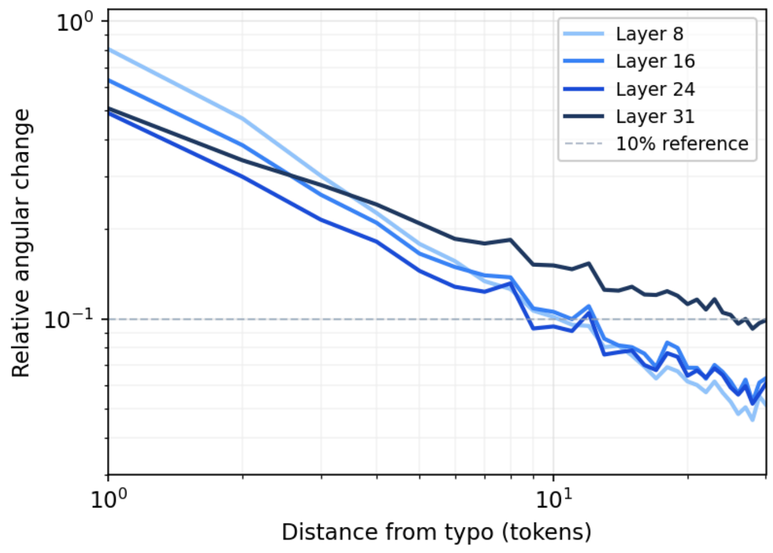


Perturbation fades to <15% within ~10 tokens downstream

THE 5-SECOND TAKEAWAY

Ordinary typos quietly break LLM safety probes. But the break is local, so it's cheaply repairable.

The model barely notices the typo, yet the probe's readout rotates ~50° at that token. The shift fades within ~10 tokens, so a fixed ~30-token suffix recovers **95%** of the lost detection.

Latent Undertow: How Ordinary Typos Break Probes

Elad David · Max Fomin · Amit LeVi
Zenity, Tel Aviv, Israel | eladd@zenity.io

1 TL;DR

THREE TAKEAWAYS

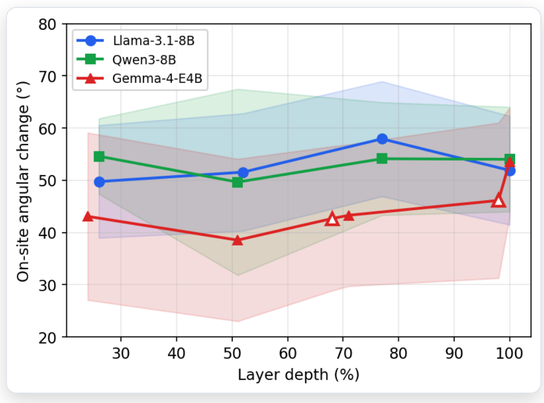
- A single typo rotates the hidden state **43–56°** at that token (norm unchanged), across three model families, yet the LLM's intent is unchanged (**94.6%**, LLM-judge).
- Stacking **~3 everyday typos** cuts a **single-position** probe's **TPR@FPR=1%** by **12.0pp**. Recalibration can't fix it.
- But the shift decays **<15% within ~10 tokens**: a ~30-token **KV-cache fork** reads past it and recovers **95%** (–0.6pp), 10× better than augmentation, plus **+4pp** out-of-distribution.

2 Background & question

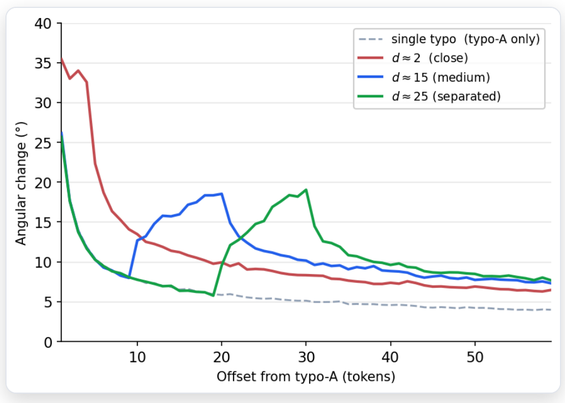
- Activation probes are deployed to monitor LLMs at inference for prompt injection and unsafe intent.
- **>40%** of real user inputs carry typos or grammatical noise.
- The model reads through typos fine, but do the probes that read its internals? **When surface noise perturbs activations without changing intent, what does the probe read out?**

3 Mechanism: rotate, then decay

- **On-site:** 43–56° rotation at the perturbed token, norm ~unchanged, so the signal is in the direction, not the scale (10° vs 30° = 3× the probe-score swing).
- **Proximity is everything:** the shift decays <15% within ~10 tokens, so a mid-sequence typo is mild. But a one-character `? → /` *at* the readout reaches **26.4° (76%** of the between-prompt baseline).



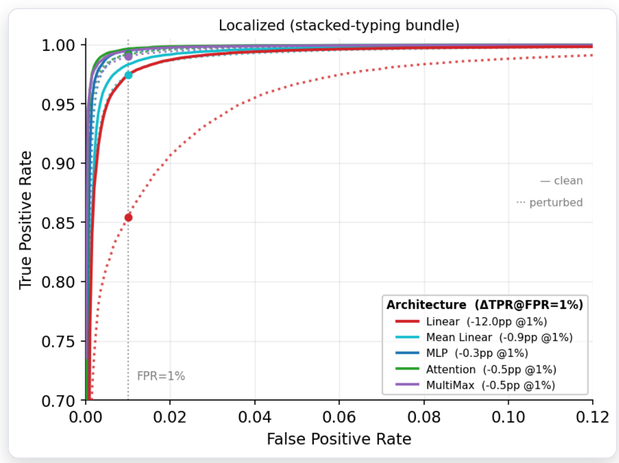
On-site rotation (43–56°), consistent across families/depths.



Two typos accumulate downstream (~1.6–2.0×); locality is additive.

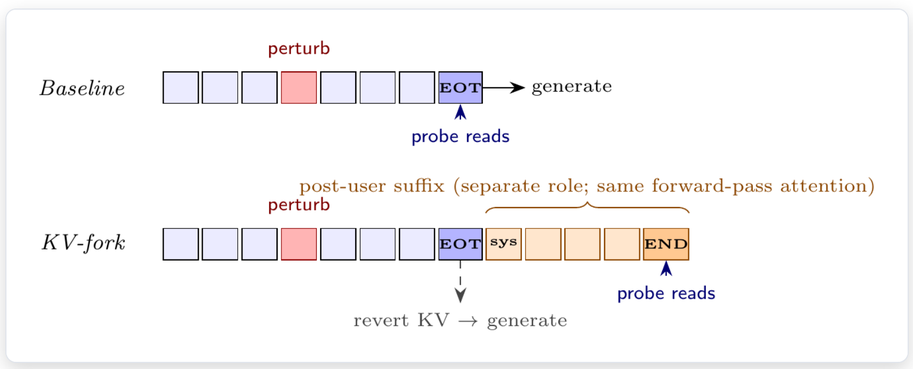
4 Consequence

- **Single-position probes break; capacity doesn't save them.** The linear probe loses **12pp** TPR@FPR=1%; an MLP at the same position loses the same. It's **where you read, not how you classify**, and the damage is class-asymmetric: it manufactures **false negatives**.
- **Aggregation helps, unevenly.** Pooling over positions cures *localized* noise but not *distributed* noise: naive mean/MLP pooling can do worse; only attention/max stay robust.



ROC (5-fold CV), localized bundle. Solid: clean; dotted: perturbed; dots: FPR=1% operating point. Single-position Linear (red) collapses; aggregators hold.

5 Defense: KV-cache fork



Append a fixed ~30-token post-user suffix; the probe reads at its end (downstream of the perturbation), then the KV-cache reverts so generation is untouched.

- Residual fragility **–0.6pp** (in-distribution), an order of magnitude better than augmentation (–3.7pp), with **no architecture change**.
- Two-for-one: also lifts out-of-distribution accuracy **+4.0pp** (77.5→81.5%), into the best aggregators' tier.

Defense	Residual TPR@FPR=1%	Cost
Architecture switch	≤0.5pp (localized)	full pipeline retrofit
Augmented training	–3.7pp	retrain probe
KV-cache fork	**–0.6pp**	+suffix, <4 ms, no retrofit

Aggregators still drop ~3.8pp on *distributed* noise; the fork's gain is general, not tied to a perturbation distribution.

6 Limitations

- Probe evaluation is on a single model/layer (Llama-3.1-8B), though the activation mechanism replicates across all three families.
- Headline probe numbers are **in-distribution** (5-fold CV) and do not decompose tokenization from attention-dilution.

PAPER
openreview.net/forum?id=gGozVMUXAd

CODE
github.com/eladd-ai/latent-undertow

zenity Labs