A Fix for Softmax and Cross-Entropy

Why tiny gradients disappear – and how to retain them

Paper: Jonathan T. Barron, Dan Gnanapragasam, Michael Tschannen

Research group discussion

arXiv:2610.31642v1 22 Oct 2026

Publication not found as of 27 Oct 2026 (author listing: arXiv). Author JAX examples: project page; no standalone official repository found.

Two errors at opposite ends of the gradient scale

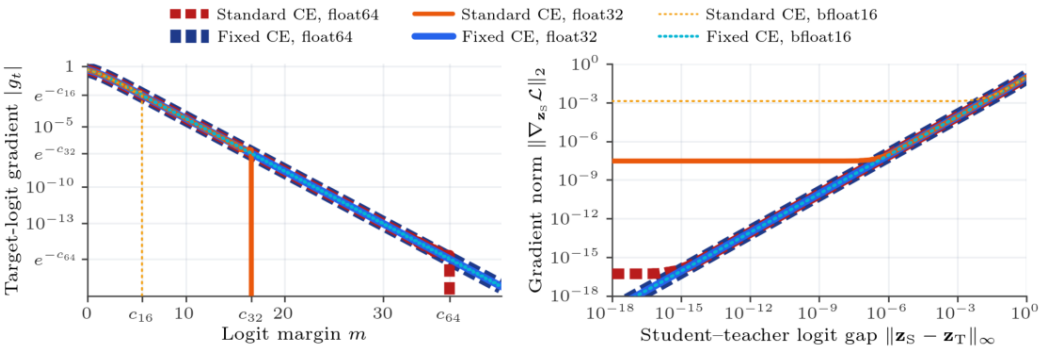
One-sentence takeaway

Stable exponentiation does not imply stable gradients: preserve small probability mass and compute near-equal probability differences from logits.

- **Collapse:** high confidence rounds the target gradient to zero.
- **Cancellation:** near teacher–student agreement creates a gradient noise floor.

Paper, Sec. 1, Fig. 1, PDF p. 1.

The two failure regimes are visible numerically



(a) Softmax Collapse: $|g_t|$ vs. margin m.

(b) Softmax Cancellation: $\|\nabla_{\mathbf{z}_S}\mathcal{L}\|_2$ as $\mathbf{z}_S \to \mathbf{z}_T$.

- Left: $|g_t|$ should decay smoothly as margin m grows.
- Right: $\|\nabla_{z_S}\mathcal{L}\|_2$ should vanish as $z_S \to z_T$.

Paper, Fig. 1, PDF p. 1. Original PDF figure crop.

Cross-entropy gradients must sum to zero

$$q = \mathrm{softmax}(z), \qquad \mathcal{L}(z, p) = -\sum_k p_k \log q_k, \qquad g = \nabla_z \mathcal{L} = q - p.$$

Translation invariance

$\mathcal{L}(z + c\mathbf{1}, p) = \mathcal{L}(z, p)$, hence $\sum_k g_k = 0$.

- Standard code subtracts $\max_k z_k$ to prevent exponential overflow.
- Rounding in later operations can still violate the gradient identity.

Paper, Sec. 1, Eq. (1), PDF p. 1; Sec. 2, PDF p. 2.

Collapse begins when $1 + e^{-m}$ rounds to 1

$$m = z_t - \log \sum_{j \neq t} e^{z_j}, \qquad q_t = \frac{1}{1 + e^{-m}}.$$

Standard target gradient

$\tilde{g}_t = q_t - 1 = 0$ after rounding.

Non-target gradients survive

$\tilde{g}_j \simeq e^{z_j - z_t} > 0.$

Precision cliffs in margin m: bf16 ≈ 5.5, fp32 ≈ 16.6, fp64 ≈ 36.7.

Paper, Sec. 2, Eq. (2)–(4), Fig. 1a, PDF pp. 1–2.

A three-class example exposes the missing component

fp32, $z = (9, -9, -9)$, target $t = 0$

$e^{-m} = 2e^{-18} \approx 3.046 \times 10^{-8}$: small near 1, representable near 0.

	Loss	Gradient (g_0, g_1, g_2)
standard CE	0	$(0,\ 1.523,\ 1.523) \times 10^{-8}$
fixed CE	3.046×10^{-8}	$(-3.046,\ 1.523,\ 1.523) \times 10^{-8}$

The standard result keeps both non-target entries but loses the target entry.

Paper, Sec. 2, numerical JAX example, PDF p. 3.

Fix hard-target CE by preserving the small mass

$$m = -\,\mathrm{LSE}_{j \neq t}(z_j - z_t), \qquad \boxed{\mathcal{L}(z,t) = \mathrm{softplus}(-m)}$$

$$g_j = \frac{e^{z_j - z_t}}{1 + e^{-m}} \;\; (j \neq t), \qquad \boxed{g_t = -\sum_{j \neq t} g_j}.$$

- Exclude the target from log-sum-exp; never form $q_t - 1$.
- Autodiff through $z_j - z_t$ recovers the negative sum at the target.

Paper, Sec. 2, Eq. (6)–(7), PDF p. 3.

The same construction fixes standalone log-softmax

Choose $i_{\max} = \arg\max_i z_i$ and define

$$\tilde{z}_k = \mathbf{1}[k \neq i_{\max}](z_k - z_{i_{\max}}), \quad m = -\operatorname{LSE}_{j \neq i_{\max}} \tilde{z}_j.$$

$$\boxed{\operatorname{logsoftmax}(z)_k = \tilde{z}_k - \operatorname{softplus}(-m)}$$

- Replace the maximum slot by exact zero; exclude it from the LSE.
- Then $\operatorname{softmax}(z) = \exp(\operatorname{logsoftmax}(z))$.
- A rounded forward probability of 1 can still have a useful gradient.

Paper, Sec. 2, Eq. (5), JAX listing, PDF p. 3.

Cancellation loses information before subtraction

$$p = \mathrm{softmax}(z_T), \quad q = \mathrm{softmax}(z_S), \quad \nabla_{z_S}\mathcal{L} = q - p.$$

- Standard soft-target CE evaluates p and q separately, then subtracts.
- For $z_S \approx z_T$, rounding probabilities erases the small difference.
- Different forward and backward formulas can leave $q - p \neq 0$ even at $z_S = z_T$.

Better input to the gradient calculation

Use the directly available $\delta = z_S - z_T$.

Paper, Sec. 3, Eq. (8), PDF pp. 3–4.

Compute $q - p$ from the logit gap

$$d_k = \delta_k - \sum_i p_i \delta_i, \qquad s_k = d_k - \operatorname{log1p}\left(\sum_i p_i \operatorname{expm1}(d_i) \right).$$

$$s_k = \log(q_k / p_k), \qquad u_k = p_k \operatorname{expm1}(s_k), \qquad g_k = u_k - q_k \sum_i u_i.$$

- $\operatorname{expm1}(x)$ and $\operatorname{log1p}(x)$ retain accuracy near zero.
- Centering makes $\sum_i p_i d_i = 0$; the final term reduces sum drift.
- At $\delta = 0$, every intermediate difference vanishes.

Paper, Sec. 3, Eq. (9), PDF p. 4.

The forward loss is unchanged; the backward path is replaced

Implementation pattern in the paper

Compute the usual soft-target CE value, then attach g as its custom gradient using `stop_gradient`.

- A small fp32 teacher–student perturbation produces a wrong sign in one standard gradient entry.
- The fixed gradient matches the fp64 reference in the paper's three-class example.
- At exactly equal logits, it returns zero and does not push the student away.

Paper, Sec. 3, JAX listing and numerical example, PDF p. 4.

Collapse creates Slingshot through Adam's memory

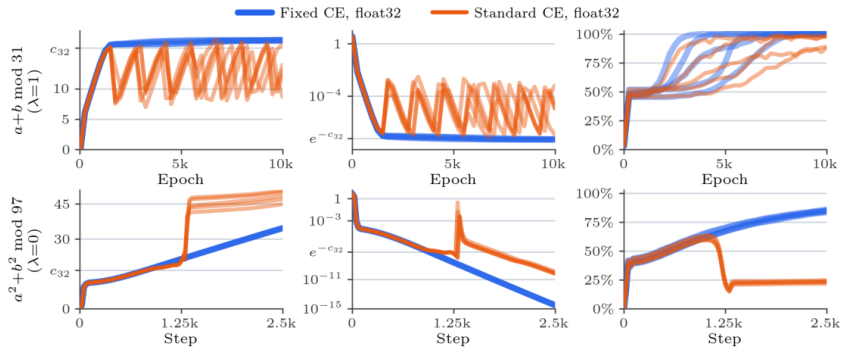
1. After memorization, margins cross the fp32 cliff; target gradients vanish.
2. Adam's second moment $\hat{v}$ decays while parameter updates nearly stop.
3. Some examples recross the cliff: restored gradients meet a small $\sqrt{\hat{v}} + \epsilon$.
4. The resulting large step spikes the loss and resets $\hat{v}$.

Prediction

Keeping $g_t \approx -e^{-m}$ smooth should remove this numerical cycle.

Paper, Sec. 4, PDF pp. 4–5. Mechanistic explanation proposed by the authors.

The reported Slingshot trajectories support that mechanism



(a) Mean logit margin m. (b) Training loss. (c) Test accuracy.

Paper, Fig. 2, PDF p. 5. Top: $a + b \bmod 31$, $\lambda = 1$; bottom: $a^2 + b^2 \bmod 97$, $\lambda = 0$. Original PDF figure crop.

The margin plateau comes from ϵ and weight decay

At an AdamW stationary point, $\hat{m}_i = g_i$ and $\sqrt{\hat{v}_i} = |g_i|$:

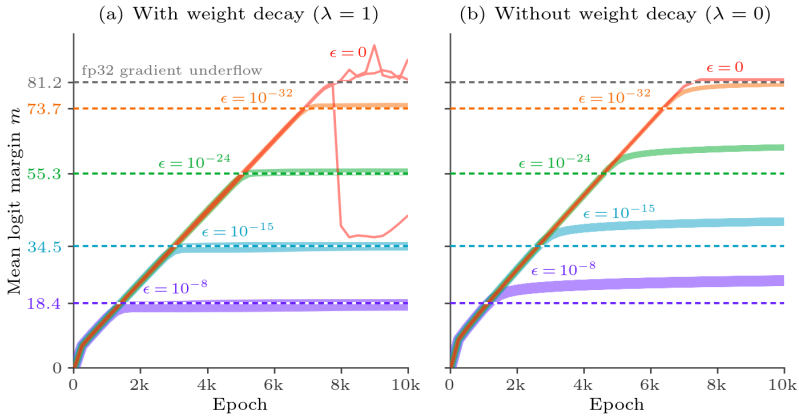
$$\frac{g_i}{|g_i| + \epsilon} + \lambda\theta_i = 0.$$

- After memorization, $|g_i| \propto e^{-m}$.
- Once $|g_i| \ll \epsilon$, the normalized update scales as e^{-m}/ϵ.
- In the studied regime, balance gives $m^*(\epsilon) \approx \log(1/\epsilon)$.

For $\epsilon = 10^{-8}$, $m^* \approx 18.4$, just above the fp32 cliff 16.6.

Paper, Sec. 5, Eq. (10)–(11), Fig. 3, PDF pp. 5–6. Approximation suppresses model-dependent factors.

The ϵ sweep separates equilibrium from a precision cliff



- With $\lambda = 1$, positive-ϵ margins level near $\log(1/\epsilon)$.
- With $\lambda = 0$, margins keep growing past that level.

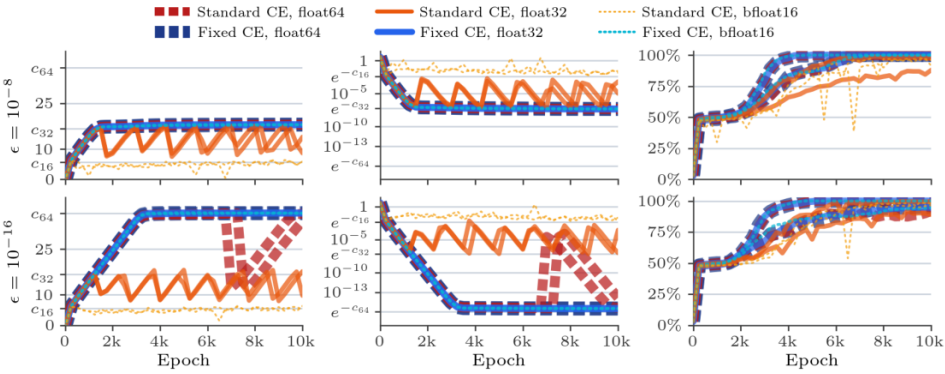
Paper, Fig. 4, PDF p. 6. Original author figure; LaProp updates $\sqrt{\hat{v}}$ directly.

Main grokking experiment varies loss, precision, and ϵ

Task	$a + b$ mod 31, 50% training split
Model	1-layer width-64 ReLU Transformer
Optimizer	full-batch AdamW, 10,000 epochs
Shared settings	$\eta = 10^{-3}$, $\beta = (0.9, 0.98)$, $\lambda = 1$
Variables	standard/fixed CE; bf16/fp32/fp64; $\epsilon = 10^{-8}, 10^{-12}, 10^{-16}$
Outcomes	test accuracy (higher better); final logit margin

Paper, Sec. 6, Fig. 5 and Table 1, PDF pp. 6–7.

The precision cliff shifts with the number format



(a) Mean logit margin m. (b) Training loss. (c) Test accuracy.

Paper, Fig. 5, PDF p. 7. Top: $\epsilon = 10^{-8}$; bottom: 10^{-16}. Original PDF figure crop.

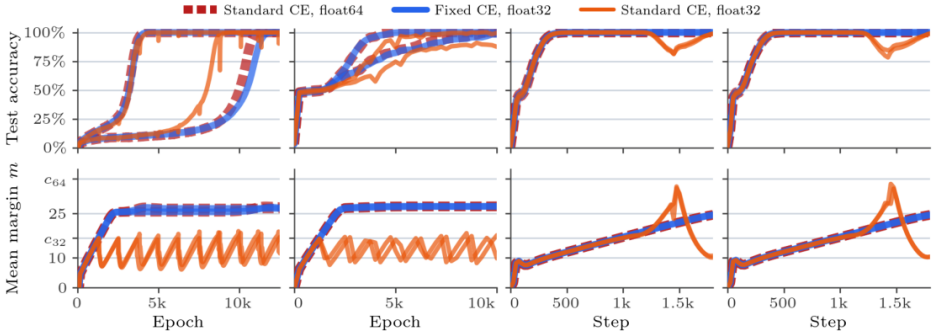
Fixed CE improves accuracy without changing the loss shape

Test accuracy (5 seeds)	$\epsilon = 10^{-8}$	$\epsilon = 10^{-16}$
fp32 fixed CE	$100.0 \pm 0.0\%$	$99.0 \pm 2.3\%$
fp32 standard CE	$94.3 \pm 5.4\%$	$96.5 \pm 4.3\%$
bf16 fixed CE	$100.0 \pm 0.1\%$	$98.9 \pm 2.4\%$
bf16 standard CE	$93.3 \pm 7.4\%$	$94.2 \pm 8.3\%$
fp64 fixed CE	$100.0 \pm 0.0\%$	$99.0 \pm 2.0\%$
fp64 standard CE	$100.0 \pm 0.0\%$	$97.2 \pm 3.2\%$

At $\epsilon = 10^{-12}$ in fp32: zero-sum projection $48.9 \pm 2.6\%$, StableMax $95.6 \pm 2.0\%$, fixed CE $99.6 \pm 0.7\%$.

Paper, Table 1, PDF p. 7. Accuracy in percent; $a + b \bmod 31$, $\lambda = 1$.

The effect repeats on four modular tasks



(a) $a+b$ mod 113 Nanda et al. (2023).

(b) $a+b$ mod 31 Varma et al. (2023).

(c) a^2+b^2 mod 97 Doshi et al. (2024).

(d) a^3+b mod 97 Doshi et al. (2024).

- Fixed fp32 CE tracks standard fp64 CE; standard fp32 CE repeatedly spikes.

Paper, Fig. 6, PDF p. 8. $\epsilon = 10^{-12}$; 1-layer Transformers and 2-layer MLPs. Original PDF figure crop.

Identical student and teacher make numerical drift measurable

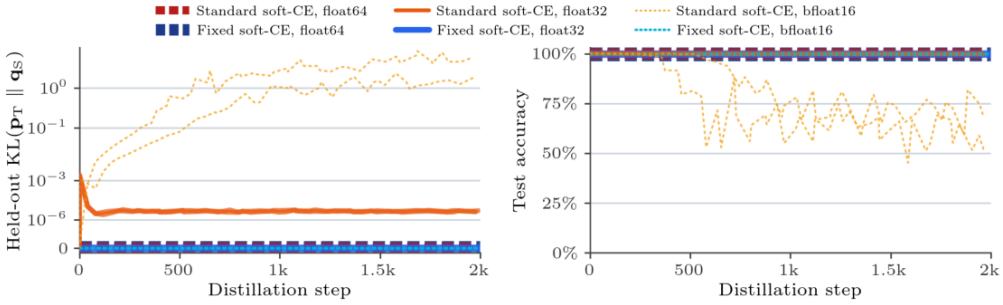
- Train teacher to 100% test accuracy on $a + b \bmod 53$.
- Copy teacher weights to student: $\theta_S^{(0)} = \theta_T$.
- Distill for 2,000 Adam steps: $\eta = 2 \times 10^{-3}$, $\lambda = 0$, $\epsilon = 10^{-8}$, temperature $T = 4$.
- Measure held-out $\mathrm{KL}(p_T \| q_S)$ and test accuracy.

The diagnostic

Since $z_S = z_T$ initially, the true student gradient is zero. Any move at initialization is numerical.

Paper, Sec. 7, PDF p. 8. $T = 4$ keeps scaled margins below the bf16 collapse cliff.

Fixed soft-target CE prevents teacher–student drift



(a) Held-out $\mathrm{KL}(\mathbf{p}_T \parallel \mathbf{q}_S)$.

(b) Test accuracy.

- Standard bf16 CE drives KL upward and degrades accuracy.
- Fixed CE stays at KL $= 0$ and 100% accuracy in all three precisions.

Paper, Fig. 7, PDF p. 8. Two of three seeds shown; original PDF figure crop.

Take-home messages and evidence boundary

1. Rewrite algebra before precision is lost: softplus for high confidence, expm1 / log1p for near equality.
2. This removes reported Slingshot spikes and distillation drift in the tested modular tasks.
3. The $m \approx \log(1/\epsilon)$ plateau comes from AdamW plus weight decay, not fp32's representable range.

Scope

The paper tests controlled small tasks and an exactly matched student–teacher initialization; it does not measure large-scale training speed or benefit.

Paper, Secs. 4–8, PDF pp. 4–9.

Paper and implementation sources

- Barron, Gnanapragasam, Tschannen. *A Fix for Softmax and Cross-Entropy*. arXiv:2610.31642v1, 2026.
- Author project page: paper text, JAX function examples, interactive demo.
- Author listing: labeled arXiv, 2026 as of 27 Oct 2026.

Numerical implementation detail

The current project-page soft-target snippet includes a fallback for large log-ratios to avoid $0 \cdot \infty$; check the intended dynamic range when copying code.