ANON: EXTRAPOLATING ADAPTIVITY BEYOND SGD AND ADAM

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive optimizers such as Adam and non-adaptive methods like SGD, the dominant first-order element-wise optimizers, exhibit distinct generalization capabilities across different architectures. Prior tunable optimizers attempt to bridge this gap by strictly interpolating between SGD and Adam, effectively confining adaptivity within the 0-to-1 bound. However, this restricted interpolation leaves performance on the table: the optimal adaptivity of some tasks lies outside this interval, and no existing tunable optimizer can reach it. Extrapolating adaptivity violates the strict non-decreasing pre-conditioner assumption underlying the convergence guarantees of existing adaptive methods, so standard guarantees no longer apply. To break this barrier, we propose Anon, an optimizer that achieves fully continuous adaptivity extrapolation across the entire real-number spectrum. To guarantee provable stability in these out-of-bound regimes, we introduce Incremental Delay Update (IDU), a novel mechanism that bypasses hard max-tracking strategies. We theoretically establish Anon's convergence in both convex and non-convex settings. Empirically, Anon demonstrates highly competitive and scalable performance among state-of-the-art element-wise optimizers on representative image classification, diffusion, and large language modeling tasks. Our code is available at https://anonymous.4open.science/r/Anon-6511/.

1 INTRODUCTION

Modern deep learning models rely heavily on optimization algorithms for effective training. Despite the wide success of adaptive optimizers such as Adam (Kingma & Ba, 2014) in large-scale models like diffusion networks (Nichol & Dhariwal, 2021) and large language models (LLMs) (Brown et al., 2020; Llama, 2023), they are often outperformed by non-adaptive methods such as SGD (Robbins & Monro, 1951) in classical architectures like CNNs (Wilson et al., 2017). Prior efforts attempt to bridge this generalization gap by interpolating between the behaviors of SGD and Adam (Chen & Gu, 2018; Ozkara et al., 2024; Savarese et al., 2021). However, large-scale independent benchmarks reveal that achieving the "best of both worlds" remains highly elusive in practice (Schmidt et al., 2021): interpolation-based methods often fail to match well-tuned SGD on classical vision or Adam on massive architectures. This empirical bottleneck raises a critical question: *How can we break this interpolation barrier to achieve a higher performance upper bound across diverse architectures?* Throughout, we study the family of first-order element-wise (diagonal) optimizers, which contains SGD, Adam, and their tunable variants.

To answer this question, we identify a fundamental mechanical difference between SGD and Adam: how they adjust step sizes in response to historical gradient magnitudes. We formalize this "scale compensation degree" as *adaptivity* (§2.2). Existing optimizers merely represent fixed points or safe interpolations within a restricted spectrum: SGD exhibits zero adaptivity ($A = 0$), while Adam exhibits full adaptivity ($A \approx 1$). Consequently, prior tunable methods strictly confine themselves to this $[0, 1]$ bound. However, the optimal adaptivity of some tasks lies outside this bound, and an optimizer confined to interpolation can never reach it.

However, naively extrapolating adaptivity violates the non-decreasing assumption (Reddi et al., 2019) underlying the convergence guarantees of existing adaptive methods, frequently triggering divergence. Traditional workarounds, like the hard max-tracking in AMSGrad, theoretically prevent divergence but compromise empirical performance. To overcome this limitation, we propose **Anon** (**A**daptivity

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

Non-restricted **O**ptimizer with a **N**ovel convergence technique). To our knowledge, Anon is the first optimizer that enables real-valued, continuous adaptivity extrapolation ($\gamma \in \mathbb{R}$) via a novel **Incremental Delay Update (IDU)** mechanism. IDU bypasses the non-decreasing assumption with a provably stable multi-scale accumulator, ensuring robust convergence across the entire real-number spectrum.

In summary, our main contributions are threefold:

- **A Unified View of Extrapolated Adaptivity:** We formalize adaptivity as a mathematically rigorous property. We demonstrate that merely interpolating between SGD and Adam is insufficient, and that continuous extrapolation ($\gamma \in \mathbb{R}$) is a valuable extension: matching SGDM at $\gamma = 0$ and Adam-level adaptivity at $\gamma = 1$, Anon makes the full real axis reachable, with the per-task optima in our experiments lying just outside the historical $[0, 1]$ interval.
- **The Anon Optimizer and IDU Mechanism:** We propose Anon powered by the novel IDU mechanism. IDU safely bypasses the strict non-decreasing pre-conditioner assumption, guaranteeing theoretical convergence matching AMSGrad and ideal Adam (under standard assumptions) even in extreme extrapolated regimes.
- **A Highly Competitive Element-wise Baseline:** Through evaluations on CNNs, diffusion models, and a proof-of-concept study on 1B-parameter LLMs, Anon explores previously unreachable adaptivity landscapes, establishing a robust and scalable baseline among state-of-the-art element-wise optimizers.

2 PRELIMINARIES

2.1 REVIEW OF THE FRAME OF OPTIMIZERS

We focus on *first-order optimizers*, which are widely used to train deep learning models. To facilitate a unified understanding of their differences and commonalities, we introduce a generic framework, summarized in Algorithm 1.

Algorithm 1 Generic Optimizer Method Frame

Require: $\boldsymbol{\theta}$, η, $\{\phi_t, \psi_t\}_{t=1}^{\infty}$
while $\boldsymbol{\theta}_t$ not converged **do**
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_t)$
  $\boldsymbol{m}_t \leftarrow \left(\phi_t(\boldsymbol{g}_{1:t,1}), \ldots, \phi_t(\boldsymbol{g}_{1:t,d})\right)^\top$
  $S_t \leftarrow \mathrm{diag}\left(\psi_t(\boldsymbol{g}_{1:t,1}), \ldots, \psi_t(\boldsymbol{g}_{1:t,d})\right)$
  $\boldsymbol{\theta}_t \leftarrow \Pi_{\mathcal{F},S_t}\left(\boldsymbol{\theta}_{t-1} - \eta(t) S_t^{-1} \boldsymbol{m}_t\right)$

Here, $\mathcal{F}$ denotes the convex feasible set. $\boldsymbol{\theta} \in \mathcal{F}$ is the parameter to be optimized. Define $f(\boldsymbol{\theta})$ as a real-valued scalar function to minimize. S_t is a diagonal matrix where $S_{t,i,i} := \psi_t(\boldsymbol{g}_{1:t,i})$. ψ_t is the pre-conditioner function. $\prod_{\mathcal{F},S}(y) = \mathrm{argmin}_{x\in\mathcal{F}}\|S^{1/2}(x-y)\|$ denotes the projection of y onto $\mathcal{F}$ under the scaling matrix S. The scheduler η controls the per-step learning rate, constant or cosine-annealed (Loshchilov & Hutter, 2016). $\boldsymbol{g}_t$ is the gradient at step t. $\boldsymbol{m}_t$ is a vector where $\boldsymbol{m}_{t,i} := \phi_t(\boldsymbol{g}_{1:t,i})$. The momentum operator $\phi_t : \mathbb{R}^t \to \mathbb{R}$ is typically implemented as a moving average of past gradients. The two common variants are:

$$\mathit{EMA}(\boldsymbol{x}_{1:t};\beta) = \frac{1-\beta}{(1-\beta^t)}\sum_{i=1}^{t}\beta^{t-i}x_i, \boldsymbol{M}(\boldsymbol{x}_{1:t};\beta) = \sum_{i=1}^{t}\beta^{t-i}x_i, \tag{1}$$

where *EMA* denotes the exponential moving average with bias correction. ***M*** refers to the classical momentum without normalization. Both operators serve to smooth the gradient history.

Since the smoothing behavior of ϕ is similar across optimizers, the key differentiator lies in the design of the pre-conditioner $\psi_t : \mathbb{R}^t \to \mathbb{R}_+$, which directly modulates the effective step size and direction of each parameter and shapes the optimizer's convergence, stability, and adaptability to heterogeneous parameter landscapes. Thus, we focus our subsequent analysis on the properties and effects of ψ. We summarize the designs of ϕ and ψ for representative optimizers in Table 1.

As shown in Table 1, if the bias correction factor $1/(1-\beta^t)$ is omitted in Exponential Moving Average (EMA), it essentially reduces to a classical momentum M scaled by a constant coefficient $1-\beta$. Therefore, for the remainder of this paper, we will primarily focus on analyzing the characteristics of the preconditioner ψ, assuming a shared momentum ϕ across all optimizers unless otherwise specified.

Table 1: Summary of momentum functions and pre-conditioners for representative optimizers (Polyak, 1964; Luo et al., 2019; Zhuang et al., 2020). For full expressions of complex terms (A_t^{AMSGrad}, A_t^{AdaBound}, $A_t^{\text{AdaBelief}}$, A_t^{Anon}), please refer to Table 5 of Appendix B.1.

OPTIMIZER	$\boldsymbol{\phi}_t(x)$	$\boldsymbol{\psi_t}(x)$	$\boldsymbol{A_t(\psi, x)}$
SGD	x_t	1	0
SGDM	$M(\boldsymbol{x};\beta)$	1	0
RMSPROP	x_t	$\sqrt{EMA(\boldsymbol{x}^2;\beta_2)}+\epsilon$	$\frac{1}{1+\epsilon/\sqrt{EMA(\boldsymbol{x}^2;\beta_2)}}$ (≈ 1)
ADAM	$EMA(\boldsymbol{x};\beta_1)$	ψ_t^{RMSPROP}	A_t^{RMSPROP}
AMSGRAD	ϕ_t^{ADAM}	$\max_{i\in[t]}\{\psi_i^{\text{RMSPROP}}\}$	$[0,1]$ (≈ 1)
ADABOUND	ϕ_t^{ADAM}	$\text{CLIP}(\psi_t^{\text{RMSPROP}}, f_l(t), f_u(t))$	$[0,1]$ $(1\to 0)$
ADABELIEF	ϕ_t^{ADAM}	$\Big(EMA((\boldsymbol{x}-\boldsymbol{\phi}^{\text{ADAM}})^2 +\epsilon/(1-\beta_2);\beta_2)\Big)^{1/2}+\epsilon$	$[0,1]$ (≈ 1)
ANON	ϕ_t^{ADAM}	ψ_t^{ANON} (5)	$\approx\gamma$

Extensive empirical evidence has shown that SGD and SGDM often generalize better than Adam on classical architectures such as ResNet (He et al., 2016), whereas Adam typically outperforms SGD on more complex architectures such as transformers. Many hypotheses have been proposed, from Adam escaping saddle points more efficiently (Staib et al., 2019) to SGD's bias toward flatter minima (Wilson et al., 2017), but a unified measure of this scaling behavior is lacking. **We therefore formally define an adaptivity metric to mathematically distinguish the scaling mechanisms of SGD and Adam and guide the design of extrapolative optimizers.** We study how adaptivity affects optimization in § 3.2, after giving its formal definition.

2.2 THE ADAPTIVITY OF EXISTING OPTIMIZERS

We formalize the concept of adaptivity based on the framework described in Algorithm 1.

Definition 2.1. Suppose the pre-conditioner ψ_n is continuous and strictly positive, i.e., $\psi_n : \mathbb{R}^n \to \mathbb{R}_{>0}$. For any optimizer following Algorithm 1, we define the adaptivity A of its pre-conditioner ψ as

$$A_n(\psi, \boldsymbol{x}_{1:n}) = \nabla_k \ln \psi_n(k\boldsymbol{x}_{1:n})\big|_{k=1}. \tag{2}$$

Furthermore, we define two pre-conditioners ψ and ψ' to be equivalent if and only if $A_n(\psi, \boldsymbol{x}_{1:n}) = A_n(\psi', \boldsymbol{x}_{1:n})$ for all $\boldsymbol{x}_{1:n} \in \mathbb{R}^n$ and $n \in \mathbb{N}_+$.

Intuition Explanation A measures the pre-conditioner's "response" to changes in the gradient's scale: $A = 0$ (like SGD) means the pre-conditioner is "non-reactive" and ignores the overall gradient scale, while $A \approx 1$ (like Adam) means it is "compensatory", adjusting itself with strength 1 to offset gradient-scale changes. Notably, A depends not only on the functional form of ψ, but also on the gradient history $\boldsymbol{g}_{1:t}$: pre-conditioning is inherently dynamic, and even for a fixed ψ its adaptivity can vary as the gradient distribution evolves. We also introduce an equivalence notion: two optimizers using different ψ functions may still be essentially equivalent from an adaptivity perspective.

Theorem 2.2. *If ψ and ψ' are from the same equivalence class, there exists $f : \mathbb{N}_+ \to \mathbb{R}_+$ such that $\psi_n(\boldsymbol{x}_{1:n}) = \psi'_n(\boldsymbol{x}_{1:n}) f(n)$ for any $\boldsymbol{x}_{1:n} \in \mathbb{R}^n$ and $n \in \mathbb{N}_+$.*

Decoupling from Scheduler Theorem 2.2 shows that if two pre-conditioners yield the same adaptivity for any input, then they are equivalent. Specifically, if there exists a scheduler adjustment that can eliminate the difference between two pre-conditioners (e.g., $\psi' = k\psi$ corresponds to $\eta'(t) = k\eta(t)$), we regard them as equivalent strategies. The proof of Theorem 2.2 is deferred to Appendix E.

Based on these definitions, we can characterize the adaptivity of several widely used optimizers: For SGD(M), the adaptivity is $A = 0$ in all dimensions, indicating no explicit scaling of the loss landscape. In contrast, for Adam and its variants (e.g., RMSProp, AdaBelief), the adaptivity is approximately $A = 1$, as the contribution of the small ϵ term is negligible compared to the accumulated gradient statistics most of the time. A more intricate case is AdaBound (Luo et al., 2019), whose adaptivity

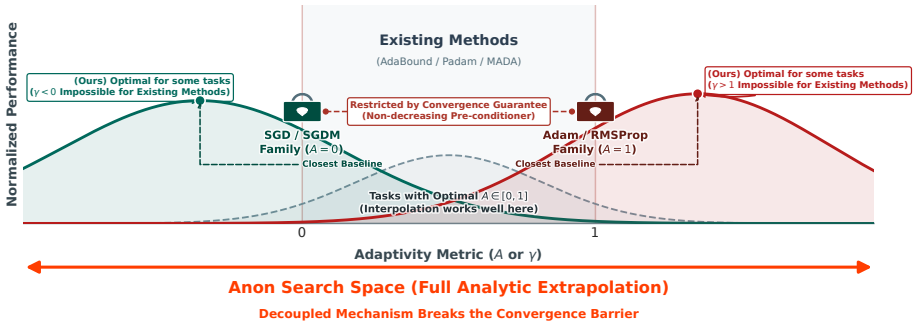


Figure 1: Visualizing the Dilemma: the distribution of existing optimizers' adaptivity and the task-dependence of the optimal adaptivity observed in our experiments.

transitions dynamically from $A \approx 1$ toward $A \approx 0$ as training proceeds. Specifically, AdaBound clamps the pre-conditioner ψ_t between shrinking bounds $\eta_l(t)$ and $\eta_u(t)$:

$$A_t(\psi^{\text{AdaBound}}, \boldsymbol{x}) = \begin{cases} A_t^{\text{RMSProp}}, & \text{if } \eta_l < \psi_t^{\text{RMS}} < \eta_u, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

As the bounds tighten over time, AdaBound behaves increasingly like SGD. This is supported by both evidence from Zhuang et al. (2020) and our experiments (Table 3), which indicates that AdaBound struggles in tasks such as GAN and diffusion model training, where high adaptivity is critical. These observations suggest the following: Optimizers with $A = 0$ (e.g., SGD) tend to generalize better on classical architectures such as CNNs, while those with $A = 1$ (e.g., Adam) perform better in complex modern architectures. However, whether $A = 0$, $A = 1$, or other values yield better performance remains an open question, which we explore in the next section.

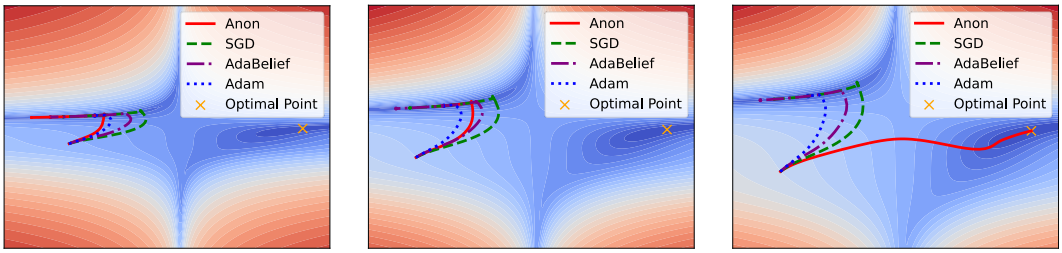
2.3 The Optimal Adaptivity for Tasks

We argue that the optimal adaptivity is inherently task-dependent and need not lie within the standard range: while mainstream optimizers typically operate at $A \approx 1$ (Adam) or $A \approx 0$ (SGD), some tasks favor adaptivity values outside the $[0, 1]$ interval (i.e., $A < 0$ or $A > 1$).

However, as we show in Figure 1, existing methods are theoretically constrained to interpolation. Approaches like Padam (Chen & Gu, 2018) (whose exponent p is merely the value our measure reads out on its pre-conditioner, $A \approx 2p$, as listed in Appendix B.1) and MADA (Ozkara et al., 2024) attempt to tune adaptivity, but they are fundamentally restricted to the convex hull of SGD and Adam. Specifically, Padam's exponent is confined to $p \leq 1/2$ (i.e., $A \in [0, 1]$), since $p > 1/2$ breaks its convergence condition and diverges in practice (Chen et al., 2018); MADA tunes the mixing coefficients of base optimizers via hyper-gradient descent, remaining within their convex hull by construction. Among these interpolation methods, we benchmark AdaBound, a widely used and well-known representative, in multiple experiments (Tables 2 and 3). Extending these methods for extrapolation is non-trivial because the convergence of most adaptive optimizers relies on the critical non-decreasing assumption:

$$\frac{\psi_{t+1}(g_{1:t+1,i})}{\eta(t+1)} \geq \frac{\psi_t(g_{1:t,i})}{\eta(t)}, \quad \forall i \in [d], \forall t \in \mathbb{N}_+, \tag{4}$$

which guarantees stability even in worst-case scenarios. Naively constructing an optimizer with negative adaptivity (e.g., via negative powers $\psi = (\psi^{\text{Adam}})^\gamma$ with $\gamma < 0$) typically causes the pre-conditioner value to decrease over time, violating Eq. 4 and leading to divergence. Prior works (Chen & Gu, 2018; Chen et al., 2018) further show that Padam suffers from divergence in practice when adaptivity extends beyond $[0, 1]$. Consequently, the literature still lacks a unified framework that stably extrapolates adaptivity to match diverse task geometries without breaking convergence guarantees.



(a) $\gamma = 1.5$ (b) $\gamma = 0.5$ (c) $\gamma = -0.5$

Figure 2: Trajectories of SGDM, Adam, AdaBelief, and Anon. The color change from deep red to deep blue represents the loss from high to low. The displayed landscape is the Beale surface rescaled per coordinate by Anon's pre-conditioner at epoch 100 (the unscaled surface is the standard Beale curved valley). More empirical experiments are shown in Appendix B.2 and D.

3 Extend to All Real Numbers

3.1 Adaptivity Tunable Optimizer and Beyond

While §2.2 and §2.3 suggest that extending adaptivity beyond $[0, 1]$ can be beneficial, achieving tunable adaptivity across all real numbers with convergence guarantees remains challenging. We propose a new technique called *incremental delay update (IDU)*, and design a novel optimizer *Anon* (**A**daptivity **N**on-restricted **O**ptimizer with a **N**ovel convergence technique) on top of it, extending the allowable range of adaptivity to all real numbers.

Algorithm 2 The Anon Optimizer

Require: η, β_1, β_2, ϵ, γ
Initialize $\boldsymbol{\theta}_0$, $\boldsymbol{m}_0 \leftarrow \mathbf{0}$, $\boldsymbol{s}_0 \leftarrow \mathbf{0}$, $t \leftarrow 0$, $k \leftarrow -1$
while $\boldsymbol{\theta}_t$ not converged **do**
  $t \leftarrow t + 1$
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_t)$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$
  $\widehat{\boldsymbol{m}}_t \leftarrow \frac{\boldsymbol{m}_t}{1-\beta_1^t}$
  $\boldsymbol{s}_t \leftarrow \beta_2 \boldsymbol{s}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
  if $k + 1 = \log_2 t$ **then**
    $k \leftarrow k + 1$
    $\boldsymbol{\sigma}_k \leftarrow \boldsymbol{s}_t / (1 - \beta_2^{\max(t/2,1)}) + \epsilon$
    $\boldsymbol{v}_k \leftarrow \begin{cases} \sqrt{2/\left(\frac{1}{\boldsymbol{v}_{k-1}^2} + \boldsymbol{\sigma}_k^{\gamma}\right)}, & \text{if } k > 0, \\ \boldsymbol{\sigma}_k^{-\gamma/2}, & \text{otherwise} \end{cases}$
    $\boldsymbol{s}_t \leftarrow \mathbf{0}$
    $\boldsymbol{V}_k \leftarrow \mathrm{diag}(v_{k,1}, \ldots, v_{k,d})$
  $\boldsymbol{\theta}_t \leftarrow \Pi_{\mathcal{F}, \boldsymbol{V}_k^{-1}}(\boldsymbol{\theta}_{t-1} - \eta(t)\boldsymbol{V}_k \widehat{\boldsymbol{m}}_t)$

The pseudocode of Anon is presented in Algorithm 2, and all the operations are element-wise. Here, $\widehat{\boldsymbol{m}}_t$ corresponds to $\boldsymbol{m}_t$ in Algorithm 1. $\mathbf{V}_k$ corresponds to $\mathbf{S}_t^{-1}$ in Algorithm 1. $\boldsymbol{s}_t, \boldsymbol{\sigma}_k, \boldsymbol{v}_k$, and k are intermediate variables. γ is a hyperparameter to adjust adaptivity A. ϵ is a small hyperparameter to avoid division by 0. β_1 and β_2 are hyperparameters for *EMA*, $0 \le \beta_1, \beta_2 < 1$, typically set as 0.9 and 0.999. Let $\{a_n\}$ be an increasing sequence with $a_0 = 0$, $a_1 = 1$, and $\tilde{a}_n = \sum_{i>0} \mathbb{I}_{a_i \le n}$. The pre-conditioner of Anon can be written as (5) ($\beta_3 = 0.5, a_n = 2^{n-1}$).

$$\psi_t^{\text{Anon}}(\boldsymbol{x}) = \left(\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t - j}(1 - \beta_3 \mathbb{I}_{j>1}) \cdot \mathit{EMA}^{\gamma}(\boldsymbol{x}^2_{a_{j-1}+1:a_j} + \epsilon; \beta_2) \right)^{1/2}. \tag{5}$$

Theorem 3.1. *For the optimizer Anon described in Algorithm 2, the adaptivity of Anon in i-th dimension is* $\in [\gamma(1-k), \gamma)$, *or* $(\gamma, \gamma(1-k)]$ *if* $\gamma < 0$, *where* $k = \epsilon / (\min_{j \in [\tilde{a}_t]} \mathit{EMA}(\mathbf{g}^2_{a_{j-1}+1:a_j, i}; \beta_2) + \epsilon) \in (0, 1]$.

According to Theorem 3.1, since we also set a small ϵ by default, we can adjust the adaptivity A of Anon by adjusting the hyperparameter γ ($A \approx \gamma$). The proof of Theorem 3.1 is shown in Appendix F.

3.2 How Adaptivity Influences Behaviors of Optimizers

Empirical Validations To show how adaptivity influences the behaviors of optimizers, we conduct a simple experiment in the loss function $f(x, y) = \ln(1 + \text{Beale}(x, y))/10$, where Beale (Beale, 1958) is a commonly used function to test optimizer performance. We apply appropriate learning rates for SGDM, Adam, AdaBelief, and Anon, and draw the optimization trajectories. We also show

the loss landscapes in the view of Anon by rescaling each coordinate with the corresponding diagonal entry of Anon's pre-conditioner $\boldsymbol{V}_k$ at epoch 100, so that the landscape is displayed in the geometry perceived by Anon. The trajectories and loss landscapes after scaling are shown in Figure 2.

Effect of Scaling By changing γ from 1.5 to -0.5, the adaptivity also changes from 1.5 to -0.5, as predicted by Theorem 3.1. When $\gamma = 1.5$, Anon takes a shorter path to descend along the y-axis; when $\gamma = 0.5$, the path lies between Adam and SGDM; and when $\gamma = -0.5$, Anon descends along the x-axis and arrives at the optimal point. As γ decreases, the x-axis is increasingly scaled down relative to the y-axis, so that Anon can choose the right path to reach the optimum. This implies that Anon's optimization path may differ greatly from other optimizers, helping reach a new parameter region with better performance.

The Meaning of Negative Adaptivity Positive adaptivity typically reduces step sizes for large gradients to help escape saddle points. In contrast, negative adaptivity adopts the opposite strategy by increasing step sizes when gradients are large, which enables the optimizer to escape from sharp minima. Intuitively, higher adaptivity drives the optimization toward steeper minima, whereas lower adaptivity favors flatter regions. **Thus, adaptivity influences the optimizer not only through its path but also by altering its preference for specific minima geometries.** Empirically, the optimal adaptivity varies from task to task and can fall on either side of this interval; Anon makes both directions reachable.

3.3 Incremental Delay Update

As we state in § 2.3, it is challenging to guarantee the convergence when adaptivity is allowed to take any value. So we propose a new technique *incremental delay update* (IDU), which can be seen as using a new function $U(\boldsymbol{x}; \psi^{\text{old}})$ to replace the old pre-conditioner function ψ, with smoothing factor $\beta_3 \in (0, 1)$ over delay intervals:

$$U_t(\boldsymbol{x}; \psi_t^{\text{old}}, \{a_n\}, \beta_3) = \left(\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t - j} \Big(1 - \beta_3 \mathbb{I}_{j>1}\Big) \cdot \Big(\psi^{\text{old}}_{a_j - a_{j-1}}(\boldsymbol{x}_{a_{j-1}+1:a_j})\Big)^2 \right)^{1/2}. \tag{6}$$

The **if**-block of Algorithm 2 gives the IDU recursion, where $\beta_3 = 0.5$, $a_n = 2^{n-1}$ and $\psi^{\text{old}} = \textit{EMA}^{\gamma}(\mathbf{x}^2 + \epsilon; \beta_2)$. IDU updates the pre-conditioner using accumulated gradient information **only at specific, delayed steps**. Intuitively, IDU holds the pre-conditioner fixed within geometrically growing blocks, bounding its cumulative variation by $\sum_k \sqrt{a_{k+1}} = O(\sqrt{T})$; per-step updates would instead incur an $O(T^{3/2})$ drift and break the bound. This strategy confines unpredictable oscillations within a manageable range, thereby ensuring theoretical convergence while still permitting the pre-conditioner to change non-monotonically. Theorems 3.2 (convex) and 3.4 (non-convex) establish the convergence of Anon, with proofs in Appendices G and H.

Theorem 3.2. *(Convergence analysis for online convex optimization) Let $\{\theta_t\}$ and $\{v_k\}$ be the sequence obtained by Algorithm 2, $\gamma \in \mathbb{R}$, $\beta_1 \in [0, 1)$, $\beta_2 \in [0, 1)$, $\beta_{1,t+1} \in [0, \beta_{1,t}]$, $\beta_{1,1} = \beta_1$, $\eta(t) = \frac{\eta_0}{\sqrt{t}}$, for $\forall t \in [T]$. Assume that $\|x - y\|_\infty \le D_\infty$ for $\forall x, y \in \mathcal{F}$. Suppose $f(\theta)$ is a convex function, $\|g_t\|_\infty \le G_\infty$, for $\forall t \in [T]$, $\theta \in \mathcal{F}$. Let $C_l = \min(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$, $C_u = \max(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$, where $\epsilon \in \mathbb{R}_+$ is a small constant set in Algorithm 2. The optimal point of f is denoted as θ^*. For $\{\theta_t\}$ generated by Anon, there is a bound on the regret:*

$$\sum_{t=1}^{T}[f_t(\theta_t) - f_t(\theta^*)] \le \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T} + \sum_{k=1}^{\tilde{a}_T - 1}\sqrt{a_{k+1}}\right) + \frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)} + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}$$
$$+ \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1}\sqrt{T} + \sum_{t=1}^{T-1}\frac{\beta_{1,t+1}\mathbb{I}_{\beta_{1,t+1}>\beta_{1,t}} dD_\infty^2}{2C_l\eta_{t+1}(1-\beta_1)^2} \tag{7}$$

Corollary 3.3. *Suppose $\beta_{1,t} = \beta_1\lambda^t,\ \ 0 < \lambda < 1$ in Theorem 3.2, then we have:*

$$\sum_{t=1}^{T}[f_t(\theta_t) - f_t(\theta^*)] \le \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T} + \sum_{k=1}^{\tilde{a}_T - 1}\sqrt{a_{k+1}}\right) + \frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)} + \frac{dD_\infty G_\infty \beta_1}{(1-\beta_1)(1-\lambda)} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \tag{8}$$

It implies the regret of Anon is upper-bounded by $O(\sqrt{T})$ for convex case when $a_n = 2^{n-1}$.

Theorem 3.4. *(Convergence analysis for non-convex stochastic optimization) The update of θ_t can be described as $\theta_{t+1} = \theta_t - \eta_t V_{\tilde{a}_t} m_t$, and $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$. Under the assumptions:*

- *f is differentiable and $f^* \le f \le F$. $\nabla f(x)$ is L-Lipschitz continuous, i.e. $\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|,\ \forall x, y$.*
- *The noisy gradient is unbiased and its infinity norm is bounded by N, i.e. $\mathbb{E} g_t = \nabla f(x)$, $\|g_t\|_\infty \le N$.*

The hyperparameters are set as: $\eta_t = \eta_0 t^{-p}$, $\eta_0 > 0$, $p \in (0,1)$ where the bounds are $C_l I \preceq V_{\tilde{a}_t} \preceq C_u I$, and $0 < C_l < C_u$ ($A \preceq B$ means $B - A$ is a positive semi-definite matrix). And ϵ and N ensure C_l and C_u exist. Here we consider the unconstrained case $\mathcal{F} = \mathbb{R}^d$. For sequence $\{\theta_t\}$ generated by Anon, we have:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\Big\|\nabla f(x_t)\Big\|^2 \le \frac{1}{\eta_0 C_l} T^{p-1} \cdot \Big(F - f^* \quad + K \int_1^T t^{-2p}\,\mathrm{d}t + J + K\Big), \tag{9}$$

where $x_t \equiv \theta_t$ denotes the iterates, the expectation is taken over the stochastic gradient noise, and

$$J = \frac{\beta_1^2 d N^2}{4L(1-\beta_1)^2} + \frac{3dN^2\eta_0 C_u}{1-\beta_1} \sum_{k=1}^{\tilde{a}_T} (a_k - \mathbb{I}_{k\neq 1})^{-p}, K = \left(\frac{1}{1-\beta_1} + \frac{1}{2}\right) L\eta_0^2 N^2 C_u^2 d. \tag{10}$$

Theorem 3.4 shows when $p = 0.5$ and $a_n = 2^{n-1}$, Anon has a convergence rate of $O(\ln T/\sqrt{T})$ for non-convex cases. Note that the convergence rates shown in Theorem 3.2 and Theorem 3.4 match those of mainstream adaptive optimizers up to a logarithmic factor under the strong assumption (4) or using the technique of AMSGrad, with constants that depend on γ and deteriorate exponentially in $|\gamma|$ (quantified in Appendix A). And the assumptions and boundedness conditions are standard in the literature and consistent with those adopted in previous works like Luo et al. (2019) and Zhuang et al. (2020).

Better Noise Robustness. Other convergence-guarantee techniques must enforce (4) by alternative means, so early-stage noise can greatly influence their performance and prevent them from using the latest gradient information. In contrast, IDU is, to our knowledge, the first technique that guarantees convergence while allowing (4) to be violated, offering Anon (IDU) better noise robustness and flexibility (see Appendix A for its relation to delayed/stale-update analyses). To verify this, we compare Anon (IDU) with AMSGrad. Slightly different from Table 1, AMSGrad is usually implemented in practice in the form: $\max_{i\in[t]}\{\psi_i^{\text{RMSProp}}\sqrt{1-\beta_2^i}\}/\sqrt{1-\beta_2^t}$ (we apply in experiments). Regardless of the implementation form, AMSGrad's strategy of persistently applying the max operation is highly susceptible to noise interference. We conduct empirical experiments to verify it, and the relevant function settings include:

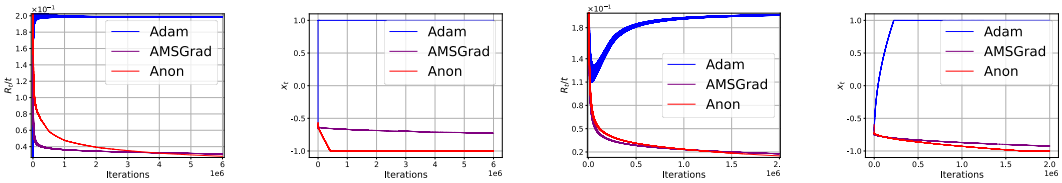
$$f_t(x) = \begin{cases} 1010x, & \text{if } t \bmod 101 = 1 \\ -10x, & \text{otherwise} \end{cases}, N_t = \begin{cases} 500/\mathrm{e}^{t-1}, & \text{if } t \bmod 2 = 1 \\ -500/\mathrm{e}^{t-1}, & \text{otherwise} \end{cases} \tag{11}$$

with the constraint set $\mathcal{F} = [-1, 1]$. The function $f_t(x)$ is the example provided in Reddi et al. (2019), which can make Adam diverge. And N_t is the noise added to the gradients g_t. We can observe that the noisy gradient is unbiased and its influence on gradients approaches 0 with the increase of t. In Figure 3, we set $\gamma = 1$ so that Anon's adaptivity matches AMSGrad and Adam with identical hyperparameters, isolating the two convergence-guarantee techniques.

In Figure 3(a)(c), the regrets divided by t of Anon and AMSGrad approach 0 (converging), while Adam's approaches a constant (diverging). Figure 3(b)(d) further shows that Anon reaches the optimal point $x = -1$ fast, whereas AMSGrad converges much slower under noise, especially when β_2 is small, proving the better noise robustness of Anon (IDU).

4 Experiments

In this section, we compare Anon with 13 baseline optimizers (SGD(M), Adam, AdamW (Loshchilov & Hutter, 2017), Yogi (Zaheer et al., 2018), AdaBound, RAdam (Liu et al., 2019), SWA (Izmailov et al., 2018), Lookahead (Zhang et al., 2019), AdaBelief, Adai (Xie et al., 2022), Lookaround (Zhang et al., 2023), Sophia (Liu et al., 2024) and HVAdam (Zhang et al., 2025)) on image classification



(a) $\beta_1 = 0.5, \beta_2 = 0.75$ (b) $\beta_1 = 0.5, \beta_2 = 0.75$ (c) $\beta_1 = 0.9, \beta_2 = 0.99$ (d) $\beta_1 = 0.9, \beta_2 = 0.99$

Figure 3: Comparison of Adam, AMSGrad, and Anon on a simple convex problem with noise. The setting of hyperparameters follows $\beta_1 < \sqrt{\beta_2}$ and $\eta(t) = 0.1/\sqrt{t}$ (Reddi et al., 2019).

with ResNet, image generation with diffusion models, and language modeling with LLMs. Our scope is element-wise (diagonal) pre-conditioners (§1): orthogonalized matrix-level methods such as Muon (Jordan et al., 2024) fall outside this framework and are compared separately in Appendix B.2. Lion (Chen et al., 2023), despite its sign-based form, is expressible in our framework with pre-conditioner $\psi_t = |\beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t|$, i.e., full adaptivity $A = 1$ (Appendix B.1), and we compare against it in Appendix B.2. Sophia-G denotes the gradient-based variant of Sophia (Liu et al., 2024) that forgoes Hessian estimates. For the diffusion experiments, all optimizers are trained by us under an identical environment and recipe. For the remaining tasks, following the standard evaluation protocol of prior optimizer studies (Luo et al., 2019; Zhuang et al., 2020; Xie et al., 2022), we source the baseline metrics from their original publications, thereby comparing Anon against each baseline at its best result tuned by the original authors. Throughout, our conclusions are grounded in controlled comparisons within each table, varying only γ or the IDU mechanism under an identical environment and recipe; a fully controlled multi-seed benchmark on CIFAR-10 (Appendix C.6) further confirms the ordering with statistical significance.

Image Classification with CNN We conduct experiments on ImageNet (Russakovsky et al., 2015) with ResNet18 and ResNet50, following the official implementations of AdaBound, AdaBelief and LookAround. Table 2 reports the top-1 accuracy for both architectures. We set a learning rate of 1 for Anon, corresponding to the 0.1 learning rate and 0.9 momentum of SGDM, since $\mathit{EMA}(\boldsymbol{x}; 0.9) \approx \mathit{M}(\boldsymbol{x}; 0.9)/10$ (1). With $\gamma = -0.1$ ($A = -0.1$), Anon surpasses SGDM ($A = 0$), confirming that negative adaptivity suits classical models like CNNs; detailed settings are in Appendix C.

Table 2: Top-1 accuracy (%) on ImageNet, with per-method differences Δ from the SGDM and Adam anchors (single run per method; controlled multi-seed results in Appendix C.6). AMSGradW denotes AMSGrad with decoupled weight decay. R18 sourcing: † Luo et al., 2019, ‡ Liu et al., 2019, ∗ Zhuang et al., 2021 (MSVAG, Balles & Hennig, 2018); R50: † Xie et al., 2022, ‡ Zhang et al., 2023, ∗ Zhang et al., 2025.

ResNet18

Method	Top-1	Δ_{SGDM}	Δ_{Adam}
Anon	**70.06**	**+0.12**	**+3.52**
SGDM	69.94	–	+3.40
AMSGradW	68.78	−1.16	+2.24
AdaBelief	69.42	−0.52	+2.88
AdaBound†	68.13	−1.81	+1.59
Yogi†	68.23	−1.71	+1.69
Adam‡	66.54	−3.40	–
MSVAG∗	65.99	−3.95	−0.55
RAdam‡	67.62	−2.32	+1.08

ResNet50

Method	Top-1	Δ_{SGDM}	Δ_{Adam}
Anon	**77.25**	**+1.02**	**+4.38**
SGDM	76.23	–	+3.36
Lookaround	76.77	+0.54	+3.90
Adam†	72.87	−3.36	–
Adai†	76.80	+0.57	+3.93
SWA‡	76.78	+0.55	+3.91
Lookahead‡	76.52	+0.29	+3.65
HVAdam∗	77.22	+0.99	+4.35

Image Generation with Diffusion Model We conduct image generation experiments on CIFAR-10 (Krizhevsky et al., 2009) with a diffusion model, searching the learning rate in $\{0.1, 0.01, ..., 0.00001\}$ for AdamW, AMSGrad, Anon, SGDM, and AdaBound. The code and other hyperparameters follow the official implementation of Nichol & Dhariwal (2021). Using a standard learning rate of 0.0001 (the optimal default for Adam-class methods) and $\gamma = 1.01$, Anon achieves a highly competitive FID of 7.75. Slight extrapolation ($\gamma > 1$) thus benefits this generative task.

Table 3: FID of diffusion models on CIFAR-10 (lower is better; single run per setting); Δ: difference from the SGDM/Adam anchor (negative = better).

Method	FID	Δ_{SGDM}	Δ_{Adam}
Adam	9.11	-3.73	–
AMSGrad	8.12	-4.72	-0.99
SGDM	12.84	–	+3.73
AdaBound	12.13	-0.71	+3.02
Anon$_{\gamma=1}$	8.03	-4.81	-1.08
Anon$_{\gamma=1.01}$	**7.75**	$\mathbf{-5.09}$	$\mathbf{-1.36}$

Table 4: Validation loss and time on OpenWebText (Anon: $\gamma = 1.1$ on GPT2-small, $\gamma = 1$ on GPT2-medium; single run per setting; Gain = AdamW $-$ Anon validation loss).

	Validation Loss ↓				Time (h) ↓		
Model	Anon	AdamW	Sophia-G	Gain	Anon	AdamW	Sophia-G
GPT2-small	**2.93283**	2.95614	2.95143	0.0233	**26.17554**	26.88118	28.98702
GPT2-medium	**2.69017**	2.70994	2.70653	0.0198	36.91487	**36.83633**	41.02486

Language Modeling We train autoregressive models on OpenWebText (Gokaslan et al., 2019) using the official Sophia implementation (Liu et al., 2024) and its exact experimental setup, with $\gamma \geq 1$ for Anon under the other optimizers' learning rate settings. As shown in Table 4, Anon obtains the lowest validation losses on both GPT2-small and GPT2-medium, demonstrating strong performance on LLM training. In contrast to many Adam variants with extra per-step overhead, Anon matches Adam's wall-clock training efficiency: IDU amortizes the per-coordinate division and square root of the pre-conditioner to $O(\log T)$ block updates, so within blocks Anon requires only element-wise multiplications; in practice this arithmetic advantage is largely masked by memory-bandwidth and kernel-launch overheads, and the measured wall-clock times of Anon and AdamW are comparable (Table 4).

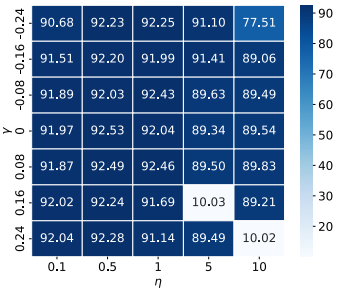


Figure 4: Hyperparameter sensitivity (ResNet20, CIFAR-10)

Comprehensive Analysis and Robustness In our experiments, the per-task optima mostly lie just outside the historical $[0, 1]$ interval; Anon reaches them, attaining the best result among all compared optimizers on every evaluated task against a different runner-up. The $\gamma = 1$ row (Adam's adaptivity) in Table 3 and the GPT2-medium row of Table 4 differ from Adam only by IDU, isolating the **improvement brought by IDU**; the further gains to the per-task optima come from adaptivity extrapolation itself, with IDU held fixed (Fig. 4; the GPT2-small anchor is in Appendix B.2).

5 Conclusion

We propose Anon, integrating tunable adaptivity with the provably stable IDU mechanism; evaluations on CNNs, diffusion, and language models establish it as a robust, competitive element-wise baseline: extrapolating the adaptivity axis just beyond the historical $[0, 1]$ interval is a small but consistently valuable extension of optimizer design, with the gains over the SGDM and Adam anchors decomposed between IDU itself and adaptivity extrapolation. IDU matches AMSGrad's convergence guarantees with better noise robustness and efficiency. In practice, γ is selected by a short directional search from the anchors, adding a single axis to the standard learning-rate budget; pairing a larger learning rate with a smaller γ (and vice versa) further improves stability. Matrix-level and higher-order extensions remain future work.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance (e.g., language polishing and grammar checking) and for code debugging support. We have not used generative AI tools for generating the core research ideas, theoretical analysis, or experimental design, and these tasks are not applicable to this work. Additionally, all AI-assisted text and code were carefully reviewed and verified for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details.

ETHICS STATEMENT

This work proposes a general-purpose optimization algorithm for training machine learning models. It does not involve human subjects, crowdsourcing, or the release of new datasets or models with a high risk for misuse, and we do not anticipate direct negative societal impacts beyond those of generic deep learning optimization research.

REPRODUCIBILITY STATEMENT

All information needed to reproduce the main experimental results is provided in the main paper and the appendix. The complete pseudo-code of Anon is given in Algorithm 2, with hyperparameter settings described in Section 4 and Appendix C. Full statements and proofs of all theoretical results are provided in Appendix E, F, G and H. Our code is available anonymously at `https://anonymous.4open.science/r/Anon-6511/`.

REFERENCES

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. *Advances in neural information processing systems*, 24, 2011.

Yang An, Baosong Yang, Binyuan Hui, Zheng Bo, Yu Bowen, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, pp. 404–413. PMLR, 2018.

Evelyn Martin Lansdowne Beale. *On an iterative method for finding a local minimum of a function of more than one variable*. Number 25. Statistical Techniques Research Group, Section of Mathematical Statistics . . . , 1958.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jinghui Chen and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In *International conference on machine learning*, pp. 7449–7479. PMLR, 2023.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019.

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

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International conference on learning representations*, volume 2024, pp. 1621–1650, 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Hugo Touvron Llama. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2407.21783*, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PmLR, 2021.

Kaan Ozkara, Can Karakus, Parameswaran Raman, Mingyi Hong, Shoham Sabach, Branislav Kveton, and Volkan Cevher. Mada: Meta-adaptive optimizers through hyper-gradient descent. *arXiv preprint arXiv:2401.08893*, 2024.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Pedro Savarese, David McAllester, Sudarshan Babu, and Michael Maire. Domain-independent dominance of adaptive methods. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16281–16290. IEEE, 2021.

Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley-benchmarking deep learning optimizers. In *International Conference on Machine Learning*, pp. 9367–9376. PMLR, 2021.

Suvrit Sra, Adams Wei Yu, Mu Li, and Alex Smola. Adadelay: Delay adaptive distributed stochastic optimization. In *Artificial Intelligence and Statistics*, pp. 957–965. PMLR, 2016.

Matthew Staib, Sashank Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. In *International Conference on Machine Learning*, pp. 5956–5965. PMLR, 2019.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

Zeke Xie, Xinrui Wang, Huishuai Zhang, Issei Sato, and Masashi Sugiyama. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International conference on machine learning*, pp. 24430–24459. PMLR, 2022.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10665–10673, 2021.

Yun Yue, Zhiling Ye, Jiadi Jiang, Yongchao Liu, and Ke Zhang. Agd: an auto-switchable optimizer using stepwise gradient difference for preconditioning matrix. *Advances in Neural Information Processing Systems*, 36:45812–45832, 2023.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Jiangtao Zhang, Shunyu Liu, Jie Song, Tongtian Zhu, Zhengqi Xu, and Mingli Song. Lookaround optimizer: k steps around, 1 step average. *Advances in Neural Information Processing Systems*, 36:28678–28698, 2023.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pp. 9593–9604, 2019.

Yiheng Zhang, Shaowu Wu, Yuanzhuo Xu, Jiajun Wu, Shang Xu, Steve Drew, and Xiaoguang Niu. Hvadam: A full-dimension adaptive optimizer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22623–22631, 2025.

Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020.

Juntang Zhuang, Yifan Ding, Tommy Tang, Nicha Dvornek, Sekhar C Tatikonda, and James Duncan. Momentum centering and asynchronous update for adaptive gradient methods. *Advances in Neural Information Processing Systems*, 34:28249–28260, 2021.

APPENDIX

A LIMITATIONS AND FUTURE WORK

While we have demonstrated that adaptivity is a critical attribute for first-order optimizers, our current Adaptivity Definition 2.1 covers the generic framework outlined in Algorithm 1. A small subset of first-order optimizers, such as HVAdam, deviate from this framework and are therefore not currently covered. We aim to propose a more generalized definition of adaptivity in future work to encompass these cases.

Additionally, due to computational resource constraints, our hyperparameter search for Anon was not exhaustive. For instance, in our diffusion model experiments, we observed that a learning rate of 0.0001 combined with an adaptivity of 1.02 hindered the reduction of training loss in the early stages. Conversely, a lower learning rate of 10^{-5} allowed for higher adaptivity values, such as 1.15. This coupling arises because the pre-conditioner scales as $(\textit{EMA}(\mathbf{g}^2) + \epsilon)^{\gamma/2}$, so increasing γ amplifies the scale compensation and effectively enlarges the step-size magnitude whenever gradient statistics deviate from unity. As a practical rule of thumb, we recommend pairing a larger learning rate with a smaller γ and a smaller learning rate with a larger γ, which we found to be more stable. Time constraints prevented a deeper investigation into these observations; thus, further research is required to fully exploit Anon's potential across different regimes.

When $\gamma < 0$, coordinates with vanishing gradients rely on ϵ to keep the pre-conditioner $(\textit{EMA}(\mathbf{g}^2) + \epsilon)^{\gamma/2}$ bounded; we use $\epsilon = 10^{-16}$, which is near the precision limit of FP32. We observed no numerical instability in any experiment, but a systematic study of the interaction between ϵ and negative γ is left to future work. We note that the condition number of the bound, C_u/C_l, grows as $(G_\infty^2/\epsilon)^{|\gamma|/2}$, so the theoretical constants deteriorate exponentially as $|\gamma|$ grows and the guarantees are most informative for moderate extrapolation, which matches the small-to-moderate $|\gamma|$ values that worked best in our experiments. In addition, the adaptivity $A \approx \gamma$ of Theorem 3.1 is state-dependent: for coordinates whose gradient statistics are close to ϵ ("dead coordinates"), the effective adaptivity degenerates toward 0 regardless of γ, so Anon behaves non-adaptively on such coordinates; we did not observe this to be a limiting factor in our experiments, but a systematic study under lower-precision (BF16) states is left to future work.

In terms of memory, Anon maintains three parameter-sized optimizer states (the same count as AMSGrad) versus two for Adam, i.e., 12 vs. 8 bytes per parameter in FP32 optimizer states. Since activations dominate end-to-end training memory in our workloads, the increase in total training memory is far below 50%.

Regarding the choice of γ, our results across all tasks provide a directional search prior: for conventional CNNs, start at $\gamma = 0$ and move to smaller values; for diffusion and language modeling, start at $\gamma = 1$ and move to larger values; if validation performance does not improve in that direction, fall back to the conventional $[0, 1]$ interval. Since β_1, β_2 and ϵ are fixed throughout all experiments, γ and the learning rate are the only searched quantities, so the added budget is one extra axis explored in a known direction rather than a sweep over $\mathbb{R}$.

Finally, we clarify the relation to lazy or stale second-moment updates: IDU differs in that the pre-conditioner is held fixed over geometrically growing blocks and the block values are combined through the β_3-EMA recurrence, which is precisely what yields the $O(\sqrt{T})$ bound on cumulative pre-conditioner variation (Theorem 3.2). A naive stale update without block structure or EMA smoothing does not admit such a bound, and our ablation (Table 9) shows the block schedule and smoothing factor both matter empirically. More broadly, IDU's notion of delay is distinct from delayed/stale *gradient* analyses (Agarwal & Duchi, 2011; Sra et al., 2016), which quantify the error of outdated gradients in asynchronous or distributed optimization: IDU delays and block-averages the *pre-conditioner* itself, replacing a monotonicity requirement with an explicit bounded-variation budget, in the spirit of cumulative step-size analyses for non-convex adaptive methods (Zhou et al., 2018; Duchi et al., 2011). Beyond tunable pre-conditioners, schedule-free training (Defazio et al., 2024) and learning-rate-free methods (Defazio & Mishchenko, 2023) relax the limitations of Adam along orthogonal axes; Anon is complementary, modifying the pre-conditioner rather than the schedule or step-size selection.

We also hope this work contributes to the broader exploration of deep learning model design. Our experiments reveal that different model architectures exhibit distinct preferences for specific adaptivity levels. This suggests that some architectural modifications previously deemed "ineffective" might simply have been mismatched with the optimizer's fixed adaptivity (typically confined to the [0, 1] range). In such scenarios, Anon's capability for extensive adaptivity tuning could potentially unlock the latent capabilities of these architectures.

B Adaptivity of Optimizers

B.1 Summary of Adaptivity

We present the detailed adaptivity expressions for the optimizers discussed in the main paper in Table 5. For Lion, the state substitution $c_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$ turns its delayed buffer refresh into the equivalent synchronous recursion $c_t = \beta_2 c_{t-1} + (\beta_1 - \beta_2)\boldsymbol{g}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$ (exact under zero initialization, $c_1 = (1-\beta_1)\boldsymbol{g}_1$), so Lion is an exact instance of Algorithm 1 with momentum c_t and pre-conditioner $\psi_t = |c_t|$ together with decoupled weight decay; since c_t is linear in the input sequence, ψ_t is 1-homogeneous and $A = 1$ exactly. For the max-based pre-conditioners (AMSGrad, Padam), the maximizing index is invariant under the positive rescaling $k > 0$, so $\psi(k\boldsymbol{x})$ rescales homogeneously in k and differentiating $\ln \psi(k\boldsymbol{x})$ at $k = 1$ yields the listed closed forms exactly, with no approximation.

Table 5: Summary of adaptivity expressions for representative optimizers.

Optimizer	$\boldsymbol{A_t(\psi, x)}$
SGD	0
SGDM	0
RMSProp	$\dfrac{1}{1+\epsilon/\sqrt{\mathit{EMA}(\boldsymbol{x}^2;\beta_2)}}$
Adam	Same as RMSProp
AMSGrad	$\dfrac{1}{1+\epsilon/\max_{i\in[t]}\sqrt{\mathit{EMA}(\boldsymbol{x}^2_{1:i};\beta_2)}}$
Padam	$\dfrac{2p}{1+\epsilon/\max_{i\in[t]}\sqrt{\mathit{EMA}(\boldsymbol{x}^2_{1:i};\beta_2)}}$
AdaBound	$\begin{cases} A_t^{\text{RMSProp}}, & \text{if } \eta_l(t) < \psi_t^{\text{RMSProp}} < \eta_u(t), \\ 0, & \text{otherwise.} \end{cases}$
AdaBelief	$\dfrac{1}{1+\epsilon\cdot\frac{\frac{1}{1-\beta_2}+\sqrt{\mathit{EMA}((\boldsymbol{x}-\boldsymbol{\phi}^{\text{Adam}})^2+\frac{\epsilon}{1-\beta_2};\beta_2)}}{\mathit{EMA}((\boldsymbol{x}-\boldsymbol{\phi}^{\text{Adam}})^2;\beta_2)}}$
Anon	Derived in (13) ($\approx \gamma$)
Lion	$1 \quad (\psi_t = \lvert\beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{x}_t\rvert)$

For Algorithm 2, we provide an equivalent formulation in Algorithm 3. While this formulation yields no speedup, it offers a clearer representation of the underlying mechanism. Note that the accumulator w_k in Algorithm 3 corresponds to $1/v_k^2$ in Algorithm 2, i.e., an arithmetic average of σ^γ over delay intervals (the debiasing term $1-\beta_2^{\max(t/2,1)}$ in Algorithm 2 equals $1-\beta_2^{2^k-a}$ at update steps $t = 2^k$); accordingly, $V_k = \operatorname{diag}(\sqrt{w_k})$ here is exactly the inverse of the update matrix V_k in Algorithm 2, so the two formulations produce identical iterates. The two algorithms trigger the pre-conditioner update at exactly the same steps $t \in \{1, 2, 4, 8, \ldots\}$, consistent with $a_n = 2^{n-1}$ ($a_0 = 0$) in Eq. (5): the conditions $k + 1 = \log_2 t$ in Algorithm 2 and $t = 2^k$ in Algorithm 3 differ only by the counter initialization ($k = -1$ vs. $k = 0$), an off-by-one in indexing rather than in the update times.

Algorithm 3 The Anon Optimizer (Equivalent Formulation)

Require: η, β_1, β_2, ϵ, γ

Initialize $\boldsymbol{\theta}_0$, $\boldsymbol{m}_0 \leftarrow \mathbf{0}$, $\boldsymbol{s}_0 \leftarrow \mathbf{0}$, $t \leftarrow 0$, $k \leftarrow 0$, $a \leftarrow 0$,
while $\boldsymbol{\theta}_t$ not converged **do**
  $t \leftarrow t + 1$
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_t)$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$
  $\widehat{\boldsymbol{m}}_t \leftarrow \frac{\boldsymbol{m}_t}{1-\beta_1^t}$
  $\boldsymbol{s}_t \leftarrow \beta_2 \boldsymbol{s}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2$
  if $t = 2^k$ **then**
    $\boldsymbol{\sigma}_k \leftarrow \boldsymbol{s}_t / (1-\beta_2^{2^k - a}) + \epsilon$
    $a \leftarrow 2^k$
    $\boldsymbol{w}_k \leftarrow \begin{cases} \frac{\boldsymbol{w}_{k-1} + \boldsymbol{\sigma}_k^\gamma}{2}, & \text{if } k > 0, \\ \boldsymbol{\sigma}_k^\gamma, & \text{otherwise} \end{cases}$
    $\boldsymbol{s}_t \leftarrow \mathbf{0}$
    $\boldsymbol{V}_k \leftarrow \mathrm{diag}(\sqrt{w_{k,1}}, \ldots, \sqrt{w_{k,d}})$
    $k \leftarrow k + 1$
  $\boldsymbol{\theta}_t \leftarrow \Pi_{\mathcal{F}, \boldsymbol{V}_{k-1}} \left(\boldsymbol{\theta}_{t-1} - \eta(t) \boldsymbol{V}_{k-1}^{-1} \widehat{\boldsymbol{m}}_t\right)$

B.2 The Effect of Adaptivity

To intuitively illustrate the impact of adaptivity, we present a visualization in Figure 5.

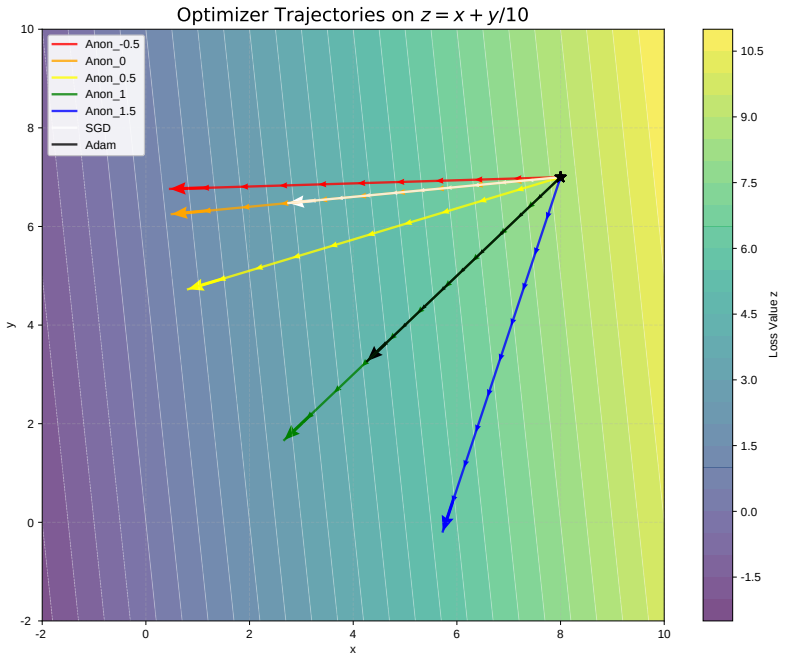


Figure 5: Optimization trajectories of SGDM, Adam, and Anon with varying γ. The gradient from yellow to purple indicates decreasing loss values. Different learning rates are applied to clearly visualize the distinct update directions.

We further demonstrate that varying adaptivity drives optimizers toward distinct regions of the parameter space by training GPT2-small on OpenWebText. The validation loss results are presented in Table 6. Additionally, we analyze the cosine similarity of the trained model parameters in Table 7. Notably, Anon with $\gamma = 1.15$ exhibits the lowest similarity when compared to Adam, Lion (Chen

et al., 2023), and Muon (Jordan et al., 2024), suggesting it discovers a unique solution (single runs; the similarities are illustrative rather than distributional).

Table 6: Validation loss on OpenWebText (GPT2-small, single run per optimizer). This measurement accompanies the solution-similarity study below and includes Lion and Muon, which are outside the element-wise scope of the main tables; it was conducted on a different hardware platform from Table 4 in the main text, so absolute values are not comparable across tables and conclusions should be drawn from within-table comparisons. Within this table, the $\gamma = 1$ anchor (2.937) sits 0.003 above Adam (2.934), a difference below the observed cross-platform variation (the same $\gamma = 1.1$ setting measures 2.933 in Table 4); $\gamma = 1.1$ improves over both anchors.

	$\text{Anon}_{\gamma=1}$	$\text{Anon}_{\gamma=1.1}$	$\text{Anon}_{\gamma=1.15}$	Adam	Lion	Muon
Loss	2.937	**2.927**	2.932	2.934	2.992	3.092

Table 7: Cosine similarity on OpenWebText. The upper number in each cell represents the cosine similarity of **all parameters**, while the lower number represents the cosine similarity of the **weights only**.

	$\text{Anon}_{\gamma=1}$	$\text{Anon}_{\gamma=1.1}$	$\text{Anon}_{\gamma=1.15}$	Adam	Lion	Muon
$\text{Anon}_{\gamma=1}$	1 1	0.597 0.317	0.355 0.201	0.914 0.511	0.857 0.161	0.901 0.194
$\text{Anon}_{\gamma=1.1}$	0.597 0.317	1 1	0.425 0.328	0.573 0.248	0.522 0.088	0.539 0.098
$\text{Anon}_{\gamma=1.15}$	0.355 0.201	0.425 0.328	1 1	0.355 0.156	0.299 0.056	0.306 0.063
Adam	0.914 0.511	0.573 0.248	0.355 0.156	1 1	0.880 0.191	0.923 0.222
Lion	0.857 0.161	0.522 0.088	0.299 0.056	0.880 0.191	1 1	0.925 0.132
Muon	0.901 0.194	0.539 0.098	0.306 0.063	0.923 0.222	0.925 0.132	1 1

C Experimental Details and Additional Results

C.1 Image Classification

ResNet18 We report results from the sources cited in the main paper. We adopted the experimental setup from the official AdaBelief implementation[1], reproducing the SGDM and AdaBelief results under their protocol; for SGDM this includes the learning-rate search over $\eta \in \{10, 1, 0.1, 0.01, 0.001\}$ with momentum 0.9 (Zhuang et al., 2020), so the SGDM row of Table 2 reports its best configuration under this sweep. For AMSGrad (with decoupled weight decay), we searched the learning rate in {0.1, 0.01, 0.001}, finding 0.01 to be optimal. For Anon, we set the learning rate to 1 and searched γ in {-0.1, -0.05, 0, 0.05}, with -0.1 yielding the best performance.

ResNet50 We report results from the sources cited in the main paper. We used the setup from the official LookAround implementation[2] and reproduced the SGDM and LookAround results under their protocol (SGDM: learning rate 0.1 with momentum 0.9) (Zhang et al., 2023). Due to computational constraints, we performed a limited search for Anon, setting $\eta = 1$ and $\gamma = -0.1$.

[1] https://github.com/juntang-zhuang/Adabelief-Optimizer

[2] https://github.com/Ardcy/Lookaround

C.2 IMAGE GENERATION

Diffusion Model We adopted the experimental setup from the official implementation[3] (Unconditional CIFAR-10 with L_hybrid objective and cosine noise schedule). We searched the learning rate in {0.1, 0.01, ..., 0.00001} for all optimizers; in particular, the SGDM result in Table 3 is its best configuration under this sweep. For Anon, we additionally searched $\gamma \in \{1, 1.1, 1.01\}$. The optimal configuration found was $\eta = 0.0001$ and $\gamma = 1.01$.

C.3 LANGUAGE MODELING

GPT2 We followed the setup in the official Sophia implementation[4][5], setting 'nproc_per_node=4' due to resource limits. We observed that applying the Sophia learning rate scheduler to AdamW on GPT2-medium resulted in lower loss, so we applied this setting to both AdamW and Anon. We set $\gamma = 1.1$ for GPT2-small and $\gamma = 1$ for GPT2-medium. All optimizers used decoupled weight decay.

C.4 FEASIBILITY STUDY ON 1B-PARAMETER LLMS: HYBRID OPTIMIZATION WITH MUON

To evaluate the compatibility of Anon with state-of-the-art matrix-level optimizers on Transformer architectures, we conduct a preliminary feasibility study training a $\sim$1B parameter language model (Qwen2 (An et al., 2024)) on the OpenWebText dataset based on Liu et al. (2025).

Recent optimizers like Muon (Jordan et al., 2024) strictly update 2D parameters via orthogonalization but structurally rely on standard element-wise optimizers (typically AdamW) as a fallback mechanism for 1D parameters (e.g., biases, normalization layers, and embeddings). Because Anon is intrinsically designed as an element-wise optimizer, it is mathematically orthogonal to Muon's 2D updates. In this experiment, we deploy Anon as a direct, drop-in replacement for the 1D AdamW fallback.

To ensure a strictly controlled comparison, we utilize the default recommended hyperparameters for the Muon+AdamW baseline. For our method, we apply a default adaptivity of $\gamma = 1.1$ for Anon without further tuning. Due to the computational constraints associated with 1B-scale training, this experiment is executed as a single run to serve as a proof-of-concept rather than a comprehensive statistical benchmark.

Table 8: Validation loss of the 1B-parameter LLM (Qwen2) trained on OpenWebText. Anon serves as the 1D parameter optimizer in the hybrid setup.

Hybrid Optimizer Configuration	**Validation Loss ↓**
Muon (2D) + AdamW (1D) [Baseline]	3.81
Muon (2D) + Anon (1D) [Ours]	**3.79**

As presented in Table 8, the hybrid Muon+Anon configuration achieves a validation loss of 3.79, compared to 3.81 for the standard Muon+AdamW baseline. These preliminary results suggest two empirical observations: (1) Anon maintains optimization stability in 1B-parameter regimes; and (2) Anon can function as a structurally compatible 1D fallback to enhance existing matrix-level optimizers.

C.5 ABLATION STUDY ON IDU HYPERPARAMETERS

We conducted an ablation study to investigate the impact of the hyperparameters $\{a_n\}$ and β_3 in IDU. Experiments were performed using ResNet20 on CIFAR-10, with results summarized in Table 9. These results demonstrate that IDU is robust to hyperparameter variations; indeed, certain configurations (e.g., $\beta_3 = 0.3, a_n = 4^{n-1}$) even outperform our default setting ($\beta_3 = 0.5, a_n = 2^{n-1}$).

[3] https://github.com/openai/improved-diffusion

[4] https://github.com/Liuhong99/Sophia

[5] https://github.com/karpathy/nanoGPT

Table 9: Ablation study on the hyperparameters $\{a_n\}$ and β_3 of IDU.

	$\beta_3 = 0.1$	$\beta_3 = 0.3$	$\beta_3 = 0.5$	$\beta_3 = 0.7$	$\beta_3 = 0.9$
$a_n = 2^{n-1}$	91.76	91.98	<u>92.42</u>	92.43	92.16
$a_n = 3^{n-1}$	92.26	92.23	92.28	92.34	92.38
$a_n = 4^{n-1}$	91.97	**92.44**	92.25	92.12	92.31

C.6 Controlled Multi-Seed Benchmark on CIFAR-10

To validate the central comparisons with statistical significance, we run a fully controlled benchmark on CIFAR-10 in a single codebase (the official AdaHessian repository (Yao et al., 2021), which the AGD paper (Yue et al., 2023) identifies for its CIFAR experiments), with three independent runs per configuration. Each baseline uses the best configuration reported by its own source for this benchmark. Anon is selected over the grid already shown in Figure 4 ($\gamma \in \{-0.24, -0.16, \ldots, 0.24\} \times \eta \in \{0.1, 0.5, 1, 5, 10\}$), choosing $(\gamma = -0.08, \eta = 1)$ for ResNet20 and $(\gamma = -0.16, \eta = 0.5)$ for ResNet32. Results are reported in Table 10.

Table 10: Top-1 accuracy (%, mean $\pm$ std over 3 runs) on CIFAR-10. Every entry except the last row is our own run in a single codebase.

Method	**ResNet20**	**ResNet32**
SGDM	92.33 ± 0.09	92.95 ± 0.04
AdamW	91.96 ± 0.15	92.62 ± 0.16
AMSGradW	92.13 ± 0.10	92.84 ± 0.19
AGD (our reproduction)	92.25 ± 0.18	93.02 ± 0.13
Anon w/o IDU	92.23 ± 0.29	93.14 ± 0.06
Anon w/ IDU	$\mathbf{92.47 \pm 0.05}$	$\mathbf{93.20 \pm 0.08}$
AGD, as reported by Yue et al. (2023)	92.35 ± 0.24	93.12 ± 0.18

Anon with IDU attains the highest mean on both architectures and the tightest spread on ResNet20 (std 0.05, against 0.09–0.29 for the others), so the ordering is consistent across seeds rather than resting on a single run. Our AGD reproduction matches the published value to within one standard deviation on both architectures (92.25 vs. 92.35; 93.02 vs. 93.12), providing direct evidence that environment differences do not materially shift cross-paper numbers on this benchmark; this supports the baseline-sourcing protocol used in Table 2.

C.7 Ablation of IDU across Operating Regimes

IDU targets a specific failure mode: persistent non-monotone decrease of the pre-conditioner, which is what breaks Adam's convergence in the construction of Reddi et al. (2019). Figure 3 identifies when this regime is entered, namely small β_2 with noisy gradients. We therefore ablate IDU in that regime first, and then across four adaptivity regimes. Throughout, "without IDU" means the pre-conditioner state is updated at every step instead of at the delayed block boundaries; everything else is unchanged, and (γ, lr) is held fixed within each row so that the IDU switch is the only difference. Three runs per cell, ResNet20 on CIFAR-10.

Table 11 reports the regime IDU is designed for: holding $\gamma = 1.1$ and $\mathrm{lr} = 0.01$ fixed while reducing β_2. IDU improves the mean at all three reduced values of β_2 ($+0.12$ to $+0.32$). Notably, with IDU the accuracy stays essentially flat as the second-moment window is shortened from 0.99 to 0.75 ($91.84/91.79/91.89$), whereas without IDU it sits 0.1–0.3 points lower at every value.

Table 11: Top-1 accuracy (%, mean $\pm$ std over 3 runs) with and without IDU at reduced β_2 ($\gamma = 1.1$, lr $= 0.01$, ResNet20).

$(\gamma, \beta_2, \text{lr})$	w/o IDU	w/ IDU	diff
$(1.1,\ 0.99,\ 0.01)$	91.58 ± 0.01	91.84 ± 0.19	$+0.26$
$(1.1,\ 0.90,\ 0.01)$	91.67 ± 0.04	91.79 ± 0.31	$+0.12$
$(1.1,\ 0.75,\ 0.01)$	91.57 ± 0.16	91.89 ± 0.12	$+0.32$

Table 12 reports the same ablation at the standard $\beta_2 = 0.999$ over four fixed diagnostic points spanning the negative, interpolation, Adam-endpoint and super-adaptive regimes. Here the differences are small and mixed in sign: none of the four settings encountered the worst case that IDU exists to prevent, namely a divergence or a persistent non-monotone collapse of the pre-conditioner, so IDU has nothing to correct at these points. As an implementation check, Anon at $\gamma = 1$ without IDU is algorithmically AdamW and obtains 92.04 ± 0.14, consistent with the 91.96 ± 0.15 we measure for AdamW under this protocol.

Table 12: Top-1 accuracy (%, mean $\pm$ std over 3 runs) with and without IDU at $\beta_2 = 0.999$ across adaptivity regimes (ResNet20).

(γ, lr)	w/o IDU	w/ IDU	diff
$(-0.1,\ 1)$	92.10 ± 0.09	92.15 ± 0.18	$+0.05$
$(0.5,\ 0.1)$	92.17 ± 0.13	92.07 ± 0.11	-0.10
$(1.0,\ 0.01)$	92.04 ± 0.14	92.20 ± 0.23	$+0.16$
$(1.1,\ 0.01)$	91.91 ± 0.23	91.87 ± 0.05	-0.04

The same regime-dependence is characteristic of other convergence-protection mechanisms: under this protocol AMSGradW differs from AdamW by $+0.17/+0.22$ on ResNet20/32, while on the diffusion task (Table 3) AMSGrad improves over Adam by 0.99 FID. IDU is therefore best understood as a safeguard whose benefit becomes visible when its protected failure mode is approached, and as the mechanism that makes the whole range $\gamma \in \mathbb{R}$ usable with a convergence guarantee, rather than as an accuracy improvement in every regime. Since the protected pathology already exists at $\gamma = 1$, IDU is also applicable as a stabilization mechanism for Adam-type optimizers beyond Anon.

A related remark concerns the tail of the block schedule: with $a_n = 2^{n-1}$, the final block spans roughly the last half of training, during which the pre-conditioner is frozen. In all our experiments this freeze coincides with cosine annealing of the learning rate, so the frozen pre-conditioner acts on steps whose magnitude is annealed toward zero; refining the final block (e.g., switching to per-step pre-conditioner updates for the last epochs) is a possible extension that we leave to future work.

C.8 Training Dynamics on GPT-2

Table 13 reports the GPT-2 Small validation loss at 20k-step intervals through the end of training. From the first reported checkpoint onward the ordering is unchanged, with no crossings, and the margin stays between 0.022 and 0.035 over AdamW and between 0.019 and 0.047 over Sophia-G. The final value is thus the endpoint of a separation established well before the end of training, rather than a lead built in the closing checkpoints. The same ordering also holds for the training loss and across γ: at every reported checkpoint, both the training and the validation loss are ordered $\gamma = 1.1 < \gamma = 1.0 < \gamma = 0.9$, with smoothly decreasing trajectories, indicating that the validation gains of larger γ are consistent with the optimization behavior during training rather than arising from a late validation-curve crossing.

Table 13: GPT-2 Small validation loss on OpenWebText at 20k-step intervals.

Steps	**Anon** ($\gamma = 1.1$)	**AdamW**	**Sophia-G**
20k	3.19062	3.22548	3.23539
40k	3.08144	3.11093	3.12880
60k	3.01662	3.04006	3.04837
80k	2.96273	2.98518	2.99619
100k	**2.93283**	2.95614	2.95143

D BENCHMARK FUNCTION VISUALIZATION

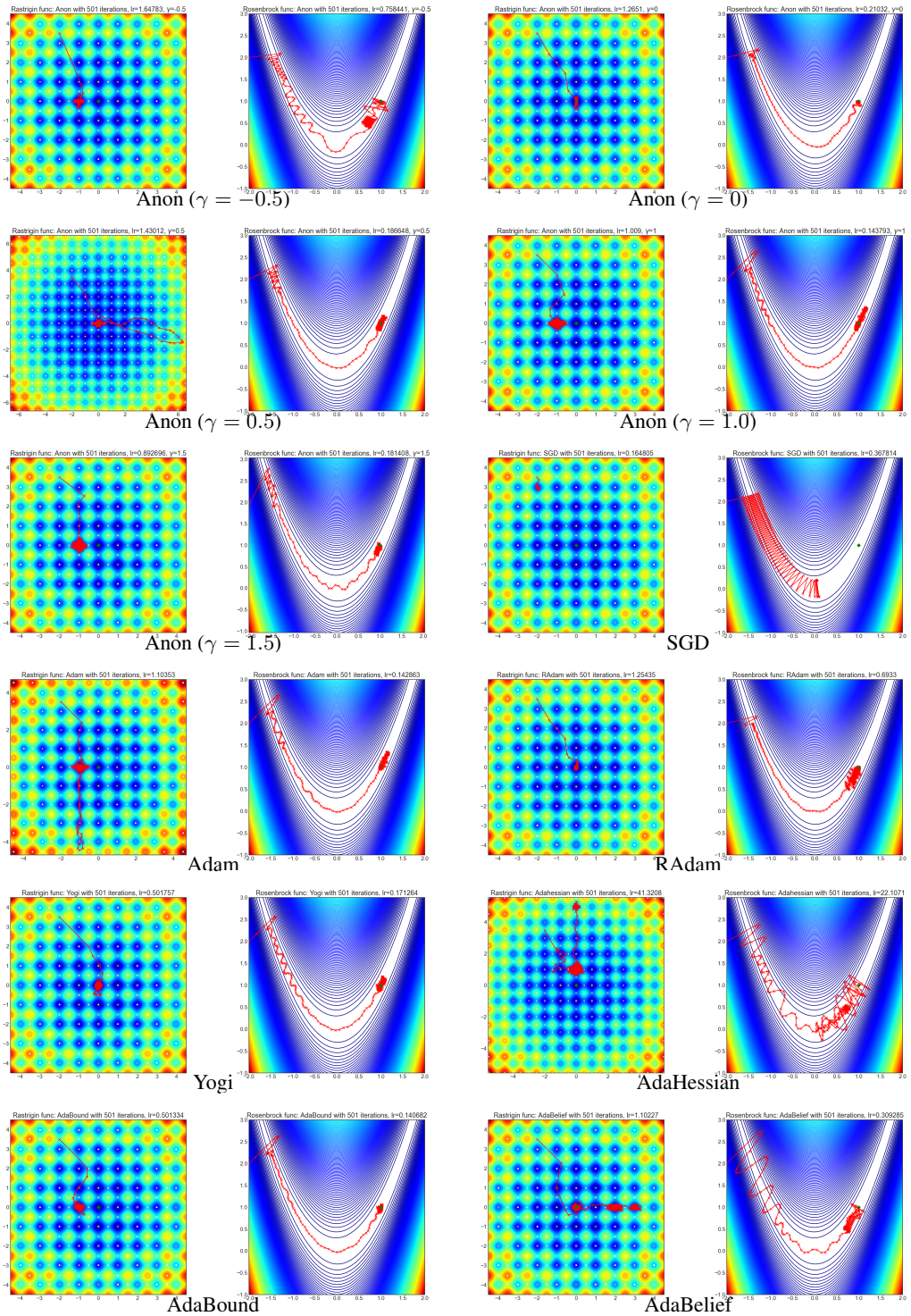
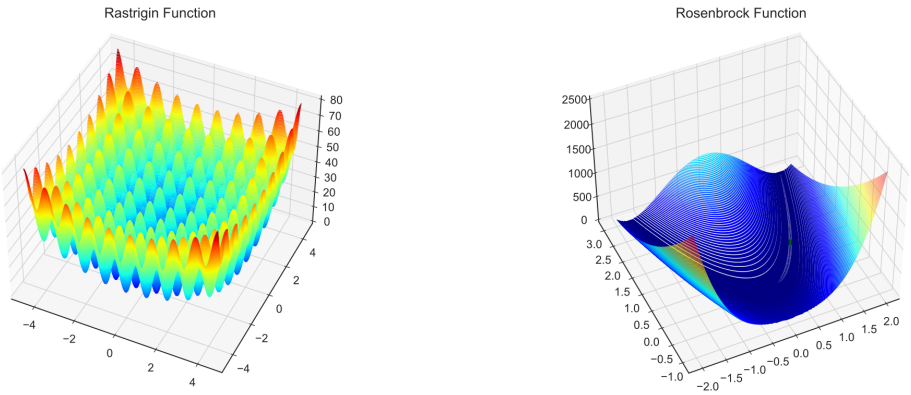


Figure 6: Optimization trajectory comparison using searched hyperparameters. The grid shows Rastrigin (columns 1, 3) and Rosenbrock (columns 2, 4) functions. Anon ($\gamma \leq 0$) tends to explore flatter regions, while Anon ($\gamma \geq 1$) and adaptive methods converge quickly to sharp minima.

To better understand optimizer behavior in complex landscapes, we visualize their trajectories on two classical benchmark functions: Rosenbrock and Rastrigin. Rosenbrock tests the ability to follow narrow, curved valleys to a global minimum at (1, 1), evaluating stability and curvature sensitivity. Rastrigin challenges optimizers with a rugged landscape filled with deceptive local minima, with the global minimum at (0, 0).



(a) Rastrigin Function (b) Rosenbrock Function

Figure 7: 3D visualization of the benchmark functions.

E THEOREM 1 IN MAIN PAPER

Theorem E.1. *If ψ and ψ' are from the same equivalence class, there is a function $f : \mathbb{N}_+ \to \mathbb{R}_+$ that makes $\psi_n(\boldsymbol{x}_{1:n}) = \psi'_n(\boldsymbol{x}_{1:n})f(n)$ for any $\boldsymbol{x}_{1:n} \in \mathbb{R}^n$ and any $n \in \mathbb{N}_+$.*

Proof. Let $h(k; \mathbf{g}_{1:n}) = \ln \psi_n(k\mathbf{g}_{1:n}) - \ln \psi'_n(k\mathbf{g}_{1:n}), h : \mathbb{R} \to \mathbb{R}$. Because ψ_n and ψ'_n are continuous, h is continuous.

When $k \neq 0$, we have

$$
\begin{aligned}
h^{'}(k; \mathbf{x}_{1:n}) =& \lim_{\Delta k \to 0} \frac{\ln \psi_n((k+\Delta k)\mathbf{x}_{1:n}) - \ln \psi'_n((k+\Delta k)\mathbf{x}_{1:n}) - \ln \psi_n(k\mathbf{x}_{1:n}) + \ln \psi'_n(k\mathbf{x}_{1:n})}{\Delta k} \\
=& \frac{1}{k} \lim_{\Delta k \to 0} \frac{\ln \psi_n((1+\Delta k/k)k\mathbf{x}_{1:n}) - \ln \psi'_n((1+\Delta k/k)k\mathbf{x}_{1:n}) - \ln \psi_n(k\mathbf{x}_{1:n}) + \ln \psi'_n(k\mathbf{x}_{1:n})}{\Delta k/k} \\
=& \frac{1}{k} \lim_{\Delta k \to 0} \frac{[\ln \psi_n((1+\Delta k/k)k\mathbf{x}_{1:n}) - \ln \psi_n(k\mathbf{x}_{1:n})] - [\ln \psi'_n((1+\Delta k/k)k\mathbf{x}_{1:n}) - \ln \psi'_n(k\mathbf{x}_{1:n})]}{\Delta k/k} \\
=& \frac{1}{k}\Big[A_n(\psi, \mathbf{x}_{1:n}) - A_n(\psi', \mathbf{x}_{1:n})\Big] \\
=& \frac{1}{k} \cdot 0 \quad \Big(Since\ \psi\ and\ \psi'\ are\ in\ the\ same\ class\Big) \\
=& 0
\end{aligned} \tag{12}
$$

So $h(k; \mathbf{x}_{1:n}) = C_1$ when $k > 0$, $h(k; \mathbf{x}_{1:n}) = C_2$ when $k < 0$. And because h is continuous, we have $C_1 = C_2 = h(0; \mathbf{x}_{1:n}) = \ln \frac{\psi_n(0)}{\psi'_n(0)}$.

Therefore, we have $\frac{\psi_n(k\mathbf{x}_{1:n})}{\psi'_n(k\mathbf{x}_{1:n})} = \frac{\psi_n(0)}{\psi'_n(0)}$ for $\forall k \in \mathbb{R}$.

And since $\mathbf{x}_{1:n}$ can be any vector $\in \mathbb{R}^n$ and any $n \in \mathbb{N}_+$, we have $\frac{\psi_n(\mathbf{x}_{1:n})}{\psi'_n(\mathbf{x}_{1:n})} = \frac{\psi_n(0)}{\psi'_n(0)}$ for $\forall \mathbf{x}_{1:n} \in \mathbb{R}^n$, $\forall n \in \mathbb{N}_+$.

Let $f(n) = \frac{\psi_n(0)}{\psi'_n(0)}$, we have $\psi_n(\boldsymbol{x}_{1:n}) = \psi'_n(\boldsymbol{x}_{1:n})f(n)$ for any $\boldsymbol{x}_{1:n} \in \mathbb{R}^n$ and any $n \in \mathbb{N}_+$. □

F THEOREM 2 IN MAIN PAPER

Theorem F.1. *For the optimizer Anon described in Algorithm 2, the adaptivity of Anon in i-th dimension is* $\in [\gamma(1-k), \gamma)$*, or* $(\gamma, \gamma(1-k)]$ *if* $\gamma < 0$*, where* $k = \epsilon/(\min_{j\in[\tilde{a}_t]} \textit{EMA}(\mathbf{g}^2_{a_{j-1}+1:a_j,i}; \beta_2) + \epsilon) \in (0, 1]$.

Proof. We let $f_{n,\gamma}(\boldsymbol{x}) = \beta_3^{-n}(1-\beta_3\mathbb{I}_{n>1})\textit{EMA}^\gamma(\boldsymbol{x}^2_{a_{n-1}+1:a_n} + \epsilon; \beta_2)$, so we have

$$
\begin{aligned}
A(\psi, \boldsymbol{g}_{1:t,i}) &= \nabla_k \ln \left(\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(k\boldsymbol{g}_{1:t,i}) \right)^{1/2} \Bigg|_{k=1} \\
&= \frac{\gamma \sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma-1}(\boldsymbol{g}_{1:t,i}) \textit{EMA}(\boldsymbol{g}^2_{a_{j-1}+1:a_j,i}; \beta_2)}{\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i})} \\
&= \frac{\gamma \sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma-1}(\boldsymbol{g}_{1:t,i}) [\textit{EMA}(\boldsymbol{g}^2_{a_{j-1}+1:a_j,i} + \epsilon; \beta_2) - \epsilon]}{\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i})} \\
&= \frac{\gamma \sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i}) - \gamma\epsilon \sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma-1}(\boldsymbol{g}_{1:t,i})}{\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i})} \\
&= \gamma \left(1 - \epsilon \cdot \frac{\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma-1}(\boldsymbol{g}_{1:t,i})}{\sum_{j=1}^{\tilde{a}_t} \beta_3^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i})} \right) \\
&= \gamma \left(1 - \epsilon \cdot \frac{\sum_{j=1}^{\tilde{a}_t} f_{j,\gamma-1}(\boldsymbol{g}_{1:t,i})}{\sum_{j=1}^{\tilde{a}_t} f_{j,\gamma}(\boldsymbol{g}_{1:t,i})} \right)
\end{aligned} \tag{13}
$$

Since $k = \epsilon/(\min_{j\in[\tilde{a}_t]} \textit{EMA}(\mathbf{g}^2_{a_{j-1}+1:a_j,i}; \beta_2) + \epsilon) \in (0, 1]$, we have $A(\psi, \boldsymbol{g}_{1:t,i}) \in [\gamma(1-k), \gamma)$, or $(\gamma, \gamma(1-k)]$ if $\gamma < 0$. □

G THEOREM 3 IN MAIN PAPER

For simplicity, we omit the debiasing step in theoretical analysis as in Reddi et al. (2019). It is easy to prove that the analysis also applies to the de-biased version.

Lemma G.1. *(McMahan & Streeter, 2010) For any* $Q \in S_+^d$ *and convex feasible set* $\mathcal{F} \subset \mathbb{R}^d$*, suppose* $u_1 = \min_{x\in\mathcal{F}} \left\| Q^{1/2}(x - z_1) \right\|$ *and* $u_2 = \min_{x\in\mathcal{F}} \left\| Q^{1/2}(x - z_2) \right\|$*, then we have* $\left\| Q^{1/2}(u_1 - u_2) \right\| \leq \left\| Q^{1/2}(z_1 - z_2) \right\|$.

Theorem G.2. *(Convergence analysis for online convex optimization) Let* $\{\theta_t\}$ *and* $\{v_k\}$ *be the sequence obtained by Algorithm 2,* $\gamma \in \mathbb{R}$*,* $\beta_1 \in [0, 1)$*,* $\beta_2 \in [0, 1)$*,* $\beta_{1,t+1} \in [0, \beta_{1,t}]$*,* $\beta_{1,1} = \beta_1$*,* $\eta(t) = \frac{\eta_0}{\sqrt{t}}$*, for* $\forall t \in [T]$*. Assume that* $\|x - y\|_\infty \leq D_\infty$ *for* $\forall x, y \in \mathcal{F}$*. Suppose* $f(\theta)$ *is a convex function,* $\|g_t\|_\infty \leq G_\infty$ *, for* $\forall t \in [T]$*,* $\theta \in \mathcal{F}$*. Let* $C_l = \min(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$*,* $C_u = \max(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$*, where* $\epsilon \in \mathbb{R}_+$ *is a small constant set in Algorithm 2. The optimal point of* f *is denoted as* θ^**. For* $\{\theta_t\}$ *generated by Anon, there is a bound on the regret:*

$$
\begin{aligned}
\sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)] \leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0} \left(\sqrt{T} + \sum_{k=1}^{\tilde{a}_T - 1} \sqrt{a_{k+1}} \right) + \sum_{t=1}^{T-1} \left[\frac{\beta_{1,t+1}\mathbb{I}_{\beta_{1,t+1}>\beta_{1,t}} dD_\infty^2}{2C_l\eta_{t+1}(1-\beta_1)^2} \right] \\
& + \frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)} + \frac{dD_\infty G_\infty}{1-\beta_1} \sum_{t=1}^{T} \beta_{1,t} + \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1} \sqrt{T}
\end{aligned}
$$

Proof.

$$
\boldsymbol{v}_k = \sqrt{2/(\frac{1}{\boldsymbol{v}_{k-1}^2} + \boldsymbol{\sigma}_k^\gamma)} \text{ if } k > 0 \text{ else } \boldsymbol{\sigma}_k^{-\gamma/2}
$$

$$
\begin{aligned}
\frac{1}{\boldsymbol{v}_k^2} &= \frac{\frac{1}{\boldsymbol{v}_{k-1}^2} + \boldsymbol{\sigma}_k^\gamma}{2} \text{ if } k > 0 \text{ else } \boldsymbol{\sigma}_k^\gamma \\
\frac{1}{\boldsymbol{v}_k^2} &= \sum_{i=0}^{k} \frac{\boldsymbol{\sigma}_i^\gamma}{2^{\min(k-i+1,k)}} \\
\frac{1}{\boldsymbol{v}_k^2} &= \sum_{i=0}^{k} \frac{\boldsymbol{EMA}^\gamma(\boldsymbol{g}^2_{\lfloor 2^{k-1}+1 \rfloor:2^k} + \epsilon; \beta_2)}{2^{\min(k-i+1,k)}}
\end{aligned} \tag{14}
$$

Since $\|g_t\|_\infty \leq G_\infty$, $C_l = \min(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$ and $C_u = \max(G_\infty^{-\gamma}, \epsilon^{-\gamma/2})$, from 14, we have:

$$
\begin{aligned}
\frac{1}{v_{k,i}^2} &\in \left[\sum_{i=0}^{k} \frac{C_u^{-2}}{2^{\min(k-i+1,k)}}, \sum_{i=0}^{k} \frac{C_l^{-2}}{2^{\min(k-i+1,k)}} \right] \\
\frac{1}{v_{k,i}^2} &\in \left[C_u^{-2}, C_l^{-2} \right] \\
v_{k,i} &\in [C_l, C_u]
\end{aligned} \tag{15}
$$

Let $\eta_t = \eta(t)$.

$$
\theta_{t+1} = \prod_{\mathcal{F}, V_{\tilde{a}_t}^{-1}} (\theta_t - \eta_t V_{\tilde{a}_t} m_t) = \min_{\theta \in \mathcal{F}} \left\| V_{\tilde{a}_t}^{-1/2} (\theta - (\theta_t - \eta_t V_{\tilde{a}_t} m_t)) \right\|
$$

Note that $\prod_{\mathcal{F}, V_{\tilde{a}_t}^{-1}}(\theta^*) = \theta^*$ since $\theta^* \in \mathcal{F}$. Use θ_i^* and $\theta_{t,i}$ to denote the i-th dimension of θ^* and θ_t respectively. From lemma (G.1), using $u_1 = \theta_{t+1}$ and $u_2 = \theta^*$, we have:

$$
\begin{aligned}
\left\| V_{\tilde{a}_t}^{-1/2} (\theta_{t+1} - \theta^*) \right\|^2 \leq & \left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \eta_t V_{\tilde{a}_t} m_t - \theta^*) \right\|^2 \\
= & \left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 + \eta_t^2 \left\| V_{\tilde{a}_t}^{1/2} m_t \right\|^2 - 2\eta_t \langle m_t, \theta_t - \theta^* \rangle \\
= & \left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 + \eta_t^2 \left\| V_{\tilde{a}_t}^{1/2} m_t \right\|^2 \\
& - 2\eta_t \langle \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t, \theta_t - \theta^* \rangle
\end{aligned} \tag{16}
$$

Note that $\beta_1 \in [0, 1)$ and $\beta_2 \in [0, 1)$, rearranging inequality (16), we have:

$$
\begin{aligned}
\langle g_t, \theta_t - \theta^* \rangle \leq & \frac{1}{2\eta_t (1 - \beta_{1,t})} \left(\left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 - \left\| V_{\tilde{a}_t}^{-1/2} (\theta_{t+1} - \theta^*) \right\|^2 \right) \\
& + \frac{\eta_t}{2(1 - \beta_{1,t})} \left\| V_{\tilde{a}_t}^{1/2} m_t \right\|^2 + \frac{\beta_{1,t}}{1 - \beta_{1,t}} \langle m_{t-1}, \theta^* - \theta_t \rangle \\
\leq & \frac{1}{2\eta_t (1 - \beta_{1,t})} \left(\left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 - \left\| V_{\tilde{a}_t}^{-1/2} (\theta_{t+1} - \theta^*) \right\|^2 \right) \\
& + \frac{\eta_t}{2(1 - \beta_{1,t})} \left\| V_{\tilde{a}_t}^{1/2} m_t \right\|^2 + \frac{\beta_{1,t}}{1 - \beta_{1,t}} \left\| m_{t-1} \right\| \left\| \theta^* - \theta_t \right\| \\
& \left(\textit{Cauchy-Schwartz's inequality: } \langle u, v \rangle \leq \left\| u \right\| \left\| v \right\| \right) \\
\leq & \frac{1}{2\eta_t (1 - \beta_{1,t})} \left(\left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 - \left\| V_{\tilde{a}_t}^{-1/2} (\theta_{t+1} - \theta^*) \right\|^2 \right) \\
& + \frac{\eta_t}{2(1 - \beta_{1,t})} \left\| V_{\tilde{a}_t}^{1/2} m_t \right\|^2 + \frac{\beta_{1,t}}{1 - \beta_{1,t}} \left\| m_{t-1} \right\| \sqrt{d} D_\infty \\
& \left(\textit{Since } \|x - y\|_\infty \leq D_\infty, \textit{for } \forall x, y \in \mathcal{F} \right) \\
= & \frac{1}{2\eta_t (1 - \beta_{1,t})} \left(\left\| V_{\tilde{a}_t}^{-1/2} (\theta_t - \theta^*) \right\|^2 - \left\| V_{\tilde{a}_t}^{-1/2} (\theta_{t+1} - \theta^*) \right\|^2 \right)
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\eta_t}{2(1-\beta_{1,t})}\left\|V_{\tilde{a}_t}^{1/2}m_t\right\|^2+\frac{\beta_{1,t}\sqrt{d}D_\infty}{1-\beta_{1,t}}\sqrt{\sum_{i=1}^{d}EMA^2(g_{1:t-1,i};\beta_1)}\\
\leq&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\eta_t}{2(1-\beta_{1,t})}\left\|V_{\tilde{a}_t}^{1/2}m_t\right\|^2+\frac{\beta_{1,t}\sqrt{d}D_\infty}{1-\beta_{1,t}}\sqrt{\sum_{i=1}^{d}G_\infty^2}\\
&\Big(\textit{Since } \|g_t\|_\infty\leq G_\infty\Big)\\
\leq&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\eta_t}{2(1-\beta_{1,t})}\left\|V_{\tilde{a}_t}^{1/2}m_t\right\|^2+\frac{\beta_{1,t}dD_\infty}{1-\beta_{1,t}}G_\infty\\
=&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\beta_{1,t}dD_\infty G_\infty}{1-\beta_{1,t}}+\frac{\eta_t}{2(1-\beta_{1,t})}m_t^\top V_{\tilde{a}_t}m_t\\
=&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\beta_{1,t}dD_\infty G_\infty}{1-\beta_{1,t}}+\frac{\eta_t}{2(1-\beta_{1,t})}\sum_{i=1}^{d}m_{t,i}^2v_{\tilde{a}_t,i}\\
\leq&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\beta_{1,t}dD_\infty G_\infty}{1-\beta_{1,t}}+\frac{\eta_t}{2(1-\beta_{1,t})}\sum_{i=1}^{d}m_{t,i}^2C_u\\
&\Big(\textit{Apply formula } (15)\Big)\\
\leq&\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\\
&+\frac{\beta_{1,t}dD_\infty G_\infty}{1-\beta_{1,t}}+\frac{dG_\infty^2C_u\eta_t}{2(1-\beta_{1,t})}
\end{aligned}
\tag{17}
$$

By convexity of f, we have:

$$
\begin{aligned}
\sum_{t=1}^{T}f_t(\theta_t)-f_t(\theta^*)\leq&\sum_{t=1}^{T}\langle g_t,\theta_t-\theta^*\rangle\\
\leq&\sum_{t=1}^{T}\left[\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\right.\\
&\left.+\frac{\beta_{1,t}dD_\infty G_\infty}{1-\beta_{1,t}}+\frac{dG_\infty^2C_u\eta_t}{2(1-\beta_{1,t})}\right]\\
&\Big(\textit{By formula } (17)\Big)\\
\leq&\sum_{t=1}^{T}\left[\frac{1}{2\eta_t(1-\beta_{1,t})}\left(\left\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\right\|^2-\left\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right)\right]\\
&+\frac{1}{1-\beta_1}\sum_{t=1}^{T}\left(\beta_{1,t}dD_\infty G_\infty+\frac{dG_\infty^2C_u\eta_t}{2}\right)
\end{aligned}
$$

$$
\begin{aligned}
&\Big(\textit{Since } 0 \le \beta_{1,t} \le \beta_1 < 1\Big) \\
=&\sum_{t=1}^{T}\left[\frac{1}{2\eta_t(1-\beta_{1,t})}\Big(\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\Big\|^2-\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\Big)\right] \\
&+\frac{1}{1-\beta_1}\sum_{t=1}^{T}\left(\beta_{1,t}dD_\infty G_\infty+\frac{dG_\infty^2C_u\eta_0}{2\sqrt{t}}\right) \\
\le&\sum_{t=1}^{T}\left[\frac{1}{2\eta_t(1-\beta_{1,t})}\Big(\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\Big\|^2-\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\Big)\right] \\
&+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\int_0^T\frac{1}{2\sqrt{t}}\,\mathrm{d}t \\
&\Big(\textit{Since } \eta_t=\eta_0/\sqrt{t}\Big) \\
=&\sum_{t=1}^{T}\left[\frac{1}{2\eta_t(1-\beta_{1,t})}\Big(\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_t-\theta^*)\Big\|^2-\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\Big)\right] \\
&+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\le&\sum_{t=1}^{T-1}\left[\frac{1}{2\eta_{t+1}(1-\beta_{1,t+1})}\Big\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2-\frac{1}{2\eta_t(1-\beta_{1,t})}\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\right] \\
&+\frac{1}{2\eta_1(1-\beta_1)}\Big\|V_1^{-1/2}(\theta_1-\theta^*)\Big\|^2+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\sqrt{T} \\
=&\sum_{t=1}^{T-1}\left[\frac{1}{2\eta_{t+1}(1-\beta_{1,t})}\Big\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2-\frac{1}{2\eta_t(1-\beta_{1,t})}\Big\|V_{\tilde{a}_t}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\right. \\
&\left.+\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\Big\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\right] \\
&+\frac{1}{2\eta_1(1-\beta_1)}\Big\|V_1^{-1/2}(\theta_1-\theta^*)\Big\|^2+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\sqrt{T} \\
=&\sum_{t=1}^{T-1}\left\{\frac{1}{2(1-\beta_{1,t})}\left[(\theta_{t+1}-\theta^*)^\top\left(\frac{V_{\tilde{a}_{t+1}}^{-1}}{\eta_{t+1}}-\frac{V_{\tilde{a}_t}^{-1}}{\eta_t}\right)(\theta_{t+1}-\theta^*)\right]\right\} \\
&+\sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\Big\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\right] \\
&+\frac{1}{2\eta_1(1-\beta_{1,t})}\Big\|V_1^{-1/2}(\theta_1-\theta^*)\Big\|^2+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\sqrt{T} \\
=&\sum_{k=1}^{\tilde{a}_T}\sum_{t=a_k}^{\min(T,a_{k+1})-1}\left\{\frac{1}{2(1-\beta_{1,t})}\left[(\theta_{t+1}-\theta^*)^\top\left(\frac{V_{\tilde{a}_{t+1}}^{-1}}{\eta_{t+1}}-\frac{V_{\tilde{a}_t}^{-1}}{\eta_t}\right)(\theta_{t+1}-\theta^*)\right]\right\} \\
&+\sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\Big\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\Big\|^2\right] \\
&+\frac{1}{2\eta_1(1-\beta_1)}\Big\|V_1^{-1/2}(\theta_1-\theta^*)\Big\|^2+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2C_u\eta_0}{1-\beta_1}\sqrt{T}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{k=1}^{\tilde{a}_T} \sum_{t=a_k}^{\min(T,a_{k+1})-2} \left\{ \frac{1}{2(1-\beta_{1,t})} \left[(\theta_{t+1}-\theta^*)^\top \left(\frac{V_{\tilde{a}_{t+1}}^{-1}}{\eta_{t+1}} - \frac{V_{\tilde{a}_t}^{-1}}{\eta_t} \right) (\theta_{t+1}-\theta^*) \right] \right\} \\
&\quad + \sum_{k=1}^{\tilde{a}_T-1} \left\{ \frac{1}{2(1-\beta_{1a_{k+1}-1})} \left[(\theta_{a_{k+1}}-\theta^*)^\top \left(\frac{V_{k+1}^{-1}}{\eta_{a_{k+1}}} - \frac{V_k^{-1}}{\eta_{a_{k+1}-1}} \right) (\theta_{a_{k+1}}-\theta^*) \right] \right\} \\
&\quad + \sum_{t=1}^{T-1} \left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})} \left\| V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*) \right\|^2 \right] \\
&\quad + \frac{1}{2\eta_1(1-\beta_1)} \left\| V_1^{-1/2}(\theta_1-\theta^*) \right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1} \sum_{t=1}^{T} \beta_{1,t} + \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1} \sqrt{T} \\
&= \sum_{k=1}^{\tilde{a}_T} \sum_{t=a_k}^{\min(T,a_{k+1})-2} \left\{ \frac{1}{2(1-\beta_{1,t})} \left[(\theta_{t+1}-\theta^*)^\top \left(\frac{V_k^{-1}}{\eta_{t+1}} - \frac{V_k^{-1}}{\eta_t} \right) (\theta_{t+1}-\theta^*) \right] \right\} \\
&\quad + \sum_{k=1}^{\tilde{a}_T-1} \left\{ \frac{1}{2(1-\beta_{1a_{k+1}-1})} \left[(\theta_{a_{k+1}}-\theta^*)^\top \left(\frac{V_{k+1}^{-1}}{\eta_{a_{k+1}}} - \frac{V_k^{-1}}{\eta_{a_{k+1}-1}} \right) (\theta_{a_{k+1}}-\theta^*) \right] \right\} \\
&\quad + \sum_{t=1}^{T-1} \left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})} \left\| V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*) \right\|^2 \right] \\
&\quad + \frac{1}{2\eta_1(1-\beta_1)} \left\| V_1^{-1/2}(\theta_1-\theta^*) \right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1} \sum_{t=1}^{T} \beta_{1,t} + \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1} \sqrt{T} \\
&\le \sum_{k=1}^{\tilde{a}_T} \sum_{t=a_k}^{\min(T,a_{k+1})-2} \left\{ \frac{1}{2(1-\beta_1)} \left[D_\infty e_d^\top \left(\frac{V_k^{-1}}{\eta_{t+1}} - \frac{V_k^{-1}}{\eta_t} \right) D_\infty e_d \right] \right\} \\
&\quad + \sum_{k=1}^{\tilde{a}_T-1} \left\{ \frac{1}{2(1-\beta_1)} \left[D_\infty e_d^\top \left(\frac{C_l^{-1}\mathbf{I}_d}{\eta_{a_{k+1}}} \right) D_\infty e_d \right] \right\} \\
&\quad + \sum_{t=1}^{T-1} \left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})} \left\| V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*) \right\|^2 \right] \\
&\quad + \frac{1}{2\eta_1(1-\beta_1)} \left\| V_1^{-1/2}(\theta_1-\theta^*) \right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1} \sum_{t=1}^{T} \beta_{1,t} + \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1} \sqrt{T} \\
&\quad \Big(\textit{Since } \eta_t = \eta_0/\sqrt{t}, \textit{ and } 0 \le \beta_{1,t} \le \beta_1 < 1 \Big) \\
&= \sum_{k=1}^{\tilde{a}_T-1} \left\{ \frac{1}{2(1-\beta_1)} \left[D_\infty e_d^\top \left(\frac{V_k^{-1}}{\eta_{a_{k+1}-1}} - \frac{V_k^{-1}}{\eta_{a_k}} \right) D_\infty e_d \right] \right\} \\
&\quad + \frac{1}{2(1-\beta_1)} \left[D_\infty e_d^\top \left(\frac{V_{\tilde{a}_T}^{-1}}{\eta_T} - \frac{V_{\tilde{a}_T}^{-1}}{\eta_{a_{\tilde{a}_T}}} \right) D_\infty e_d \right] \\
&\quad + \sum_{k=1}^{\tilde{a}_T-1} \left\{ \frac{1}{2(1-\beta_1)} \left[D_\infty e_d^\top \left(\frac{C_l^{-1}\mathbf{I}_d}{\eta_{a_{k+1}}} \right) D_\infty e_d \right] \right\} \\
&\quad + \sum_{t=1}^{T-1} \left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})} \left\| V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*) \right\|^2 \right] \\
&\quad + \frac{1}{2\eta_1(1-\beta_1)} \left\| V_1^{-1/2}(\theta_1-\theta^*) \right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1} \sum_{t=1}^{T} \beta_{1,t} + \frac{dG_\infty^2 C_u \eta_0}{1-\beta_1} \sqrt{T}
\end{aligned}
$$

$$
\begin{aligned}
\leq & 2\sum_{k=1}^{\tilde{a}_T-1}\left\{\frac{1}{2(1-\beta_1)}\left[D_\infty e_d^\top\left(\frac{C_l^{-1}\mathbf{I}_d}{\eta_{a_{k+1}}}\right)D_\infty e_d\right]\right\} \\
& + \frac{1}{2(1-\beta_1)}\left[D_\infty e_d^\top\left(\frac{C_l^{-1}\mathbf{I}_d}{\eta_T}\right)D_\infty e_d\right] \\
& + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{1}{2\eta_1(1-\beta_1)}\left\|V_1^{-1/2}(\theta_1-\theta^*)\right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}} \\
& + \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\sqrt{T} \\
& + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{1}{2\eta_1(1-\beta_1)}\left\|V_1^{-1/2}(\theta_1-\theta^*)\right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right) \\
& + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{1}{2\eta_1(1-\beta_1)}\left\|V_1^{-1/2}(\theta_1-\theta^*)\right\|^2 + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right) + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{1}{2\eta_1(1-\beta_1)}\left(D_\infty e_d^\top V_1^{-1}D_\infty e_d\right) + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right) + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}-\beta_{1,t}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{dD_\infty^2 C_l^{-1}}{2\eta_1(1-\beta_1)} + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right) + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}\mathbb{I}_{\beta_{1,t+1}>\beta_{1,t}}}{2\eta_{t+1}(1-\beta_{1,t})(1-\beta_{1,t+1})}\left\|V_{\tilde{a}_{t+1}}^{-1/2}(\theta_{t+1}-\theta^*)\right\|^2\right] \\
& + \frac{dD_\infty^2 C_l^{-1}}{2\eta_1(1-\beta_1)} + \frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t} + \frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \\
\leq & \frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right) + \sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}\mathbb{I}_{\beta_{1,t+1}>\beta_{1,t}}}{2\eta_{t+1}(1-\beta_1)^2}\left(D_\infty e_d^\top C_l^{-1}I_d D_\infty e_d\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{dD_\infty^2 C_l^{-1}}{2\eta_1(1-\beta_1)}+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T}\\
\leq&\frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right)+\sum_{t=1}^{T-1}\left[\frac{\beta_{1,t+1}\mathbb{I}_{\beta_{1,t+1}>\beta_{1,t}}dD_\infty^2}{2C_l\eta_{t+1}(1-\beta_1)^2}\right]\\
&+\frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)}+\frac{dD_\infty G_\infty}{1-\beta_1}\sum_{t=1}^{T}\beta_{1,t}+\frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \qquad (18)
\end{aligned}
$$

□

Corollary G.3. *Suppose $\beta_{1,t}=\beta_1\lambda^t,\ \ 0<\lambda<1$ in Theorem G.2, then we have:*

$$
\begin{aligned}
\sum_{t=1}^{T}f_t(\theta_t)-f_t(\theta^*)\leq&\frac{dD_\infty^2 C_l^{-1}}{(1-\beta_1)\eta_0}\left(\sqrt{T}+\sum_{k=1}^{\tilde{a}_T-1}\sqrt{a_{k+1}}\right)+\frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)}\\
&+\frac{dD_\infty G_\infty\beta_1}{(1-\beta_1)(1-\lambda)}+\frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \qquad (19)
\end{aligned}
$$

Proof. It is easy to prove using:

$$
\sum_{t=1}^{T}\beta_{1,t}=\sum_{t=1}^{T}\beta_1\lambda^{t-1}<\sum_{t=1}^{\infty}\beta_1\lambda^{t-1}\leq\frac{\beta_1}{1-\lambda} \qquad (20)
$$

Plugging (20) into (18), we can derive the results above. □

Corollary G.4. *Suppose $a_n=2^{n-1},\ \ \beta_3=\frac{1}{2}$ in (19), then we have:*

$$
\begin{aligned}
\sum_{t=1}^{T}f_t(\theta_t)-f_t(\theta^*)\leq&\frac{(3+\sqrt{2})dD_\infty^2}{(1-\beta_1)C_l\eta_0}\sqrt{T}+\frac{dD_\infty^2}{2C_l\eta_1(1-\beta_1)}\\
&+\frac{dD_\infty G_\infty\beta_1}{(1-\beta_1)(1-\lambda)}+\frac{dG_\infty^2 C_u\eta_0}{1-\beta_1}\sqrt{T} \qquad (21)
\end{aligned}
$$

Proof. It is easy to prove using:

$$
\sum_{t=1}^{T}a^{t-1}=\frac{1-a^T}{1-a} \qquad (22)
$$

□

Remark (momentum schedule). As is standard in Adam-type convergence analyses (Kingma & Ba, 2014; Reddi et al., 2019), the $O(\sqrt{T})$ regret rate requires a non-increasing momentum schedule, e.g., $\beta_{1,t}=\beta_1\lambda^{t-1}$ as in Corollary G.3. With the constant $\beta_1=0.9$ used in our experiments, the bound retains an additional term of order $\frac{\beta_1}{1-\beta_1}dD_\infty G_\infty T$; this gap between theory and implementation is shared by the existing convergence analyses of Adam and AMSGrad, whose practical implementations likewise use constant momentum, and does not affect the role of IDU, which bounds the cumulative pre-conditioner variation (the $\sum_k\sqrt{a_{k+1}}$ term) independently of the momentum schedule.

H THEOREM 4 IN MAIN PAPER

Lemma H.1. *(Zhuang et al., 2021) Let $m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t$, let $Q_t\in\mathbb{R}^d$, then*

$$
\left\langle Q_t,g_t\right\rangle=\frac{1}{1-\beta_1}\left(\left\langle Q_t,m_t\right\rangle-\left\langle Q_{t-1},m_{t-1}\right\rangle\right)+\left\langle Q_{t-1},m_{t-1}\right\rangle+\frac{\beta_1}{1-\beta_1}\left\langle Q_{t-1}-Q_t,m_{t-1}\right\rangle \qquad (23)
$$

Theorem H.2. *(Convergence analysis for non-convex stochastic optimization) The update of θ_t can be described as $\theta_{t+1} = \theta_t - \eta_t V_{\tilde{a}_t} m_t$, and $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$.*
Under the assumptions:

- *f is differentiable and $f^* \le f \le F$. $\nabla f(x)$ is L-Lipschitz continuous, i.e. $\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|,\ \forall x, y$.*
- *The noisy gradient is unbiased and its infinity norm is bounded by N, i.e. $\mathbb{E} g_t = \nabla f(x)$, $\|g_t\|_\infty \le N$.*

The hyperparameters are set as: $\eta_t = \eta_0 t^{-p}$, $\eta_0 > 0$, $p \in (0,1)$ where the bounds are $C_l I \preceq V_{\tilde{a}_t} \preceq C_u I$, and $0 < C_l < C_u$ ($A \preceq B$ means $B - A$ is a positive semi-definite matrix). And the ϵ and N ensure C_l and C_u exist. For sequence $\{\theta_t\}$ generated by Anon, we have:

$$\tfrac{1}{T}\sum_{t=1}^{T} \mathbb{E}\Big\|\nabla f(x_t)\Big\|^2 \le \tfrac{1}{\eta_0 C_l} T^{p-1}\Big(F - f^* + K\int_1^T t^{-2p}\,\mathrm{d}t + J + K\Big)$$

where

$$J = \tfrac{\beta_1^2 d}{4L(1-\beta_1)^2} N^2 + \tfrac{3dN^2}{1-\beta_1}\eta_0 C_u \sum_{k=1}^{\tilde{a}_T} (a_k - \mathbb{I}_{k\neq 1})^{-p},\ K = \Big(\tfrac{1}{1-\beta_1} + \tfrac{1}{2}\Big) L \eta_0^2 N^2 C_u^2 d$$

Proof. Let $x_t \equiv \theta_t$ denote the iterates. Let $A_t = V_{\tilde{a}_t},\ Q_t = \eta_t A_t \nabla f(x_t)$ and let $Q_0 = Q_1$, we have

$$\begin{aligned}
\sum_{t=1}^{T}\Big\langle Q_t, g_t\Big\rangle &= \frac{1}{1-\beta_1}\Big\langle Q_T, m_T\Big\rangle + \sum_{t=1}^{T}\Big\langle Q_{t-1}, m_{t-1}\Big\rangle + \frac{\beta_1}{1-\beta_1}\sum_{t=1}^{T}\Big\langle Q_{t-1} - Q_t, m_{t-1}\Big\rangle \\
&= \frac{\beta_1}{1-\beta_1}\Big\langle Q_T, m_T\Big\rangle + \sum_{t=1}^{T}\Big\langle Q_t, m_t\Big\rangle + \frac{\beta_1}{1-\beta_1}\sum_{t=0}^{T-1}\Big\langle Q_t - Q_{t+1}, m_t\Big\rangle \qquad (24)
\end{aligned}$$

First we derive a lower bound for (24).

$$\begin{aligned}
\Big\langle Q_t, g_t\Big\rangle =& \Big\langle \eta_t A_t \nabla f(x_t), g_t\Big\rangle \\
=& \Big\langle \eta_{t-1} A_{t-1}\nabla f(x_t), g_t\Big\rangle - \Big\langle (\eta_{t-1}A_{t-1} - \eta_t A_t)\nabla f(x_t), g_t\Big\rangle \\
\ge& \Big\langle \eta_{t-1} A_{t-1}\nabla f(x_t), g_t\Big\rangle - \Big\|\nabla f(x_t)\Big\|_\infty d\Big\|\eta_{t-1}A_{t-1} - \eta_t A_t\Big\|_1\Big\|g_t\Big\|_\infty \\
& \Big(\textit{By Hölder's inequality}\Big) \\
\ge& \Big\langle \eta_{t-1} A_{t-1}\nabla f(x_t), g_t\Big\rangle - dN^2\mathbb{I}_{t\neq a_{\tilde{a}_t}}\Big(\Big\|\eta_{t-1}A_{t-1}\Big\| - \Big\|\eta_t A_t\Big\|_1\Big) \\
& - dN^2\mathbb{I}_{t=a_{\tilde{a}_t}}\Big(\Big\|\eta_{t-1}A_{t-1} - \eta_t A_t\Big\|_1\Big) \qquad (25) \\
& \Big(\textit{Since } \Big\|g_t\Big\|_\infty \le N,\ \eta_{t-1} \ge \eta_t > 0, A_{t-1} = A_t \textit{ when } t \neq a_{\tilde{a}_t}\Big)
\end{aligned}$$

Perform telescope sum, we have

$$\begin{aligned}
\sum_{t=1}^{T}\Big\langle Q_t, g_t\Big\rangle \ge& \sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), g_t\Big\rangle - dN^2\sum_{k=1}^{\tilde{a}_T - 1}\Big(\Big\|\eta_{a_k}A_{a_k}\Big\|_1 - \Big\|\eta_{a_{k+1}-1}A_{a_{k+1}-1}\Big\|_1\Big) \\
& - dN^2\sum_{k=1}^{\tilde{a}_T}\Big\|\eta_{a_k-1}A_{a_k-1} - \eta_{a_k}A_{a_k}\Big\|_1 - dN^2\Big(\Big\|\eta_{a_{\tilde{a}_T}}A_{a_{\tilde{a}_T}}\Big\|_1 - \Big\|\eta_T A_T\Big\|_1\Big) \\
\ge& \sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), g_t\Big\rangle - dN^2\sum_{k=1}^{\tilde{a}_T - 1}\Big\|\eta_{a_k}A_{a_k}\Big\|_1 \\
& - dN^2\sum_{k=1}^{\tilde{a}_T}\Big\|\eta_{a_k-1}A_{a_k-1} - \eta_{a_k}A_{a_k}\Big\|_1 - dN^2\Big\|\eta_{a_{\tilde{a}_T}}A_{a_{\tilde{a}_T}}\Big\|_1
\end{aligned}$$

$$
\begin{aligned}
&\geq \sum_{t=1}^{T} \Big\langle \eta_{t-1} A_{t-1} \nabla f(x_t), g_t \Big\rangle - dN^2 \sum_{k=1}^{\tilde{a}_T} \Big\| \eta_{a_k} A_{a_k} \Big\|_1 \\
&\quad - dN^2 \sum_{k=1}^{\tilde{a}_T} \Big(\Big\| \eta_{a_k - 1} A_{a_k - 1} \Big\|_1 + \Big\| \eta_{a_k} A_{a_k} \Big\|_1 \Big) \\
&= \sum_{t=1}^{T} \Big\langle \eta_{t-1} A_{t-1} \nabla f(x_t), g_t \Big\rangle - 2dN^2 \sum_{k=1}^{\tilde{a}_T} \Big\| \eta_{a_k} A_{a_k} \Big\|_1 - dN^2 \sum_{k=1}^{\tilde{a}_T} \Big\| \eta_{a_k - 1} A_{a_k - 1} \Big\|_1 \\
&\geq \sum_{t=1}^{T} \Big\langle \eta_{t-1} A_{t-1} \nabla f(x_t), g_t \Big\rangle - 3dN^2 \sum_{k=1}^{\tilde{a}_T} \eta_{a_k - 1} C_u
\end{aligned} \tag{26}
$$

Next, we derive an upper bound for $\sum_{t=1}^{T} \Big\langle Q_t, g_t \Big\rangle$ by deriving an upper-bound for the RHS of (24). We derive an upper bound for each part.

$$
\begin{aligned}
\Big\langle Q_t, m_t \Big\rangle &= \Big\langle \eta_t A_t \nabla f(x_t), m_t \Big\rangle = \Big\langle \nabla f(x_t), \eta_t A_t m_t \Big\rangle \\
&= \Big\langle \nabla f(x_t), x_t - x_{t+1} \Big\rangle \\
&\leq f(x_t) - f(x_{t+1}) + \frac{L}{2} \Big\| x_{t+1} - x_t \Big\|^2 \\
&\quad \Big(\textit{By L-smoothness of } f \Big)
\end{aligned} \tag{27}
$$

Perform telescope sum, we have

$$
\sum_{t=1}^{T} \Big\langle Q_t, m_t \Big\rangle \leq f(x_1) - f(x_{T+1}) + \frac{L}{2} \sum_{t=1}^{T} \Big\| \eta_t A_t m_t \Big\|^2 \tag{28}
$$

$$
\begin{aligned}
\Big\langle Q_t - Q_{t+1}, m_t \Big\rangle =& \Big\langle \eta_t A_t \nabla f(x_t) - \eta_{t+1} A_{t+1} \nabla f(x_{t+1}), m_t \Big\rangle \\
=& \Big\langle \eta_t A_t \nabla f(x_t) - \eta_t A_t \nabla f(x_{t+1}), m_t \rangle \\
&+ \Big\langle \eta_t A_t \nabla f(x_{t+1}) - \eta_{t+1} A_{t+1} \nabla f(x_{t+1}), m_t \rangle \\
=& \Big\langle \nabla f(x_t) - \nabla f(x_{t+1}), \eta_t A_t m_t \Big\rangle + \Big\langle (\eta_t A_t - \eta_{t+1} A_{t+1}) \nabla f(x_t), m_t \Big\rangle \\
&= \Big\langle \nabla f(x_t) - \nabla f(x_{t+1}), x_t - x_{t+1} \Big\rangle + \Big\langle \nabla f(x_t), (\eta_t A_t - \eta_{t+1} A_{t+1}) m_t \Big\rangle \\
\leq & L \Big\| x_{t+1} - x_t \Big\|^2 + \Big\langle \nabla f(x_t), (\eta_t A_t - \eta_{t+1} A_{t+1}) m_t \Big\rangle \\
&\Big(\textit{By smoothness of } f \Big) \\
\leq & L \Big\| x_{t+1} - x_t \Big\|^2 + \Big\| \nabla f(x_t) \Big\|_\infty d \Big\| \eta_t A_t - \eta_{t+1} A_{t+1} \Big\|_1 \Big\| m_t \Big\|_\infty \\
&\Big(\textit{By Hölder's inequality} \Big) \\
\leq & L \Big\| x_{t+1} - x_t \Big\|^2 + dN^2 \mathbb{I}_{t+1 \neq a_{\tilde{a}_{t+1}}} \Big(\Big\| \eta_t A_t \Big\|_1 - \Big\| \eta_{t+1} A_{t+1} \Big\|_1 \Big) \\
&+ dN^2 \mathbb{I}_{t+1 = a_{\tilde{a}_{t+1}}} \Big(\Big\| \eta_t A_t - \eta_{t+1} A_{t+1} \Big\|_1 \Big) \\
&\Big(\textit{Since } \eta_t \geq \eta_{t+1} > 0, A_{t+1} = A_t \textit{ when } t+1 \neq a_{\tilde{a}_{t+1}} \Big)
\end{aligned} \tag{29}
$$

Perform telescope sum, we have

$$
\sum_{t=1}^{T-1} \Big\langle Q_t - Q_{t+1}, m_t \rangle \leq L \sum_{t=1}^{T-1} \Big\| \eta_t A_t m_t \Big\|^2 + dN^2 \sum_{k=1}^{\tilde{a}_T - 1} \Big(\Big\| \eta_{a_k} A_{a_k} \Big\|_1 - \Big\| \eta_{a_{k+1} - 1} A_{a_{k+1} - 1} \Big\|_1 \Big)
$$

$$
\begin{aligned}
&+ dN^2 \sum_{k=1}^{\tilde{a}_T-1} \left\| \eta_{a_{k+1}-1} A_{a_{k+1}-1} - \eta_{a_{k+1}} A_{a_{k+1}} \right\|_1 \\
&+ dN^2 \Big(\left\| \eta_{a_{\tilde{a}_T}} A_{a_{\tilde{a}_T}} \right\|_1 - \left\| \eta_T A_T \right\|_1 \Big) \\
\leq & L \sum_{t=1}^{T-1} \left\| \eta_t A_t m_t \right\|^2 + dN^2 \sum_{k=1}^{\tilde{a}_T-1} \left\| \eta_{a_k} A_{a_k} \right\|_1 \\
&+ dN^2 \sum_{k=1}^{\tilde{a}_T-1} \Big(\left\| \eta_{a_{k+1}-1} A_{a_{k+1}-1} \right\|_1 + \left\| \eta_{a_{k+1}} A_{a_{k+1}} \right\|_1 \Big) \\
&+ dN^2 \left\| \eta_{a_{\tilde{a}_T}} A_{a_{\tilde{a}_T}} \right\|_1 \\
\leq & L \sum_{t=1}^{T-1} \left\| \eta_t A_t m_t \right\|^2 + dN^2 \sum_{k=1}^{\tilde{a}_T} \left\| \eta_{a_k} A_{a_k} \right\|_1 \\
&+ dN^2 \sum_{k=1}^{\tilde{a}_T-1} \Big(\left\| \eta_{a_{k+1}-1} A_{a_{k+1}-1} \right\|_1 + \left\| \eta_{a_{k+1}} A_{a_{k+1}} \right\|_1 \Big) \\
\leq & L \sum_{t=1}^{T-1} \left\| \eta_t A_t m_t \right\|^2 + 2dN^2 \sum_{k=1}^{\tilde{a}_T} \left\| \eta_{a_k} A_{a_k} \right\|_1 \\
&+ dN^2 \sum_{k=1}^{\tilde{a}_T-1} \left\| \eta_{a_{k+1}-1} A_{a_{k+1}-1} \right\|_1 \\
\leq & L \sum_{t=1}^{T-1} \left\| \eta_t A_t m_t \right\|^2 + 3dN^2 \sum_{k=1}^{\tilde{a}_T} \eta_{a_k-1} C_u
\end{aligned} \tag{30}
$$

We also have

$$
\begin{aligned}
\Big\langle Q_T, m_T \Big\rangle = & \Big\langle \eta_T A_T \nabla f(x_T), m_T \Big\rangle = \Big\langle \nabla f(x_T), \eta_T A_T m_T \Big\rangle \\
\leq & L \frac{1-\beta_1}{\beta_1} \left\| \eta_T A_T m_T \right\|^2 + \frac{\beta_1}{4L(1-\beta_1)} \left\| \nabla f(x_T) \right\|^2 \\
& \Big(\textit{By Young's inequality} \Big) \\
\leq & L \frac{1-\beta_1}{\beta_1} \left\| \eta_T A_T m_T \right\|^2 + \frac{\beta_1 d}{4L(1-\beta_1)} N^2
\end{aligned} \tag{31}
$$

Combine (28), (30) and (31) into (24), we have

$$
\begin{aligned}
\sum_{t=1}^{T} \Big\langle Q_t, g_t \Big\rangle \leq & L \left\| \eta_T A_T m_T \right\|^2 + \frac{\beta_1^2 d}{4L(1-\beta_1)^2} N^2 \\
& + f(x_1) - f(x_{T+1}) + \frac{L}{2} \sum_{t=1}^{T} \left\| \eta_t A_t m_t \right\|^2 \\
& + \frac{\beta_1}{1-\beta_1} L \sum_{t=1}^{T-1} \left\| \eta_t A_t m_t \right\|^2 + \frac{3\beta_1}{1-\beta_1} dN^2 \sum_{k=1}^{\tilde{a}_T} \eta_{a_k-1} C_u \\
\leq & f(x_1) - f(x_{T+1}) + \Big(\frac{1}{1-\beta_1} + \frac{1}{2} \Big) L \sum_{t=1}^{T} \left\| \eta_t A_t m_t \right\|^2 \\
& + \frac{\beta_1^2 d}{4L(1-\beta_1)^2} N^2 + \frac{3\beta_1}{1-\beta_1} dN^2 \sum_{k=1}^{\tilde{a}_T} \eta_{a_k-1} C_u
\end{aligned} \tag{32}
$$

Combine (26) and (32), we have

$$\begin{aligned}\sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), g_t\Big\rangle - 3dN^2\sum_{k=1}^{\tilde{a}_T}\eta_{a_k-1}C_u \le& \sum_{t=1}^{T}\Big\langle Q_t, g_t\Big\rangle \\ \le& f(x_1) - f(x_{T+1}) + \Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\sum_{t=1}^{T}\Big\|\eta_t A_t m_t\Big\|^2 \\ &+ \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3\beta_1}{1-\beta_1}dN^2\sum_{k=1}^{\tilde{a}_T}\eta_{a_k-1}C_u \end{aligned} \tag{33}$$

Hence we have

$$\begin{aligned}\sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), g_t\Big\rangle \le& f(x_1) - f(x_{T+1}) + \Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\sum_{t=1}^{T}\Big\|\eta_t A_t m_t\Big\|^2 \\ &+ \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\sum_{k=1}^{\tilde{a}_T}\eta_{a_k-1}C_u \\ \le& f(x_1) - f^* + \Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\eta_0^2N^2C_u^2 d\sum_{t=1}^{T}t^{-2p} \\ &+ \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u\sum_{k=1}^{\tilde{a}_T}(a_k - \mathbb{I}_{k\neq 1})^{-p} \\ \le& f(x_1) - f^* + \Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\eta_0^2N^2C_u^2 d\Big(1+\int_1^T t^{-2p}\,\mathrm{d}t\Big) \\ &+ \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u\sum_{k=1}^{\tilde{a}_T}(a_k - \mathbb{I}_{k\neq 1})^{-p} \\ \le& f(x_1) - f^* + \Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\eta_0^2N^2C_u^2 d\int_1^T t^{-2p}\,\mathrm{d}t \\ &+ \underbrace{\frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u\sum_{k=1}^{\tilde{a}_T}(a_k - \mathbb{I}_{k\neq 1})^{-p}}_{J} \\ &+ \underbrace{\Big(\frac{1}{1-\beta_1}+\frac{1}{2}\Big)L\eta_0^2N^2C_u^2 d}_{K} \\ \le& f(x_1) - f^* + K\int_1^T t^{-2p}\,\mathrm{d}t + J + K \end{aligned} \tag{34}$$

Take expectations on both sides, we have

$$\begin{aligned}\sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), \nabla f(x_t)\Big\rangle \le& \mathbb{E}f(x_1) - f^* + K\int_1^T t^{-2p}\,\mathrm{d}t + J + K \\ \le& F - f^* + K\int_1^T t^{-2p}\,\mathrm{d}t + J + K \end{aligned} \tag{35}$$

Note that we have η_t decays monotonically with t, hence

$$\sum_{t=1}^{T}\Big\langle \eta_{t-1}A_{t-1}\nabla f(x_t), \nabla f(x_t)\Big\rangle \ge \eta_0 T^{-p}\sum_{t=1}^{T}\Big\langle A_{t-1}\nabla f(x_t), \nabla f(x_t)\Big\rangle \tag{36}$$

$$\ge \eta_0 T^{1-p}C_l\frac{1}{T}\sum_{t=1}^{T}\Big\|\nabla f(x_t)\Big\|^2 \tag{37}$$

Combine (35) and (37), using $f \leq F$, we have

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla f(x_t)\right\|^2 \leq \frac{1}{\eta_0 C_l}T^{p-1}\Big(F - f^* + K\int_1^T t^{-2p}\,\mathrm{d}t + J + K\Big) \tag{38}$$

And it is easy to prove when $a_n = 2^{n-1}$, we have

$$J = \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \sum_{k=1}^{\tilde{a}_T}(a_k - \mathbb{I}_{k\neq 1})^{-p} \tag{39}$$

$$\leq \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \sum_{k=1}^{\infty}(a_k - \mathbb{I}_{k\neq 1})^{-p} \tag{40}$$

$$= \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \left(1 + \sum_{k=2}^{\infty}(2^{k-1} - \mathbb{I}_{k\neq 1})^{-p}\right) \tag{41}$$

$$\leq \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \left(1 + \sum_{k=2}^{\infty}(2^{k-2})^{-p}\right) \tag{42}$$

$$= \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \left(1 + \sum_{k=1}^{\infty}(2^{-p})^{k-1}\right) \tag{43}$$

$$= \frac{\beta_1^2 d}{4L(1-\beta_1)^2}N^2 + \frac{3dN^2}{1-\beta_1}\eta_0 C_u \left(1 + \frac{1}{1-2^{-p}}\right) \tag{44}$$

□