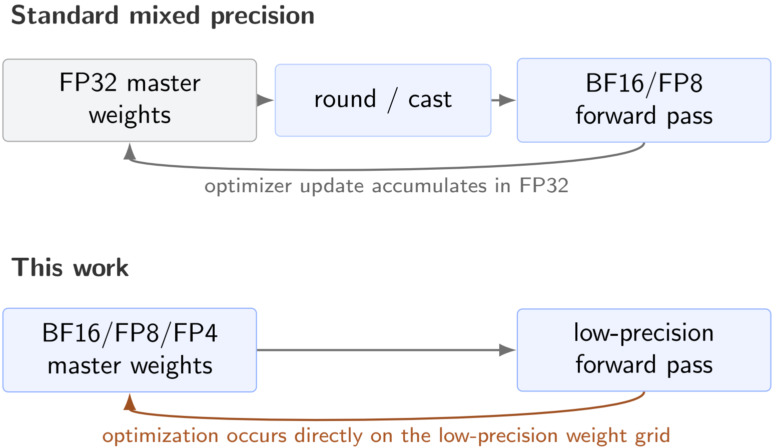
# M+Adam: Low-Precision Training via Additive–Multiplicative Optimization

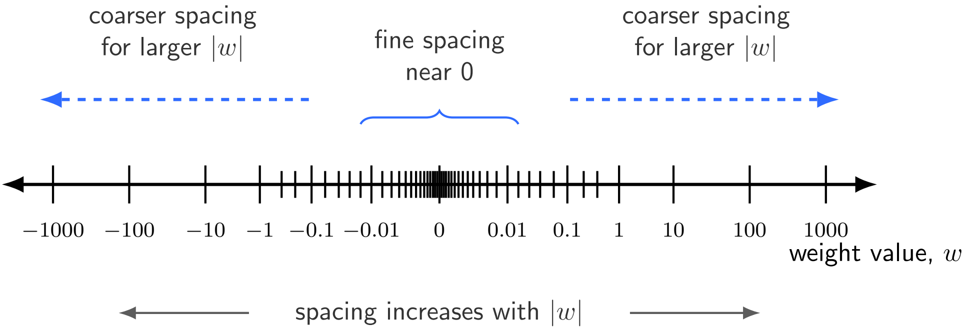
Xiaoyuan Liang[1], Sebastian Loeschcke[2], Mads Toftrup[3], Anima Anandkumar[1]

[1]California Institute of Technology [2]University of Copenhagen [3]Aarhus University

# Mixed precision usually hides updates from rounding



# Low precision makes update size depend on weight scale



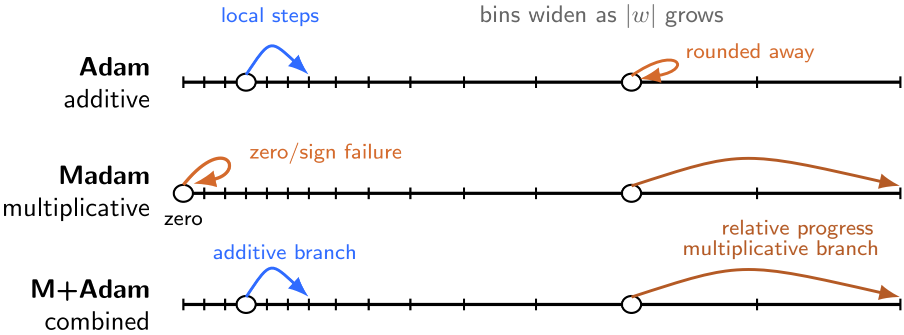
# Additive and multiplicative updates fail differently

## Key idea

M+ADAM updates each weight by combining two kinds of movement:

$$\text{new weight} = \text{old weight} + \underbrace{\text{relative scale change}}_{\text{multiplicative branch}} + \underbrace{\text{local displacement}}_{\text{additive branch}}.$$

$$w_{t+1} = w_t + \underbrace{w_t\, u_t^{(m)}}_{\text{multiplicative update}} + \underbrace{u_t^{(a)}}_{\text{additive update}}.$$



# Use additive updates near zero and multiplicative updates at scale

## Additive branch

Adam-style local correction in weight space:

$$u^{(a)} \approx -\eta_a \frac{\hat{u}}{\sqrt{\hat{v}} + \epsilon}.$$

Use it for small weights, zero revival, and sign changes.

## Multiplicative branch

Madam-style relative scale movement:

$$g^{(e)} = (\ln 2)\, wg, \qquad w \mapsto w(1 + u^{(m)}).$$

Use it when additive updates vanish in wide bins.

## One combined update

$$w_{t+1} = w_t + w_t u^{(m)} + u^{(a)}.$$

# Complementary failures motivate M+Adam

| | AdamW | Madam | M+Adam |
|---|---|---|---|
| Additive updates | ✓ | × | ✓ |
| Multiplicative updates | × | ✓ | ✓ |
| Can escape zero | ✓ | × | ✓ |
| Non-zero update at large $\lvert w \rvert$ | × | ✓ | ✓ |

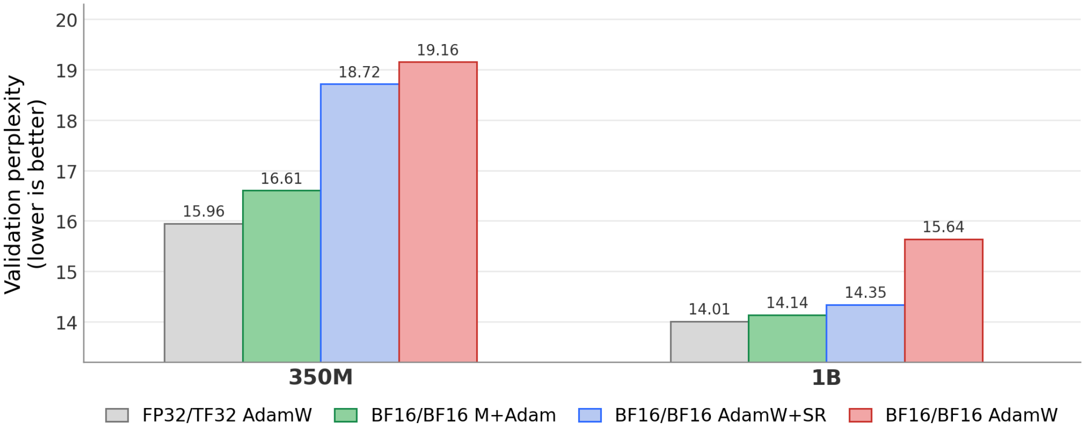
### Where baselines fail

AdamW steps can round away in coarse bins; Madam cannot revive exact zeros or cross signs.

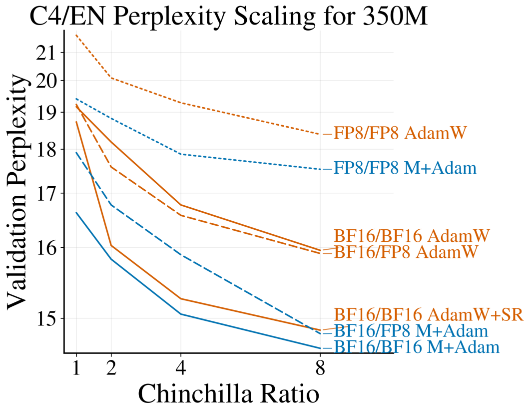
### Where M+Adam improves

Additive updates fix zero/sign/small-weight cases; multiplicative updates preserve large-scale progress.

# M+Adam reduces the BF16 gap to FP32



# Gains persist across longer training budgets




## Representative panel

- 350M model
- 1–8× Chinchilla budgets
- BF16/BF16, BF16/FP8, FP8/FP8 regimes

**Broader sweep.** The same budget sweep is repeated for 60M and 130M models.

# Main limitations are state and scale

### Current limitations

- Adds one multiplicative second-moment state.
- Full $1$–$8\times$ budget sweep is up to 350M.
- Focuses on optimizer stability, not throughput engineering.

### Next directions

- Compress the extra state more aggressively.
- Scale to larger models and longer training.
- Test broader downstream tasks.

# Low-precision training needs low-precision-aware updates

**Adam**

Good local additive correction, but updates can vanish in wide bins.

**Madam**

Good scale-aware progress, but fails at zeros and sign changes.

**M+Adam**

Combines additive refinement with multiplicative scale updates.

M+Adam makes direct BF16/FP8/FP4 weight updates more stable and accurate.