Cautious Optimizers

Kaizhao Liang* Lizhang Chen* Bo Liu Qiang Liu

University of Texas at Austin

One Line of Code. Any Optimizer. Faster Training.

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \epsilon_t \cdot \mathbf{u}_t \odot \phi(\mathbf{u}_t \odot \mathbf{g}_t), \qquad \phi(\mathbf{x}) = \frac{\dim(\mathbf{x})}{\mathrm{nnz}(\mathbf{x} > 0) + \xi} \cdot \mathbb{I}(\mathbf{x} > 0)$$

Mask momentum updates that disagree with the gradient. Drop-in for AdamW, Lion, LaProp, MARS. No new hyperparameters.

The Problem with Momentum

- ✗ **Momentum fights the gradient** — update opposes gradient on some coordinates
- ✗ **Causes oscillations** — loss temporarily increases, wasting compute
- ✗ **No monotonic loss guarantee** — only the Hamiltonian decreases

Cautious Optimizers

- ✓ **One-line change**, zero new hyperparameters
- ✓ **Masks misaligned coordinates** — only updates where $\mathbf{u}_t$ and $\mathbf{g}_t$ agree
- ✓ **Universal** — works with AdamW, Lion, LaProp, MARS
- ✓ **Provably stable** (Lyapunov) + monotonic loss decrease

The One-Line Change (PyTorch)

```
# Standard optimizer update:
p.add_(update, alpha=-lr)
# Cautious optimizer update:
mask = (update * grad > 0)
p.add_(update * mask / (mask.mean() + eps),
alpha=-lr)
```

Lyapunov Stability Guarantee

Theorem. For any momentum optimizer with Hamiltonian $\mathcal{H}$, the cautious mask ϕ guarantees: **(1)** $\mathcal{H}$ **decreases faster** than the original; **(2)** loss $\mathcal{L}$ **decreases monotonically**; **(3) convergence preserved** to stationary points. Applies to GDM, Adam, Lion, and any optimizer with Hamiltonian + dissipation structure.

Downstream Evaluation (1.2B, LM Eval-Harness)

Task	Metric	AdamW	C-AdamW
ARC-Easy	acc	0.608	**0.609**
ARC-Challenge	acc_norm	0.288	**0.298**
HellaSwag	acc_norm	0.417	**0.419**
OpenBookQA	acc	0.234	**0.236**
MMLU	acc	0.253	**0.254**
Lambada	acc	**0.331**	0.323
PIQA	acc_norm	**0.677**	0.677

C-AdamW wins **5 of 7** benchmarks. Bold = better.

Scan for paper

2D Toy: Cautious Removes Oscillations

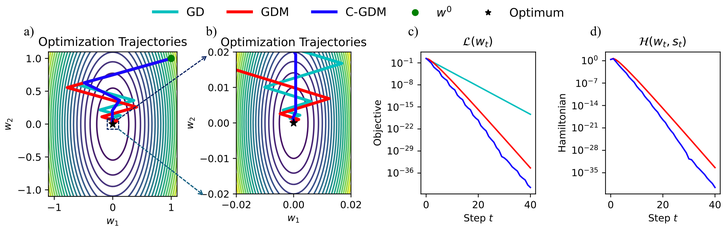

Figure 1. GD (cyan), GDM (red), C-GDM (blue) on $\mathcal{L}(\mathbf{w}) = 4w_1^2 + w_2^2$. C-GDM eliminates overshooting and ensures **monotonic** decrease of both $\mathcal{L}$ and $\mathcal{H}$.

Post-Training: Instruction Tuning & RLHF

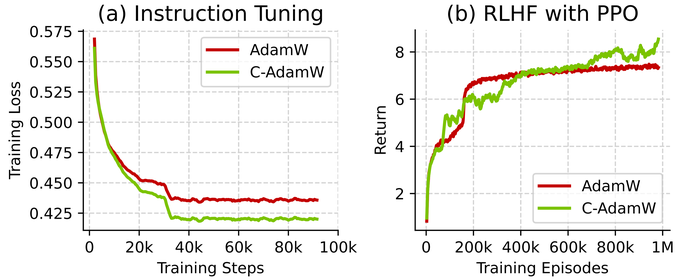

Figure 2. C-AdamW (green) vs. AdamW (red). Faster instruction tuning and higher RLHF return with PPO.

Why Does It Work?

During training, $\sim$45% of coordinates have $\mathbf{u}_t \odot \mathbf{g}_t < 0$ (momentum opposes gradient). Masking these guarantees:

$$\mathcal{L}(\mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{w}_t) \approx -\epsilon_t (\mathbf{u}_t \odot \mathbf{g}_t)^\top \phi(\mathbf{u}_t \odot \mathbf{g}_t) \leq 0$$

Every step decreases the loss — no wasted oscillations.

LLM Pretraining (C4, 100M LLaMA)

lr	1e-4	3e-4	1e-3	3e-3	1e-2	2e-2
AdamW	85.1	24.4	19.2	19.0	**19.0**	19.6
C-AdamW	–	–	19.1	18.8	**18.7**	18.8

lr	1e-4	3e-4	6e-4	1e-3
Lion	28.3	**21.4**	22.0	†
C-Lion	21.4	**19.8**	20.4	21.0

Eval perplexity (↓). 50B tokens (25× Chinchilla). †diverged.
Cautious is robust across lr and tolerates higher learning rates.

Scaling Results (FineWeb-Edu, 1× Chinchilla)

Scale	AdamW	C-AdamW	Improvement
130M	27.39	**27.30**	0.33%
300M	18.30	**18.28**	0.10%
520M	15.07	**14.92**	1.00%
1.2B	11.36	**11.32**	0.32%

Perplexity (↓). C-AdamW consistently outperforms at every scale.

Image Classification (Mini-ImageNet, ViT)

Method	Top-1 Acc.	Δ
AdamW → C-AdamW	72.11 → **73.52**	+1.41
LaProp → C-LaProp	71.73 → **73.92**	+2.19
MARS → C-MARS	74.06 → **74.91**	+0.85

Cautious variants improve *every* base optimizer.

Key Takeaway

Cautious optimizers are a **one-line, hyperparameter-free** change that consistently improves AdamW, Lion, LaProp, and MARS across pretraining, vision, and post-training.