MGUP: A Momentum-Gradient Alignment Update Policy for Stochastic Optimization

Da Chang[12], Ganzhao Yuan[123]

1:Shenzhen Institute of Advanced Technology, China;

2:Pengcheng Laboratory, China;

3:Shenzhen University of Advanced Technology, China.

Motivation

- Efficient optimization is critical for training today's large-scale models.

- Recent methods explore selective updates, like **freezing layers, but fine-grained, intra-layer control is still an open area.**

- Existing methods that use this alignment, like Cautious Optimizers, are intuitive but lack theoretical convergence guarantees in the complex stochastic optimization setting. This creates a gap between a good idea and a reliable tool.

Preliminaries on Selective Updates

- The "Cautious" Approach to Optimization
- Consider the standard stochastic optimization problem:

$$\min f(\mathbf{x}) = \mathbb{E}_{\xi \sim D}[f(\mathbf{x};\xi)]$$

- **Key Idea: Cautious Optimizers**
  - This strategy selectively applies updates based on a simple rule:

Is the momentum moving in the same direction as the current gradient?

  - Update Rule
    - $$\phi_t = \alpha \cdot \mathbb{I}(\mathbf{m}_t \odot \mathbf{g}_t > 0)$$ $$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{m}_t \odot \phi_t$$
    - This means updates are either applied (often scaled up) or completely skipped (set to zero)

The Pitfall of Zeroing Out Updates

- Completely nullifying updates for misaligned parameters (i.e., setting the step size decay factor $\gamma = 0$) can be catastrophic and cause the optimizer like **Adam** to fail to converge
- We consider a counterexample in previous literature(***Zhang et al.***):

$$\boxed{\begin{aligned} f_i(x) &= \begin{cases} nx, & x \geq -1 \\ \frac{n}{2}(x+2)^2 - \frac{3n}{2}, & x < -1 \end{cases} \quad \text{for} \quad i = 0 \\ f_i(x) &= \begin{cases} -x, & x \geq -1 \\ \frac{n}{2}(x+2)^2 - \frac{3n}{2}, & x < -1 \end{cases} \quad \text{for} \quad i > 0 \end{aligned}}$$

- This is a classical finite sum problem in stochastic optimization: $f(x) = \sum_{i=0}^{n-1} f_i(x)$

Zhang et al. Adam can converge without any modification on update rules. NeurIPS 2022.

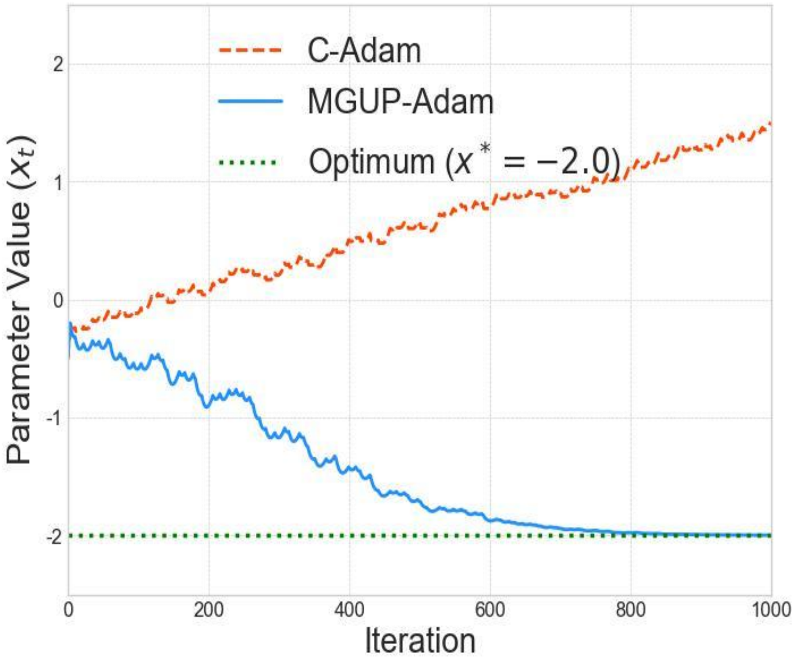
Counterexample

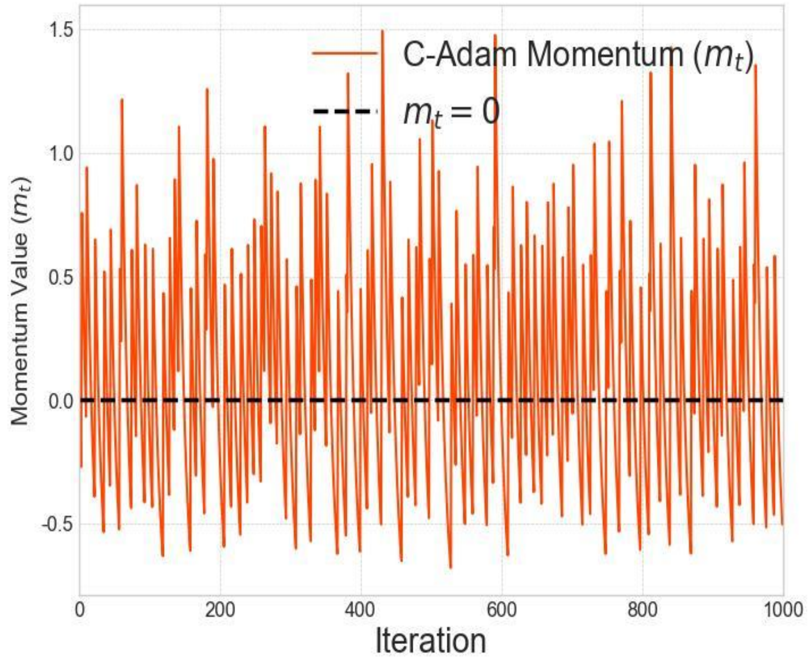
$$f_i(x) = \begin{cases} nx, & x \geq -1 \\ \frac{n}{2}(x+2)^2 - \frac{3n}{2}, & x < -1 \end{cases} \quad \text{for} \quad i = 0$$

$$f_i(x) = \begin{cases} -x, & x \geq -1 \\ \frac{n}{2}(x+2)^2 - \frac{3n}{2}, & x < -1 \end{cases} \quad \text{for} \quad i > 0$$

Init: $x = -0.5$

Optimum: $x = -2$





Unified momentum-gradient aligned mask

$$\phi_t = \begin{cases} \alpha, & \mathbf{m}_{t,i} \cdot \mathbf{g}_{t,i} > 0 \\ \gamma, & \mathbf{m}_{t,i} \cdot \mathbf{g}_{t,i} \leq 0 \end{cases}$$

where $\alpha > 1$ and $\gamma \in [0,1]$

Red Line (Diverging): Cautious Adam ($\gamma = 0$)

Blue Line (Converging): MGUP-Adam ($\gamma > 0$)

Setting misaligned updates to zero can cause divergence.

How to Ensure Convergence? A Differentiated Greedy Update

- **The MGUP Mechanism:** Instead of a binary "update vs. no update" decision, we propose a "major update vs. minor update" strategy.
- **Update Rule:**
  - **Step 1(Score):** Calculate alignment scores $\mathbf{s}_{t,i} = \mathbf{m}_{t,i} \cdot \mathbf{g}_{t,i}$ for all parameters
  - **Step 2(Rank):** Identify the top K parameters with the highest scores
  - **Step 3(Differentiate):**
    - Apply a larger step size ($\alpha \cdot \eta_t$, where $\alpha > 1$) to the top K parameters
    - Apply a smaller, but non-zero, step size ($\gamma \cdot \eta_t$ where $0 < \gamma < 1$) to the rest

MGUP Converges Where Others Fail

Theorem 1 & 2 (Simplified): For MGUP-AdamW (without weight decay), **under standard assumptions in a stochastic setting**, we provide rigorous convergence guarantees

Theorem 1 $\min_{t=1,\cdots,T} \mathbb{E}[\| \nabla f(\mathbf{x}_{t+1})\|_2^2] \leq \hat{G}$

where $\hat{G} = \frac{3L^2\eta^2 + 3\rho^2\epsilon^2}{\rho^2\epsilon^2 T}\left(\frac{f(\mathbf{x}_1) - f(\mathbf{x}^*) + 2\sigma^2 L^{-1}\log(T+1)}{\varepsilon_{\min}}\sqrt{T} - 2(\sqrt{T} - 1)\right)$

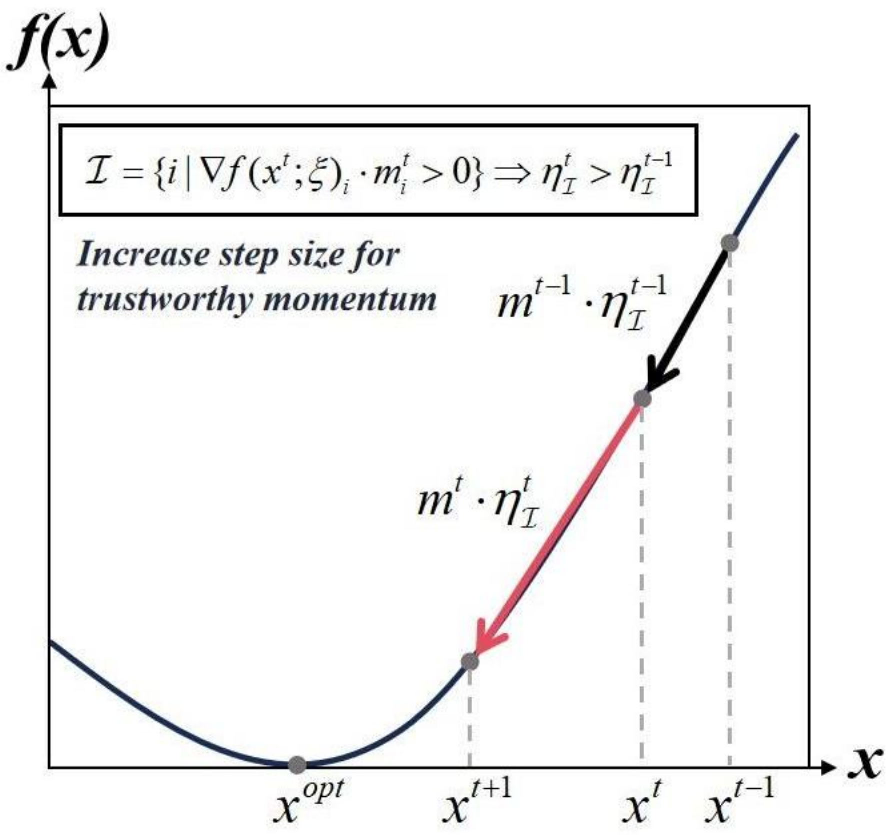
Theorem 2 For any given $\delta \in (0, 1/2)$, it holds that with probability at least $1 - 2\delta$,

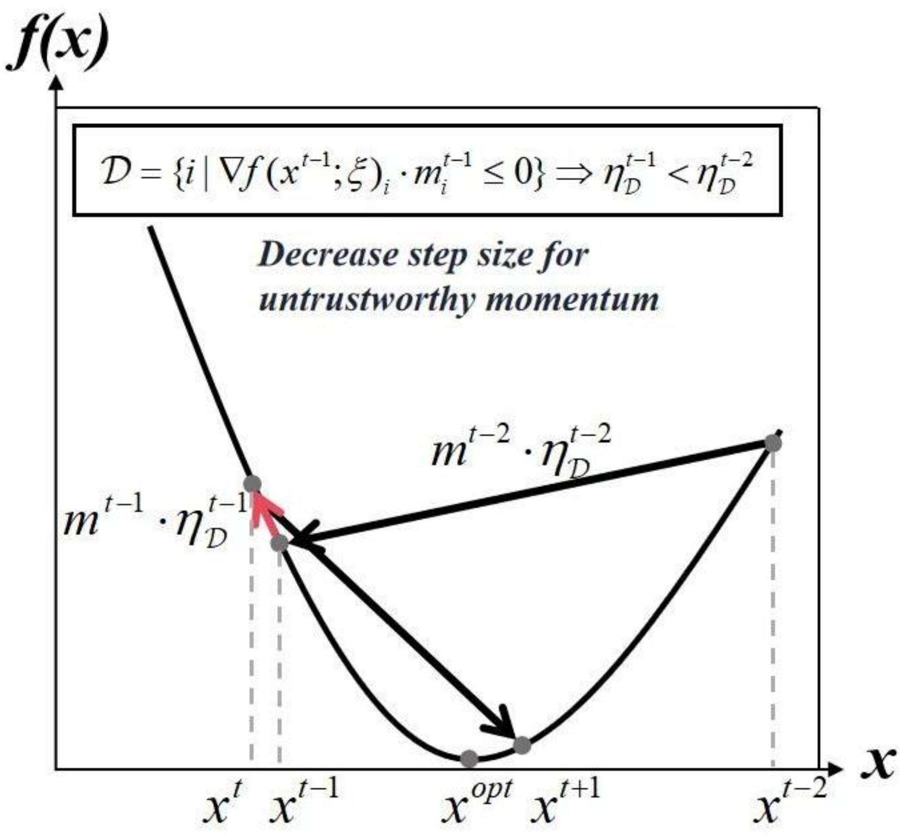
$\frac{1}{T}\sum_{s=1}^{T} \| \nabla f(\mathbf{x}_s)\|_2^2 \leq \widetilde{\mathcal{O}}(T^{-1/2}).$

Remark 3 $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta_t \phi_t \odot \frac{\mathbf{g}_t}{\mathbf{b}_t} + \frac{\beta_1}{1-\beta_1}\left(\frac{\eta_t \mathbf{b}_{t-1} \boxed{\odot \phi_t}}{\eta_{t-1}\mathbf{b}_t \boxed{\odot \phi_{t-1}}} - \mathbf{1}_d\right) \odot (\mathbf{x}_t - \mathbf{x}_{t-1})$

Not updating in the previous step causes the denominator to divide by zero

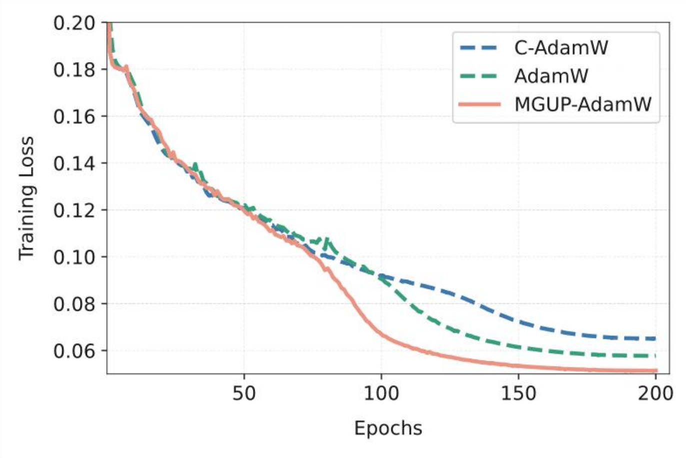
The MGUP Intuition: Greedy with Trustworthy vs. Untrustworthy Momentum

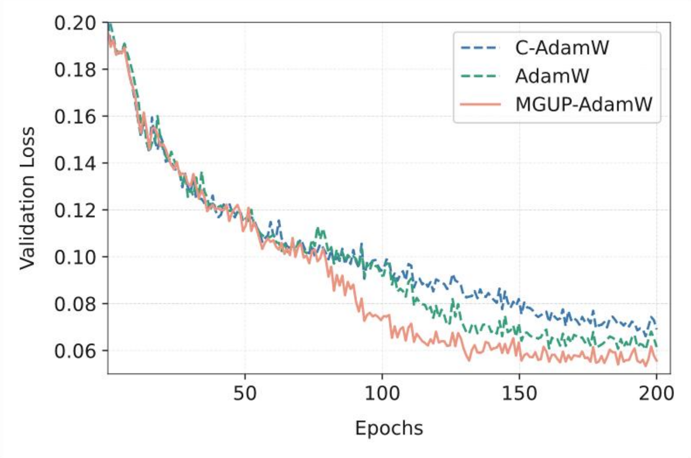




Experimental Validation

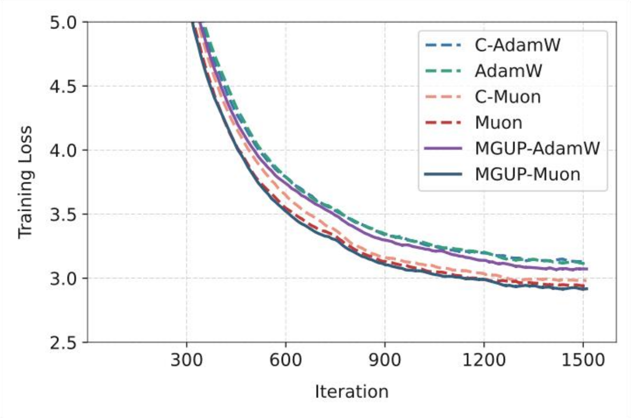
ViT-27M MAE Pre-traing on CIFAR10

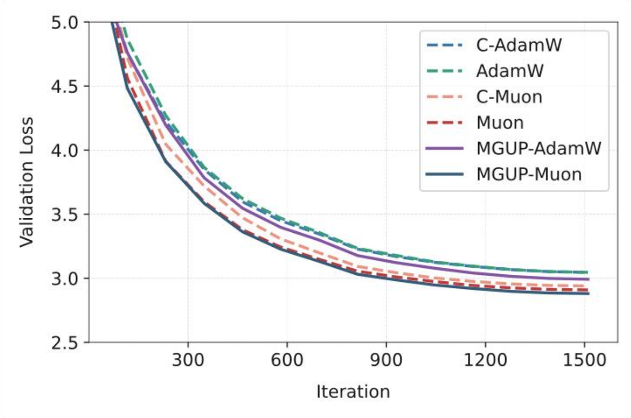


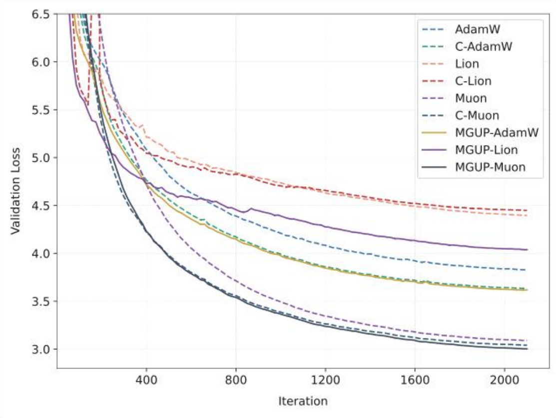


MGUP-AdamW achieved better training loss and validation loss during the training process. In contrast, C-AdamW may be gradually inferior to AdamW.

Qwen2.5-150M/LLaMA2-71M Pre-traing on Wikitext-103



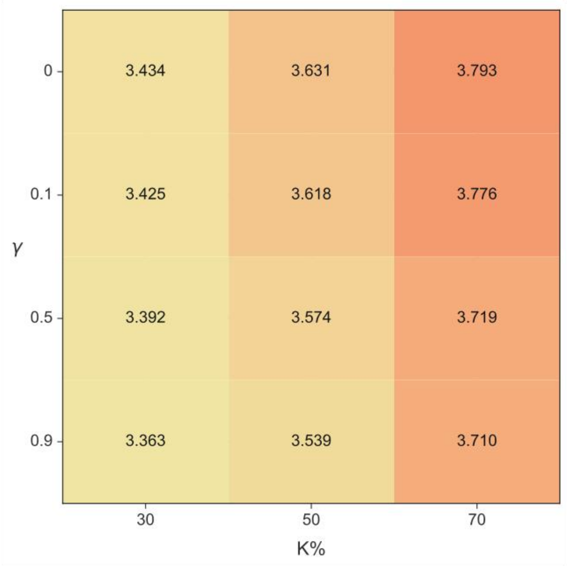




MGUP-AdamW demonstrated a higher speedup than standard AdamW and better generalization than C-AdamW.

Experimental Validation

Hyperparameter sensitivity analysis on Wikitext-103

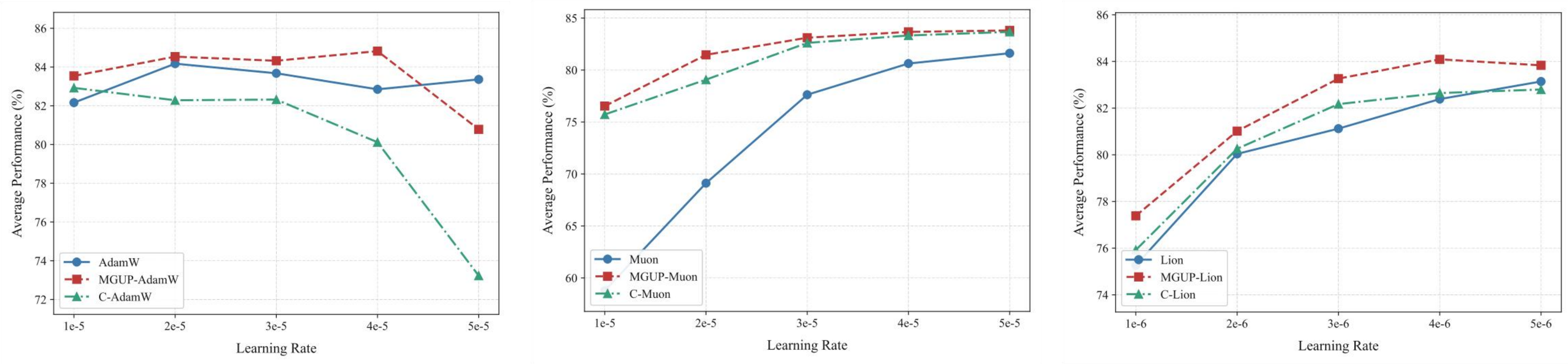


While the optimal value may vary slightly depending on the specific setting (e.g., **task type, model size, or base optimizer**), strong performance is typically maintained within the range $\tau \in [0.3, 0.7]$, suggesting that extensive tuning is unnecessary in practice.

$$\phi_t = \begin{cases} 1/\tau, & \mathbf{m}_{t,i} \cdot \mathbf{g}_{t,i} > 0 \\ \tau, & \mathbf{m}_{t,i} \cdot \mathbf{g}_{t,i} \leq 0 \end{cases}$$

Cautious-MGUP can be used as an adaptive alternative in practical tasks！

RoBERTa Fine-Tuning on GLUE



Implications for Practitioners

- **Use MGUP with confidence: It is a theoretically justified method to improve your existing AdamW, Lion, or Muon optimizers.**
- **A simple way to boost performance: By adding MGUP, you can often achieve faster convergence and better generalization without complex changes to your training pipeline.**
- **Robust Starting Point: The default setting of $\tau = 0.5$ (updating the top 50% of parameters aggressively and the bottom 50% cautiously) proved effective across our diverse experiments and is a great place to start.**