GRAMS: GRADIENT DESCENT WITH ADAPTIVE MOMENTUM SCALING

Yang Cao[*] **Xiaoyu Li**[†] **Zhao Song**[‡]

ABSTRACT

We introduce **Gr**adient Descent with **A**daptive **M**omentum **S**caling (**Grams**), a novel optimization algorithm that decouples the direction and magnitude of parameter updates in deep learning. Unlike traditional optimizers that directly integrate momentum into updates, Grams separates the update direction, derived from current gradients, from momentum, which is used solely for adaptive magnitude scaling. This approach enables Grams to achieve improved loss descent compared to state-of-the-art cautious and momentum-based optimizers. We theoretically demonstrate that Grams descents faster than other state-of-the-art optimizers and establish a global convergence guarantee for Grams. We also validate its effectiveness through extensive empirical evaluations. The results demonstrate Grams' superior performance, including faster convergence and better generalization, compared to widely-used optimizers such as Adam, Lion, and their cautious variants. Our results highlight Grams' potential as a transformative approach for efficiently training and fine-tuning large language models. Code is available at https://github.com/Gunale0926/Grams.

1 INTRODUCTION

Optimization plays a pivotal role in modern machine learning, serving as the cornerstone for training and fine-tuning models across diverse applications. Over the past decade, the introduction of adaptive optimizers like Adam (Kingma & Ba, 2014) and its variant AdamW (Loshchilov & Hutter, 2017) has significantly shaped the landscape of optimization. These algorithms have become the de facto choices for a variety of tasks, ranging from pre-training Large Language Models (LLMs) (Touvron et al., 2023) to fine-tuning models for text-to-image diffusion (Rombach et al., 2022). Despite the advent of new methods, AdamW has maintained its dominance, particularly in large-scale training regimes, thanks to its robust convergence properties and general applicability.

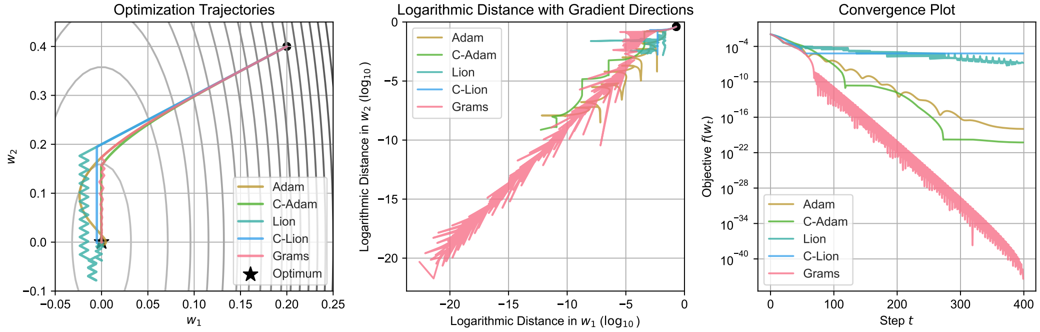
Recent innovations, such as SHAMPOO (Gupta et al., 2018), Schedule Free (Defazio et al., 2024), Lion (Chen et al., 2024), SOAP (Vyas et al., 2024), and ADOPT (Taniguchi et al., 2024), have pushed the boundaries of optimization by introducing novel update rules, momentum mechanisms, and regularization techniques. These methods promise substantial improvements in training efficiency and model performance, particularly in specialized scenarios. The cautious (Liang et al., 2024) mechanism addresses optimization challenges by adaptively masking the momentum term u_t to align with the gradient g_t, preventing conflicts that hinder training. This approach extends to Adam and Lion, resulting in variants like Cautious Adam (C-Adam) and Cautious Lion (C-Lion).

In this paper, we propose Gradient Descent with Adaptive Momentum Scaling (Grams), a novel optimization algorithm designed to address the limitations of existing methods. Unlike traditional optimizers that directly couple momentum with gradient updates, Grams decouples the direction and magnitude of parameter updates. This approach allows the update direction to be derived solely from current gradients while momentum is utilized to scale the update magnitude. Such decoupling enhances stability and robustness, particularly in dynamic optimization landscapes.

[*] ycao4@wyomingseminary.org. Wyoming Seminary.

[†] xli216@stevens.edu. Stevens Institute of Technology.

[‡] magic.linuxkde@gmail.com. Simons Institute for the Theory of Computing at UC Berkeley.


Optimization Trajectories

Logarithmic Distance with Gradient Directions

Convergence Plot

Figure 1: Convergence comparison on a simple convex function $f(w) := (0.5w_1)^2 + (0.1w_2)^2$. Learning rate $\eta = 0.01$ for Grams, Adam, and C-Adam, and $\eta = 0.001$ for Lion and C-Lion. β_1 and β_2 are default values for all optimizers. The graph on the left is the optimizing trajectories; the graph in the middle graph is the distance between current weight and optimum weight; the graph on the right is the training objectives.

By integrating insights from momentum-based methods, adaptive optimizers, and sign-based updates, Grams bridges the gap between theoretical rigor and practical performance, offering a promising direction for scalable and efficient optimization in modern machine learning.

2 PRELIMINARIES

2.1 SIGN FUNCTION

We formally define the sign function, which will be used later in our optimizer Grams.

Definition 2.1 (Sign function)**.** *Given a vector $a = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$, the sign function of a, denoted as* $\mathrm{sign}(a)$*, is defined component-wise as:*

$$\mathrm{sign}(a) = (\mathrm{sign}(a_1), \mathrm{sign}(a_2), \dots, \mathrm{sign}(a_n)),$$

where the scalar sign function $\mathrm{sign}(a_i)$ *is given by:*

$$\mathrm{sign}(a_i) = \begin{cases} 1, & \textit{if } a_i > 0, \\ 0, & \textit{if } a_i = 0, \\ -1, & \textit{if } a_i < 0. \end{cases}$$

2.2 CAUTIOUS OPTIMIZERS

Cautious mechanism (Liang et al., 2024) addresses a key challenge in optimization dynamics: when the momentum term u_t moves in a different direction from the current gradient g_t, it can potentially impede training progress. To mitigate this issue, the Cautious mechanism introduces an adaptive masking mechanism that modifies the momentum term based on its alignment with the gradient direction. Cautious mechanism could apply to Adam and Lion, which form Cautious Adam (C-Adam) and Cautious Lion (C-Lion).

Definition 2.2 (Cautious Mechanism Parameter Update)**.** *The general parameter update rule for the Cautious mechanism is given by:*

$$\begin{aligned} \widehat{u}_t &:= u_t \circ \mathbf{1}_{u_t \circ g_t \geq 0} \\ w_t &:= w_{t-1} - \eta_t \widehat{u}_t, \end{aligned} \tag{1}$$

where w_t is the weight at time step t, $\circ$ denotes Hadamard product. For C-Adam, u_t is from Definition B.4; For C-Lion, u_t is from Definition B.5. g_t is the current gradient.

The Cautious mechanism in Definition 2.2 modifies the parameter updates to ensure they align with the gradient direction, thereby reducing the risk of adverse updates that could impede convergence. To analyze the impact of this mechanism, we introduce Definition 2.3, which quantifies the change in the loss function after an update.

Definition 2.3. *For any loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$, we define $\Delta \mathcal{L}_{w_{t+1}, w_t} := \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)$, where w_{t+1} is updated from any update rule.*

As shown in Liang et al. (2024), the Cautious mechanism ensures that the updated parameters result in a non-negative inner product with the gradient, leading to a monotonic decrease in the loss function when the step size is sufficiently small. Specifically, using a Taylor approximation, it can be expressed as $\Delta \mathcal{L}_{w_{t+1}, w_t} \approx -\eta_t (u_t \circ g_t)^\top \phi(u_t \circ g_t) \le 0$, where $\phi(\cdot)$ represents the alignment mask introduced by the Cautious mechanism. This guarantees that $\mathcal{L}(w_{t+1}) \le \mathcal{L}(w_t)$, ensuring a decrease in loss.

3 Gradient Descent with Adaptive Momentum Scaling

We propose *Gradient Descent with Adaptive Momentum Scaling* (**Grams**). Grams decouples the direction and magnitude of the update by using the direction from gradients while scaling it with the norm of momentum. This section formalizes the Grams update rule, introduces its key components, and provides theoretical guarantees in both loss descent and Hamiltonian dynamics for its performance.

3.1 Definitions

We define the parameter updating rule of Grams formally as below.

Definition 3.1 (Grams Parameter Update)**.** *The parameter update rule for Grams is:*

$$\begin{aligned} m_t &:= \beta_1 m_{t-1} + (1 - \beta_1) g_t, & v_t &:= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\ \widehat{m}_t &:= \frac{m_t}{1 - \beta_1^t}, & \widehat{v}_t &:= \frac{v_t}{1 - \beta_2^t}, \\ u_t &:= \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}, & \widehat{u}_t &:= \operatorname{sign}(g_t) \circ |u_t|, \\ w_t &:= w_{t-1} - \eta_t \widehat{u}_t, \end{aligned} \tag{2}$$

where w_t is the weight at time step t, $g_t = \nabla_w \mathcal{L}_t(w_{t-1})$ is the current gradient, $|\cdot|$ is element-wise absolute value, $\circ$ denotes Hadamard product, and $\operatorname{sign}(\cdot)$ *is defined in Definition 2.1.*

3.2 Loss Descent

In this subsection, we analyze the loss descent properties of the Grams algorithm. Understanding how the loss function decreases over optimization steps provides insights into the efficiency and stability of the method. Below, we formalize the relationship between the step size, gradients, and the resulting decrease in the loss value, leveraging the L-smoothness property of the objective function.

Lemma 3.2 (Informal version of Lemma D.2)**.** *Suppose that $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Let $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t}$ be defined in Definition 2.3, w_{t+1}^{Grams} is updated from w_t using Eq.* (2)*. Then we have the following:*

- *Part 1. It holds that $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \le -\eta_t \langle |g_t|, |u_t| \rangle + \frac{L\eta_t^2}{2} \|u_t\|_2^2$.*
- *Part 2. It holds that $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \ge -\eta_t \langle |g_t|, |u_t| \rangle$.*
- *Part 3. If $\eta_t \le \frac{2}{L\|u_t\|^2} \langle |g_t|, |u_t| \rangle$, then we have $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \le 0$.*

Then, we compare the loss descent between Grams and C-Adam.

Theorem 3.3 (Loss Descent Comparison, informal version of Theorem D.3)**.** *Suppose that $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. For any parameter vector w at optimization step t, let w_t^{Grams} and w_t^{C} be the update of Grams in Definition 3.1 and Cautious optimizers in Definition 2.2, respectively. If the stepsize η_t satisfies $\eta_t \le \frac{2}{L\|u_t\|^2} \cdot \min\{\langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \ge 0} \rangle, \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t < 0} \rangle\}$, then we have $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \le \Delta \mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \le 0$.*

Remark 3.4. *Theorem 3.3 shows that Grams achieves strictly better descent in the loss landscape in the discrete analysis compared to Cautious optimizers. This theoretical guarantee suggests that Grams may converge faster and achieve better minima in practice.*

4 Empirical Experiments

We conducted comprehensive experiments across both pre-training and fine-tuning stages to evaluate the performance of our proposed Grams optimizer. Comparisons were made against several baseline optimizers, including Adam (Kingma & Ba, 2014), Lion (Chen et al., 2024), C-Adam, C-Lion (Liang et al., 2024), and, in some experiments, RMSprop (Hinton et al., 2012; Ruder, 2016).

For Lion and C-Lion, we followed the recommendation from (Chen et al., 2024), setting their learning rates to $\frac{1}{10}$ × Adam learning rate. Additional details and hyperparameters of our experiments can be found in Section G.

4.1 Pre-Training

We train the Llama 60M model (Dubey et al., 2024) using the first $2,048,000$ rows of data from English subset of the C4 dataset (Raffel et al., 2020) to assess Grams' optimization capability for Transformer-based (Vaswani et al., 2017) natural language generation (NLG) tasks. Due to the limited computing resources, we trained $1,000$ steps using constant with warm-up scheduler, in order to simulate the beginning part of regular pre-training.We used the first $10,000$ rows of validation data from the English section of the C4 dataset for evaluation. See Table 1 for evaluation results.

Table 1: Evaluation results of Llama 60M pre-training experiments.

Optimizer	Perplexity↓
Adam	49.83
C-Adam	43.21
Lion	50.25
C-Lion	53.21
Grams (ours)	**38.60**

The evaluation results of the Llama 60M pre-training experiments, as presented in Table 1, reveal that the Grams optimizer achieves the lowest perplexity (38.60) compared to other state-of-the-art optimizers, including Adam (49.83), C-Adam (43.21), Lion (50.25), and C-Lion (53.21). This substantial reduction in perplexity highlights the effectiveness of Grams in optimizing language model performance. While C-Adam and Lion exhibit improvements over their respective base optimizers, Adam and C-Lion, Grams outperforms all variants, underscoring its ability to enhance convergence and generalization. The result demonstrates Grams' superiority in both training efficiency and model quality for large-scale machine learning tasks.

For computer vision tasks, we trained and evaluated the WideResNet-50-2 model (Zagoruyko & Komodakis, 2016) on the CIFAR-10 dataset (Krizhevsky, 2009). Table 2 provides the final accuracy results.

Table 2: Evaluation results of WideResNet-50-2 pre-training experiments.

Optimizer	Final Acc↑
RMSprop	84.47%
Adam	87.56%
C-Adam	88.78%
Lion	89.21%
C-Lion	89.42%
Grams (ours)	**90.55%**

Table 2 highlight the performance of various optimizers—RMSprop, Adam, C-Adam, Lion, C-Lion, and Grams—on the WideResNet-50-2 model trained on the CIFAR-10 dataset. The final accuracy results are presented in Table 2, where Grams achieves the highest accuracy of 90.55%, surpassing Lion (89.21%), C-Lion (89.42%), Adam (87.56%) and C-Adam (88.78%). These results emphasize the effectiveness of Grams in accelerating optimization while achieving superior generalization, making it a robust choice for computer vision tasks.

4.2 Fine-Tuning

We performed full fine-tuning (FT) experiments on the Llama 3.2 1B model (Dubey et al., 2024) using the MetaMathQA dataset (Yu et al., 2023). To evaluate the model, we measured accuracy on the GSM-8K dataset (Cobbe et al., 2021). Results are reported in Table 3.

Table 3: Evaluation results of Llama 3.2 1B FT experiments.

Optimizer	GSM-8K↑
Adam	48.90%
C-Adam	49.81%
Grams (ours)	**51.02%**

The results in Table 3 showcase the performance of different optimizers during the full FT experiments on the Llama 3.2 1B model using the MetaMathQA dataset. The model's accuracy was evaluated on the GSM-8K dataset. Among the optimizers, Grams achieved the highest accuracy of 51.02%, outperforming both Adam (48.90%) and C-Adam (49.81%). These results highlight the effectiveness of Grams in fine-tuning tasks, particularly in improving the model's ability to handle complex datasets like GSM-8K. The superior performance of Grams demonstrates its capacity to achieve better generalization and optimization efficiency in fine-tuning scenarios.

We conducted parameter-efficient fine-tuning (PEFT) experiments on the Llama 3.2 3B model using the SORSA method (Cao, 2024) and the first 100,000 rows of data from the MetaMathQA dataset (Yu et al., 2023). The evaluation was performed on the MATH dataset (Hendrycks et al., 2021), with the results summarized in Table 4.

Table 4: Evaluation results of Llama 3.2 3B PEFT experiments.

Optimizer	MATH↑
Adam	**17.80%**
C-Adam	16.62%
Grams (ours)	**17.80%**

Grams achieved an accuracy of 17.80%, matching the performance of Adam and outperforming C-Adam (16.62%). These results indicate that Grams performs comparably to Adam in PEFT scenarios, maintaining its robust optimization capabilities while offering the additional benefits of parameter efficiency. This consistency further emphasizes Grams' versatility in various fine-tuning settings.

5 Conclusion and Future Work

In this paper, we introduced Gradient Descent with Adaptive Momentum Scaling (Grams), a novel optimization algorithm designed to decouple the direction and magnitude of parameter updates. By leveraging this decoupling, Grams demonstrated superior performance in both theoretical convergence guarantees and empirical evaluations, outperforming state-of-the-art optimizers such as Adam Loshchilov & Hutter (2017), Lion Chen et al. (2024), and their Cautious variants Liang et al. (2024). The results across various tasks highlight Grams' potential as a efficient optimization framework for large language model training and fine-tuning.

References

Anonymous. Improving adaptive moment optimization via preconditioner diagonalization. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=NdNuKMEv9y`. under review.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

Yang Cao. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *arXiv preprint arXiv:2409.00055*, 2024.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations*, 2022.

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, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Aaron Defazio, Xingyu Alice Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *arXiv preprint arXiv:2405.15682*, 2024.

Timothy Dozat. Incorporating nesterov momentum into adam, 2016.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pp. 1042–1085. PMLR, 2018.

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

Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. *Advances in neural information processing systems*, 28, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of adam under relaxed assumptions. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 52166–52196, 2023.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. *arXiv preprint arXiv:2411.16085*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

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

Kai Lv, Yuqing Yang, Tengxiao Liu, Qi jie Gao, Qipeng Guo, and Xipeng Qiu. Full parameter fine-tuning for large language models with limited resources. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:259187846.

Chris J Maddison, Daniel Paulin, Yee Whye Teh, Brendan O'Donoghue, and Arnaud Doucet. Hamiltonian descent methods. *arXiv preprint arXiv:1809.05042*, 2018.

James Martens and Roger Baker Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2015. URL https://api.semanticscholar.org/CorpusID:11480464.

Arvind Neelakantan, Luke Vilnis, Quoc V. Le, Lukasz Kaiser, Karol Kurach, Ilya Sutskever, and James Martens. Adding gradient noise improves learning for very deep networks, 2017.

Yurii Evgen'evich Nesterov. A method for solving the convex programming problem with convergence rate o $(1/\kappa\hat{}\ 2)$. In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Son Nguyen, Lizhang Chen, Bo Liu, and Qiang Liu. H-fac: Memory-efficient optimization with factorized hamiltonian descent. *arXiv preprint arXiv:2406.09958*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Shohei Taniguchi, Keno Harada, Gouki Minegishi, Yuta Oshima, Seong Cheol Jeong, Go Nagahara, Tomoshi Iiyama, Masahiro Suzuki, Yusuke Iwasawa, and Yutaka Matsuo. Adopt: Modified adam can converge with any β_2 with the optimal rate. *arXiv preprint arXiv:2411.02853*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Belinda Tzen, Anant Raj, Maxim Raginsky, and Francis Bach. Variational principles for mirror descent and mirror langevin dynamics. *IEEE Control Systems Letters*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

Ashia C Wilson, Benjamin Recht, and Michael I Jordan. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Procedings of the British Machine Vision Conference 2016*, 2016.

Jiawei Zhao, Zhenyu (Allen) Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *ArXiv*, abs/2403.03507, 2024. URL `https://api.semanticscholar.org/CorpusID:268253596`.

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.

Appendix

Roadmap. In Section A, we present related work. In Section B, we provide some background knowledge in optimization. In Section C, we provide some useful facts, which are utilized in the results. Section D presents a formal analysis of loss descent for Grams optimizers. In Section E, we illustrate the the property of Grams optimizer in the landscape of Hamiltonian dynamics. In Section F, we show the formal proof for the global convergence guarantee of Grams optimizer. Finally, we list the details of our experiments in Section G.

A Related Work

Adam Variants and Memory-Efficient Optimization Adam and its numerous variants have been pivotal in addressing optimization challenges across diverse applications Kingma & Ba (2014); Liu et al. (2019). Among these, AdamW Liu et al. (2019) introduced a crucial modification by decoupling weight decay from gradient updates, restoring the original intent of weight regularization. NAdam Dozat (2016) integrated Nesterov momentum, and AdaBelief Zhuang et al. (2020) refined the second moment estimation for improved generalization. Adan Xie et al. (2024) extended these advancements with an additional momentum term, balancing performance with memory overhead. Schedule-free optimizers Defazio et al. (2024) have further simplified the optimization process by dynamically adjusting learning rates without pre-defined schedules, enhancing adaptability across tasks. More recent efforts, such as ADOPT Taniguchi et al. (2024), streamlined first-order momentum updates through normalization.

Memory-efficient strategies have addressed the growing resource demands of large-scale models. AdaFactor Shazeer & Stern (2018) factorize second-order statistics, achieving sublinear memory usage. K-Fac Martens & Grosse (2015) approximates the Fisher information matrix using Kronecker-factored representations. Innovations such as fused gradient computation Lv et al. (2023) and GaLore Zhao et al. (2024) leverage low-rank gradient structures to optimize memory efficiency.

Regularization Techniques Regularization plays a critical role in improving generalization and robustness in optimization. Lion Chen et al. (2024) introduced sign-based updates with uniform magnitudes, offering inherent noise regularization Neelakantan et al. (2017); Foret et al. (2021); Chen et al. (2022). Earlier methods, such as signSGD Bernstein et al. (2018), explored similar ideas but focused on reducing communication costs in distributed optimization. Despite its efficiency, signSGD often underperformed in deep learning tasks, such as ConvNet training, where Lion demonstrated superior performance through advanced momentum mechanisms.

Building on these ideas, the Cautious mechanism Liang et al. (2024) adaptively masks momentum terms to ensure alignment with gradient directions, mitigating conflicts. This approach has led to new variants, including Cautious Adam (C-Adam) and Cautious Lion (C-Lion), which combine regularization benefits with robust convergence guarantees.

Hamiltonian Dynamics in Optimization Hamiltonian dynamics provides a robust theoretical framework for understanding momentum-based optimization Nesterov (1983); Sutskever et al. (2013); Nguyen et al. (2024); Anonymous (2024). The seminal work of Sutskever et al. (2013) provided a physical interpretation of momentum methods, linking the oscillatory behavior of algorithms like Nesterov's and Polyak's methods Nesterov (1983) to principles of dynamical systems. While traditional gradient descent guarantees a monotonic decrease in objective function values, momentum-based methods exhibit non-monotonic dynamics that require more advanced analytical tools Jin et al. (2018). This has motivated the development of Lyapunov-based approaches for convergence analysis in convex optimization Krichene et al. (2015); Wilson et al. (2016).

Recent studies have further formalized these connections by modeling optimization processes as continuous-time ODEs, uncovering inherent Hamiltonian structures Maddison et al. (2018); Nguyen et al. (2024). These insights have significantly enhanced the theoretical understanding of classical momentum-based algorithms and provided a foundation for exploring new optimization frameworks Anonymous (2024). Moreover, Hamiltonian principles have been extended to analyze convergence rates for accelerated methods Jin et al. (2018) and have inspired broader applications in optimization. In parallel, Mirror Descent, while distinct from Hamiltonian dynamics, leverages variational principles and maintains efficiency with a mild dependence on the dimensionality of decision variables, making it well-suited for large-scale problems Krichene et al. (2015); Tzen et al. (2023).

Algorithm 1 Gradient Descent with Adaptive Momentum Scaling (Grams)

```
Require: parameter $w$, step sizes $\{\eta_t\}$, dampening factors $\beta_1, \beta_2 \in [0, 1)$, $\epsilon > 0$, weight decay $\gamma \geq 0$
Initialize $t = 0$, $m_0 = v_0 = \mathbf{0}$
while $w_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_w \mathcal{L}_t(w_{t-1})$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$
    $u_t \leftarrow \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$
    $\widehat{u}_t \leftarrow \mathrm{sign}(g_t) \circ |u_t|$
    $w_t \leftarrow w_{t-1} - \eta_t \widehat{u}_t$
    $w_t \leftarrow w_t - \eta_t \gamma w_t$                ▷ Add weight decay Loshchilov & Hutter (2017)
end while
```

B Backgrounds on Optimization

B.1 Notations

For two vectors $u, v \in \mathbb{R}^d$, we use $\langle u, v \rangle$ to denote the standard inner product in the Euclidean space. We use $\|u\|_2$ to denote the ℓ_2-norm of u and use $\|u\|_\infty$ to denote the ℓ_∞-norm of u. For a matrix A, we use $\|A\|_F$ to denote the Frobenius norm of A. For a twice differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, we use $\nabla f(x)$ and $\nabla^2 f(x)$ to denote the gradient and Hessian of f, respectively. Given a vector $x \in \mathbb{R}^d$, we use $\mathbf{1}_{x \geq 0} \in \mathbb{R}^d$ to denote the vector where each entry indicates whether the corresponding entry of x is non-negative, i.e., for each $i \in [d]$, $(\mathbf{1}_{x \geq 0})_i = 1$ if $x_i \geq 0$, and $(\mathbf{1}_{x \geq 0})_i = 0$ otherwise.

B.2 Basic Definition

We define the L-smoothness of functions as below.

Definition B.1 (L-smooth)**.** *We say that a function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if $\|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L \|x_1 - x_2\|_2$ for all $x_1, x_2 \in \mathbb{R}^d$.*

We state a common fact of L-smooth functions as follow.

Fact B.2. *If a function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth, then we have*

$$f(x_2) \leq f(x_1) + \langle \nabla f(x_1), x_2 - x_1 \rangle + \frac{L}{2} \|x_2 - x_1\|_2^2,$$
$$f(x_2) \geq f(x_1) + \langle \nabla f(x_1), x_2 - x_1 \rangle - \frac{L}{2} \|x_2 - x_1\|_2^2.$$

We also define PL-condition as below.

Definition B.3 (PL-condition)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the μ-Polyak–Łojasiewicz (PL) condition with constant $\mu > 0$ if the following inequality holds for all $x \in \mathbb{R}^d$:*

$$\|\nabla f(x)\|^2 \geq 2\mu (f(x) - f^*),$$

where f^* is the minimum value of the function f, i.e., $f^* = \inf_{x \in \mathbb{R}^d} f(x)$.

B.3 Adam Optimizer

Adam (Adaptive Moment Estimation) Kingma & Ba (2014) is a widely-used optimizer that combines the benefits of RMSprop Hinton et al. (2012) and momentum by maintaining both first and second moment estimates of the gradients. The algorithm adapts the learning rates for each parameter using these estimates.

Definition B.4 (Adam)**.** *The parameter update rule for Adam is given by:*

$$\begin{aligned} m_t &:= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &:= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \widehat{m}_t &:= \frac{m_t}{1-\beta_1^t} \\ \widehat{v}_t &:= \frac{v_t}{1-\beta_2^t} \\ u_t &:= \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t}+\epsilon} \\ w_{t+1} &:= w_t - \eta_t u_t, \end{aligned}$$

where w_t is the weight at time step t, m_t and v_t are the first and second momentum estimates respectively, $g_t = \nabla_w \mathcal{L}_t(w_{t-1})$ is the current gradient, β_1 and β_2 are decay rates for the moment estimates, ϵ is a small constant for numerical stability, and η_t is the learning rate at step t.

B.4 Lion Optimizer

Evolved Sign Momentum (Lion) Chen et al. (2024) is an efficient optimizer that leverages momentum and sign-based updates. Lion's key innovation lies in its update rule, which combines both current and momentum gradients through sign operations.

Definition B.5 (Lion Parameter Update)**.** *The parameter update rule for Lion is given by:*

$$\begin{aligned} u_t &:= \mathrm{sign}(\beta_1 m_{t-1} + (1-\beta_1) g_t) \\ w_t &:= w_{t-1} - \eta_t \cdot u_t \\ m_t &:= \beta_2 m_{t-1} + (1-\beta_2) g_t, \end{aligned}$$

where w_t is the weight at time step t, m_{t-1} is the momentum term, $g_t = \nabla_w \mathcal{L}_t(w_{t-1})$ is the current gradient, β_1 and β_2 are the momentum coefficients, η_t is the learning rate at step t, and sign *is defined in Definition 2.1,*

Lion's efficiency stems from its memory-efficient design - it only needs to maintain a single momentum term and operates primarily through sign operations. This makes it particularly suitable for large-scale training where memory constraints are significant. The optimizer has demonstrated strong performance in training large language models and vision transformers, often achieving comparable or better results than Adam while using less memory.

C Useful Facts

Fact C.1. *Given vectors $a, b, c \in \mathbb{R}^d$, we have*

$$\langle a, b \circ c \rangle = \langle a \circ b, c \rangle.$$

Fact C.2. *Let two vectors $a, b \in \mathbb{R}^n$, then:*

$$\langle a, -\mathrm{sign}(a) \circ |b| \rangle = -\langle |a|, |b| \rangle$$

Proof. For the left side of the equation:

$$\begin{aligned} \langle a, -\mathrm{sign}(a) \circ |b| \rangle &= \sum_{i=1}^n -a_i \mathrm{sign}(a_i) |b|_i \\ &= -\sum_{i=1}^n |a|_i |b|_i \\ &= -\langle |a|, |b| \rangle \end{aligned}$$

where the first step comes from the definition of inner product, the second step uses Fact C.5, and the final step uses the definition of inner product again. □

Fact C.3. *Let two vectors $a, b \in \mathbb{R}^n$, then:*

$$\langle a, b\rangle - \langle |a|, |b|\rangle \leq 0.$$

Proof.

$$\begin{aligned}\langle a, b\rangle - \langle |a|, |b|\rangle &= \sum_{i=1}^{n} a_i b_i - |a|_i |b|_i \\ &= \sum_{i=1}^{n} \begin{cases} 0 & \text{if } a_i \text{ and } b_i \text{ have the same sign} \\ -2|a_i||b_i| & \text{if } a_i \text{ and } b_i \text{ have opposite signs} \end{cases} \\ &\leq 0,\end{aligned}$$

where the first step uses the definition of inner product, the second step discusses the only two cases we have for signs, and the final inequality comes from basic algebra. □

Fact C.4. *Let $x = a \circ b$ be an element-wise product of two vectors $a, b \in \mathbb{R}^n$, then:*

$$\langle a, b\rangle - \langle |a|, |b|\rangle - \langle a \circ b, \mathbf{1} - \mathbf{1}_{a \circ b > 0}\rangle \leq 0$$

Proof.

$$\begin{aligned}&\langle a, b\rangle - \langle |a|, |b|\rangle - \langle a \circ b, \mathbf{1} - \mathbf{1}_{a \circ b > 0}\rangle \\ &= \sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} |a_i||b_i| - (\sum_{i=1}^{n} a_i b_i - \sum_{i: a_i b_i > 0}^{n} a_i b_i) \\ &= \sum_{i: a_i b_i > 0}^{n} a_i b_i - \sum_{i=1}^{n} |a_i||b_i|,\end{aligned}$$

where the first step expands the terms, and the second step simplifies by splitting the sum based on the sign of $a_i b_i$.

If all $a_i b_i \geq 0$, then $\sum_{i: a_i b_i > 0}^{n} a_i b_i = \sum_{i=1}^{n} |a_i||b_i|$, so the expression is 0. Otherwise, $\sum_{i=1}^{n} |a_i||b_i| > \sum_{i: a_i b_i > 0}^{n} a_i b_i$, so the expression is negative.

Thus,

$$\langle a, b\rangle - \langle |a|, |b|\rangle - \langle a \circ b, \mathbf{1}_d - \mathbf{1}_{a \circ b > 0}\rangle = \sum_{i: a_i b_i > 0}^{n} a_i b_i - \sum_{i=1}^{n} |a_i||b_i| \leq 0.$$

The proof is complete. □

Fact C.5. *Given a scalar $a \in \mathbb{R}$, we have:*

$$a \cdot \mathrm{sign}(a) = |a|.$$

Proof. Let $a \in \mathbb{R}$. By Definition 2.1:

$$\mathrm{sign}(a) = \begin{cases} 1, & \text{if } a > 0, \\ 0, & \text{if } a = 0, \\ -1, & \text{if } a < 0. \end{cases}$$

Consider the following cases:

- If $a > 0$, then $\mathrm{sign}(a) = 1$, so:

$$a \cdot \mathrm{sign}(a) = a \cdot 1 = a = |a|.$$

- If $a = 0$, then $\mathrm{sign}(a) = 0$, so:

$$a \cdot \mathrm{sign}(a) = 0 \cdot 0 = 0 = |a|.$$

- If $a < 0$, then $\operatorname{sign}(a) = -1$, so:

$$a \cdot \operatorname{sign}(a) = a \cdot (-1) = -a = |a|.$$

Thus, in all cases, $a \cdot \operatorname{sign}(a) = |a|$. □

Fact C.6. *Given a vector $a = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$, we have:*

$$a \circ \operatorname{sign}(a) = |a|,$$

where the operations are applied component-wise.

Proof. Let $a = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$. By Definition 2.1, the sign function is applied component-wise:

$$\operatorname{sign}(a) = (\operatorname{sign}(a_1), \operatorname{sign}(a_2), \dots, \operatorname{sign}(a_n)).$$

Expanding the Hadamard product $a \circ \operatorname{sign}(a)$ component-wise:

$$a \circ \operatorname{sign}(a) = (a_1 \cdot \operatorname{sign}(a_1), a_2 \cdot \operatorname{sign}(a_2), \dots, a_n \cdot \operatorname{sign}(a_n)).$$

By Fact C.5 (the scalar version), for each i:

$$a_i \cdot \operatorname{sign}(a_i) = |a_i|.$$

Thus:

$$a \circ \operatorname{sign}(a) = (|a_1|, |a_2|, \dots, |a_n|) = |a|,$$

where the absolute value $|a|$ is applied component-wise. □

D Loss Descent

Lemma D.1. *Suppose that $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Let $\Delta\mathcal{L}_{w^{\mathrm{C}}_{t+1}, w_t}$ be defined in Definition 2.3, w^{C}_{t+1} is updated from w_t using Definition 2.2. Then we have the followings:*

- *Part 1. It holds that*

$$\Delta\mathcal{L}_{w^{\mathrm{C}}_{t+1}, w_t} \leq -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle + \frac{L\eta_t^2}{2} \|u_t\|_2^2, \tag{3}$$

- *Part 2. It holds that*

$$\Delta\mathcal{L}_{w^{\mathrm{C}}_{t+1}, w_t} \geq -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle.$$

- *Part 3. If $\eta_t \leq \frac{2}{L\|u_t\|_2^2} \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle$, then $\Delta\mathcal{L}_{w^{\mathrm{C}}_{t+1}, w_t} \leq 0$.*

Proof. **Proof of Part 1.** We can show that

$$\begin{aligned}
\Delta\mathcal{L}_{w^{\mathrm{C}}_{t+1}, w_t} &= \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \\
&\leq \mathcal{L}(w_t) + \langle g_t, w_{t+1} - w_t \rangle + \frac{L}{2} \|w_{t+1} - w_t\|_2^2 - \mathcal{L}(w_t) \\
&= \langle g_t, w_{t+1} - w_t \rangle + \frac{L}{2} \|w_{t+1} - w_t\|_2^2 \\
&= \langle g_t, -\eta_t u_t \circ \mathbf{1}_{u_t \circ g_t \geq 0} \rangle + \frac{L}{2} \|\eta_t u_t \circ \mathbf{1}_{u_t \circ g_t \geq 0}\|_2^2 \\
&= -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle + \frac{L}{2} \|\eta_t u_t \circ \mathbf{1}_{u_t \circ g_t \geq 0}\|_2^2 \\
&\leq -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle + \frac{L\eta_t^2}{2} \|u_t\|_2^2 \qquad (4)
\end{aligned}$$

where the first step follows from Definition 2.3, the second step follows from that $\mathcal{L}$ is L-smooth and Fact B.2, the third step follows from basic algebra, the fourth step follows from Definition 2.2, the fifth step follows from Fact C.1, and the last step follows from basic algebra.

Proof of Part 2. Next, we can show that

$$\begin{aligned}
\Delta \mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} &= \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \\
&\geq \mathcal{L}(w_t) + \langle g_t, w_{t+1} + w_t \rangle - \frac{L}{2}\|w_{t+1} - w_t\|_2^2 - \mathcal{L}(w_t) \\
&\geq \langle g_t, w_{t+1} - w_t \rangle \\
&= \langle g_t, -\eta_t u_t \circ \mathbf{1}_{u_t \circ g_t \geq 0} \rangle \\
&= -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle
\end{aligned} \tag{5}$$

where the first step follows from Definition 2.3, the second step follows from that $\mathcal{L}$ is L-smooth and Fact B.2, the third step follows from basic algebra, the fourth step follows from Definition 2.2, the last step follows from Fact C.1.

Proof of Part 3. By rearranging the Eq. (4), it is clear that if $\eta_t \leq \frac{2}{L\|u_t\|_2^2} \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle$, then we have $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \leq 0$. □

Lemma D.2 (Formal version of Lemma 3.2)**.** *Suppose that $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Let $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t}$ be defined in Definition 2.3, w_{t+1}^{Grams} is updated from w_t using Eq.* (2)*. Then we have the following:*

- *Part 1. It holds that*

$$\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \leq -\eta_t \langle |g_t|, |u_t| \rangle + \frac{L\eta_t^2}{2}\|u_t\|_2^2. \tag{6}$$

- *Part 2. It holds that*

$$\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \geq -\eta_t \langle |g_t|, |u_t| \rangle.$$

- *Part 3. If $\eta_t \leq \frac{2}{L\|u_t\|^2} \langle |g_t|, |u_t| \rangle$, then we have $\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \leq 0$.*

Proof. **Proof of Part 1.** We can show that

$$\begin{aligned}
\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} &= \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \\
&\leq \mathcal{L}(w_t) + \langle g_t, w_{t+1} - w_t \rangle + \frac{L}{2}\|w_{t+1} - w_t\|_2^2 - \mathcal{L}(w_t) \\
&= \langle g_t, w_{t+1} - w_t \rangle + \frac{L}{2}\|w_{t+1} - w_t\|_2^2 \\
&= \langle g_t, -\eta_t \cdot \mathrm{sign}(g_t) \circ |u_t| \rangle + \frac{L}{2}\|\eta_t \cdot \mathrm{sign}(g_t) \circ |u_t|\|_2^2 \\
&= -\eta_t \langle g_t \circ \mathrm{sign}(g_t), |u_t| \rangle + \frac{L}{2}\|\eta_t u_t\|_2^2 \\
&\leq -\eta_t \langle |g_t|, |u_t| \rangle + \frac{L\eta_t^2}{2}\|u_t\|_2^2
\end{aligned} \tag{7}$$

where the first step follows from Definition 2.3, the second step follows from that $\mathcal{L}$ is L-smooth and Fact B.2, the third step follows from basic algebra, the fourth step follows from Definition 3.1, the fifth step follows from the Fact C.1, and the last step follows from $g_t \circ \mathrm{sign}(g_t) = |g_t|$.

Proof of Part 2. Next, we can show that

$$\begin{aligned}
\Delta \mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} &= \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \\
&\geq \mathcal{L}(w_t) + \langle g_t, w_{t+1} + w_t \rangle - \frac{L}{2}\|w_{t+1} - w_t\|_2^2 - \mathcal{L}(w_t) \\
&\geq \langle g_t, w_{t+1} - w_t \rangle
\end{aligned}$$

$$\begin{aligned}
&= \langle g_t, -\eta_t \cdot \mathrm{sign}(g_t) \circ |u_t| \rangle \\
&= -\eta_t \langle |g_t|, |u_t| \rangle
\end{aligned} \tag{8}$$

where the first step follows from Definition 2.3, the second step follows from that $\mathcal{L}$ is L-smooth and Fact B.2, the third step follows from basic algebra, the fourth step follows from Definition 3.1, the last step follows from the Fact C.1 and Fact C.6.

Proof of Part 3. By rearranging the Eq. (7), it is clear that if $\eta_t \leq \frac{2}{L\|u_t\|_2^2}\langle |g_T|, |u_t| \rangle$, then we have $\Delta\mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \leq 0$. □

Theorem D.3 (Loss Descent Comparison, formal version of Theorem 3.3)**.** *Suppose that $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. For any parameter vector w at optimization step t, let w_t^{Grams} and w_t^{C} be the update of Grams in Definition 3.1 and Cautious optimizers in Definition 2.2, respectively. If the stepsize η_t satisfies*

$$\eta_t \leq \frac{2}{L\|u_t\|^2} \cdot \min\{\langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle, \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t < 0} \rangle\},$$

then we have

$$\Delta\mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \leq \Delta\mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \leq 0.$$

Proof. We define the index sets:

$$\begin{aligned}
I^+ &= \{i \in [d] : u_{t,i}, g_{t,i} \geq 0\}; \\
I^- &= \{i \in [d] : u_{t,i}, g_{t,i} < 0\}.
\end{aligned}$$

By Part 1. of Lemma D.2, we have

$$\Delta\mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} \leq -\eta_t \langle |g_t|, |u_t| \rangle + \frac{L\eta_t^2}{2}\|u_t\|_2^2. \tag{9}$$

By Part 2. of Lemma D.1, we have

$$\Delta\mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \geq -\eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle. \tag{10}$$

Then we can show that

$$\begin{aligned}
\Delta\mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} - \Delta\mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \leq &- \eta_t \langle |g_t|, |u_t| \rangle + \eta_t \langle u_t \circ g_t, \mathbf{1}_{u_t \circ g_t \geq 0} \rangle + \frac{L\eta_t^2}{2}\|u_t\|_2^2 \\
= &- \eta_t \sum_{i=1}^d |u_{t,i}||g_{t,i}| + \eta_t \sum_{i \in I^+} u_{t,i} g_{t,i} + \frac{L\eta_t^2}{2}\|u_t\|_2^2 \\
= &- \eta_t \sum_{i \in I^+} |u_{t,i}||g_{t,i}| - \eta_t \sum_{i \in I^-} |u_{t,i}||g_{t,i}| + \eta_t \sum_{i \in I^+} u_{t,i} g_{t,i} + \frac{L\eta_t^2}{2}\|u_t\|_2^2 \\
= &- \eta_t \sum_{i \in I^+} u_{t,i} g_{t,i} - \eta_t \sum_{i \in I^-} |u_{t,i}||g_{t,i}| + \eta_t \sum_{i \in I^+} u_{t,i} g_{t,i} + \frac{L\eta_t^2}{2}\|u_t\|_2^2 \\
= &- \eta_t \sum_{i \in I^-} |u_{t,i}||g_{t,i}| + \frac{L\eta_t^2}{2}\|u_t\|_2^2
\end{aligned}$$

where the first step follows from Eq. (10) and Eq. (9), the second step expands vectors element-wise, the third step follows from that $[d]$ is the disjoint union of I^+ and I^-, the fourth step follows from that $|u_{t,i}||g_{t,i}| = u_{t,i} g_{t,i}$ for $i \in I^+$, and the last step follows from basic algebra.

To ensure $\Delta\mathcal{L}_{w_{t+1}^{\mathrm{Grams}}, w_t} - \Delta\mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \leq 0$, it suffices to have

$$-\eta_t \sum_{i \in I^-} |u_{t,i}||g_{t,i}| + \frac{L\eta_t^2}{2}\|u_t\|_2^2 \leq 0.$$

Rearranging the above inequality gives

$$\begin{aligned}\eta_t &\le \frac{2}{L\|u_t\|_2^2} \sum_{i \in I^-} |u_{t,i}||g_{t,i}| \\ &= \frac{2}{L\|u_t\|_2^2} \langle g_t \circ u_t, \mathbf{1}_{u_t \circ g_t < 0} \rangle,\end{aligned}$$

where the last step follows from the definition of I^- and basic algebra.

Note that by Part 3 of Lemma D.1, if $\eta_t \le \frac{2}{L\|u_t\|_2^2} \langle g_t \circ u_t, \mathbf{1}_{g_t \circ u_t \ge 0} \rangle$, we have $\mathcal{L}_{w_{t+1}^{\mathrm{C}}, w_t} \le 0$. □

E HAMILTONIAN DYNAMICS

Definition E.1 (Section 2.1 from Liang et al. (2024))**.** *Momentum-based algorithms can be typically viewed as monotonic descending algorithms on an augmented loss $H(W, S)$, which satisfies $\min_S H(W, S) = \mathcal{L}(W)$, so that minimizing $\mathcal{L}(W)$ is equivalent to minimizing $H(W, S)$. A typical choice is*

$$H(w, s) = \mathcal{L}(w) + \mathcal{K}(s),$$

where $\mathcal{K}(\cdot)$ is any lower bounded function. The continuous-time form of most momentum-based algorithms can be written into a Hamiltonian descent form:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} w_t &= -\nabla \mathcal{K}(s_t) - \Phi_t(\nabla \mathcal{L}(w_t)) \\ \frac{\mathrm{d}}{\mathrm{d}t} s_t &= \nabla \mathcal{L}(w_t) - \Psi_t(\nabla \mathcal{K}(s_t))\end{aligned} \tag{11}$$

where $H(W, S)$ is a Hamiltonian (or Lyapunov) function that satisfies

$$\min_S H(W, S) = \mathcal{L}(W), \quad \forall W,$$

so that minimizing $\mathcal{L}(W)$ reduces to minimizing $H(W, S)$; and $\Phi(\cdot), \Psi(\cdot)$ are two monotonic mappings satisfying

$$\langle x, \Phi(x) \rangle \ge 0, \qquad \langle x, \Psi(x) \rangle \ge 0, \qquad \forall x \in X.$$

With $\Phi(X) = \Psi(X) = 0$, the system in (11) *reduces to the standard Hamiltonian system that keeps $H(W_t, S_t) =$ const along the trajectory. When adding the descending components with Φ and Ψ, the system then keeps $H(W, S)$ monotonically decreasing:*

$$\frac{\mathrm{d}}{\mathrm{d}t} H(w_t, s_t) = \Delta_H(w_t, s_t) \le 0,$$

where

$$\Delta_H(w_t, s_t) := -\langle x, \Phi(x) \rangle - \langle x, \Psi(x) \rangle. \tag{12}$$

On the other hand, $\mathcal{L}(w)$, which is the true objective, is not necessarily decreasing monotonically.

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathcal{L}(w_t) = -\Delta_{\mathcal{L}}(w_t, s_t),$$

where

$$\Delta_{\mathcal{L}}(w_t, s_t) := \langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle + \langle \nabla \mathcal{L}(w_t), \Phi_t(\nabla \mathcal{L}(w_t)) \rangle. \tag{13}$$

Definition E.2 (Grams Hamiltonian Dynamics)**.** *We could modify Hamiltonian dynamics with Grams' optimizing scheme,*

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} w_t &:= -\operatorname{sign}(\nabla \mathcal{L}(w_t) \circ |\nabla \mathcal{K}(s_t)| - \Phi_t(\nabla \mathcal{L}(w_t)) \\ \frac{\mathrm{d}}{\mathrm{d}t} s_t &:= \nabla \mathcal{L}(w_t) - \Psi_t(\nabla \mathcal{K}(s_t)),\end{aligned}$$

where $|\cdot|$ denotes element-wise absolute value, $\circ$ is the Hadamard product, and Φ_t, Ψ_t are scaling functions.

Theorem E.3 (Theorem 2.3 in Liang et al. (2024)). *For Hamiltonian dynamics of Cautious optimizer (in Definition 2.2), we have:*

$$\begin{aligned}\Delta_H^C(w_t,s_t) := \frac{\mathrm{d}}{\mathrm{d}t}H(w_t,s_t) &= \langle x_t, \mathbf{1} - \mathbf{1}_{x_t\geq 0}\rangle - \Delta_H(w_t,s_t).\\ \Delta_{\mathcal{L}}^C(w_t) := \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(w_t) &= -\langle x_t, \mathbf{1}_{x_t\geq 0}\rangle - \langle \nabla\mathcal{L}(w_t), \Phi_t(\nabla\mathcal{L}(w_t))\rangle\\ &= \langle x_t, \mathbf{1} - \mathbf{1}_{x_t\geq 0}\rangle - \Delta_{\mathcal{L}}(w_t,s_t).\end{aligned}$$

where $\Delta_H(w_t,s_t)$ and $\Delta_{\mathcal{L}}(w_t)$ represent the decreasing rates of H and $\mathcal{L}$ in accordance with the system in Definition E.1.

Hence:

- *If $\langle x_t, (\mathbf{1}_d - \mathrm{sign}(x_t))\rangle \leq 0$ for any $x \in \mathbb{R}^d$, then both H and $\mathcal{L}$ decrease faster than the original system:*

$$\begin{aligned}\Delta_H^C(w_t,s_t) &\leq -\Delta_H(w_t,s_t) \leq 0,\\ \Delta_{\mathcal{L}}^C(w_t) &\leq -\Delta_{\mathcal{L}}(w_t,s_t).\end{aligned}$$

- *If $\langle x_t, \mathrm{sign}(\nabla\mathcal{L}(w_t))\rangle \geq 0$ for any $x \in \mathbb{R}^d$, then $\mathcal{L}$ decreases monotonically:*

$$\Delta_{\mathcal{L}}^C(w_t) \leq 0.$$

Theorem E.4 (Convergence of Grams Hamiltonian Dynamics). *Following the dynamics in Definition E.2, we have*

$$\begin{aligned}\Delta_H^{\textit{Grams}}(w_t,s_t) &:= \frac{\mathrm{d}}{\mathrm{d}t}H(w_t,s_t) \leq 0,\\ \Delta_{\mathcal{L}}^{\textit{Grams}}(w_t) &:= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(w_t) \leq -\Delta_{\mathcal{L}}(w_t,s_t),\end{aligned}$$

where $\Delta_H(w_t,s_t)$ and $\Delta_{\mathcal{L}}(w_t,s_t)$ represent the decreasing rates of H and $\mathcal{L}$ in accordance with the system in Definition E.1.

Proof. Recall Eq. (12) and (13):

$$\begin{aligned}\Delta_H(w_t,s_t) &:= \langle \nabla\mathcal{L}(w_t), \Phi(\nabla\mathcal{L}(w_t))\rangle + \langle \mathcal{K}(s_t), \Psi(\mathcal{K}(s_t))\rangle\\ \Delta_{\mathcal{L}}(w_t,s_t) &:= \langle \nabla\mathcal{L}(w_t), \nabla\mathcal{K}(s_t)\rangle + \langle \nabla\mathcal{L}(w_t), \Phi_t(\nabla\mathcal{L}(w_t))\rangle.\end{aligned}$$

Following the dynamics in Definition E.2, we can calculate the derivative of $H(w_t,s_t)$ with respect to t:

$$\begin{aligned}\Delta_H^{\mathrm{Grams}}(w_t,s_t) &= \langle \nabla\mathcal{L}(w_t), \frac{\mathrm{d}}{\mathrm{d}t}w_t\rangle + \langle \nabla\mathcal{K}(s_t), \frac{\mathrm{d}}{\mathrm{d}t}s_t\rangle\\ &= \langle \nabla\mathcal{L}(w_t), -\mathrm{sign}(\nabla\mathcal{L}(w_t)) \circ |\nabla\mathcal{K}(s_t)| - \Phi_t(\nabla\mathcal{L}(w_t))\rangle\\ &\quad + \langle \mathcal{K}(s_t), \nabla\mathcal{L}(w_t) - \Psi_t(\nabla\mathcal{K}(s_t))\rangle\\ &= \langle \nabla\mathcal{L}(w_t), -\mathrm{sign}(\nabla\mathcal{L}(w_t)) \circ |\nabla\mathcal{K}(s_t)|\rangle + \langle \nabla\mathcal{K}(s_t), \nabla\mathcal{L}(w_t)\rangle - \langle \nabla\mathcal{L}(w_t), \Phi_t(\nabla\mathcal{L}(w_t))\rangle\\ &\quad - \langle \nabla\mathcal{K}(s_t), \Psi_t(\nabla\mathcal{K}(s_t))\rangle\\ &= \langle \nabla\mathcal{L}(w_t), \nabla\mathcal{K}(s_t)\rangle - \langle |\nabla\mathcal{L}(w_t)|, |\nabla\mathcal{K}(s_t)|\rangle - \Delta_H(w_t,s_t)\\ &\leq 0,\end{aligned}$$

where the first step follows from the chain rule for the time derivative of the Hamiltonian H, the second step substitutes the dynamics from Definition E.2, the third step separates the inner products for clearer analysis, the fourth step follows the definition of $\Delta H(w_t,s_t)$ and Fact C.2, and the last step follows Fact C.3, and $-\Delta H(w_t,s_t) \leq 0$.

Then, we calculate the derivative of $\mathcal{L}(w_t)$ with respect to t.

$$\begin{aligned}\Delta_{\mathcal{L}}^{\mathrm{Grams}}(w_t) &= \langle \nabla\mathcal{L}(w_t), -\mathrm{sign}(\nabla\mathcal{L}(w_t)) \circ |\nabla\mathcal{K}(s_t)| - \Phi_t(\nabla\mathcal{L}(w_t))\rangle\\ &= \langle \nabla\mathcal{L}(w_t), -\mathrm{sign}(\nabla\mathcal{L}(w_t)) \circ |\nabla\mathcal{K}(s_t)|\rangle - \langle \nabla\mathcal{L}(w_t), \Phi_t(\nabla\mathcal{L}(w_t))\rangle\end{aligned}$$

$$
\begin{aligned}
&= -\langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle - \langle \nabla \mathcal{L}(w_t), \Phi_t(\nabla \mathcal{L}(w_t)) \rangle \\
&= \langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle \\
&\quad - (\langle \nabla \mathcal{L}(w_t), \Phi_t(\nabla \mathcal{L}(w_t)) \rangle + \langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle) \\
&= \langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle - \Delta_{\mathcal{L}}(w_t, s_t)
\end{aligned}
$$

where the first step follows from the chain rule, and the second step separates the inner products. The third step follows Fact C.2, the fourth step adds and subtracts the term $\langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle$ simultaneously, the fifth step follows the definition of $\Delta_{\mathcal{L}}(w_t, s_t)$ from Eq. (13).

Since we know $\langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle \le 0$ from Fact C.3,

$$
\langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle \le -\Delta_{\mathcal{L}}(w_t, s_t)
$$

Thus we complete the proof. □

Theorem E.5 (Convergence Comparison of Hamiltonian Dynamics between Grams and Cautious Optimizers)**.** *From Theorem E.4 and E.3, recall $\Delta_{\mathcal{L}}^{Grams}(w_t)$ and $\Delta_{\mathcal{L}}^{C}(w_t)$:*

$$
\Delta_{\mathcal{L}}^{Grams}(w_t) \le \Delta_{\mathcal{L}}^{C}(w_t).
$$

Proof. We calculate the difference between $\Delta_{\mathcal{L}}^{\text{Grams}}(w_t)$ and $\Delta_{\mathcal{L}}^{\text{C}}(w_t)$:

$$
\Delta_{\mathcal{L}}^{\text{Grams}}(w_t) - \Delta_{\mathcal{L}}^{\text{C}}(w_t) = \langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{L}(w_t)| \rangle - \langle x_t, \mathbf{1} - \mathbf{1}_{x_t \ge 0} \rangle,
$$

where $x_t = \nabla \mathcal{L}(w_t) \circ \nabla \mathcal{K}(s_t)$.

By applying Fact C.4, we know:

$$
\langle \nabla \mathcal{L}(w_t), \nabla \mathcal{K}(s_t) \rangle - \langle |\nabla \mathcal{L}(w_t)|, |\nabla \mathcal{K}(s_t)| \rangle - \langle x_t, \mathbf{1} - \mathbf{1}_{x_t \ge 0} \rangle \le 0,
$$

with equality if all components of $\nabla \mathcal{L}(w_t) \circ \nabla \mathcal{K}(s_t) \ge 0$.

Thus:

$$
\Delta_{\mathcal{L}}^{\text{Grams}}(w_t) - \Delta_{\mathcal{L}}^{\text{C}}(w_t) \le 0,
$$

which implies:

$$
\Delta_{\mathcal{L}}^{\text{Grams}}(w_t) \le \Delta_{\mathcal{L}}^{\text{C}}(w_t).
$$

Thus we complete the proof. □

F Global Convergence

In this subsection, we establish the global convergence properties of the Grams optimizer. By analyzing the update rules and assumptions on the optimization landscape, we demonstrate that Grams converges to a stationary point of the objective function. This analysis underscores the optimizer's robustness and effectiveness in a wide range of optimization scenarios.

F.1 Assumptions

To ensure theoretical rigor, we base our analysis on the following standard assumptions commonly used in optimization theory. These assumptions define the properties of the loss function and the optimization setting, enabling precise derivations of convergence guarantees.

Assumption F.1 (Lower bound of loss)**.** *The Loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is differentiable and closed within its open domain $\mathrm{dom}(\mathcal{L}) \subseteq \mathbb{R}^d$ and is bounded from below, i.e., $\mathcal{L}^* := \inf_w \mathcal{L}(w) > -\infty$.*

Assumption F.2 (Bounded gradient)**.** *The Loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ satisfies $\nabla \mathcal{L}(w) \le G$ for all $w \in \mathrm{dom}(\mathcal{L})$.*

Assumption F.3 (L-smooth)**.** *The Loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth for some $L > 0$.*

Assumption F.4 (μ-PL-condition)**.** *The Loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ satisfies μ-PL-condition for some $\mu > 0$.*

F.2 CONVERGENCE

In this subsection, we provide a detailed analysis of the convergence properties of the Grams optimizer. We begin by revisiting the convergence guarantee of the widely-used Adam optimizer as established in Li et al. (2023). Using this as a foundation, we extend the analysis to Grams, highlighting its enhanced convergence behavior under the same assumptions.

Lemma F.5 (Convergence of Adam, Section 5.3 in Li et al. (2023))**.** *Suppose that Assumptions F.1, F.2, and F.3 hold. Given initial weight w_1 with initial optimality gap $\Delta_1 := \mathcal{L}(w_1) - \mathcal{L}^* < \infty$, choose an large enough G such that $G \geq \max\{\epsilon, 3\sqrt{L\Delta_1}\}$, a small enough fixed step size $\eta > 0$, and $\beta = \Theta(\eta G^{1/2})$. Consider that the weight w_t is updated by Adam for each $t \in [T]$. Then we have*

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla \mathcal{L}(w_t)\|_2^2 \leq \frac{8G\Delta_1}{\eta T}.$$

The result in Lemma F.5 establishes a baseline for the convergence of Adam under standard assumptions. Building on this, we extend the analysis to Grams by leveraging its unique update mechanism, which decouples the direction and magnitude of updates. The following theorem demonstrates that Grams achieves global convergence, meaning that it is guaranteed to reach the optimal objective value from any initial point with finite initial optimality gap.

Theorem F.6. *Suppose that Assumptions F.1, F.2, F.3 and F.4 hold. Given initial point w_1 with initial optimality gap $\Delta_1 := \mathcal{L}(w_1) - \mathcal{L}^* < \infty$, choose large an enough G such that $G \geq \max\{\epsilon, 3\sqrt{L\Delta_1}\}$, a small enough fixed step size $\eta > 0$, and $\beta = \Theta(\eta G^{1/2})$. Consider that the weight w_t is updated by Grams (Algorithm 1) for each $t \in [T]$. Then we have*

$$\mathcal{L}(w_T) - \mathcal{L}^* \leq \frac{4G}{\mu\eta T}(\mathcal{L}(w_1) - \mathcal{L}^*).$$

Proof. Given initial weight w_1, we denote $w'_1, w'_2, \dots, w'_T$ be the weights updated by Adam where $w'_1 := w_1$. By Lemma F.5, we have

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla \mathcal{L}(w'_t)\|_2^2 \leq \frac{8G\Delta_1}{\eta T}. \tag{14}$$

For each $t \in [T]$, we can show that

$$\begin{aligned}
\|\nabla \mathcal{L}(w'_t)\|_2^2 &\geq 2\mu(\mathcal{L}(w'_t) - \mathcal{L}^*) \\
&= 2\mu(\mathcal{L}(w'_t) - \mathcal{L}(w'_{t-1}) + \mathcal{L}(w'_{t-1}) - \mathcal{L}(w'_{t-2}) + \cdots + \mathcal{L}(w'_2) - \mathcal{L}(w'_1) + \mathcal{L}(w'_1) - \mathcal{L}^*) \\
&= 2\mu(\Delta\mathcal{L}_{w'_t, w_{t-1}} + \Delta\mathcal{L}_{w'_{t-1}, w_{t-2}} + \cdots + \Delta\mathcal{L}_{w'_2, w_1} + \mathcal{L}(w'_1) - \mathcal{L}^*) \\
&\geq 2\mu(\Delta\mathcal{L}_{w_t, w_{t-1}} + \Delta\mathcal{L}_{w_{t-1}, w_{t-2}} + \cdots + \Delta\mathcal{L}_{w_2, w_1} + \mathcal{L}(w'_1) - \mathcal{L}^*) \\
&= 2\mu(\Delta\mathcal{L}_{w_t, w_{t-1}} + \Delta\mathcal{L}_{w_{t-1}, w_{t-2}} + \cdots + \Delta\mathcal{L}_{w_2, w_1} + \mathcal{L}(w_1) - \mathcal{L}^*) \\
&\geq 2\mu(\mathcal{L}(w_t) - \mathcal{L}(w_{t-1}) + \mathcal{L}(w_{t-1}) - \mathcal{L}(w_{t-2}) + \cdots + \mathcal{L}(w_2) - \mathcal{L}(w_1) + \mathcal{L}(w_1) - \mathcal{L}^*) \\
&= 2\mu(\mathcal{L}(w_t) - \mathcal{L}^*),
\end{aligned} \tag{15}$$

where the first step follows from Assumption F.4, the second step follows from basic algebra, the third step follows from Definition 2.3, the fourth step follows form Theorem D.3, the fifth step follows from $w'_1 = w_1$, the sixth step follows from Definition 2.3, and the last step follows from basic algebra.

Combining Eq. (14) and Eq. (15) gives

$$\mathcal{L}(w_T) - \mathcal{L}^* \leq \frac{4G}{\mu\eta T}(\mathcal{L}(w_1) - \mathcal{L}^*).$$

Thus we complete the proof.

□

G Experiments Details

For the Lion and C-Lion optimizers, we set the learning rate to $\frac{1}{10} \times$ Adam learning rate, as recommended in Chen et al. (2024).

G.1 Pre-Training

For the pre-training experiments with Llama 3.2 60M Dubey et al. (2024), we used the first $2,048,000$ rows of training data from the English section of the C4 dataset Raffel et al. (2020). We used the first $10,000$ rows of validation data from the English section of the C4 dataset for evaluation. Table 5 provides a detailed summary of the hyperparameters employed.

Table 5: Hyperparameters for Llama 3.2 60M pre-training experiments.

Optimizers	**Grams/AdamW/CAdamW**	**Lion/CLion**
	Training	
Epoch	1	1
Learning Rate	6e-3	6e-4
Weight Decay	0.0	0.0
Batch Size	2048	2048
Model Precision	BF16	BF16
Mix Precision	BF16&TF32	BF16&TF32
Scheduler	Constant with warm-up	Constant with warm-up
Warm-up Steps	50	50
Grad Clipping	1.0	1.0
β_1	0.9	0.9
β_2	0.95	0.95
ϵ	1e-6	1e-6
Seq-len	256	256
	Evaluating	
Precision	BF16	
Seq-len	256	

For the computer vision experiments, we used the CIFAR-10 dataset Krizhevsky (2009) to train and evaluate the WideResNet-50-2 model Zagoruyko & Komodakis (2016). Table 6 outlines the corresponding hyperparameters.

G.2 Fine-Tuning

For fine-tuning experiments of the Llama 3.2 1B model, Table 7 provides the detailed hyperparameters.

For PEFT of the Llama 3.2 3B model, Table 7 provides the detailed hyperparameters.

Table 6: Hyperparameters for WideResNet-50-2 pre-training experiments.

Optimizers	**Grams/AdamW/CAdamW**	**Lion/CLion**
Training		
Epoch	10	10
Learning Rate	2e-3	2e-4
Weight Decay	0.0	0.0
Batch Size	128	128
Model Precision	FP32	FP32
Mix Precision	None	None
Scheduler	Linear	Linear
Warm-up Steps	100	100
Grad Clipping	1.0	1.0
β_1	0.9	0.9
β_2	0.999	0.99
ϵ	1e-6	1e-6
Evaluating		
Precision	FP32	

Table 7: Hyperparameters for Llama 3.2 1B fine-tuning experiments.

Optimizers	**Grams/AdamW/CAdamW**
Training	
Epoch	1
Learning Rate	1e-4
Weight Decay	0.0
Batch Size	64
Model Precision	BF16
Mix Precision	BF16&TF32
Scheduler	Cosine
Warm-up Ratio	0.03
Grad Clipping	1.0
β_1	0.9
β_2	0.999
ϵ	1e-6
Seq-len	512
Evaluating	
Precision	BF16
Seq-len	1024

Table 8: Hyperparameters for Llama 3.2 3B PEFT experiments.

Optimizers	**Grams/AdamW/CAdamW**
Training	
Epoch	1
Learning Rate	1e-4
Weight Decay	0.0
Batch Size	128
Model Precision	BF16
Mix Precision	BF16&TF32
Scheduler	Cosine
Warm-up Ratio	0.03
Grad Clipping	1.0
β_1	0.9
β_2	0.999
ϵ	1e-6
Seq-len	512
Rank	128
SORSA Cao (2024) γ	1e-3
Evaluating	
Precision	BF16
Seq-len	2048