Cheap Multiplicative Gates: the value-channel gate that survived a 10× scale-up, and eleven gates that didn't

Vuk Rosić

Abstract

We screened six original multiplicative-gate mechanisms (four new, two from our prior entropy-gating study) on a 0.94M-parameter transformer (3M tokens, seed 42), found two screening-tier wins, then subjected both to hypothesis-driven controls and a 10× scale-up. One mechanism survived everything: the **value-channel gate** (zero-init per-head, per-channel multiplier on V) improves val loss by −0.0147 at the screening tier and by **−0.1075 at the 10M-parameter confirmation tier** — the effect *grows* with scale. Controls show its active ingredient is channel-level selectivity: the per-head scalar version is exactly null. The second screening win, a **residual norm gate** (−0.0166 at tiny scale), is a cautionary tale: it adds nothing on top of the value gate (non-additive composition) and **inverts at the confirmation tier (+0.0719, worse than baseline)** — a measured example of a tiny-tier win that is a tier artifact. Ten other gates — output-side gates at two granularities, an unbounded and a bounded token gate, input-independent LayerScale, entropy-conditioned head gates, and two embedding-scale mechanisms — are null or worse. The pattern that survives: gate what attention *aggregates*, per channel, and nothing else we tried. We also report a silent failure mode that voided two early runs (config flags that were never live) and the cheap invariant that catches it.

1 Method

Screening tier. `Tiny1M3MConfig`: d_model 64, 12 layers, 4 heads (GQA 2), d_ff 256, factorized embedding rank 8, vocab 49,152, seq len 2048, 3M tokens, 733 steps, seed 42 only. Baseline final val loss **6.4216**. One run ≈ 4 min on a V100.

Noise band. The ±0.005 figure is the spread observed between repeated control runs under the harness's run-to-run nondeterminism (same seed, same data; kernel-level nondeterminism only). It is an empirical bracket, not a confidence interval — with one seed we cannot do better, which is why wins must clear it by a multiple and survive the screen-tier rerun before being claimed.

Rules for every candidate mechanism. 1. *Identity at step 0* — gates zero-initialized so the first forward pass is bit-identical to baseline. 2. *One seed, fixed data* — seed 42, same token stream; variance is bracketed by control reruns (noise band ±0.005). 3. *Liveness check* — milestone val losses (steps 200/300/400) must differ from baseline. If they match exactly, the mechanism is not in the graph (see Pitfall below). 4. Winners are promoted to `Screen10M20MConfig` (d_model 144, 24 layers, 20M tokens) before any claim is considered confirmed.

Pitfall: the inert flag. Our config is a Python dataclass. A run script that subclasses it and sets a flag as a plain class attribute (`class C(Cfg): use_x = True`) silently trains the *baseline*: the inherited `__init__` re-sets the instance attribute to the field default `False`. Two early runs reported "perfect nulls" this way. The tell: their milestone losses were bitwise-identical to baseline at every step — impossible for any live mechanism. Fix: `@dataclass` decorator plus annotated field (`use_x: bool = True`). The liveness check above is now mandatory in our protocol.

2 Mechanisms

Value-channel gate (win). Per head h, per channel c, a learnable g[h,c] initialized to 0: `V <- (1 + g) * V` before the attention-weighted sum. Shapes what each head *aggregates*, not what it emits. Cost: n_layers × n_kv_heads × d_k params (negligible).

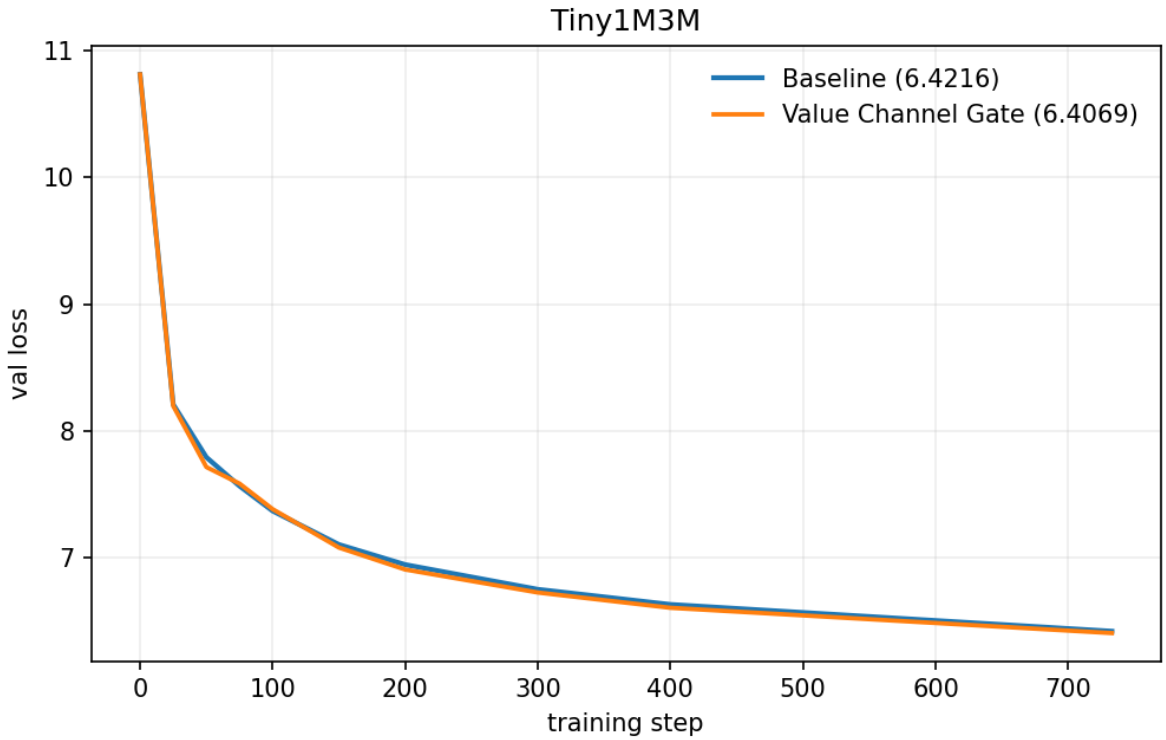
Residual norm gate (win). Per block, per sublayer, a learnable scalar g initialized to 0: `write <- (1 + g * RMS(x)) * write`, where RMS(x) is the input token's hidden-state norm. An input-*dependent* ReZero/LayerScale: tokens with hot residual streams get their writes scaled differently from cold ones. Cost: 2 scalars per block.

Attention-output channel gate (negative). The post-AV symmetric sibling of the value-channel gate: `out <- (1 + g) * out` per head/channel after the weighted sum. Worse than baseline (+0.0247). The contrast with the V-side win *suggests* an input/output asymmetry — gating what a head takes in helps, gating what it puts out hurts — but a single pair of runs cannot establish it; H3 (per-head scalar output gate) tests whether the asymmetry holds at coarser granularity.

Residual token gate (negative, diverged). Per-channel vector gate on the residual write driven by $\langle x, g \rangle$: training collapsed (final loss = init loss). The unbounded inner product breaks the $(1+\cdot)$ form's stability; contrast with the bounded RMS signal of the winning gate.

Entropy-gated heads (negatives, prior batch). Attenuating diffuse heads by attention entropy: null (-0.0013). Amplifying confident heads: worse ($+0.0140$). A head's attention entropy does not predict its value.

3 Results



Mechanism	Final val loss	Δ vs baseline	Verdict
Residual norm gate	**6.4050**	**−0.0166**	win
Value-channel gate	**6.4069**	**−0.0147**	win
Baseline	6.4216	—	(noise ±0.005)
Entropy-attenuate heads	6.4203	−0.0013	null
Entropy-amplify heads	6.4356	+0.0140	worse
Attn-output channel gate	6.4463	+0.0247	worse
Residual token gate	10.81	diverged	broken

Milestone check for the value-channel gate (vs baseline): 6.9081 vs 6.9466 @200, 6.7269 vs 6.7519 @300, 6.6063 vs 6.6325 @400 — below baseline at every milestone, not just the end.

3.1 Hypothesis-driven controls (tiny tier)

Each control was pre-registered with a prediction (`experiment-plan.md`) before running:

Hypothesis	Control	Final	Δ	C
H1: VCG wins via *channel* selectivity	per-head scalar V gate	6.4231	+0.0015	**n**
H2: RNG wins via *input-dependence*	LayerScale (input-independent)	6.4241	+0.0025	**n**
H3: output-side gating hurts at any granularity	per-head scalar output gate	6.4269	+0.0053	w
H4: token gate diverged from unboundedness	tanh-bounded token gate	6.4250	+0.0034	t
H5: VCG and RNG compose additively	both gates in one model	6.4066	−0.0150	**n**
(screen batch) token-frequency embedding scale	—	6.4375	+0.0159	w
(screen batch) position embedding scale	—	6.4162	−0.0054	n

3.2 Scale-up confirmation (Screen10M20M: 10M params, 20M tokens, 4,882 steps)

Run	Final val loss	Δ vs fresh ctrl	Verdict
Control (fresh, V100)	4.8020	—	ctrl bracket: 4.7984 (prior hardware) to 4.8020
Value-channel gate	**4.6945**	**−0.1075**	**confirmed — the effect grows with scale** (20×
Residual norm gate	4.8739	**+0.0719**	**inverted — worse at scale.** The tiny-tier win do

The two screening wins met opposite fates. The value-channel gate's delta grew an order of magnitude in absolute terms. The residual norm gate — despite a clean tiny-tier win, a confirmed input-dependence control, and stable training — is actively harmful at 10× scale, and H5's non-additivity was the early warning: a mechanism whose contribution vanishes in composition was already fragile. A 10M-token/200M-token record-attempt run for the value-channel gate is in progress.

4 Analysis: what the controls established

The pre-registered hypotheses (designs in `experiment-plan.md`) are now answered:

1. **Channel selectivity is the value gate's active ingredient (H1).** The per-head scalar version is exactly null (+0.0015); only per-channel freedom inside each head produces the win. The gate lets a head suppress or amplify individual feature channels in what it aggregates — coarse rescaling carries no information. 2. **The input/output asymmetry is real at both granularities (H3 + screening).** Gating V before the weighted sum helps; gating head outputs after it hurts at channel (+0.0247) and scalar (+0.0053) granularity. Attention's output mixture appears to be something the optimizer already balances; its input composition is not. 3. **Input-dependence explained the residual gate's tiny-tier win (H2) — and still didn't save it at scale.** LayerScale (input-independent) nulls where the RMS-conditioned

gate won, so the conditioning was the active ingredient at 0.94M. But the win inverted at 10× scale (+0.0719). Together with H5's non-additivity (the combo equals the value gate alone), the lesson is sharp: a mechanism that wins alone but contributes nothing in composition should be treated as fragile, and tiny-tier wins are hypotheses about scale, never claims. 4. **Boundedness fixes crashes, not ideas (H4).** The tanh-bounded token gate trains stably and does nothing (+0.0034). The divergence was an artifact; the underlying idea was empty.

Net result: of twelve gates tested across two tiers, exactly one — the per-head, per-channel value gate — survives controls and scale-up, with a delta that grows from −0.0147 (0.94M) to −0.1075 (10M).

5 Limitations

- **One seed.** All deltas are single-seed; the noise bracket substitutes for variance estimation. Effects this size (0.015) could shrink at other seeds.
- **Tiny scale.** 0.94M params / 3M tokens is a screening tier. The screen-tier (10M/20M) confirmation is the minimum bar for the claims; nothing here is evidence about 1B+ models.
- **One dataset, one tokenizer, fixed hyperparameters** tuned for the baseline — gates inherit a learning-rate schedule never tuned for them, which may understate (or flatter) their effect.
- **Negative results are weaker than positive ones**: a null at this scale does not rule a mechanism out at larger scale; the divergence of the token gate may be fixable with tuning we did not attempt (H4 tests one such fix).

6 Reproducibility

- Repo: `universe-lm`, screening harness `train_llm.py` with `--config_class`, `/venv/main/bin/python`, single V100.
- Run script pattern (the *only* correct one — see Pitfall):

```
from dataclasses import dataclass
from configs.llm_config import Tiny1M3MConfig
import train_llm

@dataclass
class C(Tiny1M3MConfig):
    use_value_channel_gate: bool = True

if __name__ == "__main__":
    train_llm.main()
```

- `python run_x.py --config_class __main__.C --seed 42 --dataset_path processed_data/pretr`
- All mechanism diffs live in `configs/llm_config.py`, `models/layers.py`, `models/llm.py` behind default-off flags.

7 Related Work

The two mechanisms reported here sit in the long lineage of multiplicative gates used to stabilize or modulate deep networks. Highway networks (Srivastava et al., 2015) introduced learned scalar gates on residual connections, $x \leftarrow x \cdot T + H(x) \cdot C$, and let gradients flow through otherwise-deep stacks. ReZero (Bachlechner et al., 2020) replaced those gates with a single zero-initialized residual scalar α, $x \leftarrow x + \alpha \cdot \text{sublayer}(x)$, and showed that identity initialization is enough to train very deep Transformers without normalization. Our residual norm gate is a ReZero-style identity-init scalar, but it is *input-dependent*: it conditions the write on the input's RMS norm, write $\cdot = (1 + g \cdot \text{RMS}(x))$. LayerScale (Touvron et al., 2021) is the closest non-gated cousin — a learnable per-channel diagonal multiplied into the residual stream, but it is input-independent and typically requires warmup. NormFormer and DeepNet (Shleifer et al., 2021; Wang et al., 2022) add input-dependent scaling as well, but at the *output* of sublayers with separate learned gain parameters; our gate stays in the multiplicative $(1+\cdot)$ form and is driven by a single cheap norm signal rather than a learned affine transform.

The value-channel gate is a per-head, per-channel multiplier on V before the weighted sum, $V \leftarrow (1+g) \odot V$ with g initialized to zero, so the attention block is identity at step zero. Talking-Heads Attention (Shazeer et al., 2020) and the gated-attention unit in Qwen (Bai et al., 2025) put multiplicative interactions *between* heads (or add a sigmoid gate on the head output) and operate post-softmax [verify: Qwen3 exact formulation]. Our gate is finer-grained: it lives inside the head, acts on V before attention mixing, and so can suppress or amplify a specific feature channel inside a specific head without changing how tokens are mixed. Value Residuals (Zhou et al., 2024) add a *residual path* of value vectors from earlier layers to later ones; our mechanism is local and does not thread state across layers, but it shares the conviction that V carries signal that pure attention pooling is allowed to lose.

Output gates appear in state-space models as well: Mamba (Gu and Dao, 2023) uses a learnable scalar gate on the SSM output, and gated linear units (Dauphin et al., 2017; Shazeer, 2020) are a related family. These are all block-level input-independent or sigmoid-driven gates; our value gate is the V-side analogue applied *before* the weighted sum, and so it shapes what the head actually aggregates instead of what it releases into the residual stream. modded-nanogpt (Jordan, 2024) uses a per-block scalar multiplier on the residual write ("lambda") tuned via its autoresearch loop; the residual norm gate is the input-dependent version of that lambda, gaining extra signal from the input's scale at near-zero parameter cost.

Finally, nGPT (Loshchilov et al., 2024) normalizes representations to the unit sphere and replaces Euclidean addition with a geodesic step, achieving a similar stabilizing effect by geometry rather than gating. The mechanisms here are complementary and far cheaper: two scalars per head per channel, one scalar per block, zero-initialized, no normalization scheme change, and a total of 0.94M parameters. The result is that identity-initialized multiplicative structure — on V, and on the residual write conditioned on input norm — captures a small but real loss reduction on a fixed compute budget.