p,j} \tag{5}$$
$$\left(w^m w^p e^{-(h_{m,m}+h_{p,p})t} + A_{m,p}(t) + A_{pm}(t)\right) + \mathcal{O}(\bar{\epsilon}^2),$$

where $A_{m,p}(t)$ and $A_{p,m}(t)$ are defined in lemma 2. 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 (see details in the previous notations paragraph). The approximation consists mainly of two parts: the first is gradient-based and not time-dependent, and the second is Hessian-based, which decays in time. The proof of the proposition appears in Appendix 10.5. The influence of the batch size B on the noise appears in the denominator of Eq. 5. Suggesting that larger values of B will attenuate the absolute values in the covariance matrix, as expected.

4.3 Jump Intensity

We use $\beta_l(t)$ to denote 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 $\beta_l(t)^{-1}\lambda_{\Psi_l}$, where λ is the levy measure, and the jump intensity is formulated as:

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

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:

$$P(s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J^l_1 \notin [d^-_l, d^+_l]) = \frac{m_l(t)\Phi_l s_t^{\alpha_l-1}}{\beta_l(t)},$$

Model	B	Gauss	$S\alpha S$	$S\alpha S$ Wins
EfficientNet-b2	32	0.0180	**0.0009**	99.74%
EfficientNet-b3	32	0.0241	**0.0010**	99.63%
EfficientNet-b4	32	0.0344	**0.0021**	99.68%
FlexVit	32	0.0340	**0.0021**	99.62%
Vit base	32	0.0649	**0.0066**	99.74%
Vit small	32	0.0287	**0.0030**	99.56%
EfficientNet-b2	64	0.0108	**0.0010**	99.58%
EfficientNet-b3	64	0.0139	**0.0013**	99.46%
EfficientNet-b4	64	0.0206	**0.0019**	99.56%
FlexVit	64	0.0250	**0.0039**	99.13%
Vit base	64	0.0458	**0.0042**	99.59%
Vit small	64	0.0208	**0.0021**	99.30%

Table 1: The fitting error between the empirical SGN and Gaussian or $S\alpha S$ distributions. We evaluate six models on a subset of ImageNet with 200k images and average the error over 10,000 network parameters. Lower values indicate a better fit. $S\alpha S$ Wins- counts the numbers of parameters that are better fitted by $S\alpha S$ distribution. B is the batch size. See Appendix 9 for the standard deviations.

where $m_l(t) = \frac{(\mathbf{1}^T \Sigma_l(t))^{\frac{1}{\alpha_l}} \epsilon_l^{\alpha_l}}{\alpha_l}$, and $\Phi_l = (-d_l^-)^{-\alpha_l} + (d_l^+)^{-\alpha_l}$.

5 Theorems

The following section provides a theoretical analysis of SGD dynamics during DNN training. Our analysis is based on two empirical pieces of evidence demonstrated in this work; the first is that SGN is indeed heavy-tailed (Tables 1,2,3). The second is that each parameter in the DNN's training process has a different stability parameter α (Figures 1,5), which can significantly affect the noise properties.

Our work assumes that the training process can exit from the domain only at times that coincide with large jumps; please see Sec. 9 for the mathematical evidence. Following this assumption, 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 used in 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: $C_{l,\nu} \triangleq \frac{2+(\gamma-1)(\alpha_l - 1 + \rho(\alpha_l - \alpha_\nu))}{1+(\gamma-1)(\alpha_l-1)}$, where $C_{l,\nu}$ 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 1. *Given $C_{l,\nu}$ and $A_{l,\nu}$, let s_t be an exponential scheduler $s_t = t^{\gamma-1}$, the mean transition time from the domain $\mathcal{G}$:*

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

Where Γ is the gamma function, $\bar{m}_l = \frac{\bar{\Sigma}_l^{\alpha_l} \epsilon_l^{\alpha_l}}{\alpha_l}$ and $\bar{\beta}_l = \frac{2\epsilon_l^{\rho\alpha_l}}{\alpha_l}$ is the time independent jump intensity. See Appendix 10.1 for the full proof.

It can be observed from Thm. 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 1. *Using Thm. 1, if the cooling rate is negligible, i.e $\gamma \to 1$, the mean transition time:*

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

Col. 1 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 edge $(max(-d_i^-, d_i^+))$ of the domain $\mathcal{G}$ does

Model	B	Gauss	$S\alpha S$	$S\alpha S$ Wins
Clip-b	32	0.0038	**0.0028**	96.60%
Clip-b	64	0.0034	**0.0029**	96.80%
Clip-b	256	0.0040	**0.0036**	96.88%
Clip-l	32	0.0033	**0.0028**	96.67%

Table 2: Fitting experiment on Clip-l (Clip large) and Clip-b (clip-base). SGN is estimated on a subset of Laion400M, and we average the fitting error over 10,000 parameters. $S\alpha S$ Wins- counts the portion of network parameters that are better fitted by $S\alpha S$ distribution.

Model	B	Gauss	$S\alpha S$	$S\alpha S$ Wins
SD2.0	4	0.0073	**0.0068**	94.92%
SDXL1.0	2	0.0056	**0.0050**	96.58%
SDXL DDPO	2	0.0046	**0.0041**	88.41%

Table 3: The fitting error between the empirical of SGN and $S\alpha S$ or Gaussian parametric distribution. The table shows SD2 and SDXL1.0 (Podell et al., 2023) are finetuned using, and SDXL DDPO is finetuned using the DDPO (Black et al., 2023) . The empirical distribution of SGN is the average of at least 10,000 parameters. Note the $B = 4$ is the largest batch size that can be processed by our hardware.

not affect the mean escape time. Furthermore, one can note that the escape time dependency on the learning rate is polynomial.

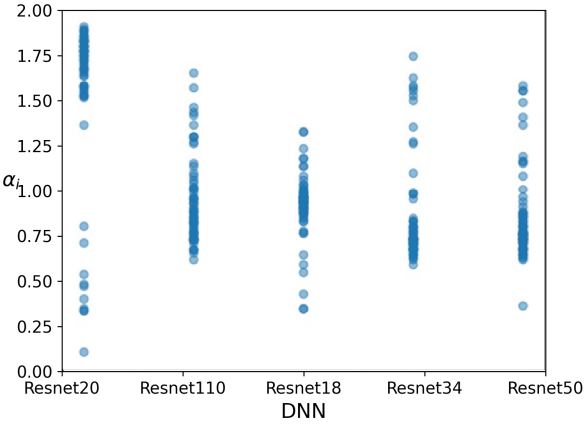

Figure 1: Each dot represents the distribution parameter α_i of a single weight in the DNN. Values on the x-axis represent five different DNNs, left to right: ResNet20/110/18/34/50 He et al. (2015); this plot confirms that distinct weights in a DNN lead to different noise distributions during training.

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 2. *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:*

$$P(W_\sigma \in \Omega_i^+(\delta)) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{m}_i \Phi_i}{\bar{\beta}_i} (d_i^+)^{-\alpha_i} \qquad (6)$$
$$\frac{\beta_l^2 (\bar{m}_l \Phi_l)^{-C_l}}{\beta_S((\gamma-1)(\alpha_l-1)+1)} \Gamma(C_l).$$

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 in Eq. 6. 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 evaluate the escaping ratio of two parameters.

Corollary 2. *The ratio of probabilities for exiting the local minima from two different DNN parameters is:*

$$\frac{P(W_\sigma \in \Omega_l^+(\delta))}{P(W_\sigma \in \Omega_j^+(\delta))} \leq \frac{\mathbf{1}^T \Sigma_l}{\mathbf{1}^T \Sigma_j} \eta^{(\alpha_l-\alpha_j)(1-\rho)} \frac{(d_l^+)^{-\alpha_l}}{(d_j^+)^{-\alpha_j}}.$$

We remind the reader that (d_i^+) is a function of the horizontal distance from the domain's edge. Therefore, we conclude 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:

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

where $\Delta_{l,j} = \alpha_l - \alpha_j$. Our experimental findings indicate that parameters from layers closer to the output, particularly the classifier, display a heavier-tailed SGN distribution. This observation suggests a higher probability of exiting from one of the final layers than from others. More details and experiments are presented in Section 6

The next theorem evaluates the probability of exiting the basin after time u.

Theorem 3. *Let the scheduler be $s_t = t^{\gamma-1}$, where γ is the cooling rate; let us denote two constants that express the effect of the scheduler: $\gamma_l \triangleq 1+(\gamma-1)(\alpha_l-1)$ and $\kappa \triangleq \frac{1+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{\gamma_l}$, for $u > 0$:*

$$P(\sigma > u) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{\beta}_l \bar{m}_l \Phi_l}{\bar{\beta}_S \gamma_l (\bar{m}_l \Phi_l)^\kappa} \Gamma(\kappa, \bar{m}_l \Phi_l u^{\gamma_l}) \quad .$$

In the following corollary, we now show that the probability of exiting a basin after u iterations decay exponentially with respect to u, $\bar{m}_l$, and Φ_l.

Corollary 3. *Using Thm. 3, for $\gamma \to 1$:*

$$P(\sigma > u) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{\beta}_l}{\bar{\beta}_S} e^{-\bar{m}_l \Phi_l u} \quad .$$

The value Φ_l describes the horizontal width of the basin, and $\bar{m}_l$ is a function of the learning rate and the noise covariance matrix. Our proof appears in Appendix 11.4.

6 Experiments

This section presents the core experimental results supporting our analysis; additional experiments can

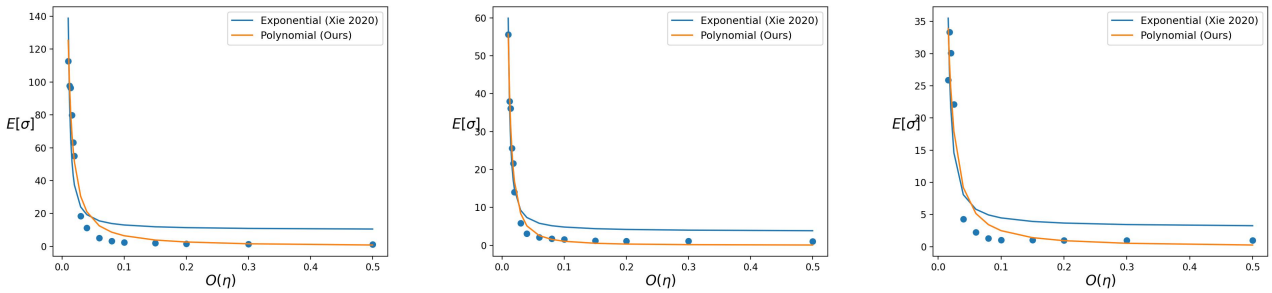

Figure 2: The mean escape time of SGD on Breastw (left), Cardio (middle), and Satellite (right) datasets. The dots represent the empirical mean escape time for different learning rates. The fitted curves are based on our model (orange) and (Xie et al., 2020) (blue) with a batch size of 32. Each dot represents an average of over 100 random seeds for each learning rate. Our theory better explains the empirical results for all three datasets examined.

be found in Appendix 8. To demonstrate the heavy-tailed nature of SGN, we have explored multiple datatypes and architectures, including multi-layer perception (MLP), residual NN (ResNet), CLIP (Radford et al., 2021), latent diffusion models (von Platen et al., 2022), Bert (Devlin et al., 2018), vision transformers (Fan et al., 2021), and more.

Stochastic gradient noise distribution We empirically show that SGN is better characterized using the $S\alpha S$ Lévy distribution. Our evaluation follows Xie et al. (2020), calculating the noise of each parameter separately using multiple mini-batches, as opposed to (Simsekli et al., 2019) that calculated the noise of multiple parameters on one mini-batch and averages over all parameters and batches to characterize the distribution of SGN. We use four extensive experiments to provide solid empirical evidence that SGN is indeed heavy-tailed. To quantify the goodness of fit of our noise model, we present the fitting error of the empirical distribution of SGN to Gaussian and $S\alpha S$ distributions. In each experiment, we also quantify the portion of NN parameters that are better characterized by the $S\alpha S$ distribution ($S\alpha S$ Wins).

The first experiment is based on an image recognition task using a subset of the ImageNet Krizhevsky et al. (2012) dataset. Here, we evaluate six different models and two different batch sizes. Our results are presented in Tab. 1.

In the second experiment, we use Clip (Radford et al., 2021) variants on the Laion400M (Schuhmann et al., 2021) dataset; this task involves both text and images as input data. Two variants of Clip are used: Clip base (3 batch size variations) and Clip-large. In Tab. 2, we present the results.

The third experiment examines the SGN for generative models. We use two Latent diffusion models: Stable Diffusion2 [1](SD2) and latest SDXL1.0 (Podell et al., 2023) model, both finetuned on "pokemon-blip-captions" dataset [2] using Diffusers (von Platen et al., 2022) regimes, Finally, evaluate SDXL1.0(Podell et al., 2023) (Podell et al., 2023) finetuned using denoising diffusion policy optimization (DDPO) (Black et al., 2023). The results are presented in Tab. 3.

Our numeric results (Tables 1-3), which cover many commonly used architectures, datasets, and tasks, strongly support that SGN is heavy-tailed. Specifically, in all our evaluations, the fitting error was smaller for the $S\alpha S$ distribution than for the Gaussian model. Moreover, $S\alpha S$ led to a better fit in most individual parameter evaluations.

In the fourth experiment, we trained three ResNet variants and a Bert-based architecture that were trained on CINIC10 (Darlow et al., 2018), CIFAR100 (Krizhevsky, 2009), and the CoLA (Warstadt et al., 2018) dataset. In this experiment, we further compare to an $S\alpha S$ distribution with different values of α for distinct parameters. In Fig. 3 we show qualitative results and numeric results of this evaluation appear in Tab. 6. Our results demonstrate that our proposed SGN model of an $S\alpha S$ distribution with parameters specific α_i values leads to the best fit. More technical details appear in Appendix 8.1.

Different parameters hold different noise distributions? In this section, we perform two experiments; the first demonstrates that distinct DNN parameters lead to different SGN properties. The second shows how the nature of SGN can change for different layers. We randomly sampled 10,000 parameters from five different DNNs in the first experiment. Then, we calculated the SGN and estimated α_i for each parameter; Fig. 1 depicts the results. We observe that different parameters have noise that distributes differently during training. We can further notice that the vari-

[1]https://huggingface.co/stabilityai/stable-diffusion-2

[2]https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions

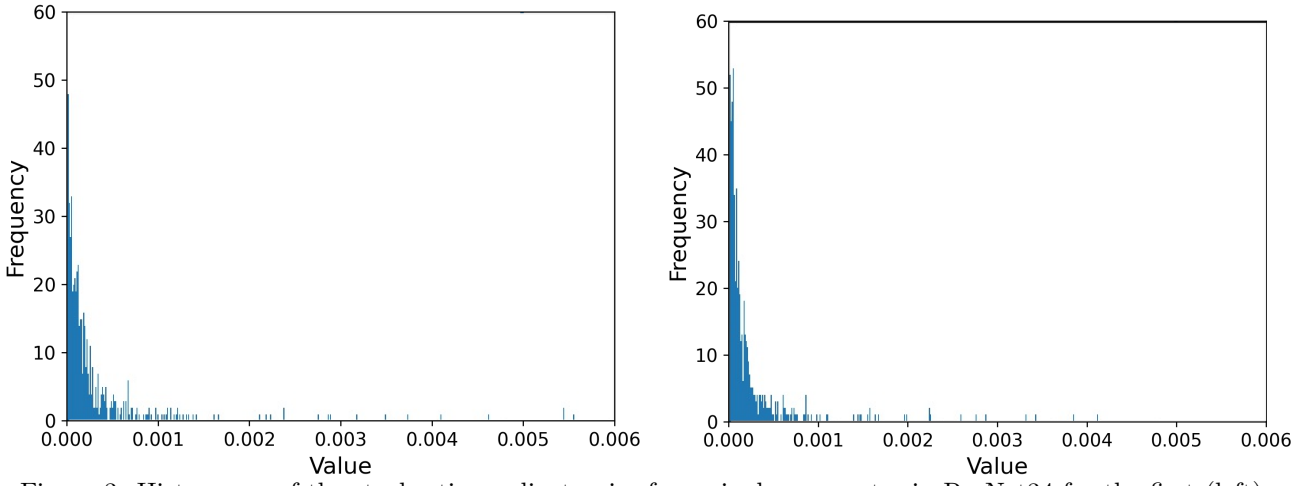

Figure 3: Histograms of the stochastic gradient noise for a single parameter in ResNet34 for the first (left) and second (right) layers. The plots qualitatively show that SGN presents a heavy tail nature.

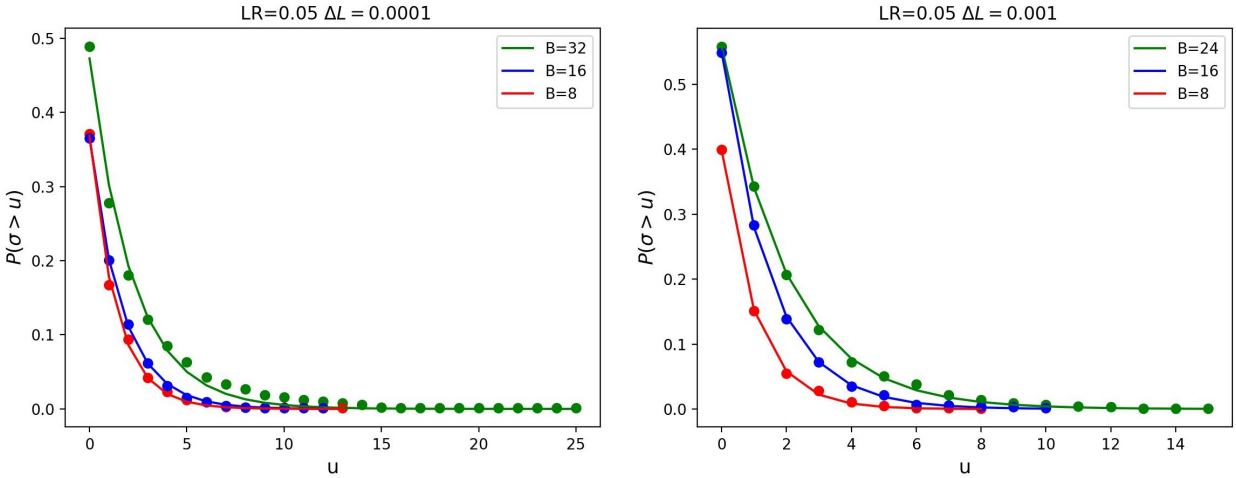

Figure 4: Validation of Thm.3. x-axis represents the number of iterations,y-axis represents the probability of exiting the basin. We train the same model, 1000 seeds, and store the iteration that corresponds to a basin escape. The left plot shows results on the Cardio dataset with different mini-batch sizes, and the right plot shows the same on the Speech dataset. The exponential decay predicted by our theorem (lines) coincides with the empirical results (dots).

ability of the heavy-tail indicator is stretched on large segments of α_i values; this is another evidence that strengthens the heavy-tail hypothesis.

In the second experiment, we randomly sampled 20 parameters from each layer in the DNN, calculated the noise, and fitted α_i. Then, we averaged the α_i values over a specific layer. The plots of α as a function of the layer index can be seen in the appendix 8.6. The plots illustrate two interesting phenomenons; first, DNN's final layers exhibit heavier-tail SGN. Second, in the mobilenetV3 (Koonce and Koonce, 2021) experiments, it can be seen the Squeeze-and-Excite layers have much lower α, one element that may cause this effect is the "Hard Sigmoid" activation, which consists with clipping.

This implies that building a framework that considers the DNN as one homogeneous system is insufficient; each parameter in the DNN has its characteristics, and we should consider this when modeling the noise. Models were trained as detailed in Appendix 8.1.

Mean escape time The following experiment validates Thm 1. We trained a three-layer neural network with Relu activation on "BreastW," "Satellite," and "Cardio" datasets (Dua and Graff, 2017). We first train the model using SGD with a batch size of 256 until reaching a local minimum (see discussion Appendix 9.3). After reaching the critical point, we decrease the mini-batch size to 32 and try to escape the critical minimum, Fig 2 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 (Xie et al., 2020), with the same amount of free parameters. The results in Fig 2 show the mean escape time using a batch size of 32; we observe that our theory better explains the

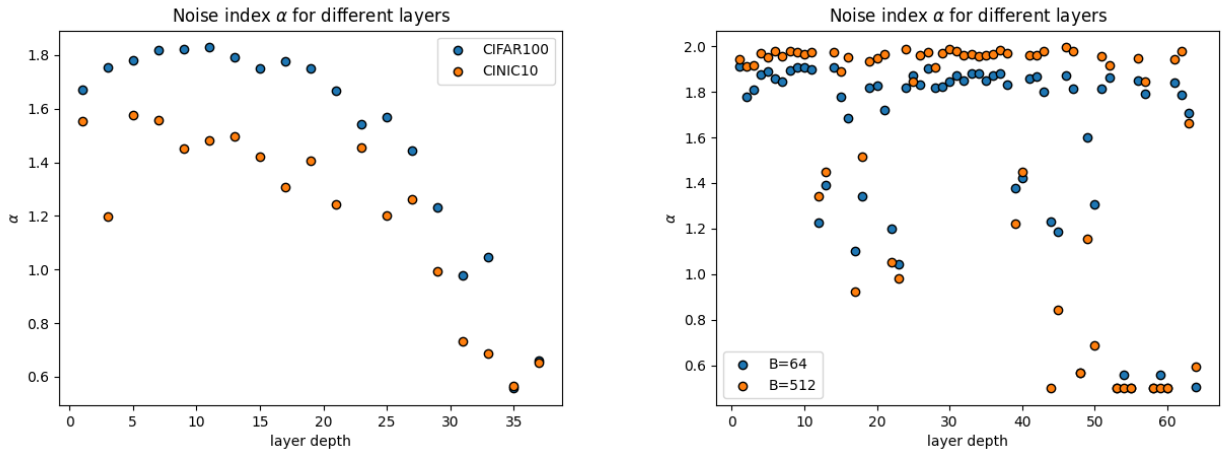

Figure 5: All plots show the heavy-tail level of the SGN per layer, where low index are layers closer to the input. The left image shows ResNet18 on both CINIC10 and CIFAR10 datasets; a clear pattern is that layers closer to the prediction layer hold heavier SGN, suggesting that those layers are more likely 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.

empirical results on all three datasets. Our method shows limitations when using small batch sizes, as depicted in Appendix 8.4.

Probability of escaping after time u. The following experiment validates Thm. 3. We trained a three-layer neural network with Relu activation on Speech, Cardio, and dataset (Dua and Graff, 2017) using SGD with a learning rate of 0.05 and batch size of 128 until convergence to local minima. We measure the time to escape the local minimum on 1000 seeds and plot the probability distribution to exit as a function of time in Fig. 4. These results demonstrate that our theoretical results coincide with the empiric evidence.

Learning rate decay The heavy tail behavior of SGN may prevent the training process from converging to a critical point due to the large jump process; hence, reducing the frequency and size of the large jumps may be crucial for good convergence. This experiment demonstrates that learning rate decay can improve generalization by attenuating the SGN. We show that the performance gain stems from attenuating the noise magnitude and variance and not from reducing the deterministic step towards $-\nabla U$. The details and results are presented in Appendix. 8.2.

Escape Axis In this experiment, we demonstrate that the optimization process is more probable to escape from the axis with lower α_i, as proposed in Col. 2. The details of and result of this experiment are presented in Appendix 8.7.

7 Conclusions

We revisit the noise model of SGD and present extensive systematic evaluations demonstrating that SGN is indeed heavy-tailed. We further show that distinct parameters are characterized by different distribution parameters, namely α values. Our experiments corroborate 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 our 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 better characterize 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. We also presented experiments that support the claim that a significant feature of LR schedulers comes from reducing the fluctuations of the SGN. Finally, we show that parameters in the DNN that hold noise that distributes with low α_i have a unique role in the training process, helping the training process escape local minima.

Limitations and Future Research 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. Another interesting question for future work involves analyzing what causes specific parameters to have smaller α values than others. We believe that addressing such questions could improve the architectural design of DNNs.

Acknowledgments LW was supported by a grant from the Tel Aviv University Center for AI and Data Science (TAD).

References

Amann, H. (2011). *Gewöhnliche differentialgleichungen*. Walter de Gruyter.

Bally, V. and Talay, D. (1996). The law of the euler scheme for stochastic differential equations: Ii. convergence rate of the density.

Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. (2023). Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

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

Brockmann, D. and Sokolov, I. (2002). Lévy flights in external force fields: from models to equations. *Chemical Physics*, 284(1-2):409–421.

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

Chaudhari, P. and Soatto, S. (2018). 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.

Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J. (2018). CINIC-10 is not ImageNet or CIFAR-10. *arXiv e-prints*, page arXiv:1810.03505.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Ditlevsen, P. D. (1999). Observation of α-stable noise induced millennial climate changes from an ice-core record. *Geophysical Research Letters*, 26(10):1441–1444.

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

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835.

Freidlin, M., Szücs, J., and Wentzell, A. (2012). *Random Perturbations of Dynamical Systems*. Grundlehren der mathematischen Wissenschaften. Springer.

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

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

He, F., Liu, T., and Tao, D. (2019a). Control batch size and learning rate to generalize well: Theoretical and empirical evidence.

He, H., Huang, G., and Yuan, Y. (2019b). Asymmetric valleys: Beyond sharp and flat local minima. *arXiv preprint arXiv:1902.00744*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385.

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

Hu, W., Junchi Li, C., Li, L., and Liu, J.-G. (2017). On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv e-prints*, page arXiv:1705.07562.

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

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

Imkeller, P. and Pavlyukevich, I. (2008). Metastable behaviour of small noise lévy-driven diffusions. *ESAIM: Probability and Statistics*, 12:412–437.

Imkeller, P., Pavlyukevich, I., and Stauch, M. (2010). First exit times of non-linear dynamical systems in d perturbed by multifractal lévy noise. *Journal of Statistical Physics*, 141(1):94–119.

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

Jana, S., Li, H., Yamada, Y., and Lindenbaum, O. (2023). Support recovery with projected stochastic gates: Theory and application for linear models. *Signal Processing*, 213:109193.

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

Koonce, B. and Koonce, B. (2021). Mobilenetv3. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pages 125–144.

Kramers, H. A. (1940). Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

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

Li, H., Lindenbaum, O., Cheng, X., and Cloninger, A. (2020). Variational diffusion autoencoders with random walk sampling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 362–378. Springer.

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

Li, Q., Tai, C., and E, W. (2015). Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv e-prints*, page arXiv:1511.06251.

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

Lindenbaum, O. and Steinerberger, S. (2021). Randomly aggregated least squares for support recovery. *Signal Processing*, 180:107858.

Lindenbaum, O. and Steinerberger, S. (2022). Refined least squares for support recovery. *Signal Processing*, 195:108493.

Mandt, S. and Blei, D. M. (2015). Continuous-time limit of stochastic gradient descent revisited.

Mandt, S., Hoffman, M. D., and Blei, D. M. (2016). A Variational Analysis of Stochastic Gradient Algorithms. *arXiv e-prints*, page arXiv:1602.02666.

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

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

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

Mori, T., Ziyin, L., Liu, K., and Ueda, M. (2021). Power-law escape rate of SGD. *arXiv e-prints*, page arXiv:2105.09557.

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

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

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

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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

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

Sato, I. and Nakagawa, H. (2014a). Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 982–990, Bejing, China. PMLR.

Sato, I. and Nakagawa, H. (2014b). 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.

Scalas, E., Gorenflo, R., and Mainardi, F. (2000). Fractional calculus and continuous-time finance.

Physica A: Statistical Mechanics and its Applications, 284(1-4):376–384.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (2021). Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

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

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

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

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*.

Svirsky, J. and Lindenbaum, O. (2023). Interpretable deep clustering. *arXiv preprint arXiv:2306.04785*.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. (2022). Diffusers: State-of-the-art diffusion models.

Warstadt, A., Singh, A., and Bowman, S. R. (2018). Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Wightman, R. (2019). Pytorch image models. `https://github.com/rwightman/pytorch-image-models`.

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

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

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

You, K., Long, M., Wang, J., and Jordan, M. I. (2019). How Does Learning Rate Decay Help Modern Neural Networks? *arXiv e-prints*, page arXiv:1908.01878.

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

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

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

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

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

Supplementary Material

8 Experimental Section

This section includes further experiments, details on the experiments presented in the main paper, and more results and visual evidence.

8.1 Full Technical details

We trained several CNNs on the CINIC dataset Darlow et al. (2018) and the BERT base model on CoLA Warstadt et al. (2018) dataset. All models are trained until reaching convergence. Using the pre-trained weights, we sample $10,000$ 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; the Sum of square error (SSE) is used to evaluate the quality of our fit. In Xie et al. (2020), 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. Specifically, since we want to estimate the escape time, we argue that a pre-trained DNN would better characterize this property.

We trained four ResNet variants Resnet18/34/50. Those models were trained using SGD optimizer, a learning rate of 0.01, and a batch size of 400. We examine the SGD noise of the BERT model, which was fine-tuned on CoLA Warstadt et al. (2018) 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. 8.1. Visual examples for the heavy-tailed nature of SGN can be seen in Fig. 3, and additional results are presented in Sec. 8.3.

In the ImageNet experiments, we examined the SGN using six image recognition models trained on the full ImageNet dataset, and the SGN was estimated using a set of 200k images (200 images from each class) with batch sizes 32 and 64. We use the timm (Wightman, 2019) package for the pre-trained models. The complete tables, including standard deviations, can be found in tables. 4 5

In the text-image experiments we use Clip (Radford et al., 2021) variants on the Laion400M (Schuhmann et al., 2021) dataset; this task involves both text and images as input data. Two variants of Clip are used: Clip base (3 batch size variations) and Clip-large. In Tab. 7, we present the full results, including the standard deviation.

The third experiment examines the SGN for generative models. We use two Latent diffusion models: Stable Diffusion2 [3](SD2), with batch size of 4, and latest SDXL1.0 model (Podell et al., 2023) with batch size of 2 , both finetuned on "pokemon-blip-captions" dataset [4] using Diffusers von Platen et al. (2022) regimes. In the above experiments, 13k and 17k parameters were sampled, all of them from the Unet, which is the only part that is finetuned(which is the common practice). Finally, evaluate SDXL1.0Podell et al. (2023) Podell et al. (2023) finetuned using denoising diffusion policy optimization (DDPO) Black et al. (2023). The full results are presented in Tab. 8.

[3]https://huggingface.co/stabilityai/stable-diffusion-2

[4]https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions

Model	Gauss	$S\alpha S$	$S\alpha S$ Wins
EfficientNet-b2	0.0180 ± 0.0049	0.00092 ± 0.0001	99.74%
EfficientNet-b3	0.02410 ± 0.0058	0.00096 ± 0.0001	99.63%
EfficientNet-b4	0.03439 ± 0.0089	0.00213 ± 0.0006	99.68%
FlexVit	0.03399 ± 0.0156	0.00211 ± 0.0003	99.62%
Vit base	0.06495 ± 0.0264	0.00656 ± 0.00126	99.74%
Vit small	0.02870 ± 0.0131	0.0030 ± 0.0009	99.56%

Table 4: Subset of ImageNet, 200k images, all models loaded using the timm package. Batch size 32. 10,000 parameters were sampled; lower mean is better. $S\alpha S$ Wins- defines numbers of parameters better fitted by $S\alpha S$ distribution

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

Table 5: Subset of ImageNet, 200k images, all models loaded using the timm package. Batch size 64. 10,000 parameters were sampled. $S\alpha S$ Wins- defines numbers of parameters better fitted by $S\alpha S$ distribution

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

Table 6: 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 Devlin et al. (2018) 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	Gauss	$S\alpha S$	$S\alpha S$ Wins
Clip-b [B=32]	$0.0038 \pm 3.83e^{-6}$	$0.0028 \pm 2.76e^{-6}$	96.60%
Clip-b [B=64]	$0.0034 \pm 3.00e^{-6}$	$0.0029 \pm 2.44e^{-6}$	96.80%
Clip-b [B=256]	$0.0040 \pm 2.64e^{-6}$	$0.0036 \pm 2.08e^{-6}$	96.88%
Clip-l [B=32]	$0.0033 \pm 3.03e^{-6}$	$0.0028 \pm 2.41e^{-6}$	96.67%

Table 7: Fitting experiment on two variations of CLIP model: Clip-l represents clip large, Clip-b represents clip-base. SGN calculation is on a subset of Laion400M (200k images). 10,000 parameters were sampled. $S\alpha S$ Wins- defines numbers of parameters better fitted by $S\alpha S$ distribution

Model	Gauss	$S\alpha S$	$S\alpha S$ Wins
SD2.0	$0.0073 \pm 4.13e^{-6}$	$0.0068 \pm 3.67e^{-6}$	94.92%
SDXL1.0	$0.0056 \pm 3.60e^{-6}$	$0.0050 \pm 3.01e^{-6}$	96.58%
SDXL DDPO	$0.0046 \pm 3.12e^{-6}$	$0.0041 \pm 2.90e^{-6}$	88.41%

Table 8: The fitting error between SGN and $S\alpha S$/ Gaussian distribution, on image generation task. The table shows SD2.0 and SDXL1.0 are finetuned using, SDXL DDPO is finetuned using DDPO technique. The empirical distribution of SGN is the average of at least 10,000 parameters.

8.2 Learning rate decay

The heavy tail behavior of SGN may prevent the training process from converging to a critical point due to the large jump process; hence reducing the frequency and size of the large jumps may be crucial for good convergence. This paragraph aims to demonstrate that the LRdecay's effectiveness may be due to the attenuation of SGN. We show two experiments. First, we trained ResNet110 He et al. (2015) on CIFAR100 Krizhevsky (2009), on epoch 280, the learning rate is decreased by a factor of 10. Fig. 6 shows that the learning rate decay results in a lower noise amplitude and less variance. In the second experiment, a ResNet20 He et al. (2015) is trained in three different variations for 90 epochs; the first variation had LRdecay at epochs 30 and 60, the second had a batch-size increase at epochs 30 and 60, the third was trained with the same learning rate and batch size for the entire training process, the results show almost identical results on the first two cases, (i.e., LRdecay and batch increase) reaching a top-1 score of 66.7 and 66.4 on the validation set. In contrast, the third led to worse performances reaching a top-1 score of 53. Smith et al. (2017) performed similar experiments to show the similarity between decreasing the learning rate and increasing the batch size; however, their purpose was to suggest a method for improving training speed without degrading the results.

LRdecay decreases the step size and the noise amplitude; however, increasing the batch size only decreases the noise amplitude. Combining the results of the two experiments above, we may carefully deduce that the main effect of LRdecay is reducing the fluctuation in the gradient update phase and not decreasing the step size (step size is the movement of the deterministic process towards the minus of the gradient). SGN amplitude reduction enables the training process to get easier localization in the current promising domain. In figure 6 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.

8.3 Empirical evidence of the heavy tail nature of SGN

In figure 7 and 8 we present histograms demonstrating the heavy tail nature of SGN.

8.4 Additional escape time experiments

Please see Fig. 9.

8.5 α_i Variability

Figure 1 shows how different SGNs attribute different parameters in the same DNN.

8.6 α_i as a function of the layer in the DNN

Fig. 5 The caption explores the heavy-tail level of the SGN for each layer. The left figure depicts ResNet18 He et al. (2015) on CINIC10 Darlow et al. (2018) and CIFAR10 Krizhevsky (2009) , 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.

8.7 Escape axis

In this section, we demonstrate that the optimization process is more probable to escape from the axis with lower α_i. We use a 2D Ackleys function; the escape process starts at the global minimum $\vec{0}$. We apply Gradient Descent with added $S\alpha S$ noise ($S\alpha S(\alpha_{x_1})$,$S\alpha S(\alpha_{x_2})$), where $\alpha_1 = \alpha_2 - \Delta$, learning rate of $1e-4$, with no momentum or weight decay. Once the optimization process passes some predefined radius, we check which axis is larger. Fig 10 shows how probable it is to exit from x_1 based on 1000 different seeds. This result implies that as the gap Δ between the α_i values increases, the axis with the smaller value of α is more probable to lead to an escape from the local minimum.

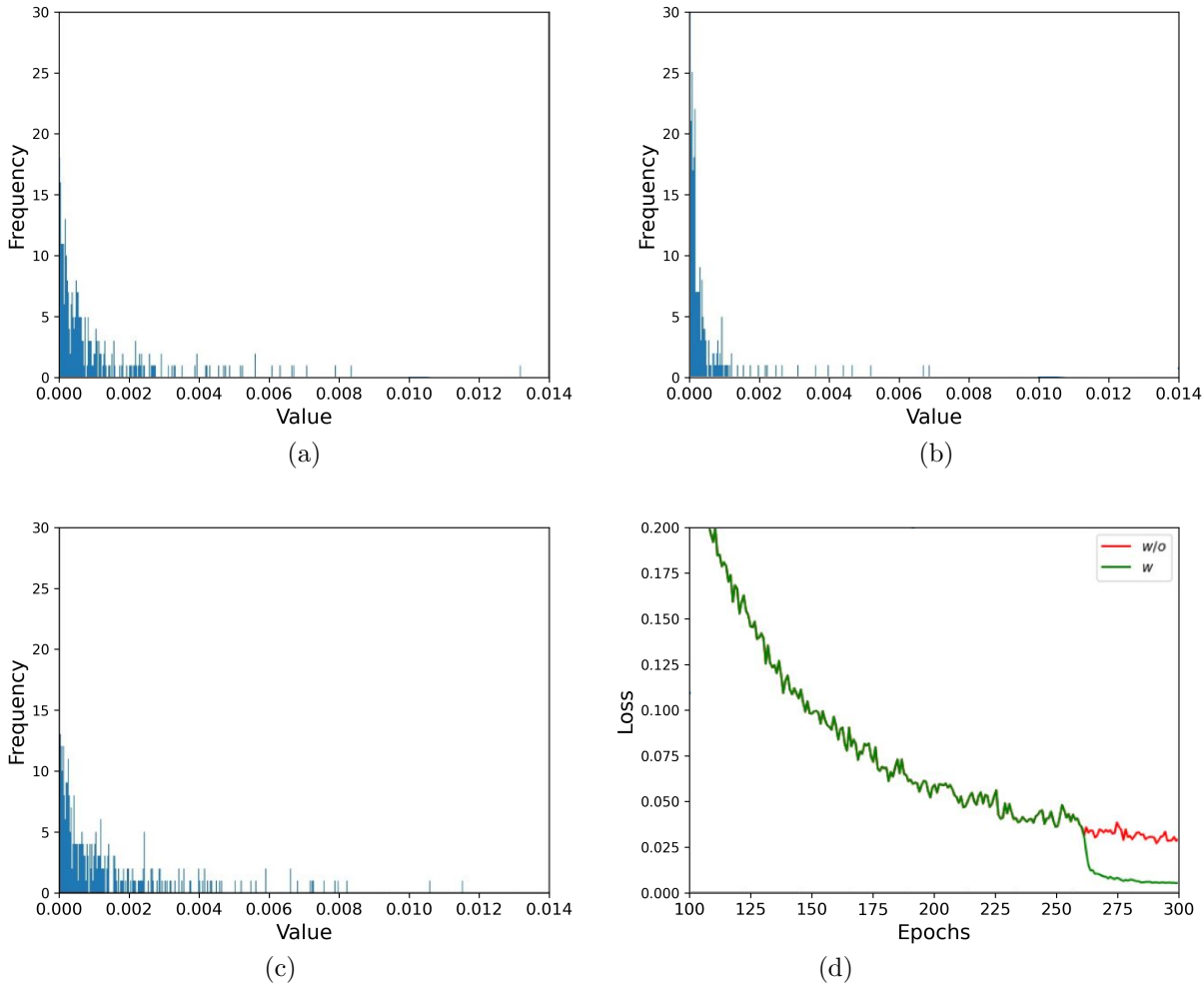

Figure 6: The stochastic gradient noise of a single parameter in ResNet110 He et al. (2015). (a) Before applying learning rate decay, at epoch 279. (b) After applying learning rate decay, at epoch 281. (c) Without learning rate decay, at epoch 280. (d) The training loss with and without learning rate decay applied at epoch 280.

9 Additional technical details

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

9.1 Notations

9.2 $S\alpha S$ 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:

$$\mathbb{E}[e^{iwL_t^l}] = exp\{-t\int_{\mathbb{R}/\{0\}}[e^{iwy} - 1 - iwy\mathbb{I}\{|y| \leq 1\}]\frac{dy}{|y|^{1+\alpha_l}}\}, \tag{7}$$

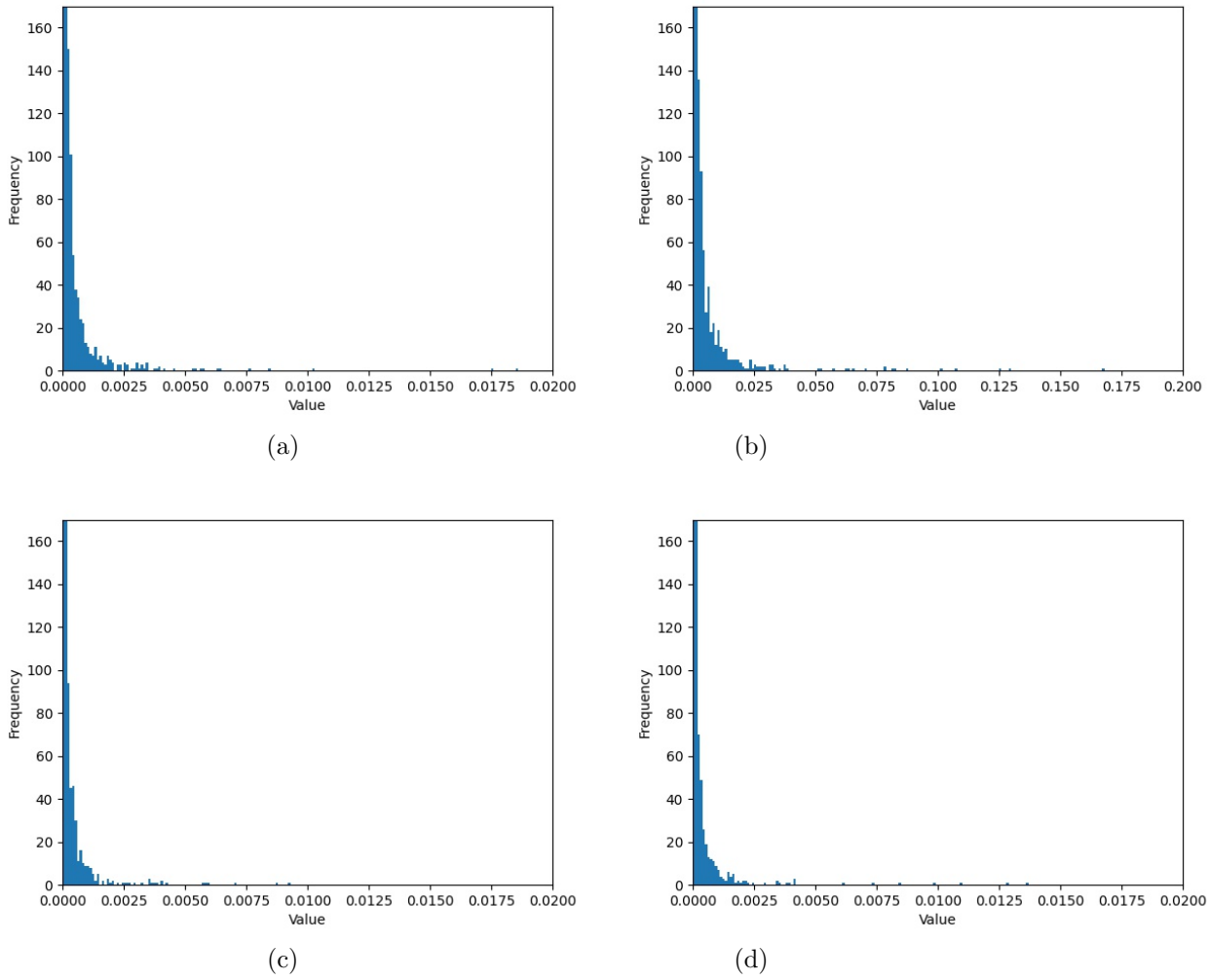

Figure 7: The stochastic gradient noise of a ResNet50 trained on CIFAR100 for four randomly sampled parameters, please zoom in in order to see the long tail behavior.

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. 4.3

9.3 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.

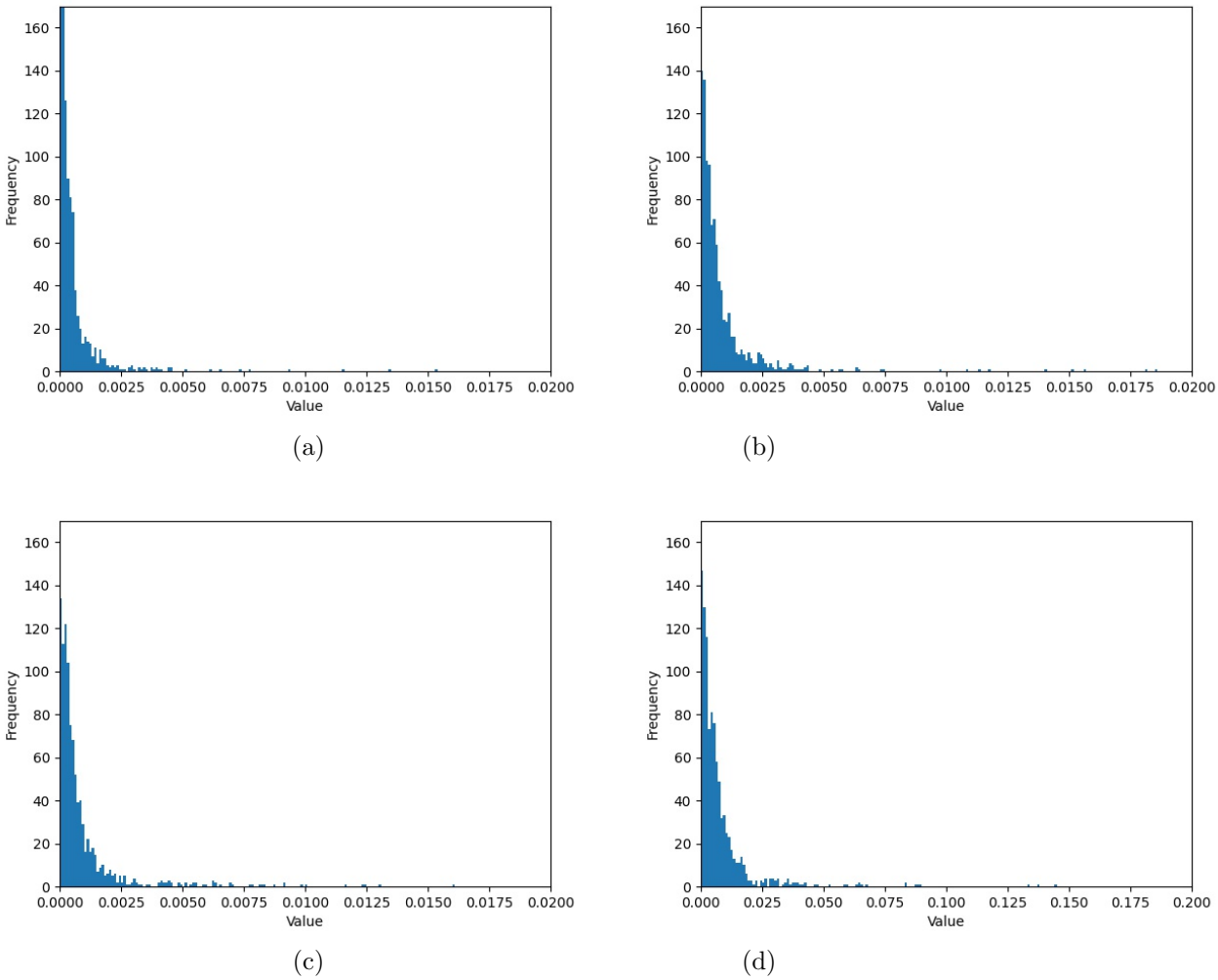

Figure 8: The stochastic gradient noise of a ResNet18 trained on CIFAR100 for four randomly sampled parameters, please zoom in in order to see the long tail behavior.

9.4 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^*:

$$U(W) = U(W^*) + \nabla U(W^*)(W - W^*) + \frac{1}{2}(W - W^*)^T H(W^*)(W - W^*) \tag{8}$$

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

9.5 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 13. Second, Y_t converges to the minimum much faster than the average temporal gap between the large jumps; third, using lemma 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 $C_1 > 0$,

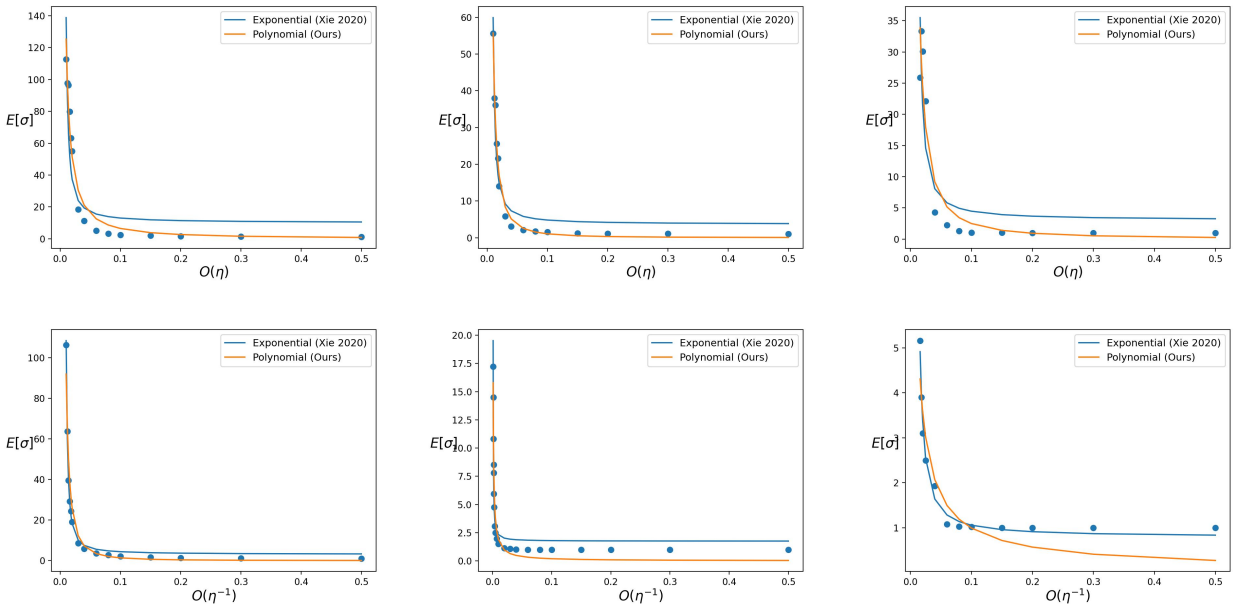

Figure 9: The mean escape time of SGD on Breastw (left), Cardio (middle), and Satellite (right) datasets. The plots show the fitting base on two methods: ours and Xie et al. (2020), on the upper row shows escaping with batch size 32, while the bottom row with batch size 8. Each dot represents the mean 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.

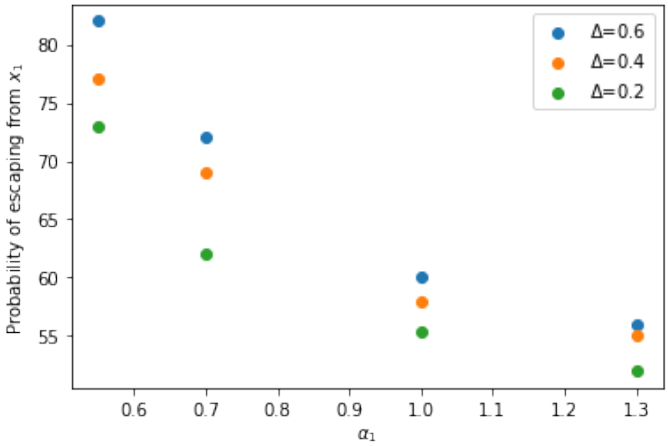

Figure 10: Four different values of α_1 and three values of Δ are selected, and the y-axis shows the probability of escaping from x_1, which is the axis with lower α. For example, the top-left most dot (blue) shows that when $\alpha_1 = 0.55$ and $\alpha_2 = 1.05$, the process's probability of escaping from axis x_1 is $\sim 82\%$.

the relaxation time is

$$T_R^l = \max\left\{\int_{d_l^-}^{-\bar{\epsilon}_l^\zeta} \frac{dy}{-U'(y)_l}, \int_{\bar{\epsilon}_l^\zeta}^{d_l^+} \frac{dy}{U'(y)_l}\right\} \leq C_1 |ln\bar{\epsilon}_l|. \tag{9}$$

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.
Σ	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

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

$$\mathbb{E}[\Pi_k^l] = \mathbb{E}[\tau_k^l - \tau_{k-1}^l] = \beta_l^{-1} = \frac{\alpha_l}{2}\bar{\epsilon_l}^{-\rho\alpha_l}. \tag{10}$$

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.

10 Proofs

10.1 Proof of Theorem 1

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

$$\begin{aligned}\mathbb{E}[\sigma_{\mathcal{G}}] &= \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*]\mathbb{I}\{\sigma_{\mathcal{G}} = \tau_k^*\}] \\ &= \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*\mathbb{I}\{\sum_{l=0}^{N} s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_1^l\mathbb{I}\{\tau_1^l = \tau_1^*\} \in \mathcal{G}, \sum_{l=0}^{N} s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_2^l\mathbb{I}\{\tau_2^l = \tau_2^*\} \in \mathcal{G} \\ &, .., \sum_{l=0}^{N} s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_k^l\mathbb{I}\{\tau_k^l = \tau_k^*\} \notin \mathcal{G}\}]\end{aligned} \tag{11}$$

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

$\mathbb{I}\{\tau_w^l = \tau_k^*\}$ 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:

$$
\begin{aligned}
\mathbb{I}\{\tau_w^l = \tau_k^*\} = \frac{\beta_l(t)}{\beta_S(t)} \binom{k-1}{w-1} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w} \\
\frac{\beta_l(t)}{\beta_S(t)} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}
\end{aligned} \tag{12}
$$

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 \\
&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 \\
&\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)} t e^{-\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)} t e^{-\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)} t e^{-\beta_l(t)t} s_t^{\alpha_l - 1} m_l(t) \Phi_l
\end{aligned} \tag{13}
$$

$$\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1} e^{-(s_t^{\alpha_l-1}m_l(t)\Phi_l t-\beta_l(t)t)}dt$$
$$\leq \sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\int_0^\infty \frac{\beta_l}{\beta_S} t s_t^{\alpha_l-1+\rho(\alpha_l-\alpha_\nu)}\bar{m}_l\Phi_l e^{-s_t^{\alpha_l-1}\bar{m}_l\Phi_l t}dt$$
$$= \sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\int_0^\infty \frac{\beta_l}{\beta_S} t^{1+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}\bar{m}_l\Phi_l e^{-t^{1+(\gamma-1)\rho(\alpha_l-1)}\bar{m}_l\Phi_l}dt$$
$$= \sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\beta_l\bar{m}_l\Phi_l}{\beta_S(1+(\gamma-1)\rho(\alpha_l-1))}$$
$$\left[(\bar{m}_l\Phi_l)^{-\frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)\rho(\alpha_l-1)}}\Gamma\left(\frac{2+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{1+(\gamma-1)\rho(\alpha_l-1)}\right)\right]dt$$
$$= \sum_{l=0}^{N} A_{l,\nu}^{-1}\frac{\beta_l\bar{m}_l\Phi_l}{\beta_S(1+(\gamma-1)\rho(\alpha_l-1))}(\bar{m}_l\Phi_l)^{-C_{l,\nu,p}}\Gamma\left(C_{l,\nu,p}\right)dt \quad .$$

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:

$$\mathbb{E}[\sigma_{\mathcal{G}}] \leq \sum_{l=0}^{N} A_{l,\nu}^{-1}\frac{1}{\beta_S(1^T\Sigma_l)^{\frac{1}{\alpha_l}}\epsilon^{\alpha_l(1-\rho)}\Phi_l}. \tag{14}$$

10.2 Proof of Theorem 2

$$P(W_\sigma \in \Omega_i^+(\delta)) = \sum_{k=1}^{\infty}\prod_{j=1}^{k-1} P(J_j^* \in \mathcal{G})P(J_k^* \in \Omega_i^+) \tag{15}$$
$$= \sum_{k=1}^{\infty}\prod_{j=1}^{k-1} P(J_j^* \in \mathcal{G})P(J_k^* > d_i^+)$$
$$= \sum_{k=1}^{\infty}(1-P(J_k^* \notin \mathcal{G}))^{k-1}P(J_k^* \geq d_i^+)$$
$$\leq \sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k-1}(1-P(s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_w^l \notin \mathcal{G}))^{w-1}$$
$$(1-P(s_t^{\frac{\alpha_\nu-1}{\alpha_\nu}}\epsilon(\mathbf{1}^T\Sigma_\nu(t))^{\frac{1}{\alpha_\nu}}J_w^v \notin \mathcal{G}))^{k-w}P(J_w^l \geq d_i^+)P(\tau_w^l = \tau_k^*)$$
$$= \sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k-1}\int_0^\infty \frac{\beta_l(t)}{\beta_S(t)}\frac{(k-1)!}{(w-1)!(k-w)!}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}\beta_l(t)$$
$$e^{-\beta_l(t)t}\frac{(\beta_l(t)t)^{w-1}}{(w-1)!}[1-\frac{s_t^{\alpha_l-1}m_l(t)}{\beta_l(t)}\Phi_l]^{w-1}[1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu]^{k-w}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i}$$
$$= \sum_{k=1}^{\infty}\sum_{l=1}^{N}\int_0^\infty \frac{\beta_l(t)}{\beta_S(t)}\beta_l(t)e^{-\beta_l(t)t}\frac{s_t^{\alpha_i-1}m_i(t)}{\beta_i(t)}(d_i^+)^{-\alpha_i}$$
$$\sum_{w=1}^{k-1}[1-\frac{s_t^{\alpha_l-1}m_l(t)}{\beta_l(t)}\Phi_l]^{w-1}\frac{(k-1)!}{(w-1)!(k-w)!}\frac{(\beta_l(t)t)^{w-1}}{(w-1)!}$$
$$\left((1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right)^{k-w}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}dt$$

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

$$
\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2(\bar{m}_l\Phi_l)^{-C_l}}{\beta_S((\gamma-1)\rho(\alpha_l-1)+1)}\Gamma\left(C_l\right) \tag{16}
$$

When $\gamma \to 1$:

$$
\sum_{l=0}^{N}A_{l,\nu}^{-1}\frac{\bar{m}_i\Phi_i}{\bar{\beta}_i}(d_i^+)^{-\alpha_i}\frac{\beta_l^2}{\beta_S(\bar{m}_l\Phi_l)^2} \quad . \tag{17}
$$

10.3 Proof of Proposition 1

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

$$
\mathbf{E}_{t,k}^i = \left\{\sup_{t\in[0,\Pi_k]}|\epsilon\xi_{t,k}^i| < C_E\right\} \quad . \tag{18}
$$

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

$$
\begin{aligned}
\left\{\sup_{t\in[0,\Pi_k]}|Z_{t,k}^i(w)-Y_{t,k}^i(w)| \geq c\bar{\epsilon}^\theta\right\} &= \left\{\sup_{t\in[0,\Pi_k]}|\bar{\epsilon}X_{t,k}^i(w)+R_{t,k}^i(w)| \geq c\bar{\epsilon}^\theta\right\} \\
&\subseteq \left\{\sup_{t\in[0,\Pi_k]}|\bar{\epsilon}X_{t,k}^i(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta\right\} \cup \left\{|R_{t,k}^i(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta\right\} \\
&\subseteq \left\{\sup_{t\in[0,\Pi_k]}|\bar{\epsilon}\xi_{t,k}^i| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta\right\} \cup \left\{\left\{|R_{t,k}^i(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta\right\} \cap \mathbf{E}_{t,k}^i\right\} \cup \left\{\left\{|R_{t,k}^i(w)| \geq \frac{c}{2}\bar{\epsilon}^\theta\right\} \cap \mathbf{E}_{t,k^i}^c\right\}
\end{aligned} \tag{19}
$$

$$\subseteq \left\{ \sup_{t\in[0,\Pi_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right\} \cup \left\{ \sup_{t\in[0,\Pi_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z\sqrt{C_R}}\bar{\epsilon}^{0.5\theta} \right\} \cup \left\{ \sup_{t\in[0,\Pi_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq C_E \right\}$$
$$\subseteq \left\{ \sup_{t\in[0,\Pi_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right\} \quad .$$

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

$$P\left(\sup_{t\in[0,\Pi_k]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right) \leq P\left(\sup_{t\in[0,\Pi_k]} |\bar{\epsilon}\xi^i_{t,k}| \geq \frac{c}{2C_Z}\bar{\epsilon}^\theta \right) \tag{20}$$
$$\leq \frac{4C_Z^2}{c^2\bar{\epsilon}^{2\theta}}\,\mathbb{E}[\bar{\epsilon}\xi^i_{t,k}]^2 = \frac{8C_Z^2}{c^2}\bar{\epsilon}^{2-2\theta}\left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)}-1}{1-\alpha_l}\right]T \leq \frac{8C_Z^2}{c^2}\left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}}{1-\alpha_l}\right]T$$
$$= \bar{C}_\theta\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}T \quad .$$

Final step:

$$P\left(\sup_{t\in[0,T]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right) = \int_0^\infty P\left(\sup_{t\in[0,\tau]} |Z^i_{t,k}(w) - Y^i_{t,k}(w)| \geq c\bar{\epsilon}^\theta \right)\beta_i e^{-\beta_i\tau}d\tau \tag{21}$$
$$= \bar{C}_\theta\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}\int_0^\infty \tau^{1-\rho(1-\alpha_l)+2-2\theta}\beta_i e^{-\beta_i\tau}d\tau$$
$$= \bar{C}_\theta\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}\frac{\Gamma(2-\rho(1-\alpha_l)+2-2\theta)}{\beta_i^{2-\rho(1-\alpha_l)+2-2\theta}} = C_\theta\bar{\epsilon}^{-\rho(1-\alpha_l)+2-2\theta}$$

.

10.4 Proof of Lemma 2

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 $\bar{\epsilon} = s_t^{\frac{\alpha-1}{\alpha}}\epsilon$:

$$Z^i_{t,k} = Y^i_{t,k} + \bar{\epsilon}X^i_{t,k} + R^i_{t,k} \quad . \tag{22}$$

Where $R^i_{t,k}$ is the error term, we will not discuss this term, for more details see Imkeller and Pavlyukevich (2006a). $X^i_{t,k}$ 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 5, 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:

$$X^i_{t,k} = \int_0^t H(Y_p(w))_{ii}Z^i_{p,k}dp + \xi^i_{p,k} \quad . \tag{23}$$

The solution to this equation:

$$X^i_{t,k} = \int_0^t e^{-\int_p^t H(Y_u(w))_{ii}du}d\xi^i_{p,k} \quad . \tag{24}$$

Using integration by parts:

$$X^i_{t,k} = \xi^i_{t,k} - \int_0^t \xi^i_{p,k}H(Y_p(w))_{ii})e^{-\int_p^t H(Y_u(w))_{ii})du}dp \tag{25}$$

$$\mathbb{E}[X^l_{t,k}] =^* \mu^l_\xi t - \int_0^t \mu^l_\xi t H(Y_p(w)))_{ll}e^{-\int_p^t H(Y_u(w))_{ll})du}dp \tag{26}$$
$$= \mu^l_\xi t - \int_0^t \mu^l_\xi p h_{ll}e^{-\int_p^t h_{ll}du}dp$$
$$= \mu^l_\xi t - \int_0^t \mu^l_\xi p h_{ll}e^{-h_{ll}(t-p)}dp$$

$$
\begin{aligned}
&= \mu_\xi^l t - [\mu_\xi^l h_{ll}[(-\frac{(h_{ll}p+1)}{h_{ll}^2})e^{-h_{ll}(t-p)}]_0^t \\
&= \mu_\xi^l t - [\mu_\xi^l h_{ll}[(-\frac{(h_{ll}t+1)}{h_{ll}^2}) + (\frac{1}{h_{ll}^2})e^{-h_{ll}(t)}] \\
&= \mu_\xi^l t + \mu_\xi^l \frac{(h_{ll}t+1)}{h_{ll}} - \mu_\xi^l \frac{1}{h_{ll}} e^{-h_{ll}t} \\
&= \mu_\xi^l (2t + \frac{1}{h_{ll}} - \frac{1}{h_{ll}} e^{-h_{ll}t}) \quad .
\end{aligned}
$$

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

$$
\mu_\xi^l(t) \triangleq \mathbb{E}[\xi_{t,k}^l] = 2t \int_1^{\bar{\epsilon}^{-\rho}} \frac{dy}{y^{\alpha_l}} = 2t[\frac{1}{1-\alpha_l} y^{1-\alpha_l}]_1^{\bar{\epsilon}^{-\rho}} = 2t \left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_l)} - 1}{1-\alpha_l}\right] \quad . \tag{27}
$$

We will keep the previous assumptions Imkeller and Pavlyukevich (2006b,a) 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:

$$
\begin{aligned}
\mathbb{E}[Z_{t,k}^i Z_{t,k}^j] &= \mathbb{E}[Y_t^i Y_t^j + \bar{\epsilon}_j Y_t^i X_{t,k}^j + \bar{\epsilon}_i Y_t^j X_{t,k}^i + \bar{\epsilon}_j \bar{\epsilon}_i X_{t,k}^j X_{t,k}^i] \\
&\approx^* \mathbb{E}[Y_t^i Y_t^j] + \bar{\epsilon}_j Y_t^i \mathbb{E}[X_{t,k}^j] + \bar{\epsilon}_i Y_t^j \mathbb{E}[X_{t,k}^i] \\
&= Y_t^i Y_t^j + \bar{\epsilon}_j Y_t^i \mathbb{E}[X_{t,k}^j] + \bar{\epsilon}_i Y_t^j \mathbb{E}[X_{t,k}^i] \\
&= Y_t^i Y_t^j + \bar{\epsilon}_j Y_t^i \mu_\xi^j (2t + \frac{1}{h_{jj}} - \frac{1}{h_{jj}} e^{-h_{jj}t}) + \bar{\epsilon}_i Y_t^j \mu_\xi^l (2t + \frac{1}{h_{ii}} - \frac{1}{h_{ii}} e^{-h_{ii}t}) \\
&\approx w_i w_j e^{-(h_{ii}+h_{jj})t} + \bar{\epsilon}_j w_i e^{-h_{ii}t} 2t \left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_j)} - 1}{1-\alpha_j}\right] (2t + \frac{1}{h_{jj}} - \frac{1}{h_{jj}} e^{-h_{jj}t}) \\
&+ \bar{\epsilon}_i w_j e^{-h_{jj}t} 2t \left[\frac{\bar{\epsilon}^{-\rho(1-\alpha_i)} - 1}{1-\alpha_i}\right] (2t + \frac{1}{h_{ii}} - \frac{1}{h_{ii}} e^{-h_{ii}t}) \quad .
\end{aligned} \tag{28}
$$

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

10.5 Proof of Proposition 2

SGD's covariance:

$$
\Sigma_t = \frac{1}{D} \left[\frac{1}{B} \sum_{i=1}^{Q} \nabla U(W_t)_i \nabla U(W_t)_i^T - \nabla U(W_t) \nabla U(W_t)^T \right] \quad . \tag{29}
$$

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

$$
U(W_t) = U(W^*) + \nabla U(W^*)(W - W^*) + \frac{1}{2}(W_t - W^*)^T \nabla^2 U(W^*)(W_t - W^*) \quad . \tag{30}
$$

Examining SGD's gradient on the b-th data point, using the approximation in 30:

$$
\nabla U(W_t)_i \approx \nabla U_d(W^*) + \nabla^2 U_d(W^*)(W_t - W^*) \quad . \tag{31}
$$

The exact gradient (of GD) is:

$$
\nabla U(W_t) \approx \nabla^2 U(W^*)(W_t - W^*) \quad . \tag{32}
$$

As a result of empirical evidence in Meng et al. (2020) 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. 29 can be written as:

$$
\Sigma(W_t) = \frac{1}{B} \left[\frac{1}{D} \sum_{d=1}^{D} \nabla U_d(W^*) \nabla U_d(W^*)^T + H_d(W^*) W_t W_t^T H_d(W^*) - H(W^*) W_t W_t^T H(W^*) \right] \tag{33}
$$

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

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

$$\begin{aligned}\Sigma_{i,j}(t) &= \frac{1}{B}\left[\sum_{k=1}^{N}\sum_{p=1}^{N}(\frac{1}{D}\sum_{d=1}^{D} h_{d,i,k}h_{d,p,j} - h_{i,k}h_{p,j})\tilde{w}_{k,p} + \frac{1}{D}\sum_{d=1}^{D}\nabla u_{d,i}\nabla u_{d,j}\right] \\ &= \frac{1}{B}\left[\sum_{k=1}^{N}\sum_{p=1}^{N}(\frac{1}{D}\sum_{d=1}^{D} h_{d,i,k}h_{d,p,j} - h_{i,k}h_{p,j})\tilde{w}_{k,p} + \tilde{u}_{d,i,j}\right]\end{aligned} \tag{34}$$

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

$$\begin{aligned}\Sigma_{i,j}(t) &= \frac{1}{B}\left[\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}W_{t,k}W_{t,p}\right] \\ &= \frac{1}{B}\left[\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}Z_{t,k}Z_{t,p}\right] \\ &= \frac{1}{B}\tilde{u}_{ij} + \sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}\,\mathbb{E}[Z_{t,k}Z_{t,p}]\end{aligned} \tag{35}$$

$$\begin{aligned}&\Sigma_{i,j}(t) = \frac{1}{BD}\tilde{u}_{ij}+ \\ &\frac{1}{B}[\sum_{k=1}^{N}\sum_{p=1}^{N}\bar{h}_{i,k,p,j}(w_k w_p e^{-(h_{kk}+h_{pp})t} + \bar{\epsilon}_p w_k e^{-h_{kk}t}\mu_\xi^p(2t + \frac{1}{h_{pp}}(1-e^{-h_{pp}t})) \\ &+ \bar{\epsilon}_k w_p e^{-h_{pp}t}\mu_\xi^k(2t + \frac{1}{h_{kk}}(1-e^{-h_{kk}t})))] + \mathcal{O}(\bar{\epsilon}^2)\end{aligned} \tag{36}$$

10.6 Proof of Lemma 1

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

$$\frac{dY_t}{dt} = -\nabla U(Y_t) \quad . \tag{37}$$

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

$$d\zeta(t) = \langle \nabla U(Y_t), dY_t\rangle = -\left\|\nabla U(Y_t)\right\|^2 \quad . \tag{38}$$

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

$$\begin{aligned}U(Y_t) - U(W^*) &\leq \frac{1}{2\mu}\left\|\nabla U(Y_t)\right\|^2 \\ -2\mu\zeta(t) &\geq d\zeta(t) \quad .\end{aligned} \tag{39}$$

Using Gronwall's lemma Gronwall (1919)::

$$U(Y_t) - U(W^*) \leq (U(w) - U(W^*))e^{-2\mu t} \quad . \tag{40}$$

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

$$\|Y_t - W^*\|^2 \leq \frac{2(U(w) - U(W^*))}{\mu}e^{-2\mu t} = \frac{2\zeta(t)}{\mu}e^{-2\mu t} \quad . \tag{41}$$

11 Additional Theorems

11.1 Mean Escape Time-Multistep scheduler

Next theorem deals with the mean transition time of a popular scheduler, the multi-step scheduler. Before stating the theorem, let us define few constants first, $\nu_{nn} \triangleq \frac{\gamma_p^{\alpha_\nu - 1}\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu$ this term express the global attributes of the DNN. Next constant $\tilde{C}_{l,\nu,p} = \frac{\frac{\bar{\beta}_l}{\bar{\beta}_S}\gamma_p^{\rho(\alpha_l-\alpha_\nu)}(\gamma_p^{\alpha_l-1}\bar{m}_l\Phi_l - \gamma_p^{\rho(\alpha_l-1)}\bar{\beta}_l)}{\left[\nu_{nn}+\frac{\bar{\beta}_l}{\bar{\beta}_S}\gamma_p^{\rho(\alpha_l-\alpha_\nu)}(1-\nu_{nn})\right]}$ utters global and single parameters attributes , last: $\bar{C}_{l,p} \triangleq \bar{\beta}_l\gamma_p^{\rho(\alpha_l-1)}$.

Theorem 4. *Let s_t be a multi-step scheduler . Further, let us notate $C_{l,\nu,p} \triangleq \bar{C}_{l,\nu,p} + \tilde{C}_{l,\nu,p}$, and $E_p \triangleq e^{-C_{l,\nu,p}T_p}T_p$. The mean transition time with a multi-step scheduler satisfies:*

$$\mathbb{E}[\sigma_{\mathcal{G}}] \approx \sum_{l=0}^{N}\sum_{p=0}^{P}\frac{\bar{\beta}_l}{\bar{\beta}_S C_{l,\nu,p}}\gamma_p^{\alpha_l(1-\rho)-1+\rho\alpha_\nu}\bar{m}_l\Phi_l A_{l,\nu}^{-1}(E_p - E_{p+1}) \quad .$$

Proof:

$$\mathbb{E}[\sigma_{\mathcal{G}}] \leq \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*\mathbb{I}\{\sigma_{\mathcal{G}} = \tau_k^*\}] \tag{42}$$

$$= \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*\mathbb{I}\{\sum_{l=0}^{N}s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_1^l\mathbb{I}\{\tau_1^l = \tau_1^*\} \in \mathcal{G},$$

$$\sum_{l=0}^{N}s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_2^l\mathbb{I}\{\tau_2^l = \tau_2^*\} \in \mathcal{G}, .., \sum_{l=0}^{N}s_t\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_k^l\mathbb{I}\{\tau_k^l = \tau_k^*\} \notin \mathcal{G}\}]$$

$$= \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*\mathbb{I}\{J_1^* \in \mathcal{G}, J_2^* \in \mathcal{G}, .., J_k^* \notin \mathcal{G}\}]$$

$$\leq \sum_{k=1}^{\infty}\mathbb{E}[\tau_k^*(1 - \mathbb{I}\{J_k^* \notin \mathcal{G}\})^{k-1}\mathbb{I}\{J_k^* \notin \mathcal{G}\}]$$

$$= \sum_{k=1}^{\infty}\sum_{l=1}^{N}\mathbb{E}[\tau_k^l(1 - \mathbb{I}\{J_k^l \notin \mathcal{G}\})^{k-1}\mathbb{I}\{J_k^l \notin \mathcal{G}\}\mathbb{I}\{\tau_k^l = \tau_k^*\}]$$

$$\leq \sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k}\mathbb{E}[\tau_w^l(1 - \mathbb{I}\{s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_w^l \notin \mathcal{G}\})^{w-1}(1 - \mathbb{I}\{s_t^{\frac{\alpha_\nu-1}{\alpha_\nu}}\epsilon(\mathbf{1}^T\Sigma_\nu(t))^{\frac{1}{\alpha_\nu}}J_w^m \notin \mathcal{G}\})^{k-w}$$

$$\mathbb{I}\{s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_w^l \notin \mathcal{G}\}\mathbb{I}\{\tau_w^l = \tau_k^*\}]$$

$$= \sum_{k=1}^{\infty}\sum_{l=0}^{N}\sum_{w=1}^{k}\int_0^{\infty}\frac{\beta_l(t)}{\beta_S(t)}\frac{(k-1)!}{(w-1)!(k-w)!}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}\left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}\beta_l(t)t$$

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

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

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

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

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

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

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

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

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

$$\sum_{l=0}^{N} \sum_{p=0}^{P} \int_{T_p}^{T_{p+1}} \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{(\alpha_l - 1 + \rho(\alpha_l - \alpha_\nu))} \bar{m}_l \Phi_l \left[\left(1 - \frac{\bar{m}_\nu}{\bar{\beta}_\nu} \Phi_\nu \right) \left(1 - \frac{\bar{\beta}_l}{\bar{\beta}_S} \right) \right]^{-1} t$$

$$e^{\left(- \frac{\frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} (\gamma_p^{\alpha_l - 1} \bar{m}_l \Phi_l - \gamma_p^{\rho(\alpha_l - 1)} \bar{\beta}_l)}{\left[\frac{\gamma_p^{\alpha_\nu - 1} \bar{m}_\nu}{\bar{\beta}_\nu} \Phi_\nu + \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} - \frac{\gamma_p^{\alpha_\nu - 1} \bar{m}_\nu \Phi_\nu}{\beta_\nu(t)} \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} \right]} - \bar{\beta}_l \gamma_p^{\rho(\alpha_l - 1)} \right) t} dt$$

Let us notate:

$$C_{l,\nu,p} \triangleq \frac{\frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} (\gamma_p^{\alpha_l - 1} \bar{m}_l \Phi_l - \gamma_p^{\rho(\alpha_l - 1)} \bar{\beta}_l)}{\left[\frac{\gamma_p^{\alpha_\nu - 1} \bar{m}_\nu}{\bar{\beta}_n u} \Phi_\nu + \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} + \frac{\gamma_p^{\alpha_\nu - 1} \bar{m}_\nu \Phi_\nu}{\beta_\nu(t)} \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{\rho(\alpha_l - \alpha_\nu)} \right]} + \bar{\beta}_l \gamma_p^{\rho(\alpha_l - 1)}, A_{l,\nu} \triangleq \left[\left(1 - \frac{\bar{m}_\nu}{\bar{\beta}_\nu} \Phi_\nu \right) \left(1 - \frac{\bar{\beta}_l}{\bar{\beta}_S} \right) \right]$$

$$\sum_{l=0}^{N} \sum_{p=0}^{P} \int_{T_p}^{T_{p+1}} \frac{\bar{\beta}_l}{\bar{\beta}_S} \gamma_p^{(\alpha_l - 1 + \rho(\alpha_l - \alpha_\nu))} \bar{m}_l \Phi_l A_{l,\nu}^{-1} t e^{-C_{l,\nu,p} t}$$

$$= \sum_{l=0}^{N} \sum_{p=0}^{P} \frac{\bar{\beta}_l}{\bar{\beta}_S C_{l,\nu,p}^2} \gamma_p^{\alpha_l - 1 + \rho(\alpha_l - \alpha_\nu)} \bar{m}_l \Phi_l A_{l,\nu}^{-1} (e^{-C_{l,\nu,p} T_p} (C_{l,\nu,p} T_p + 1) - e^{-C_{l,\nu,p} T_{p+1}} (C_{l,\nu,p} T_{p+1} + 1))$$

$$\approx \sum_{l=0}^{N} \sum_{p=0}^{P} \frac{\bar{\beta}_l}{\bar{\beta}_S C_{l,\nu,p}} \gamma_p^{\alpha_l - 1 + \rho(\alpha_l - \alpha_\nu)} \bar{m}_l \Phi_l A_{l,\nu}^{-1} (e^{-C_{l,\nu,p} T_p} T_p - e^{-C_{l,\nu,p} T_{p+1}} T_{p+1})$$

11.2 Trapping probability

Theorem 5. *Let s_t be an exponential scheduler $s_t = t^{\gamma-1}$, γ is the cooling rate. The probability of the process to be trapped in the domain $\mathcal{G}$ is upper bounded by:*

$$P(\sigma < \infty) \leq \sum_{l=0}^{N} \frac{\bar{m}_l \Phi_l}{\bar{\beta}_S} \left[\frac{\bar{\beta}_l}{\bar{\beta}_S} - \frac{\bar{\beta}_l}{\bar{\beta}_S} \frac{\bar{m}_\nu}{\bar{\beta}_\nu} \Phi_\nu \right]^{-1} \tag{43}$$

$$\frac{(2\beta_l - m_l\Phi_l)^{\frac{(\gamma-1)(\alpha_l - 1 + \rho(\alpha_\nu - 1)) + 1}{(\gamma-1)(\rho(\alpha_l-1))+1}}}{(\gamma-1)(\rho(\alpha_l-1)) + 1} \Gamma\left(\frac{(\gamma-1)(\alpha_l - 1 + \rho(\alpha_\nu - 1)) + 1}{(\gamma-1)(\rho(\alpha_l-1)) + 1}\right)$$

Proof:

$$P(\sigma < \infty) = \sum_{k=0}^{\infty} P(\sigma = \tau_k^*) \leq \sum_{k=0}^{\infty} P(J_1^* \in \mathcal{G}, J_2^* \in \mathcal{G}, .., J_k^* \notin \mathcal{G}) \tag{44}$$

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

$$= \sum_{k=1}^{\infty} \sum_{l=1}^{N} \sum_{w=1}^{k-1} (1 - P(s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon (\mathbf{1}^T \Sigma_l(t))^{\frac{1}{\alpha_l}} J_w^l \notin \mathcal{G}))^{w-1} (1 - P(s_t^{\frac{\alpha_\nu - 1}{\alpha_\nu}} \epsilon (\mathbf{1}^T \Sigma_\nu(t))^{\frac{1}{\alpha_\nu}} J_w^v \notin \mathcal{G}))^{k-w}$$

$$P(s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon (\mathbf{1}^T \Sigma_l(t))^{\frac{1}{\alpha_l}} J_w^l \notin \mathcal{G}) P(\tau_w^l = \tau_k^*)$$

$$= \sum_{k=1}^{\infty} \sum_{l=0}^{N} \sum_{w=1}^{k} \int_0^{\infty} \frac{\beta_l(t)}{\beta_S(t)} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}$$

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

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

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

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

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

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

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

$$= \sum_{l=0}^{N} \int_0^{\infty} \frac{m_l(t)\Phi_l}{\beta_S(t)} e^{-\beta_l(t)t} s_t^{\alpha_l - 1} \left[1 - \left(1 - \frac{s_t^{\alpha_\nu - 1} m_\nu(t)}{\beta_\nu(t)} \Phi_\nu\right)\left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1}$$

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

$$= \sum_{l=0}^{N} \int_0^{\infty} \frac{m_l(t)\Phi_l}{\beta_S(t)} e^{-\beta_l(t)t} s_t^{\alpha_l - 1} \left[1 - \left(1 - \frac{s_t^{\alpha_\nu - 1} m_\nu(t)}{\beta_\nu(t)} \Phi_\nu\right)\left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1}$$

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

$$\sum_{l=0}^{N}\int_0^\infty \frac{m_l(t)\Phi_l}{\beta_S(t)}e^{-\beta_l(t)t}s_t^{\alpha_l-1}\left[1-\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1} \tag{45}$$

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

$$\leq\sum_{l=0}^{N}\int_0^\infty \frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}s_t^{\alpha_l-1+\rho(\alpha_\nu-1)}$$

$$\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}-\frac{\bar{\beta}_l}{\bar{\beta}_S}\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right]^{-1}e^{-(2\beta_l-m_l\Phi_l)s_t^{\rho(\alpha_l-1)}t}dt$$

$$=\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}-\frac{\bar{\beta}_l}{\bar{\beta}_S}\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right]^{-1}$$

$$\int_0^\infty t^{(\gamma-1)(\alpha_l-1+\rho(\alpha_\nu-1))}e^{-(2\beta_l-m_l\Phi_l)t^{(\gamma-1)(\rho(\alpha_l-1))+1}}dt$$

$$=\sum_{l=0}^{N}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}-\frac{\bar{\beta}_l}{\bar{\beta}_S}\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right]^{-1}$$

$$\frac{(2\beta_l-m_l\Phi_l)^{\frac{(\gamma-1)(\alpha_l-1+\rho(\alpha_\nu-1))+1}{(\gamma-1)(\rho(\alpha_l-1))+1}}}{(\gamma-1)(\rho(\alpha_l-1))+1}\Gamma(\frac{(\gamma-1)(\alpha_l-1+\rho(\alpha_\nu-1))+1}{(\gamma-1)(\rho(\alpha_l-1))+1}) \quad .$$

11.3 Trapping probability - Multi step scheduler

Theorem 6. *Let s_t be a multistep scheduler, γ is the cooling rate. The probability of the training process to be trapped in the domain $\mathcal{G}$, namely $P(\sigma<\infty)$ is upper bounded by:*

$$\sum_{l=0}^{N}\sum_{p=0}^{P}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\gamma_p^{\rho(1-\alpha_\nu)+\alpha_l-1-\rho(\alpha_l-1)}\frac{e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}Tp}(1-e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)T}})}{(2\bar{\beta}_l-\bar{m}_l\Phi_l)} \quad . \tag{46}$$

Proof:

$$P(\sigma<\infty)\approx\sum_{k=0}^{\infty}P(\sigma=\tau_k^*)\leq\sum_{k=0}^{\infty}P(J_1^*\in\mathcal{G},J_2^*\in\mathcal{G},..,J_k^*\notin\mathcal{G}) \tag{47}$$

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

$$=\sum_{k=1}^{\infty}\sum_{l=1}^{N}\sum_{w=1}^{k-1}(1-P(s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_w^l\notin\mathcal{G}))^{w-1}(1-P(s_t^{\frac{\alpha_\nu-1}{\alpha_\nu}}\epsilon((\mathbf{1}^T\Sigma_\nu(t))^{\frac{1}{\alpha_\nu}})^{\frac{1}{\alpha_\nu}}J_w^v\notin\mathcal{G}))^{k-w}$$

$$P(s_t^{\frac{\alpha_l-1}{\alpha_l}}\epsilon(\mathbf{1}^T\Sigma_l(t))^{\frac{1}{\alpha_l}}J_w^l\notin\mathcal{G})P(\tau_w^l=\tau_k^*)$$

$$=\sum_{k=1}^{\infty}\sum_{l=0}^{N}\sum_{w=1}^{k}\int_0^\infty\frac{\beta_l(t)}{\beta_S(t)}\frac{(k-1)!}{(w-1)!(k-w)!}\left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1}\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}$$

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

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

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

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

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

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

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

$$=\sum_{l=0}^{N}\int_0^{\infty}\frac{m_l(t)\Phi_l}{\beta_S(t)}e^{-\beta_l(t)t}s_t^{\alpha_l-1}\left[1-\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1}$$

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

$$\sum_{l=0}^{N}\int_0^{\infty}\frac{m_l(t)\Phi_l}{\beta_S(t)}e^{-\beta_l(t)t}s_t^{\alpha_l-1}\left[1-\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1} \tag{48}$$

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

$$=\sum_{l=0}^{N}\int_0^{\infty}\frac{m_l(t)\Phi_l}{\beta_S(t)}e^{-\beta_l(t)t}s_t^{\alpha_l-1}\left[1-\left(1-\frac{s_t^{\alpha_\nu-1}m_\nu(t)}{\beta_\nu(t)}\Phi_\nu\right)\left(1-\frac{\beta_l(t)}{\beta_S(t)}\right)\right]^{-1}$$

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

$$\leq\sum_{l=0}^{N}\int_0^{\infty}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}s_t^{\alpha_l-1+\rho(\alpha_\nu-1)}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}e^{-(2\beta_l-m_l\Phi_l)s_t^{\rho(\alpha_l-1)}t}dt$$

$$=\sum_{l=0}^{N}\sum_{p=0}^{P}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\gamma_p^{\rho(1-\alpha_\nu)+\alpha_l-1}\int_{T_p}^{T_{p+1}}e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}t}dt$$

$$=\sum_{l=0}^{N}\sum_{p=0}^{P}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\gamma_p^{\rho(1-\alpha_\nu)+\alpha_l-1}\frac{e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}T_p}-e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}T_{p+1}}}{(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}}$$

$$=\sum_{l=0}^{N}\sum_{p=0}^{P}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\gamma_p^{\rho(1-\alpha_\nu)+\alpha_l-1}\frac{e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}Tp}-e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}T(p+1)}}{(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}}$$

$$=\sum_{l=0}^{N}\sum_{p=0}^{P}\left[\frac{\bar{\beta}_l}{\bar{\beta}_S}(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu)\right]^{-1}\frac{\bar{m}_l\Phi_l}{\bar{\beta}_S}\gamma_p^{\rho(1-\alpha_\nu)+\alpha_l-1-\rho(\alpha_l-1)}\frac{e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)}Tp}(1-e^{-(2\bar{\beta}_l-\bar{m}_l\Phi_l)\gamma_p^{\rho(\alpha_l-1)T}})}{(2\bar{\beta}_l-\bar{m}_l\Phi_l)}.$$

11.4 Probability of escaping after time u

We further investigate the probability of exiting before time u:

Theorem 7. *Let $s_t=t^{\gamma-1}$, where γ is the cooling rate,let us denote two constants that express the effect of the*

scheduler: $\gamma_l \triangleq 1+(\gamma-1)(\alpha_l-1)$ *and* $\kappa \triangleq \frac{1+(\gamma-1)(\alpha_l-1+\rho(\alpha_l-\alpha_\nu))}{\gamma_l}$*, for* $u>0$*:*

$$P(\sigma > u) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{\beta}_l \bar{m}_l \Phi_l}{\bar{\beta}_S \gamma_l (\bar{m}_l \Phi_l)^\kappa} \Gamma\left(\kappa, \bar{m}_l \Phi_l u^{\gamma_l}\right) \quad . \tag{49}$$

In order to further investigate this expression, let us temporally neglect the cooling effect.

Corollary 4. *Using Thm. 2, for* $\gamma \to 1$*:*

$$P(\sigma > u) \leq \sum_{l=0}^{N} A_{l,\nu}^{-1} \frac{\bar{\beta}_l}{\bar{\beta}_S} e^{-\bar{m}_l \Phi_l u} \quad . \tag{50}$$

As in $1d$ Imkeller and Pavlyukevich (2006a,b), it can be seen that for small ϵ, the probability depends exponentially on time u.

$$P(\sigma > u) = \sum_{k=0}^{\infty} P(\tau_k^* > u) P(\sigma = \tau_k^*) \leq \sum_{k=0}^{\infty} P(\tau_k^* > u) P(J_1^* \in \mathcal{G}, J_2^* \in \mathcal{G}, .., J_k^* \notin \mathcal{G}) \tag{51}$$

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

$$= \sum_{k=1}^{\infty} P(\tau_k^* > u)(1 - P(J_k^* \notin \mathcal{G}))^{k-1} P(J_k^* \notin \mathcal{G})$$

$$\approx \sum_{k=1}^{\infty} \sum_{l=1}^{N} \sum_{w=1}^{k-1} (1 - P(s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon (\mathbf{1}^T \Sigma_l(t))^{\frac{1}{\alpha_l}} J_w^l \notin \mathcal{G}))^{w-1} (1 - P(s_t^{\frac{\alpha_\nu - 1}{\alpha_\nu}} \epsilon (\mathbf{1}^T \Sigma_\nu(t)^{\frac{1}{\alpha_\nu}} J_w^v \notin \mathcal{G}))^{k-w}$$

$$P(s_t^{\frac{\alpha_l - 1}{\alpha_l}} \epsilon (\mathbf{1}^T \Sigma_l(t))^{\frac{1}{\alpha_l}} J_w^l \notin \mathcal{G}) P(\tau_w^l = \tau_k^*) P(\tau_w^l > u)$$

$$= \sum_{k=1}^{\infty} \sum_{l=0}^{N} \sum_{w=1}^{k} \int_u^{\infty} \frac{\beta_l(t)}{\beta_S(t)} \frac{(k-1)!}{(w-1)!(k-w)!} \left(\frac{\beta_l(t)}{\beta_S(t)}\right)^{w-1} \left(1 - \frac{\beta_l(t)}{\beta_S(t)}\right)^{k-w}$$

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

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

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

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

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

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

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

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

Exponential scheduler

$$\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\int_u^\infty \frac{\bar{\beta}_l\bar{m}_l\Phi_l}{\bar{\beta}_S}s_t^{\alpha_l-1+\rho(\alpha_l-\alpha_\nu)}e^{-s_t^{\alpha_l-1}\bar{m}_l\Phi_l t}dt \tag{52}$$

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

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

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

For $\gamma\to 1$:

$$\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\bar{\beta}_l}{\bar{\beta}_S}\Gamma\left(1,\bar{m}_l\Phi_l u\right) \tag{53}$$

$$=\sum_{l=0}^{N}\left[\left(1-\frac{\bar{m}_\nu}{\bar{\beta}_\nu}\Phi_\nu\right)\left(1-\frac{\bar{\beta}_l}{\bar{\beta}_S}\right)\right]^{-1}\frac{\bar{\beta}_l}{\bar{\beta}_S}e^{-\bar{m}_l\Phi_l u}\quad .$$

12 Extras

Lemma 3. *$\forall T\in[\Pi_j,\Pi_{j+1}]$, $\forall j\in\mathbb{N}$, and $\forall w\in[d_i^-,d_i^+]$ there exist a finite C_Z s.t:*

$$\sup_T|X_t^i(w)|\le C_z^I\sup_T|\xi_t^i|\quad . \tag{54}$$

Using stochastic asymptotic expansion:

$$|X_t^i(w)|\le\sup_{t\in[0,T]}|\xi_{t,l}|\left(1+\sup_{t\in[0,T]}\int_0^t H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp\right)\quad . \tag{55}$$

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

$$C_1=\max_{w\in\mathcal{G}}\int_0^{\hat{T}}H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp\quad . \tag{56}$$

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

$$\int_0^t H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp= \tag{57}$$

$$\int_0^{\hat{T}}H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp+\int_{\hat{T}}^t H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp$$

The estimate for the first term:

$$\int_0^{\hat{T}}H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp=e^{-\int_{\hat{T}}^t H(Y_u(w))_{ii}du}\int_0^{\hat{T}}H(Y_p(w))_{ii}e^{-\int_p^{\hat{T}}H(Y_u(w))_{ii}du}dp \tag{58}$$

$$\le e^{-m_1^i(t-\hat{T})}C_1\le C_1\quad .$$

The second sum:

$$\int_{\hat{T}}^t H(Y_p(w))_{ii}e^{-\int_p^t H(Y_u(w))_{ii}du}dp\le\int_{\hat{T}}^t m_2^i e^{-m_1^i(t-p)}dp\le\frac{m_2^i}{m_1^i}\quad . \tag{59}$$

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

13 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:*
$$\langle \nabla U(w), n(w)\rangle < -\frac{1}{C} \quad , \tag{60}$$
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:*
$$lim_{t\to\infty} Y_t(w) \to 0 \quad . \tag{61}$$

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:*
$$d^+_i(w) \triangleq inf\{t > 0 : g^i_{w,+}(t) \in \partial\mathcal{G}\} \quad . \tag{62}$$
- *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 Amann (2011) .*

14 Constructing the SDE

Let us first define our SGD iterative update rule:
$$w_k = w_{k-1} - \bar{\eta}_k \nabla U(w_k) + \bar{\eta}_k \zeta_k \quad . \tag{63}$$
$\zeta_k \in \mathbb{R}^N, w_k \in \mathbb{R}^N, \nabla U(w_k) \in \mathbb{R}^N$. Let us remind that $\Sigma^k \in R^{NxN}$ approximates the noise covariance matrix :
$$\Sigma_k = \frac{1}{D}\left[\frac{1}{B}\sum_{i=1}^{Q} \nabla U(w_k)_i \nabla U(w_k)^T_i - \nabla U(w_k)\nabla U(w_k)^T\right] \quad . \tag{64}$$
The SGN is assumed to be modeled by a Levy-stable random variable, $\zeta^l_k \sim S\alpha S(1^T\Sigma^k_l)$, note that $1^T\Sigma^k_l$ 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:
$$W_t = \int_0^t \nabla U(W_p)\, dp + \int_0^t \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_t) r_l dL^l_t \quad . \tag{65}$$
We aim to use the Euler-Maruyama method and Levy process properties to achieve Eq. 63. 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:
$$w_{\tau_k} = w_{\tau_{k-1}} - \nabla U(w_{\tau_k}, \tau_k)(\tau_k - \tau_{k-1}) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_{\tau_k}) r_l (L^l_{\tau_k} - L^l_{\tau_{k-1}}) \quad . \tag{66}$$
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.
$$w_k = w_{k-1} + \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_k) r_l S^l_k \quad . \tag{67}$$
Where $S^l_k \sim S\alpha S(\eta^{\frac{1}{\alpha_l}})$. Using $S\alpha S$ characteristic function and the fact L_t is a real value process: $S^l_k = \zeta^l_k \eta^{\frac{1}{\alpha_l}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{-\frac{1}{\alpha_l}}$, let us use this identity:
$$w_k = w_{k-1} + \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta^{\frac{\alpha_l - 1}{\alpha_l}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{\frac{1}{\alpha_l}} (W_k) r_l \zeta^l_k \eta^{\frac{1}{\alpha_l}} ((1^T\Sigma_l)^{\frac{1}{\alpha_l}})^{-\frac{1}{\alpha_l}} \quad . \tag{68}$$

$$w_k = w_{k-1} - \eta_k \nabla U(w_k) + \sum_{l=1}^{N} \eta_k r_l \zeta_k^l \quad . \tag{69}$$

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

$$w_k = w_{k-1} - \eta_k \nabla U(w_k) + \eta \zeta_k \quad . \tag{70}$$

For the convergence of the Euler-Maruyama discritization please see Jacod et al. (2005); Protter et al. (1997); Bally and Talay (1996).