Milestone 005 — The unit-hypersphere anomaly

Tyler J.H.G.

Abstract

What happens when recurrent state keeps direction but discards magnitude?

Research period: September 1–2, 2026
Historical anchor: #164

Once phase-aware attention was isolated, I kept it fixed and asked what should happen to the state as it passes repeatedly through the model.

I compared independent depth, shared recurrent computation, recurrence on a unit hypersphere, and a related construction that retained radius. My expectation was that retaining radius would help: magnitude is another quantity the model could use, and normalization explicitly discards it.

The direction-only model finished far ahead. That made the unit constraint worth investigating, particularly because the outcome contradicted the reason I had expected the retained-radius version to work.

> **Sticky note — unit hypersphere:** the set of vectors with length 1. A state on that surface can change direction while its overall magnitude stays fixed.

= Visual intuition

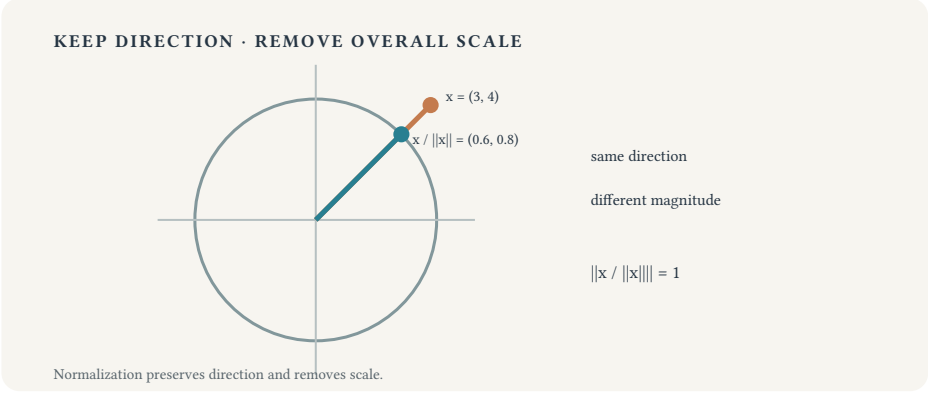


Figure 1: Toy normalization: projecting a vector onto the unit circle preserves its direction and removes its scale.

Published Sep 29, 2026

1. Frozen endpoint

After 128 training updates, validation negative log-likelihood (NLL) per original byte was approximately:

- depth-1 anchor: **5.1371**
- independent depth-3: **5.7188**
- shared generic depth-3: **5.1569**
- unit-hypersphere depth-3: **3.6584**
- retained-radius depth-3: **5.1225**

Lower NLL is better. At this endpoint, the unit model beat the comparable shared recurrent model by about **1.50 nats per byte**. Keeping radius failed to reproduce that improvement.

There was also a complication in the trajectory. The unit model learned more slowly and had worse area under the learning curve (AULC). It finished ahead after spending much of training behind.

That gives us two observations to explain together: a large endpoint advantage, and a worse learning trajectory. A useful account of the unit constraint has to accommodate both.

> **Sticky note — AULC:** area under the learning curve summarizes loss across training. A model can have a better final checkpoint while accumulating worse loss over the trajectory.

= Try a small example

The toy vector `[3, 4]` makes normalization easy to inspect: its length is 5, and its unit direction is `[0.6, 0.8]`.

```
import numpy as np
x = np.array([3.0, 4.0])
u = x / np.linalg.norm(x)
print('unit state:', u, 'norm=', np.linalg.norm(u))
# Normalization deliberately removes one scalar degree of freedom:
overall scale.
```

Saved output:

```
unit state: [0.6 0.8] norm= 1.0
```

2. Interpretation

The unit constraint changed several things at once: capacity, conditioning, rank dynamics, and optimization. The endpoint gap gave me a reason to study those changes, but it did not select a mechanism among them.

The retained-radius control weakened my initial explanation. Adding magnitude back failed to preserve the effect. I next needed to check the full learning trajectories and

compare against a Cartesian model with matched regularization: could those account for the advantage?

3. Later Muon comparison

A later three-seed comparison tested Muon on `unit_hypersphere_depth3`, using **frozen historical AdamW** results as the reference. Learning-curve area favored AdamW; final NLL favored Muon. The reviewed outcome was **mixed/inconclusive**, with the two measures favoring different optimizers.

Inspect the finalized Muon multi-seed package. That package tests optimizer behavior on this model condition. The #164 comparison above tests the choice of recurrent topology.

4. Sources and chronology

- Milestone notebook 005 — preserved chronological source
- Research chronology
- Publication dispositions and Atlas-to-article map

The matching Atlas theme is **Unit-sphere normalization**. The earlier Visual Intuition Atlas notebook preserves the source visualization.

The next chronological article is **Milestone 006 — A dramatic endpoint can still mislead**.