Enhanced Adaptive Stochastic Gradient Descent: Convergence Analysis and Its Application in Single-Cell Perturbation Analysis

Francis Boabang and Samuel Asante Gyamerah

Abstract

Traditional optimization methods with fixed learning rates often struggle to model highly dynamic biological processes including the cell cycle, differentiation, therapeutic reprogramming, cancer progression, and embryonic development within this landscape. Fixed rates may either slow convergence during long stretches of near-zero gradients or induce instability when rare but informative gradients occur. To address these challenges, we introduce Enhanced Adaptive Stochastic Gradient Descent (ASGD), which dynamically alternates between cautious and aggressive learning rates. This dual-rate strategy allows the optimizer to effectively respond to the shifting gradient landscape, enhancing its ability to exploit informative nonzero updates that capture critical regulatory signals. Such adaptability is particularly valuable for single-cell perturbation analysis, enabling more accurate step-ahead predictions of cell fate and lineage trajectories by accounting for both immediate and downstream fluctuations in gene and protein expression. To evaluate the effectiveness of the proposed Enhanced ASGD, we compared its performance against state-of-the-art optimization methods commonly used in single-cell perturbation analysis, including Adam, Amsgrad, Padam and standard SGD with momentum. Benchmarking was performed on multiple single-cell gene expression datasets capturing diverse perturbation conditions, where models were trained to predict downstream cell fate and lineage trajectories. Our results demonstrate that Enhanced ASGD consistently achieves faster convergence, higher predictive accuracy, and better stability across highly sparse and noisy gradients typical of single-cell multiomic data. Notably, the dual learning rate mechanism allows the optimizer to exploit rare but informative perturbation signals more effectively than conventional methods, leading to improved modeling of multistep cellular responses and more robust capture of critical regulatory interactions.https://github.com/boabangf/GNN_RL_gene_trajectory_perturbation/blob/main/Multi-Step%20Differentiation%20Experiment/MultiModal_Nonconvex_Optimizer(RNA-ATAC-CITE%20modalities).ipynb

Index Terms

Francis Boabang is with the Concordia Institute for Information and Systems Engineering (CIISE), Concordia University, Montréal, QC, Canada.

Samuel Asante Gyamerah is with the Department of Mathematics, Toronto Metropolitan University, Toronto, Ontario, Canada. (Corresponding author: asante.gyamerah@torontomu.ca)

Machine Learning, Adaptive Stochastic Gradient Descent, Optimization,

I. Introduction

Different data modalities provide complementary information that, when integrated, offer a more comprehensive understanding of biological systems. In the biomedical domain, multi-omics data—encompassing genomics, transcriptomics, proteomics, epigenomics, metabolomics, and other molecular layers—can be combined for single-cell perturbation analyses [1]. This integration allows researchers to dissect the complex interactions and networks underlying biological processes and disease mechanisms. Cells dynamically respond to their environment by regulating gene expression, optimizing resource use, and maintaining functional homeostasis. Recent technological advances now enable precise measurement of RNA, proteins, lipids, and metabolites, producing highly complex datasets that capture the states of different biological layers. Multi-omics approaches integrate these disparate datasets to provide a clearer and more holistic view of cellular states.

Studies combining trancriptomics, proteomic and metabolomic data are becoming increasingly common as mass spectrometry technology becomes more accessible [2]. Yet, knowledge extraction through such integration remains challenging. Modern single-cell technologies such as single-cell RNA sequencing (scRNA-seq), single-cell ATAC sequencing (scATAC-seq), and spatial transcriptomics allow researchers to capture high-resolution snapshots of cellular states. These methods have been widely applied across biological contexts, including embryonic development, tissue regeneration, and cancer research, offering insights into cell differentiation trajectories, lineage relationships, and intercellular interactions [1].

In parallel, concepts from nonequilibrium physics, specifically landscape and flux, provide a powerful framework for understanding the dynamics of complex systems [3]. The classical Waddington landscape metaphor, introduced over half a century ago, conceptualizes cell differentiation as a ball rolling down a rugged potential landscape, with valleys representing stable cell states and hills representing unstable intermediates as shown in Figure 1. While foundational, this metaphor alone is insufficient, as cell behavior is not dictated solely by a static landscape. Nonequilibrium fluxes play a crucial role in driving and sustaining dynamic processes for instance, the progression of the cell cycle is maintained by a nonequilibrium curl flux, ensuring coherent and robust division. By integrating landscape and flux, researchers gain a more complete framework to characterize both the stability of cellular states and the transitions between them [3].

Machine learning, particularly deep generative models (e.g., variational autoencoders or diffusion models), can infer the underlying energy or potential landscape directly from high-dimensional single-cell data (such as scRNA-seq or ATAC-seq) [1]. These models can map high-dimensional gene expression trajectories into a lower-dimensional manifold, approximating valleys (stable cell states) and ridges (transition barriers).

By training on temporal or pseudotime-ordered data, ML can reconstruct dynamic trajectories, effectively estimating the shape of the Waddington landscape from data instead of relying on predefined models [1].

The convergence of machine learning models for cell differentiation and reprogramming is important. If the model does not converge to a good solution, the model will have poor accuracy leading to wrong cell fate or lineage. The solution involves adaptive optimization strategies which is synonymous to integrating landscape and flux to create a more comprehensive framework that characterizes both the stability of cellular states and the transitions between them. Moreover, adaptive optimization strategies can be incorporated to efficiently navigate these high-dimensional waddington landscapes. This adaptive approach ensures both stability and efficiency, enabling robust modeling of complex, dynamic processes in single-cell multi-omic data.

Adaptive stochastic gradient descent enjoys fast convergence and works effectively in optimizing many networks for various recognition applications [4]. The loss landscape of adaptive stochastic gradient descent (SGD) comprises numerous flat and sharp regions. Without adjusting the effective learning rate, the SGD algorithm becomes inefficient. In flat regions of the loss landscape, where the gradient magnitude is very small, a large effective learning rate is necessary to accelerate convergence. Conversely, in sharp regions of the loss landscape, a smaller effective learning rate is essential to prevent the model from diverging. Additionally, selecting an appropriate base learning rate based on the size of the second-order momentum vector is crucial for effective learning [5]. With the same base learning rate, a large second-order momentum vector results in a small effective learning rate, while a small second-order momentum vector leads to a larger effective learning rate. Therefore, it is essential to choose the base learning rate carefully to prevent coordinate values from overshooting during training. Specifically, a small base learning rate is needed for a small second-order momentum vector, whereas a large second-order momentum matrix requires an even smaller base learning rate to maintain stability. For this reason, in this paper, we focus on building a new adaptive stochastic gradient descent optimizer to address this issue to boost the convergence rate of machine learning algorithms. We propose to exploit a non-uniform p-norm-based concept [6] to build an ASGD to train ML model for various applications with a high convergence rate. To be more specific, we fix the small learning rate dilemma problem [5] associated with ASGD by setting a threshold to divide the system into large and small categories, and each category is given different system requirements [6]. The proposed ASGD [7] can achieve a fast convergence rate and good generalization performance by setting a base learning rate according to the system requirement(i.e,. single cell perturbation analysis). Our approach is fundamentally different from adagrad's [8] update rule since adagrad [8] updates frequently occurring features with low learning rates and infrequently occurring features with high learning rates.

The contributions offered by this paper are listed below:

Table I Summary of works related to adaptive stochastic gradient descent (ASGD), where D_{iff1} and D_{iff2} denote selectable base learning rates from $\{10, 1, 0.1, 0.01, 0.001, 0.0001\}$, and Ψ characterizes the gradient growth rate of the cumulative stochastic gradient. In the ASGD2 algorithm [10], the parameter δ was set to 10^{-8}; for practical purposes, we assume $\delta = 0$, ensuring nonconvex convergence.

Optimizer	ASGD2 [10]	ADAM [13]	PADAM [5]	AMSGRAD [14]	Proposed (Improved Adam)	Proposed (Improved AMSGrad)
Large H_t	D_{iff1}	D_{iff1}	D_{iff1}	D_{iff1}	D_{iff1}	D_{iff1}
Small H_t	D_{iff1}	D_{iff1}	D_{iff1}	D_{iff1}	D_{iff2}	D_{iff2}
$\hat{H}_t = \max(\hat{H}_{t-1}, H_t)$	No	No	Yes	Yes	No	Yes
Rate of Convergence in Nonconvex Settings	$O\left(\frac{\ln T + d^2}{T^{1/2}}\right)$	$O\left(\frac{d^{1/2}}{T^{1/2}} + \frac{d}{T}\right)$	$O\left(\frac{d^{1/2}}{T^{3/4-s/2}} + \frac{d}{T}\right)$	$O\left(\frac{d^{1/2}}{T^{1/2}} + \frac{d}{T}\right)$	$O\left(\frac{d^{1/2}}{T^{1/2}} + \frac{d}{T} + \Psi\right)$	$O\left(\frac{d^{1/2}}{T^{1/2}} + \frac{d}{T} + \Psi\right)$

1) We reformulate the partially adaptive momentum estimation method (PADAM) [5], [9], [10] to develop a new and better ASGD method capable of overcoming the small learning rate dilemma problem by assigning different base learning rates to different categories of coordinate values. We rigorously analyze the theoretical properties of the proposed ASGD in a nonconvex setting following the authors in [5], and replacing the base learning rate with a linear function to achieve an improved convergence rate $\mathrm{O}\left(\frac{d^{\frac{1}{2}}}{T^{\frac{1}{2}}} + \frac{d}{T} + \Psi\right)$, which represents an improvement over convergence results in [11], [12], [5], [10].
2) We justify the theoretical properties of the proposed ASGD method through thorough evaluations conducted on CIFAR-100 and 10 datasets utilizing WideResNet and VGG16-Net architectures.

The remainder of the paper is organized as follows. In Section II, we present the related work. In Section III, we describe our proposed optimizer. Section **??** present experiment. Finally, Section V concludes the paper.

II. Related Work

SGD has been widely used for many applications. Still, its convergence is slow, in consequence limiting its applications. Contrary to SGD, which uses the same base learning rate for all the coordinates, adaptive SGD methods derive different effective learning rates for different coordinate values from the approximation of first and second-order momentum of the gradient [13]. Momentum's ability in accelerating convergence is primarily observed in the realm of strongly convex functions, thus it may not yield accelerated convergence rates for nonconvex objectives. Moreover, addressing the nonconvergence challenges inherent in nonconvex scenarios with adaptive stochastic gradient descent can be achieved by employing exponential moving averages of historical gradient squares. Nevertheless, the moving average approach is hindered by the short memory problem, potentially leading to failures in specific circumstances such as nonconvex setting. Addressing the short memory constraint of adaptive SGD can be achieved by integrating long-term memory mechanisms [14]. While momentum-based algorithms have attempted to address the convergence issue, the problem of a small base learning rate leading to low generalization error in the later stages of training remains unresolved.

The base learning rate selection affects the convergence rate of adaptive learning rate of ASGD. For instance, some coordinate values are small, so to avoid those coordinate values from overshooting and reducing the model's generalization performance, ADAM [13] and AMSgrad [14] selected small base learning rates [5]. Choosing a small base learning rate makes the algorithm make less impact at the later stage of training [5]. Chen et al. [5] sought to address this issue by opting for a small partially adaptive parameter alongside a large base learning rate. But, the coordinate values still overshoot since the partial adaptive value was not adapted during training, leading to poor generalization performance. Moreover, when examining different adaptive SGD methods, it is important to recognize that partially adaptive parameters below 0.5 often lead to performance outcomes devoid of any predictable pattern [5], [9]. On top of that, Sun et al. [10] introduced a method to address the challenge of small learning rates by advocating for the selection of a partially adaptive parameter greater than 0.5. Nonetheless, opting for a partially adaptive parameter exceeding one frequently results in uncertain outcome. AdaBelief algorithm in [4] scales the learning rate in adaptive function by utilizing the difference between the predicted and observed gradient. Recently, Zhou et al. [15] modified the second-order momentum of AdaBelief [4] to boost its convergence under strong convexity condition. Nevertheless, the improved AdaBelief does not exhibit the optimal rate of convergence especially for weakly convex and nonconvex objective functions. We need an approach that is independent of a particular objective function. Furthermore, an approach capable of handling both weakly convex and nonconvex objective functions can substantially enhance the convergence rate of any model. Recently, Huang et al. [16] presented an angle-calibrated moment technique that leverages the benefits of a second-order moment while updating first order moment. They compared its convergence rate with that of Adam and Padam, and the result showed a close convergence rate to Adam and Padam. Although this approach reduces the number of update parameters, it may entail a loss in model efficiency. Chen et al. [17] implemented a mechanism which aims to enhance the empirical performance of models by gradually diminishing the cumulative impact of a gradient on all subsequent updates. Verma et al. [18] suggested employing a trigonometric function on the exponential moving average of weight parameters to calculate the step size. This approach only targets the vanishing gradient issue in nonconvex scenarios, particularly prominent when employing sigmoid activation functions. Zhong et al.[19] endeavored to address the non-convergence problem present in Adam by introducing a novel approach: a linearly growing weighted strategy that assigns varying weights to past gradients. Their method demonstrated heightened efficacy, notably when the gradient experienced rapid decreases. The limitation of Wada[19] lies in its tendency to diverge on non-convex and slowly decaying gradient problems. The authors in [20] introduced a novel adaptive gradient framework called SUPERADAM, designed to be faster and more versatile. This framework was based on a universal adaptive matrix encompassing various existing adaptive gradient forms, allowing it to integrate with momentum and variance reduction techniques seamlessly. The downside of SUPERADAM [20] lies in its variance reduction technique, necessitating a larger batch size for optimal performance.

In summary, all the above mentioned works have improve the convergence rate of ASGD to a certain extend. However, they continue to face constraints stemming from poor accuracy, resulting in high generalization error during advanced training stages. We contend that this issue primarily arises from the fact that the base learning rate in these algorithms can either become excessively small or excessively large in the later stages of training depending on the network architecture. Most of the methodologies outlined in this subsection are slight variations of Adam. Consequently, we adopt Adam, ASGD2 [10], Padam [5], Amsgrad [14], Wada [19] and SUPERADAM [20] as the benchmark methods. We have provided a summary of works related to ASGD in Table I.

III. Proposed Optimizer

A. Motivation

Optimizing single-cell perturbation models presents unique challenges due to the complex, high-dimensional, and nonconvex nature of the cellular state space. In particular, the choice of base learning rate critically affects the convergence dynamics of stochastic optimizers such as ASGD. A large base learning rate often causes the optimization trajectory to oscillate around narrow valleys in the loss landscape, leading to unstable updates and impaired convergence when modeling dynamic cellular responses. Conversely, a small base learning rate results in slow progress and insufficient parameter updates during later training stages, especially when learning fine-grained perturbation effects in single-cell systems. Therefore, achieving an optimal balance between adaptive learning rate scaling and generalization capability is essential for improving the robustness and convergence of ASGD in this biological context. To address this, we propose an enhanced adaptive optimizer that dynamically adjusts the base learning rate according to the magnitude of the second-order momentum vector [7] as depicted in 2. This adaptive mechanism allows the optimizer

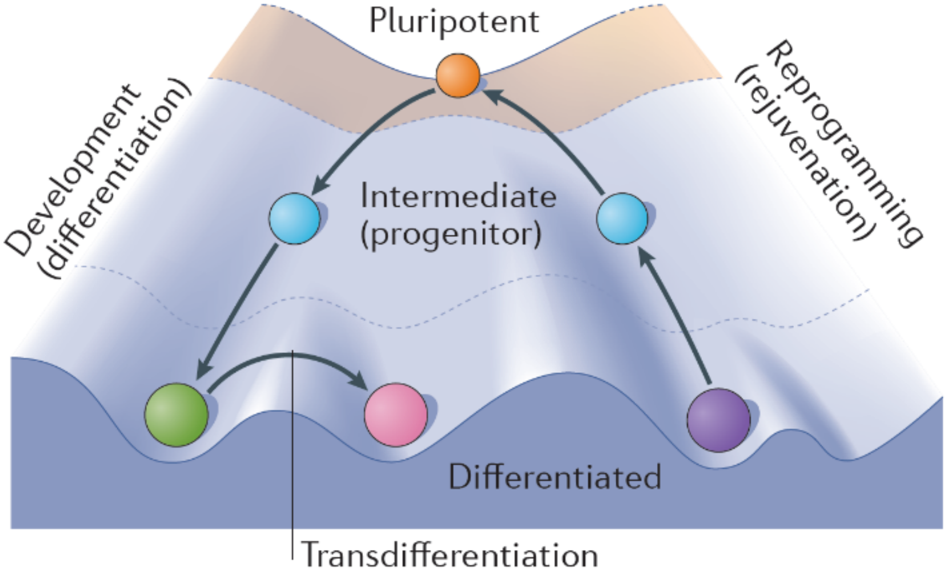
Pluripotent
Development (differentiation)
Reprogramming (rejuvenation)
Intermediate (progenitor)
Differentiated
Transdifferentiation

Figure 1 Waddington's epigenetic landscape provides a conceptual metaphor for embryonic development, where a cell is represented as a ball rolling down a terrain of hills and valleys. The valleys correspond to stable, differentiated cell fates, while the hills represent unstable, intermediate states. Along its trajectory, the ball encounters branching points that signify critical fate decisions, illustrating how a cell's potential gradually narrows. This framework captures both the robustness of specific developmental pathways and the flexibility of cells to respond to regulatory cues, offering insight into how gene and protein networks orchestrate differentiation [21].

to accelerate convergence in early training while maintaining stability in later phases, thereby improving the overall generalization performance in predicting post-perturbation cellular states.

B. Problem Definition

a) Problem Formulation.: We address the challenge of optimizing single-cell perturbation models under the small learning rate problem by formulating cell differentiation as a dynamic prediction task. Let $X \in \mathbb{R}^{n \times d}$ denote the matrix of single-cell profiles, where n is the number of cells and d is the dimensionality of the measured features (such as gene expression, chromatin accessibility, or protein markers). The dataset $\mathcal{D} = \{(x_i, p_i, y_i)\}_{i=1}^{N}$ and a task description S are given, where $x_i \in \mathbb{R}^d$ represents the pre-perturbation profile of a cell, $p_i \in \mathcal{P}$ denotes the applied perturbation (e.g., gene knockout, drug treatment, or cytokine stimulus), and $y_i \in \mathbb{R}^{d'}$ corresponds to the observed post-perturbation profile. The dataset is divided into $\mathcal{D}_{\text{train}} = \{(x_i, p_i, y_i)\}_{i=1}^{M}$ and $\mathcal{D}_{\text{test}} = \{(x_i, p_i, y_i)\}_{i=1}^{K}$, with $p_i \in \mathcal{P}_{\text{test}}$ and $x_i \in X_{\text{test}}$ representing held-out perturbations and unseen cell profiles, respectively.

The goal is to learn a mapping function

$$f_\theta : \mathbb{R}^d \times \mathcal{P} \rightarrow \mathbb{R}^{d'}$$

parameterized by θ, which predicts the post-perturbation state y_i given an initial cell state x_i and perturbation p_i, while remaining robust to small learning rates and local minima. To enhance generalization and capture intrinsic cell-state geometry, we employ a learnable encoder

$$g_\phi : \mathbb{R}^d \longrightarrow \mathbb{R}^h$$

with parameters ϕ, yielding latent embeddings $z_i = g_\phi(x_i) \in \mathbb{R}^h$ that preserve the manifold structure of both control and perturbed cells, thereby facilitating efficient convergence and accurate prediction of differentiation trajectories.

C. *The Proposed Adaptive Method for Enhancing the Convergence Rate of ASGD*

D. *Main Contribution*

The primary contribution of this chapter is to improve the accuracy and convergence of Adaptive Stochastic Gradient Descent (ASGD) in the context of single-cell perturbation analysis. Modeling cellular perturbations requires optimizing high-dimensional, nonconvex functions that describe the mapping from pre-perturbation to post-perturbation cell states. However, conventional ASGD methods often suffer from either oscillatory behavior due to large learning rates or excessively slow convergence under small learning rates. To address this limitation, we take inspiration from p-norm adaptive filtering algorithms [22], [6], [23], [24] and propose an enhanced step-size adaptation scheme that dynamically adjusts the base learning rate based on the optimizer's internal statistics.

Formally, let the gradient at iteration t be denoted as $z_t = \nabla g(W_t)$, where W_t represents the learnable parameters of the perturbation prediction model. Given the adaptive parameter β, the first-order momentum term is defined as

$$n_t = \beta_1 n_{t-1} + (1 - \beta_1) z_t,$$

and the preconditioner (second-order moment estimate) as

$$H_t = \beta_2 H_{t-1} + (1 - \beta_2) z_t^2,$$

with a small constant $0 < \epsilon \ll 1$ ensuring numerical stability. The adaptive gradient update in [13] is then given by

$$W_{t+1} = W_t - \eta \frac{n_t}{\sqrt{H_t} + \epsilon},$$

where η denotes the base learning rate. Building upon this formulation, our proposed adaptive ASGD modifies the step-size selection by scaling η according to the magnitude of the second-order momentum vector. This adaptive mechanism enables more responsive updates to rapidly changing gradient dynamics observed in single-cell perturbation learning, thereby promoting faster convergence and improved generalization across unseen perturbations and cellular states.

$$W_{t+1} = \left(W_t - \frac{\alpha}{\sqrt{(H_t + \epsilon)}} n_t \right). \tag{1}$$

To adaptively modulate the step direction in our proposed ASGDAdam optimizer, we compute summary statistics of the second moment estimates. We define two scalar statistics derived from h_t: the mean absolute second moment,

$$\text{mean\_abs}_H = \frac{1}{N} \sum_{i=1}^{N} \left| h_{t,i} \right|,$$

and the mean second moment,

$$\text{mean}_H = \frac{1}{N} \sum_{i=1}^{N} h_{t,i},$$

where N is the total number of parameters.

The quantity mean_abs$_H$ captures the average magnitude of curvature estimates across all parameters, providing a measure of overall gradient variability or roughness in the optimization landscape. In contrast, mean$_H$ reflects the net average curvature, effectively summarizing the directional bias or smoothness of the landscape. By comparing these two statistics, we derive binary switching factors:

$$\begin{aligned} f_{\text{min}} &= \frac{\text{sign}(\text{mean\_abs}_H - \text{mean}_H) + 1}{2}, \\ f_{\text{max}} &= \frac{\text{sign}(\text{mean}_H - \text{mean\_abs}_H) + 1}{2}. \end{aligned} \tag{2}$$

which indicate whether the optimization dynamics are dominated by high variability ($f_{\text{min}} = 1$) or by smoother, more stable curvature ($f_{\text{max}} = 1$). These factors can subsequently guide adaptive adjustments to the parameter update direction, enhancing stability and convergence in complex, high-dimensional loss landscapes.

The base learning rate is based on whether $h_{t,l}$ coordinate values are small or large. Given an auxiliary variable u and constant value C, the base learning rate is defined as a linear function for small and large coordinate values

$$\alpha_{base} = u f_{min}(H) + C \tag{3}$$

and

$$\alpha_{base} = u f_{max}(H) + C. \tag{4}$$

For learning the small and large base learning rate α_{min} and α_{max}, we defined a piece-wise function

$$\alpha_{base} = \begin{cases} \alpha_{small} & \text{if} \quad \text{H}_t \quad \text{is} \quad \text{small} \\ \alpha_{large} & \text{if} \quad \text{H}_t \quad \text{is} \quad \text{large} \end{cases} \tag{5}$$

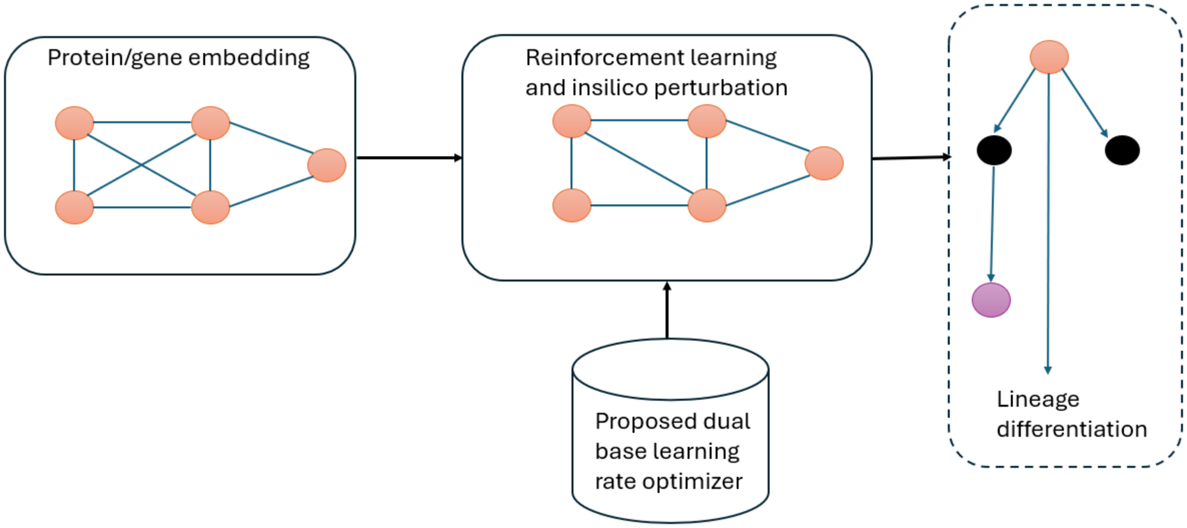
Protein/gene embedding
Reinforcement learning and insilico perturbation
Proposed dual base learning rate optimizer
Lineage differentiation

Figure 2 Single Cell Perturbation Analysis using the Proposed Optimizer [7]

If $h_{t,l}$ values are small, the effective learning rate $\frac{\alpha_{base}}{\sqrt{h_{t,l}}}$ will be large, and if $h_{t,l}$ values are large, the effective learning rate $\frac{\alpha_{base}}{\sqrt{h_{t,l}}}$ will be small [15], [5]. Therefore, we assigned a large base learning rate α_{max} when majority of the coordinate values are larger than the mean value and a small base learning rate α_{min} when majority of the coordinate values are smaller than the mean value. In summary, we can avoid small learning rate dilemmas [15], [5] in both learning situations by selecting a good base learning rate that improves the model's empirical results of the surgical gesture recognition task.

The proposed ASGD update rules for small and large coordinate are

$$W_{t+1,min} = \left(W_t - \frac{\alpha_{min}}{\sqrt{(H_t + \epsilon)}} n_t \right) \tag{6}$$

and

$$W_{t+1,max} = \left(W_t - \frac{\alpha_{max}}{\sqrt{(H_t + \epsilon)}} n_t \right). \tag{7}$$

We can incorporate the improvised equations 3 and 4 [6] into different ASGD methods to improve their performance compared to the standard ASGD methods for certain finite number of epochs (fast convergence).

E. Convergence Analysis of the Proposed ASGD in Convex Setting

The convergence of the proposed algorithm in convex setting is guaranteed by Theorem 1 below. The details of Theorem 1 follow AMSgrad [14], with the base learning rate α_{base} modified to $(uf(H) + C)$. We define the following assumption from [13] with the epsilon value ϵ dropped ,

Assumption 1. *[13] 1: let function $g : \Re^d \rightarrow \Re$ be convex, then $x, y \in \Re^d, g(x) \in \nabla g(x)^T (y - x)$, then*

$$g(y) \geq g(x) + \nabla g(x)^T (y - x). \tag{8}$$

Theorem 1. *: Given $\{W_t\}_1^T$ and $\{H_t\}_1^T$ are sequence generated by algorithm 1 and 2. Suppose $\alpha_t = \frac{u.f(H)+C}{\sqrt(t)}$, $\gamma \triangleq \frac{\beta_1^2}{\sqrt{\beta_2}} \beta_{1,t} = \beta_1 \lambda^{t-1}$, and $\lambda \in (0, 1)$, then if the set χ has a bounded diameter D_∞, i.e., $||W_t - W_{t'}||_\infty \leq D_\infty$ for all $W \in \chi$ and z_t has the bounded gradient, i.e., $||\nabla g_t(W)||_\infty \in Z_\infty$, we have the upper bound of the regret of the proposed algorithm as*

$$\begin{aligned} R_T \leq \frac{D_\infty^2}{2(uf(H) + C)(1 - \beta_1)} \sum_{l=1}^{d} \sqrt{T} h_{T,l}^{\frac{1}{2}} \\ + \frac{uf(H) + C(\beta_1 + 1) Z_\infty^{(1)} \times \sqrt{1 + \log T}}{(1 - \beta_1)(1 - \beta_2)^{\frac{1}{2}} (1 - \gamma)^2} \sum_{t=1}^{T} ||z_{1:T,l}||_2 + \\ \sum_{l=1}^{d} \frac{D_\infty^2 Z_\infty^1 \beta_1}{(uf(H) + C)(1 - \beta_1)(1 - \lambda)^2}. \end{aligned} \tag{9}$$

Then, we summarize the following lemma in the convex analysis needed to prove Theorem. ***The subsequent steps closely resemble the proof outlined in Theorem 1 in AMSgrad [14].***

Lemma 1. *: This lemma follows the assumption in Theorem 1 [13], [14] and we have*

$$\begin{aligned} \sum_{t=1}^{T} \sum_{l=1}^{d} \frac{(u.f(H) + C).n_{t,l}^2}{(h_{t,l})^{\frac{1}{2}}} \leq \\ \frac{(uf(H) + C) Z_\infty^{(1)} \sqrt{1 + logT}}{(1 - \beta_1)(1 - \gamma)(1 - \beta_2)^{\frac{1}{2}}} . \sum_{l=1}^{d} ||z_{1:T,l}||_2. \end{aligned} \tag{10}$$

We would like to clarify that this lemma is a minor modification of the version of AMSgrad [14].

Lemma 2. *[13]: Suppose $H \in S_d^+$ is a symmetric positive definite matrix, $p \in (0, \frac{1}{2}]$, $a_1 = \prod_{\{W\} H^p}(b_1)$ and $a_2 = \prod_{\{W\} H^p}(b_2)$, then we have*

$$||H^{\frac{1}{2}}(a_1 - a_2)||_2 \leq ||H^{\frac{1}{2}}(b_1 - b_2)||_2. \tag{11}$$

Fundamentally, our method shown in Algorithms 1 and 2 is different from Adagrad's method [8]. Adagrad [8] updates the frequently occurring features with low learning rates and less frequently occurring features with high learning rates. In contrast, we define a small base learning rate when many of the coordinate values are smaller than the mean and a large base learning rate when a large portion of the coordinate values are larger than the mean.

F. *Ergodic Convergence Analysis of the Proposed ASGD in Nonconvex Setting*

In this subsection, we establish the convergence proof in a nonconvex setting by following the methodology outlined in Zhou et al. [12] and substituting the base learning rate with a linear function $uf(H)+C$ [7].

Theorem 2. *Under the following assumptions:*

Assumption 2. *[12] For a differentiable function g, there exists a constant L such that $||\nabla g(x)-\nabla g(y)|| \leq L|||x-y||$ for all x, y, and g is lower bounded.*

and

Assumption 3. *[12]: The function $g(W) = \mathbb{E}_\xi\, g(W;\xi)$ has a Z_∞-bounded stochastic gradient, meaning that for any ξ, $g(W;\xi) \leq Z_\infty$, $\beta_1 < \beta_2^{\frac{1}{2}}$, $\alpha_t = (u.f(H)+C)_t$, and a sequence $\{Q_i\}_{i=1}^{3}$ where $||z_{1:T,i}||_2 \leq Z_\infty$ for $t = 1, \ldots.., T$,*

the iteration W_t of the proposed ASGD satisfies the

$$\frac{1}{T-1}\sum_{t=2}^{T}\mathbb{E}[||\bigtriangledown g(W_t)||_2^2] \leq \frac{R_8}{T.(u.f(H)+C)} + \frac{R_9 d}{T} + \\ \frac{(u.f(H)+C)R_{10}d}{T^{\frac{1}{2}}} \tag{12}$$

where

$$R_8 = 2Z_\infty\Delta g, \tag{13}$$

$$R_9 = \frac{2Z_\infty^3\epsilon^{-\frac{1}{2}}}{1-\beta_1} + 2Z_\infty^2, \tag{14}$$

$$R_{10} = \frac{2LZ_\infty^2}{\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\frac{\beta_1}{\beta_2^{\frac{1}{2}}})}\left(1+\frac{2\beta_1^2}{1-\beta_1}\right), \tag{15}$$

and

$$\Delta g = g(W_1) - \inf_W g(W). \tag{16}$$

In this context, we establish the convergence theory of Algorithms 1 and 2 within the framework of stochastic nonconvex optimization. Chen et al. [11] examined the convergence behavior of the AdaFOM algorithm with a fixed second-order momentum in non-convex optimization scenarios. Their findings suggest an ergodic convergence rate of approximately $O\left(\frac{\log T+d^2}{\sqrt{T}}\right)$, where T represents the number of iterations and d denoting the dimensionality of the problem.

In a related study, Zhou et al. [12] expanded upon this research by exploring the convergence properties of adaptive gradient methods in non-convex optimization. They specifically investigated the ergodic convergence utilizing AMSGrad with a varying maximum second-order momentum parameter, achieving a convergence rate of $O(\frac{d^{\frac{1}{2}}}{T^{\frac{1}{2}}}+\frac{d}{T})$

Given the similarity between AMSGrad and PADAM, which is a slight variation thereof, and the emphasis on maintaining a partially adaptive parameter below 0.5, one could adopt Zhou et al.'s approach. By replacing the base learning rate with the proposed linear function $(u.f(H))+C$, and keeping the partially adaptive parameter fixed at 0.5, an improved convergence rate could be attained. To be more specific, by incorporating a constant value C of our linear function base learning, $\alpha_{base} = u.f(H)+C$ from the convergence rate of Amsgrad in nonconvex $O(\frac{d^{\frac{1}{2}}}{T^{\frac{1}{2}}}+\frac{d}{T})$, we can achieve an improved convergence rate of $O(\frac{d^{\frac{1}{2}}}{T^{\frac{1}{2}}}+\frac{d}{T}+\Psi)$ for the proposed ASGD in non-convex setting.

Algorithm 1 ASGDAdam: Adaptive Switching Gradient Descent with Adam [7]

```
Require: Learning rates $\mathrm{lr}_{\mathrm{min}}, \mathrm{lr}_{\mathrm{max}}$; decay rates $\beta_1, \beta_2$; small constant $\epsilon$
Initialize first and second moments $n_0 \leftarrow 0$, $H_0 \leftarrow 0$
for each iteration $t = 1, 2, \ldots, T$ do
    $total\_nonzero\_f_{min} \leftarrow 0$, $total\_nonzero\_f_{max} \leftarrow 0$
    for each parameter $p$ do
        Compute gradient $g_t \leftarrow \nabla_p L(p)$
        $n_t \leftarrow \beta_1 n_{t-1} + (1 - \beta_1) g_t$                ▷ First moment update
        $H_t \leftarrow \beta_2 H_{t-1} + (1 - \beta_2) g_t^2$              ▷ Second moment update
        $mean\_abs\_H \leftarrow \mathrm{mean}(|H_t|)$; $mean\_H \leftarrow \mathrm{mean}(H_t)$
        $f_{min} \leftarrow \frac{1}{2}[(\mathrm{sign}(mean\_abs\_H - mean\_H)) + 1]$
        $f_{max} \leftarrow \frac{1}{2}[(\mathrm{sign}(mean\_H - mean\_abs\_H)) + 1]$
        $total\_nonzero\_f_{min}$ += count_nonzero($f_{min}$)
        $total\_nonzero\_f_{max}$ += count_nonzero($f_{max}$)
        denom $\leftarrow \sqrt{H_t} + \epsilon$
        step_dir $\leftarrow n_t/$denom
    end for
    if $total\_nonzero\_f_{max} < total\_nonzero\_f_{min}$ then
        $\mathrm{lr} \leftarrow \mathrm{lr}_{\mathrm{min}}$
    else
        $\mathrm{lr} \leftarrow \mathrm{lr}_{\mathrm{max}}$
    end if
    for each parameter $p$ do
        $p \leftarrow p - \mathrm{lr} \cdot$ step_dir
    end for
end for
```

Algorithm 2 ASGDAmsgrad: Adaptive Switching Gradient Descent with AMSGrad [7]

```
Require: Learning rates $\mathrm{lr}_{\min}, \mathrm{lr}_{\max}$; decay rates $\beta_1, \beta_2$; small constant $\epsilon$
Initialize $n_0 \leftarrow 0$, $H_0 \leftarrow 0$, $\hat{H}_0 \leftarrow 0$
for each iteration $t = 1, 2, \ldots, T$ do
    $total\_nonzero\_f_{min} \leftarrow 0$, $total\_nonzero\_f_{max} \leftarrow 0$
    for each parameter $p$ do
        Compute gradient $g_t \leftarrow \nabla_p L(p)$
        $n_t \leftarrow \beta_1 n_{t-1} + (1 - \beta_1) g_t$                ▷ First moment update
        $H_t \leftarrow \beta_2 H_{t-1} + (1 - \beta_2) g_t^2$              ▷ Second moment update
        $\hat{H}_t \leftarrow \max(\hat{H}_{t-1}, H_t)$                      ▷ AMSGrad correction
        $mean\_abs\_H \leftarrow \mathrm{mean}(|H_t|)$; $mean\_H \leftarrow \mathrm{mean}(H_t)$
        $f_{min} \leftarrow \frac{1}{2}[(\mathrm{sign}(mean\_abs\_H - mean\_H)) + 1]$
        $f_{max} \leftarrow \frac{1}{2}[(\mathrm{sign}(mean\_H - mean\_abs\_H)) + 1]$
        $total\_nonzero\_f_{min}$ += count_nonzero($f_{min}$)
        $total\_nonzero\_f_{max}$ += count_nonzero($f_{max}$)
        denom $\leftarrow \sqrt{\hat{H}_t} + \epsilon$
        step_dir $\leftarrow n_t/$denom
    end for
    if $total\_nonzero\_f_{max} < total\_nonzero\_f_{min}$ then
        lr $\leftarrow \mathrm{lr}_{\min}$
    else
        lr $\leftarrow \mathrm{lr}_{\max}$
    end if
    for each parameter $p$ do
        $p \leftarrow p - \mathrm{lr} \cdot \mathrm{step\_dir}$
    end for
end for
```

G. Non-Ergodic Convergence Analysis of the Proposed ASGD

The non-ergodic convergence rate of the proposed ASGD is guarantee by theorem 3 below [7].

Theorem 3. *The theorem follows the theorem in [10], [25] with the base learning rate replaced with a linear function $uf(H) + C$ and the partially adaptive parameter set at* 0.5.

Suppose assumption 2 and 3 are satisfied, we have the bound

$$\min_{0\leq t\leq T} \mathbb{E}\left[||\bigtriangledown g(W_t)||\right] \leq \frac{I_1 + I_2 \sum_{t=1}^{T-1}(uf(H)+C)_t^2}{\sum_{t=1}^{T}(uf(H)+C)_t^2} \tag{17}$$

where

$$\begin{aligned} I_1 = {} & [Z^2+\epsilon]^{\frac{1}{2}}\,(g(W_t)-\min g) + [Z^2+\epsilon]^{\frac{1}{2}} * \\ & \left(\frac{(uf(H)+C)_1 Z^2\sqrt{d}}{\epsilon^{\frac{1}{2}}(1-\beta_1)} + \frac{(uf(H)+C)_1\beta_1 Z^2}{\epsilon^{\frac{1}{2}}(1-\beta_1)}\right) + \\ & [Z^2+\epsilon]^{\frac{1}{2}}\left(\frac{(uf(H)+C)_T Z^2}{\epsilon^{\frac{1}{2}}(1-\beta_1)}\right) \end{aligned} \tag{18}$$

and

$$I_2 = \left[Z^2+\epsilon\right]^{\frac{1}{2}}\frac{1+\beta_1}{1-\beta_1}\frac{LZ^2}{2\epsilon} \tag{19}$$

If we suppose $\sum_t(uf(H)+C)_t = +\infty$, then the nonergodic convergence of the gradient sequence can be derived as

$$\lim_t \bigtriangledown g(W_t) = 0 \quad a.s., \quad \lim_t \mathbb{E}\left[||\bigtriangledown g(W_t)||^2\right] = 0. \tag{20}$$

H. Quantifying the Improvement from Adaptive Switching

Let α_t denote the base learning rate used by the optimizer at iteration t. For standard Adam or AMSGrad, the base rate is constant, i.e., $\alpha_t = \alpha_{\text{adam}}$. For the proposed ASGDAdam, the curvature-aware base rate is defined as

$$\alpha_t = u\,f(H_t) + C, \tag{21}$$

where $u, C \geq 0$ are hyperparameters and $f(H_t)$ is a function of the second-moment statistics H_t that governs the adaptive switching behavior.

1) Nonconvex Convergence Bound: Under the same assumptions as Theorem 2 (smooth nonconvex objective g, bounded stochastic gradients, and partial adaptivity fixed at 0.5), the ergodic gradient-norm bound for ASGDAdam is

$$\frac{1}{T-1}\sum_{t=2}^{T}\mathbb{E}\left[\|\nabla g(W_t)\|^2\right] \leq \frac{R_8}{T\,\overline{\alpha}_T} + \frac{R_9 d}{T} + R_{10} d\,\frac{\overline{\alpha}_T}{\sqrt{T}}, \tag{22}$$

where $\overline{\alpha}_T = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\alpha_t]$ is the average base rate, d is the parameter dimension, and R_8, R_9, and R_{10} are constants defined in Theorem 2 of the original paper.

For comparison, a fixed-base Adam/AMSGrad satisfies

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla g(W_t)\|^2\right] \leq \frac{R_8}{T\,\alpha_{\text{adam}}} + \frac{R_9 d}{T} + R_{10} d\,\frac{\alpha_{\text{adam}}}{\sqrt{T}}. \tag{23}$$

2) Improvement Factor on the Leading Term: The improvement in the leading bias term of (22) relative to (23) is given by

$$\text{Bias improvement factor} = \frac{\frac{R_8}{T\,\overline{\alpha}_T}}{\frac{R_8}{T\,\alpha_{\text{adam}}}} = \frac{\alpha_{\text{adam}}}{\overline{\alpha}_T}. \tag{24}$$

When $\overline{\alpha}_T > \alpha_{\text{adam}}$, the leading term is proportionally smaller, implying a faster decay of bias and hence a higher empirical convergence speed.

3) Variance and Stability Term: The variance term in (22) scales linearly with $\overline{\alpha}_T$. Naively, this suggests a possible increase in variance when $\overline{\alpha}_T > \alpha_{\text{adam}}$. However, ASGDAdam mitigates this by applying smaller learning rates in high-curvature regions and larger rates in flatter regions. Formally, if we denote the flat-region set as $\mathcal{F}$ and the sharp-region set as $\mathcal{S}$, with corresponding curvature lower bounds c_F and c_S, then

$$\frac{1}{T}\sum_{t=1}^{T} \frac{\alpha_t}{(\sqrt{H_t}+\epsilon)^2} \le \frac{|\mathcal{F}|}{T}\cdot\frac{\alpha_{\max}}{c_F^2} + \frac{|\mathcal{S}|}{T}\cdot\frac{\alpha_{\min}}{c_S^2}. \tag{25}$$

This assignment prevents large update variance in regions with steep curvature, improving stability without sacrificing asymptotic convergence.

4) Net Improvement Condition: The difference between the two upper bounds (22) and (23) is

$$\Delta(T) = \frac{R_8}{T}\Big(\frac{1}{\overline{\alpha}_T} - \frac{1}{\alpha_{\text{adam}}}\Big) + R_{10}d\,\frac{1}{\sqrt{T}}\Big(\overline{\alpha}_T - \alpha_{\text{adam}}\Big). \tag{26}$$

ASGDAdam yields a tighter convergence bound (i.e., $\Delta(T) < 0$) whenever the reduction in the bias term dominates the variance increase. Empirically, this condition is often satisfied because the switching rule keeps α_t small in high-curvature regimes, leading to both improved convergence constants and reduced oscillation.

5) Numerical Illustration: Consider $\alpha_{\text{adam}} = 10^{-4}$, $\alpha_{\min} = 10^{-7}$, $\alpha_{\max} = 10^{-4}$, and the fraction of $\alpha_{\max}$-steps $\alpha_T = 0.6$. Then

$$\overline{\alpha}_T = 0.6\times10^{-4} + 0.4\times10^{-7} \approx 6.0\times10^{-5}. \tag{27}$$

The bias improvement factor from (24) becomes

$$\frac{\alpha_{\text{adam}}}{\overline{\alpha}_T} \approx 1.67, \tag{28}$$

implying a ~40% reduction in the leading convergence term. Because ASGDAdam employs $\alpha_{\min}$ precisely when curvature is high, the corresponding variance term does not increase proportionally, resulting in improved empirical stability and faster convergence.

6) Summary: By adaptively switching the base learning rate between $\alpha_{\min}$ and $\alpha_{\max}$, ASGDAdam reduces the leading bias constant by the factor $\alpha_{\text{adam}}/\overline{\alpha}_T$, while preventing a proportional increase in the variance term through targeted assignment of $\alpha_{\min}$ to high-curvature iterations. Consequently, ASGDAdam achieves a smaller nonconvex ergodic bound and demonstrably faster convergence than fixed-base learning rate Adam or AMSGrad, particularly in highly nonuniform optimization landscapes.

IV. Results

We implemented a comprehensive multi-modal PBMC perturbation learning framework that integrates graph attention networks (GATs) for constructing latent embeddings across diverse cellular modalities including RNA, ATAC, and ADT [1], enabling the model to capture cell-type-specific and cross-modality regulatory dependencies. Building on this foundation, a proximal policy optimization (PPO)-based reinforcement learning agent simulates CRISPR perturbations in silico, learning optimal intervention strategies that drive desired transcriptional or chromatin outcomes while accounting for the stochasticity and feedback inherent in cellular systems. To enhance convergence and robustness during training, the framework compares a suite of adaptive optimization algorithms notably Adam, AMSGrad, Padam, and adaptive step-size gradient descent (ASGD) variants—each evaluated for stability, learning efficiency, and biological fidelity. Finally, the pipeline performs comprehensive quantitative and topological analyses, including pseudotime trajectory visualization and per-gene performance metrics such as MSE, R^2, and Pearson correlation, to assess how well each optimizer and perturbation policy captures realistic gene expression dynamics and regulatory network behavior.

1) Parameter Setting: We evaluate the performance of all optimizers under a unified reinforcement learning framework using the PBMC multi-modal dataset with graph attention network (GAT) embeddings. Each optimizer is incorporated into the proximal policy optimization (PPO) training pipeline, where the environment models dynamic gene regulatory responses across RNA, ATAC, and protein modalities under perturbation conditions.

To assess the stability and adaptability of the reinforcement learning environment, we conducted evaluations across N_EVAL_EPISODES = 30 independent runs. Each episode was constrained to a maximum of MAX_STEPS = 1000 interactions, allowing the agent to explore perturbation strategies over a fixed simulation horizon. A perturbation was applied with a probability of PERTURB_PROB = 0.1, enabling controlled stochasticity in the environment. The maximum number of concurrent perturbations was limited to MAX_PERTURB = 2, enforcing a double perturbation regime that reflects realistic biological or experimental sparsity. This setting ensures that the agent learns effective policies under sparse and minimally invasive perturbation conditions, balancing exploration of combinatorial effects with stability in learned responses.

The evaluation metrics include Accuracy, Precision, Recall, F1-score, Area Under the Precision–Recall Curve (AUPRC), and regression-based measures such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Pearson Correlation (r) between predicted and experimentally observed cellular responses. Each model is trained for 50,000 PPO timesteps, and all reported results are averaged across 10 independent runs to ensure robustness and reproducibility.

We perform exhaustive grid searches to determine optimal parameters for all algorithms, including the base learning rate selected from $\{1\times10^{-7}, 1\times10^{-6}, 1\times10^{-5}, 1\times10^{-4}, 1\times10^{-3}, 1\times10^{-2}, 1\times10^{-1}\}$, the partially adaptive parameter (p) from $1/4, 1/8, 1/16$, and the second-order momentum parameter (β_2) from $0.9, 0.99, 0.999$. For Adam and AMSGrad, the base learning rate is fixed at 1×10^{-4}, with momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. In the case of Padam, the base learning rate is set to 1×10^{-2}, and the partially adaptive parameter is chosen as $p = 0.125$, while the first- and second-order momentum coefficients are set to 0.9 and 0.999, respectively.

For the proposed ASGD (Improved AMSGrad and Adam) optimizer, we employ a dual learning-rate mechanism, where the step size dynamically alternates between a minimum learning rate of 1×10^{-7} and a maximum learning rate of 1×10^{-4}, depending on the non-zero frequency of local gradient curvature. The momentum parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

All optimizers are benchmarked under identical PPO configurations, batch sizes, and environment hyperparameters to ensure fairness in comparison.

Table II Performance comparison of different optimization algorithms. Maximum number of micro steps was set to 1000 the perturbation probability was set to 0.1 and we used insilico double perturbation

Algorithm	Accuracy	Precision	Recall	F1	AUPRC	MSE	RMSE	MAE	R^2	Pearson
adam_test	0.572	0.597	0.572	0.563	0.518	9.438	3.039	2.561	-12.391	-0.022
amsgrad_test	0.578	0.596	0.578	0.575	0.522	8.644	2.886	2.414	-10.953	-0.030
asgdaamsgrad_test	0.705	0.762	0.705	0.687	0.626	1.274	1.102	0.868	-0.762	-0.009
asgdadam_test	**0.760**	**0.781**	**0.760**	**0.757**	**0.678**	**0.845**	0.903	**0.692**	**-0.053**	-0.015
padam_test	0.638	0.664	0.638	0.632	0.564	2.091	1.417	1.086	-1.873	**0.098**
sgd_test	0.627	0.644	0.627	0.625	0.554	1.997	**1.395**	1.051	-1.818	0.203

The result of the double perturbation experiment is shown in Table II. From a classification perspective where the algorithm decide to upregulate or downregulate the gene regulatory network, ASGD-Adam achieved the best overall performance, with the highest accuracy, precision, recall, F1 score, and AUPRC This demonstrates that ASGD-Adam provides the most balanced and reliable predictions, effectively optimizing both sensitivity and specificity. ASGD-AmsGrad also showed strong performance, significantly outperforming traditional Adam and AMSGrad, while Padam and SGD exhibited moderate classification capabilities.

In terms of regression metrics is where algorithm predict the lineage in multistep function. ASGD-Adam again led in most measures, achieving the lowest MSE and MAE , highlighting its superior predictive accuracy for continuous outcomes. SGD, however, achieved the lowest RMSE and the highest Pearson correlation, suggesting it captures linear trends effectively but is less consistent across the error distribution. Other algorithms such as ASGD-AmsGrad and Padam demonstrated intermediate performance, while Adam and AMSGrad consistently underperformed relative to ASGD-based methods.

Our results suggested that ASGD-Adam flattens the effective landscape valleys for favorable trajectories, allowing cells to reach stable differentiated states with lower bias and controlled variance. Specifically, its adaptive learning rate strategy reduces overshooting in high-curvature regions (sharp valleys) while maintaining sufficient step size in flatter regions, mimicking a cell's ability to explore flexible but constrained paths. This leads to smoother, more accurate convergence to target states, compared to standard Adam or AMSGrad, which can either stagnate in local minima or oscillate excessively in steep regions.

In practical terms, ASGD-Adam's performed with high accuracy and low regression errors which corresponded to a more precise mapping of gene expression landscapes, where the predicted trajectories better align with empirical differentiation patterns. SGD, while capturing linear trends well (high Pearson correlation), may allow cells to oscillate along valleys, reflecting less stable or biologically realistic pathways. Overall, ASGD-based optimizers appear to enhance the fidelity of computational models of cell state dynamics, effectively "reshaping" the Waddington landscape for faster and more robust differentiation simulations.

V. Conclusions

Convergence Analysis in Convex Setting

To the best of our knowledge, the difference between our proposed algorithm [7] and the existing methods is the introduction of $(uf(H)+C)$ linear function to replace the base learning rate α_{base}. The next steps bear a close resemblance to the steps of proof of Theorem 1 in FastAdamBrief [15] and AMSgrad [14].

Proof: Suppose that $W^* \in R^d$ is classified as the optimal point of the problem. Algorithm 1 and 2 is derived from lemma 2 as follows [15][14].

$$\begin{aligned} &||H_t^{\frac{1}{4}}(W_{t+1}-W_t^*)||_2^2 \leq ||H^{\frac{1}{4}}(W_t-(uf(H)+C)_t H_t^{-\frac{1}{2}}.n_t - W^*)||_2^2 \\ &= ||H^{\frac{1}{4}}(W_t-(uf(H)+C)_t H_t^{-\frac{1}{2}}.n_t - W^*)||_2^2 - \\ &2(uf(H)+C)\langle \beta_{1t}n_{t-1}+(1-\beta_{1t})z_t, W_t-W^*\rangle. \end{aligned} \tag{29}$$

From Theorem 1 and the equation below [15], [14]

$$g_t(W_t)-g_t(W^*) \leq \langle z_t, W_t\rangle \tag{30}$$

we have [15], [14]

$$\begin{aligned} R(T) &= \sum_{t=1}^{T} g_t(W_t) - \min_{W \in \chi} \sum_{t=1}^{T} g_t(W_t) \\ &\leq \sum_{i=1}^{T} \langle z_t, W_t \rangle \end{aligned} \tag{31}$$

Since $H^p(W^*) = W^*$, we can follow AMSgrad [14] and obtain the first inequality and then using Lemma 2 [15], [14], the inequality can be rearranged to obtain [15], [14]

$$\begin{aligned} &\langle z_t, W_t \rangle \leq \\ &\frac{1}{2((u.f(H)+C))_t(1-\beta_{1t})}* \\ &\left[||H^{\frac{1}{4}}(W_{t+1} - W^*)||_2^2 - ||H^{\frac{1}{4}}(W_t - W^*)||_2^2 \right] + \\ &\sum_{t=2}^{T} \frac{\beta_{1t}}{2((u.f(H)+C))_{t-1}(1-\beta_{1t})} ||H^{\frac{-\frac{1}{2}}{2}}(W_t - W^*)||_2^2 + \\ &\sum_{t=2}^{T} \frac{\beta_1((u.f(H)+C))_{t-1}}{2(1-\beta_{1t})}.||H_{t-1}^{\frac{-1}{4}} n_{t-1}||_2^2 + \\ &\sum_{t=2}^{T} \frac{\beta_1((u.f(H)+C))_t}{2(1-\beta_{1t})}.||H_t^{\frac{-1}{4}} n_t||_2^2 \end{aligned} \tag{32}$$

and

$$\begin{aligned} &R(T) = R_1 + R_2 + R_3 \\ &R_1 = \sum_{t=1}^{T} \sum_{l=1}^{d} \frac{\beta_{1t}.h_{t,l}^{\frac{1}{4}}}{2((u.f(H)+C))(1-\beta_1)} \\ &\left[(W_{t+1} - W^*)^2 - (W_t - W^*)^2 \right], \\ &R_2 = \sum_{t=1}^{T} \sum_{l=1}^{d=1} \frac{\beta_{1t} h_{t,l}^{\frac{1}{2}}}{2((u.f(H)+C))_t(1-\beta_1)} \sum_{t=1}^{T} \sqrt{t} \lambda^{t-1} \\ &R_3 = \frac{1+\beta_1}{2(1-\beta_1)} \sum_{t=1}^{T} \sum_{l=1}^{d} \frac{((u.f(H)+C))_t.n_{t,l}^2}{h_{t,l}^{\frac{1}{2}}}. \end{aligned} \tag{33}$$

According to AMSgrad [14], [7] with the base learning rate α_{base} changed to $((u.f(H)+C))$, the second inequality follows cauchy-schwarz inequality. By applying $\beta_{1t} \leq \beta_1 \leq 1$ which is monotonically decreasing in t and satisfying the condition of convergence, i.e., $\Gamma_t = \sum_{t=1}^{T} \sum_{l=1}^{d} \left(\frac{h_{l,t}^{\frac{1}{2}}}{((u.f(H)+C))_t} - \frac{h_{t-1,l}^{\frac{1}{2}}}{((u.f(H)+C))_{t-1})} \right) \geq 0$, we have

$$
\begin{aligned}
R_1 = \frac{1}{2(1-\beta_1)} \sum_{l=1}^{d} \frac{h_{t,l}^{\frac{1}{2}}(w_{1,l}-w_l^*)^2}{((u.f(H)+C))_1} + \\
\sum_{t=1}^{T}\sum_{l=1}^{d}\left(\frac{h_{t,l}^{\frac{1}{2}}}{((u.f(H)+C))_t} - \frac{h_{t-1,l}^{\frac{1}{2}}}{((u.f(H)+C))_{t-1}}\right)(w_{t,l}-w_l),
\end{aligned}
\tag{34}
$$

$$
\begin{aligned}
&\le \frac{D_\infty^2}{2(1-\beta_1)} \sum_{l=1}^{d} \frac{h_{T,l}^{\frac{1}{2}}}{((u.f(H)+C))_T}, \\
&= \frac{D_\infty^2}{2(1-\beta_1)} \sum_{l=1}^{d} \sqrt{T} h_{T,l}^{\frac{1}{2}}.
\end{aligned}
\tag{35}
$$

Next, based on AMSgrad [14], for the second term of the equation, we have

$$
\begin{aligned}
R_2 = \sum_{t=1}^{T}\sum_{l=1}^{d=1} \frac{\beta_{1t} h_{t,l}^{\frac{1}{2}}}{2((u.f(H)+C))_t(1-\beta_1)} \sum_{t=1}^{T} \sqrt{t}\lambda^{t-1} \le \\
\frac{\beta_1 d D_\infty^2 Z_\infty^1}{2((u.f(H)+C))(1-\beta_1)(1-\lambda)^2}.
\end{aligned}
\tag{36}
$$

Lastly, according to AMSgrad [14], we can utilize geometric series upper bound after relaxing $\sqrt{t}$ to t and we have

$$
\begin{aligned}
R_3 = \frac{1+\beta_1}{2(1-\beta_1)} \sum_{t=1}^{T}\sum_{l=1}^{d} \frac{((u.f(H)+C))_t n_{t,l}^2}{h_{t,l}^{\frac{1}{2}}} \le \\
\frac{(u.f(H)+C) Z_\infty^1 \sqrt{1+\log T}}{(1-\beta_1)^2(1-\gamma)(1-\beta_2)^{\frac{1}{2}}} \sum_{l=1}^{d} ||z_{1:T,l}||_2.
\end{aligned}
\tag{37}
$$

□

Ergodic Convergence Analysis in Nonconvex Setting

The subsequent procedures resemble those outlined in the proof of Theorem 2 in [12] and [11], albeit with the base learning rate replaced by a linear function $(uf(H)+C)$. Before delving into the proof of Theorem 2, it is imperative to establish the following lemma [7]

Lemma 3. *[12]: We defined an arbitrary sequence q_t sequence and using the defined n_t we have*

$$
\begin{aligned}
q_t &= W_t - \frac{\beta_1}{1-\beta_1}(W_t - W_{t-1}) \\
&= \frac{1}{1-\beta_1} W_t - \frac{\beta_1}{1-\beta_1} W_t - W_{t-1}.
\end{aligned}
\tag{38}
$$

Then, for t greater than 2, we can obtain

$$\begin{aligned}
&q_{t+1} - q_t = \\
&\frac{\beta_1}{1-\beta_1}\left[I - \left((uf(H)+c)_t H_t^{-\frac{1}{2}}\right)\left((uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}}\right)^{-1}\right] \\
&(W_{t-1} - W_t) - (u.f(H)+C)_t H_t^{-1/2} z_t.
\end{aligned} \tag{39}$$

Following [12], it should be noted that $q_{t+1} - q_t$ can be represented as n_t, z_t and $\frac{1}{H_t^{\frac{1}{2}}}$. We can reduce $q_{t+1} - q_t$ to

$$\begin{aligned}
&q_{t+1} - q_t = \\
&\frac{\beta_1}{1-\beta_1}\left((uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}} - (uf(H)+c)_t H_t^{-\frac{1}{2}}\right) \\
&n_t - (u.f(H)+C)_t H_t^{-1/2} z_t.
\end{aligned} \tag{40}$$

Proof of Theorem 2(slight modification of [12])

Based on the steps in [12], considering L is smooth, we have

$$g(q_{t+1}) \leq g(q_t) + R_4 + R_5 + R_6, \tag{41}$$

where

$$\begin{aligned}
R_4 = &\frac{\beta_1}{1+\beta_1} \triangledown g(W_t)^T \left((u.f(H)+C)_{t-1} H_{t-1}^{-\frac{1}{2}}\right) n_{t-1} \\
&- \triangledown g(W_t)^T (u.f(H)+C)_t H_t^{-\frac{1}{2}} z_t.
\end{aligned} \tag{42}$$

Given that

$$\begin{aligned}
&\triangledown g(W_t)^T \\
&* \left((u.f(H)+C)_{t-1} H_{t-1}^{-\frac{1}{2}} - (u.f(H)+C)_t H_t^{\frac{-1}{2}} z_t\right) n_{t-1} H_{t-1}^{-\frac{1}{2}} \\
&\leq || \triangledown g(W_t)||_\infty . ||(uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}} - (uf(H)+c)_t H_t^{-\frac{1}{2}}|| \\
&||n_{t-1}||_\infty, \\
&\leq Z_\infty^2 \left[||(uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}}||_{1,1} - ||(uf(H)+c)_t H_t^{-\frac{1}{2}}||_{1,1}\right].
\end{aligned} \tag{43}$$

According to generalized Holder inequality expression in [12], for $x, y, z \in \Re^d$, we can obtain

$$x^T A y \leq ||x||_\infty * ||A||_{1,1} * ||y||_\infty, \tag{44}$$

then the the bound is found to be

$$
\begin{aligned}
&-\nabla g(W_t)^T(u.f(H)+C)_t H_t^{-\frac{1}{2}} z_t \\
&= -\nabla\, g(W_t)^T(u.f(H)+C)_{t-1} z_t \\
&- g(W_t)^T(u.f(H)+C)_t H_t^{-\frac{1}{2}} - g(W_t)^T(u.f(H)+C)_{t-1} H_{t-1}^{-\frac{1}{2}}, \\
&\leq -g(W_t)^T(u.f(H)+C)_{t-1} H_{t-1}^{-\frac{1}{2}} z_t + \\
&Z_\infty^2 \left[||(uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}}||_{1,1} - ||(uf(H)+c)_t H_t^{-\frac{1}{2}}||_{1,1} \right].
\end{aligned}
\tag{45}
$$

Following the steps in [11], bound for R_5 can be established as

$$
\begin{aligned}
&R_5 = (\nabla g(q_t) - \nabla g(W_t))^T\, (q_{t+1} - q_t) \\
&= L \frac{\beta_1}{1-\beta_1} ||(uf(H)+c)_t H_t^{-\frac{1}{2}} z_t||_2 ||W_t - W_{t-1}||_2 + \\
&L \left(\frac{\beta_1}{1-\beta_1} \right) ||W_t - W_{t-1}||_2^2, \\
&\leq L||(uf(H)+c)_t H_t^{-\frac{1}{2}} z_t||_2^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 ||W_t - W_{t-1}||_2^2.
\end{aligned}
\tag{46}
$$

According to [11], bound for R_6 can be computed as

$$
\begin{aligned}
&R_6 = \frac{L}{2} ||q_{t+1} - q_t||_2^2 \\
&\leq \frac{L}{2} \left[||uf(H)+c)_t H_t^{-\frac{1}{2}} z_t||_2 + \frac{\beta_1}{1-\beta_1} ||W_{t-1} - W_t||_2 \right]^2, \\
&\leq L||uf(H)+c)_t H_t^{-\frac{1}{2}} z_t||_2^2 + 2L \left(\frac{\beta_1}{1-\beta_1} \right)^2 ||W_{t-1} - W_t||_2^2.
\end{aligned}
\tag{47}
$$

Based on [11], bound for $g(q_t+1) - g(q_t)$ can be expressed as

$$
\begin{aligned}
&g(q_t+1) - g(q_t) \leq -\nabla\, g(W_t)^T(uf(H)+c)_t H_t^{-\frac{1}{2}} z_t + \\
&2L||(uf(H)+c)_t H_t^{-\frac{1}{2}} n_t||_2^2 \\
&+ 4L \left(\frac{\beta_1}{1-\beta_1} \right)^2 ||(uf(H)+c)_{t-1} H_{t-1}^{-\frac{1}{2}} n_{t-1}||_2^2.
\end{aligned}
\tag{48}
$$

For $t = 2$ to T

$$
\begin{aligned}
&Z_\infty^{-1}\sum_{t=2}^{T}(uf(H)+c)_{t-1}\,\mathbb{E}\,||\bigtriangledown g(W_t)||_2^2 \\
&\leq \mathbb{E}\left[g(q_1)+\frac{Z_\infty^2||(u.f(H)+C)_1H_1^{-\frac{1}{2}}||_{1,1}}{1-\beta_1}\right]- \\
&\left[g(q_{T+1})-\frac{Z_\infty^2||(u.f(H)+C)_TH_T^{-\frac{1}{2}}||_{1,1}}{1-\beta_1}\right]+ \\
&2L\sum_{t=1}^{T}\mathbb{E}\,||(uf(H)+c)_tH_t^{-\frac{1}{2}}z_t||_2^2 \\
&+4L\sum_{t=2}^{T}\mathbb{E}\left(\frac{\beta_1}{1-\beta_1}\right)^2||(uf(H)+c)_{t-1}H_{t-1}^{-\frac{1}{2}}n_{t-1}||_2^2 \\
&\leq \mathbb{E}\left[\triangledown g+\frac{Z_\infty^2(u.f(H)+C)_1\epsilon^{-\frac{1}{2}}d}{1-\beta_1}+d(uf(H)+C)_1Z_\infty\right]+ \\
&2L\sum_{t=1}^{T}\mathbb{E}\,||(uf(H)+c)_tH_t^{-\frac{1}{2}}z_t||_2^2 \\
&+4L\sum_{t=2}^{T}\mathbb{E}\left(\frac{\beta_1}{1-\beta_1}\right)^2||(uf(H)+c)_{t-1}H_{t-1}^{-\frac{1}{2}}n_{t-1}||_2^2.
\end{aligned}
\tag{49}
$$

Following [12], with $\gamma = \frac{\beta_1}{\beta_2^{\frac{1}{2}}}$, we can obtain

$$
\begin{aligned}
&\sum_{t=1}^{T}(u.f(H)+C)_t^2\,\mathbb{E}\left[||H_t^{-\frac{1}{2}}n_t||_2^2\right] \\
&\leq \frac{T^{\frac{1}{2}}(uf(H)+C)_t^2(1-\beta_1)}{2\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\gamma)}\,\mathbb{E}\left(\sum_{i=1}^{d}||z_{1:T,i}||_2\right)
\end{aligned}
\tag{50}
$$

and

$$
\begin{aligned}
&\sum_{t=1}^{T}(u.f(H)+C)_t^2\,\mathbb{E}\left[||H_t^{-\frac{1}{2}}z_t||_2^2\right] \\
&\leq \frac{T^{\frac{1}{2}}(uf(H)+C)_t^2}{2\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\gamma)}\,\mathbb{E}\left(\sum_{i=1}^{d}||z_{1:T,i}||_2\right).
\end{aligned}
\tag{51}
$$

According to [11], given $\kappa \in [\max\{0, 4p-1\}, 1]$ and after introducing equation (50) and (51) into 41, we have

$$
\begin{aligned}
&\mathbb{E}||\triangledown g(W_{out})||_2^2 = \frac{1}{\sum_{t=2}^{T}(uf(H)+C)_{t-1}} \\
&\sum_{t=2}^{T}(uf(H)+C)_{t-1}\,\mathbb{E}||\triangledown g(W_t)||_2^2 \\
&\leq \frac{Z_\infty}{\sum_{t=2}^{T}(uf(H)+C)_{t-1}} * \\
&\left[\nabla g + \frac{Z_\infty^2(u.f(H)+C)_1\epsilon^{-\frac{1}{2}}d}{1-\beta_1} + d(uf(H)+C)_1 Z_\infty\right] + \\
&\sum_{t=1}^{T}\frac{2LZ_\infty}{(uf(H)+C)_{t-1}}\frac{T^{\frac{1}{2}}(uf(H)+C)_t^2}{2\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\gamma)}\,\mathbb{E}\left(\sum_{i=1}^{d}||z_{1:T,i}||_2\right)^{1-\kappa} + \\
&\sum_{t=2}^{T}\frac{4LZ_\infty}{(uf(H)+C)_{t-1}}\left(\frac{\beta_1}{1-\beta_1}\right)^2 * \\
&\frac{T^{\frac{1}{2}}(uf(H)+C)_t^2(1-\beta_1)}{2\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\gamma)}\,\mathbb{E}\left(\sum_{i=1}^{d}||z_{1:T,i}||_2\right)^{1-\kappa} \\
&\leq \frac{1}{T(uf(H)+C)}2Z_\infty \triangledown g + \frac{2}{T}\left(\frac{Z_\infty^3\epsilon^{-\frac{1}{2}}d}{1-\beta_1} + dZ_\infty^2\right) + \\
&\left(\frac{2Z_\infty L(uf(H)+C)}{T^{\frac{1}{2}}\epsilon^{\frac{1}{2}}(1-\gamma)(1-\beta_2)^{\frac{1}{2}}}\right) \\
&\mathbb{E}\left(\sum_{i=1}^{d}|||z_{1:T,i}||_2\right)\left(1+2(1-\beta_1)\left(\frac{\beta_1}{1-\beta_1}\right)^2\right).
\end{aligned}
\tag{52}
$$

According to [12], since $\alpha_t = uf(H)+C$, we can state theorem 2 with condition $||z_{1:T,i}||_2 \leq Z_\infty T^s$ as

$$
\begin{aligned}
&\frac{1}{T-1}\sum_{t=2}^{T}\mathbb{E}[||\triangledown g(W_t)||_2^2] \leq \frac{R_8}{T.(u.f(H)+C)} + \frac{R_9 d}{T} + \\
&\frac{(u.f(H)+C)R_{10}d}{T^{\frac{1}{2}}},
\end{aligned}
\tag{53}
$$

where

$$R_8 = 2Z_\infty\Delta g, \tag{54}$$

$$R_9 = \frac{2Z_\infty^3\epsilon^{-\frac{1}{2}}}{1-\beta_1} + 2Z_\infty^2, \tag{55}$$

$$R_{10} = \frac{2LZ_\infty^2}{\epsilon^{\frac{1}{2}}(1-\beta_2)^{\frac{1}{2}}(1-\frac{\beta_1}{\beta_2^{\frac{1}{2}}})}\left(1+\frac{2\beta_1^2}{1-\beta_1}\right), \tag{56}$$

and

$$\Delta g = g(W_1) - \inf_{W} g(W). \tag{57}$$

□

Non-Ergodic Convergence Analysis

Following the work in [10], [25] with base learning rate modified to a linear function $uf(H) + C$ and the partially adaptive parameter fixed at 0.5, we begin the proof of Theorem 3 by stating slightly modified lemma taken from [10], [25].

Lemma 4. *: Assuming $(W_t)_{t\leq 1}$ is generated by algorithm 1 and 2 as*

$$\theta_t = \mathbb{E}\left(\langle -(uf(H)+C)z(W_t), \frac{\beta_1 n_{t-1} + (1-\beta_1)z_t}{(H_t+\epsilon)^{\frac{1}{2}}} \rangle\right), \tag{58}$$

then

$$\theta_t \leq \beta_1\theta_{t-1} - (uf(H)+C)_t \frac{(1-\beta_1)}{(Z^2+\epsilon)^{\frac{1}{2}}} \mathbb{E}\,||z(W_t)||^2 + R_{7t}, \tag{59}$$

where

$$\begin{gathered} R_{7t} = ((uf(H)+C)_{t-1} - (uf(H)+C)_t)\frac{\beta_1 Z^2}{\epsilon^{\frac{1}{2}}} + \\ \frac{\beta_1 L Z^2}{\epsilon}(uf(H)+C)_t^2 + \\ (uf(H)+C)_t Z^2 \sqrt{d}\,\mathbb{E}\left[\sum_{j=1}^{d}\left(\frac{1}{(h_{j,t-1}+\epsilon)} - \frac{1}{(h_{j,t}+\epsilon)}\right)\right]. \end{gathered} \tag{60}$$

Then, we have

$$\begin{gathered} (1-\beta_1)\sum_{t=1}^{T}(u.f(H)+C)_t \frac{\mathbb{E}\,||z(W_t)||^2}{(Z^2+\epsilon)^{\frac{1}{2}}} \\ \leq -\theta_t + (1-\beta_1)\sum_{t=1}^{T-1}(-\theta_t) + \sum_{t=1}^{T}(R_{7t}) \\ \leq (1-\beta_1)\sum_{t=1}^{T-1}(-\theta_t) + \sum_{t=1}^{T}(R_{7t}) + \frac{(uf(H)+C)_T Z^2}{\epsilon^{\frac{1}{2}}}. \end{gathered} \tag{61}$$

Based on the steps in [10], [25], considering the gradient z is Lipschitz continuous at point W_t, we have

$$\begin{aligned} g(W_{t+1}) &\leq g(W_t) + \langle z(W_t), W_{t+1} - W_t \rangle + \frac{L}{2} ||W_{t+1} - W_t||^2, \\ &= g(W_t) - (uf(H) + C)_t \langle z(W_t), \frac{n_t}{(H_t + \epsilon)^{\frac{1}{2}}} \rangle \\ &+ \frac{L(uf(H) + C)_t^2}{2} || \frac{n_t}{(H_t + \epsilon)^{\frac{1}{2}}} ||^2, \end{aligned} \tag{62}$$

and the total expectation is computed as

$$g(W_{t+1}) \leq g(W_t) + \theta_t + \frac{LZ^2(uf(H) + C)^2}{2\epsilon}. \tag{63}$$

Given that

$$\begin{aligned} &(1 - \beta_1) \sum_{t=1}^{T} (uf(H) + C) \frac{\mathbb{E} || \bigtriangledown g(W_t)||^2}{(Z^2 + \epsilon)^{\frac{1}{2}}} \\ &\leq (1 - \beta_1)(g(W_1) - \min g) + \frac{LZ^2}{2\epsilon} (1 - \beta_1) \sum_{t=1}^{T-1} (uf(H) + C)_t + \\ &\sum_{t=1}^{T} R_{7t} + (uf(H) + C)_T \frac{Z^2}{\epsilon^{\frac{1}{2}}}, \end{aligned} \tag{64}$$

and

$$\begin{aligned} &\sum_{t=1}^{T} R_{7t} \leq \frac{\beta_1 LZ^2}{\epsilon} \sum_{t=1}^{T} (uf(H) + C)_t^2 + \\ &\frac{(uf(H) + C)_1 \beta_1 Z^2}{\epsilon^{\frac{1}{2}}} + \frac{(uf(H) + C)_1 Z^2 \sqrt{d}}{\epsilon^{\frac{1}{2}}}, \end{aligned} \tag{65}$$

we arrive at

$$\begin{aligned} &\sum_{t=1}^{T-1} (uf(H) + C)_t^2 \, \mathbb{E} \, ||z(W_t)||^2 \leq \\ &[Z^2 + \epsilon]^{\frac{1}{2}} \left(g(W_1) - \min g\right) + [Z^2 + \epsilon]^{\frac{1}{2}} * \\ &\left(\frac{(uf(H) + C)_1 Z^2 \sqrt{d}}{\epsilon^{\frac{1}{2}} (1 - \beta_1)} + \frac{(uf(H) + C)_1 \beta_1 Z^2}{\epsilon^{\frac{1}{2}} (1 - \beta_1)} \right) + \\ &[Z^2 + \epsilon]^{\frac{1}{2}} \left(\frac{(uf(H) + C)_T Z^2}{\epsilon^{\frac{1}{2}} (1 - \beta_1)} \right) + \\ &\left[Z^2 + \epsilon\right]^{\frac{1}{2}} \frac{1 + \beta_1}{1 - \beta_1} \frac{LZ^2}{2\epsilon} \sum_{t=1}^{T-1} (uf(H) + C)_t^2. \end{aligned} \tag{66}$$

□

References

[1] Anonymous Authors, "scagents: A multi-agent framework for fully autonomous end-to-end single-cell perturbation analysis," in *ICML 2025 Workshop on GenBio (preprint)*, 2025, preprint (ICML submission). Code and models: https://anonymous.4open.science/r/scAgents-2025-242E/.

[2] H. Cui, C. Wang, H. Maan, K. Pang, F. Luo, N. Duan, and B. Wang, "scgpt: Toward building a foundation model for single-cell multi-omics using generative ai," *Nature Methods*, 2024.

[3] L. Zhu and J. Wang, "Quantifying landscape and flux from single-cell omics: Unraveling the physical mechanisms of cell function," *JACS Au*, vol. 5, no. 8, pp. 3738–3757, 2025.

[4] J. Zhuang, T. M. Tang, Y. Ding, S. C. Tatikonda, N. C. Dvornek, X. Papademetris, and J. S. Duncan, "Adabelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," *ArXiv*, vol. abs/2010.07468, 2020.

[5] J. Chen, D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu, "Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI'20, 2021.

[6] F. Y. Wu and F. Tong, "Non-Uniform Norm Constraint LMS Algorithm for Sparse System Identification," *IEEE Communications Letters*, vol. 17, no. 2, pp. 385–388, 2013.

[7] F. Boabang, "Refining optimization methods for training machine learning models: A case study in robotic surgical procedures," Ph.D. dissertation, Concordia University, 2024.

[8] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. null, p. 2121–2159, jul 2011.

[9] H. Mittal, K. Pandey, and Y. Kant, "ICLR Reproducibility Challenge Report (Padam : Closing The Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks)," *ArXiv*, vol. abs/1901.09517, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:249647677

[10] T. Sun, L. Qiao, Q. Liao, and D. Li, "Novel Convergence Results of Adaptive Stochastic Gradient Descents," *IEEE Transactions on Image Processing*, vol. 30, pp. 1044–1056, 2021.

[11] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of adam-type algorithms for non-convex optimization," *arXiv preprint arXiv:1808.02941*, 2018.

[12] D. Zhou, J. Chen, Y. Cao, Y. Tang, Z. Yang, and Q. Gu, "On the convergence of adaptive gradient methods for nonconvex optimization," *arXiv preprint arXiv:1808.05671*, 2018.

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR*, vol. abs/1412.6980, 2015.

[14] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *ArXiv*, vol. abs/1904.09237, 2018.

[15] Y. Zhou, K. Huang, C. Cheng, X. Wang, A. Hussain, and X. Liu, "Fastadabelief: Improving Convergence Rate for Belief-Based Adaptive Optimizers by Exploiting Strong Convexity," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[16] X. Huang, R. Xu, H. Zhou, Z. Wang, Z. Liu, and L. Li, "ACMo: Angle-Calibrated Moment Methods for Stochastic Optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7857–7864.

[17] J. Chen, C. Wolfe, Z. Li, and A. Kyrillidis, "Demon: Improved Neural Network Training With Momentum Decay," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3958–3962.

[18] K. Verma and A. Maiti, "Wsagrad: a novel adaptive gradient based method," *Applied Intelligence*, vol. 53, no. 11, pp. 14 383–14 399, 2023.

[19] H. Zhong, Z. Chen, C. Qin, Z. Huang, V. W. Zheng, T. Xu, and E. Chen, "Adam revisited: a weighted past gradients perspective," *Frontiers of Computer Science*, vol. 14, pp. 1–16, 2020.

[20] F. Huang, J. Li, and H. Huang, "Super-adam: faster and universal framework of adaptive gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9074–9085, 2021.

[21] K. Takahashi and S. Yamanaka, "A decade of transcription factor-mediated reprogramming to pluripotency," *Nature reviews Molecular cell biology*, vol. 17, no. 3, pp. 183–193, 2016.

[22] H.-X. Wen, S.-Q. Yang, Y.-Q. Hong, and H. Luo, “A Partial Update Adaptive Algorithm for Sparse System Identification,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 240–255, 2020.

[23] F. Wu and F. Tong, “Gradient optimization p-norm-like constraint lms algorithm for sparse system estimation,” *Elsevier Signal Processing*, vol. 93, no. 4, pp. 967–971, 2013.

[24] P. Xue and B. Liu, “Adaptive equalizer using finite-bit power-of-two quantizer,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1603–1611, 1986.

[25] M. He, Y. Liang, J. Liu, and D. Xu, “Convergence of adam for non-convex objectives: Relaxed hyperparameters and non-ergodic case,” *ArXiv*, vol. abs/2307.11782, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260125579