Attention Sinks Induce Gradient Sinks: Massive Activations as Gradient Regulators in Transformers

Yihong Chen[1] **Zhouchen Lin**[2,3*] **Quanming Yao**[1,4,5*]
[1]Department of Electronic Engineering, Tsinghua University
[2]State Key Lab of General Artificial Intelligence
[3]School of Intelligence Science and Technology, Peking University
[4]State Key Laboratory of Space Network and Communications
[5]Beijing National Research Center for Information Science and Technology
chenyihong@tsinghua.edu.cn, zlin@pku.edu.cn, qyaoaa@tsinghua.edu.cn

Abstract

Attention sinks and massive activations are recurring and closely related phenomena in Transformer models. Existing explanations have largely focused on the forward pass, yet in pre-norm Transformers, large residual-stream norms play only an indirect forward role because sublayers operate on normalized inputs. We study this relationship from the perspective of backpropagation. Empirically and theoretically, we show that under causal masking, attention sinks can induce pronounced gradient concentration, which we term *gradient sinks*. Since the RMSNorm Jacobian attenuates gradients roughly in inverse proportion to input norm, massive activations can be understood as adaptive regulators of this localized gradient pressure during training. This interpretation predicts that attenuating sink-induced gradients should weaken massive activations. We test this prediction with V-scale, a modification that adjusts backpropagated gradients on the value paths. In V-scale models and a backward-only control, attention sinks are preserved, whereas massive activations are suppressed. These results identify gradient sinks as a backward-pass counterpart of attention sinks, and massive activations as an adaptive RMSNorm-mediated response that attenuates the resulting localized training pressure. Our code is available at https://github.com/LARS-research/GradientSink.

1 Introduction

Two prominent and frequently co-occurring phenomena in Transformer-based large language models (LLMs) are *attention sinks* (AS), where a small number of tokens attract disproportionate attention mass [14, 43], and *massive activations* (MA), where activations become unusually large on a small set of tokens and features [2, 5, 36]. These phenomena are predominantly observed at the first token and are of significant practical importance for long-context inference [34, 43], fine-tuning [10, 18, 20, 21], and quantization [5, 42]. The frequent co-occurrence of AS and MA has also attracted mechanistic interest. Existing work has provided several forward-pass explanations for why AS and MA may arise or persist. For example, AS has commonly been explained as a way to implement "no-op" attention heads that contribute little to the current token [4, 14, 15] under the sum-to-one constraint of the softmax operator, while MA has been described as helping form attention-sink behavior [29, 35, 36].

However, in modern pre-norm architectures, attention and MLP sublayers operate on normalized inputs [38, 44, 48], making local forward attention depend primarily on the direction of representation rather than on its raw norm. This makes the emergence of MA appear difficult to explain from a purely forward-pass perspective. Moreover, many post-hoc interventions in the literature modify

*Co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

MLP outputs or token activations of trained models. While informative, such interventions often perturb both magnitude and direction, making it difficult to isolate the causal mechanism linking AS and MA. Recent studies of training dynamics [11, 29] and architectural decoupling [37] clarify their development and separability, but a training-time puzzle remains: *why do standard pre-norm Transformers learn such extreme activations at sink tokens?*

We approach this question from the backward pass. Residual-stream scale can be important during backpropagation because it controls how much gradient signal is transmitted through normalization in pre-norm architectures. This matters precisely at sink tokens. In an attention head under causal masking, when many later tokens attend to a sink token, their training signals are routed back to that token as well. This turns a forward attention sink into a backward concentration of gradients, which we call a gradient sink (GS).

In this view, massive activations can be understood as learned regulators that absorb sink-induced gradient pressure and help maintain stable residual-stream gradient transport. This interpretation further predicts that relieving such pressure through an alternative attenuation route should reduce reliance on MA even while AS remains present. The remainder of the paper develops this mechanism from observation and analysis to intervention.

Our contributions are threefold:

- **Gradient sinks.** We identify gradient sinks as a backward-pass counterpart of attention sinks in pre-norm Transformers. We show that concentrated gradient pressure at the first token is widespread across models.
- **Mechanism.** We provide an empirical and theoretical account of massive activations as adaptive regulators. In pre-norm architectures, RMSNorm makes activation scale an effective local factor to attenuate gradient pressure induced by attention sinks, thereby stabilizing gradient transport across the residual stream.
- **Intervention.** We propose *V-scale*, a value-path gradient valve that attenuates sink-induced gradients. V-scale and its backward-only control substantially suppress massive activations while retaining attention sinks, supporting the proposed mechanism. We also observe improvements on multi-key retrieval across several context lengths and precision settings.

2 Setup

We introduce the notation needed for the empirical measurements and theoretical analysis that follow. In this paper, token positions are indexed by $t \in \{0, \dots, T-1\}$, and $\mathcal{L}$ denotes the training loss.

Pre-norm Transformer and attention We study decoder-only Llama-like Transformers with pre-norm residual blocks, RMSNorm, RoPE positional encoding, and SwiGLU MLPs [32, 33, 38, 48]. Figure 1 (a) illustrates the forward computation. Let $H^\ell = (h_0^\ell, \dots, h_{T-1}^\ell)^\top \in \mathbb{R}^{T \times d_{\text{model}}}$ denote the hidden states input to layer ℓ. A pre-norm block updates the sequence as

$$\widetilde{H}^\ell = \mathrm{RMSNorm}(H^\ell), \qquad R^{\mathrm{attn},\ell} = \mathrm{Attention}(\widetilde{H}^\ell), \quad H^{\ell+1/2} = H^\ell + R^{\mathrm{attn},\ell},$$
$$\widetilde{H}^{\ell+1/2} = \mathrm{RMSNorm}(H^{\ell+1/2}), \quad R^{\mathrm{mlp},\ell} = \mathrm{MLP}(\widetilde{H}^{\ell+1/2}), \qquad H^{\ell+1} = H^{\ell+1/2} + R^{\mathrm{mlp},\ell}.$$

We write $r_t^{\mathrm{attn},\ell}$ and $r_t^{\mathrm{mlp},\ell}$ for the t-th rows of $R^{\mathrm{attn},\ell}$ and $R^{\mathrm{mlp},\ell}$.

The attention block can be either multi-head attention (MHA) [39] or a variant such as grouped-query attention (GQA) [1]. For simplicity, we fix one layer ℓ and one attention head h, and suppress the layer/head superscripts unless needed. With RoPE, the query, key, and value states at token position t are $q_t = R_t W_Q \widetilde{h}_t$, $k_t = R_t W_K \widetilde{h}_t$, and $v_t = W_V \widetilde{h}_t$, where $R_t \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ is the RoPE rotation matrix. The causal attention logits and weights are computed by $z_{tj} = \langle q_t, k_j \rangle / \sqrt{d_{\text{head}}} + m_{tj}$ and $a_{tj} = \mathrm{softmax}(z_{t,:})_j$, where $m_{tj} = 0$ for $j \le t$ and $m_{tj} = -\infty$ otherwise. Finally, the value aggregation and head output are $y_t = \sum_{j \le t} a_{tj} v_j$ and $o_t = W_O y_t$, respectively.

Attention sink and activation measurements For a candidate sink position s (typically the first token, i.e., $s = 0$), we follow [14, 29] and use thresholded criteria to count sink-like behavior:

$$\mathrm{Sink}_s^{\epsilon,\ell} = \frac{1}{N_{\text{head}}} \sum\nolimits_{h=1}^{N_{\text{head}}} \mathbb{I}(\alpha_s^{\ell,h} > \epsilon), \quad \alpha_s^{\ell,h} = \frac{1}{T_\epsilon} \sum\nolimits_{t=s}^{s+T_\epsilon-1} a_{ts}^{\ell,h}, \tag{1}$$

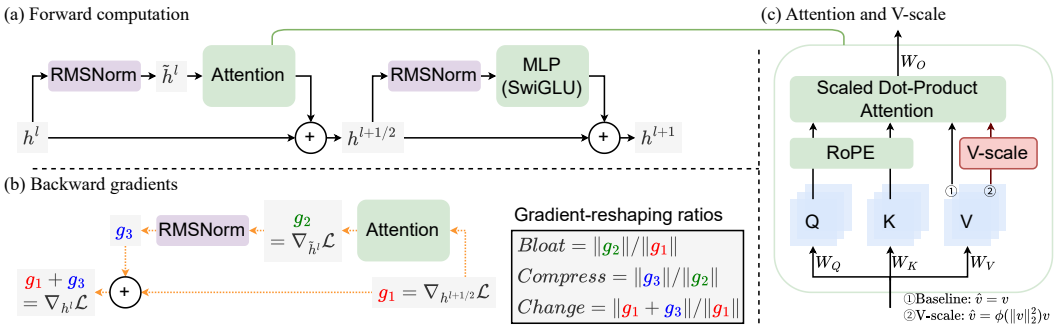


Figure 1: Pre-norm Transformer architecture, gradient measurements, and V-scale placement. (a) Forward computation through the attention and MLP sublayers. (b) Backward flow and gradient measurements for the attention sublayer. Detailed definitions are given in Section 3.2. (c) Expanded attention computation. Value paths 1 and 2 are alternatives for the baseline model and the V-scale variant proposed in Section 5, respectively.

where N_{head} is the number of attention heads in each layer. Following prior work [14] we set the threshold $\epsilon = 0.3$ and $T_\epsilon = 64$. For theory and training-dynamics measurements, we also use the attention-column mass and second moment

$$M_s^{\ell,h} = \sum_{t=s}^{T-1} a_{ts}^{\ell,h}, \quad S_s^{\ell,h} = \sum_{t=s}^{T-1} (a_{ts}^{\ell,h})^2. \tag{2}$$

Thus M_s measures the total attention mass routed through token s by all causally allowed future queries, while S_s is its second-moment analogue. In experiments, we focus on the first token ($s = 0$).

For token-wise activation analyses, we track activation norms at several computational sites, including the input of Transformer layers $\|h_{t,\text{in}}^\ell\| = \|h_t^\ell\|_2$, the hidden states after attention $\|h_{t,\text{half}}^\ell\| = \|h_t^{\ell+1/2}\|_2$, and the output of Transformer layers $\|h_{t,\text{out}}^\ell\| = \|h_t^{\ell+1}\|_2$. We also measure branch outputs $\|r_t^{\text{attn},\ell}\|_2$ and $\|r_t^{\text{mlp},\ell}\|_2$, together with value-state norms $\|v_t^\ell\|_2$, when needed.

3 Empirical Evidence

This section establishes the empirical basis for viewing massive activations as gradient regulators. We analyze typical open-source pretrained LLMs [13, 45, 46, 47], together with `LlamaForCausalLM` baselines that we pretrain from scratch on the C4 [31] dataset at scales ranging from 0.1B to 1B parameters. Detailed model and training configurations are deferred to Appendix B.

3.1 From attention sinks to gradient sinks

To trace how sink structure reappears in backpropagation, we first define gradient-sink behavior.

Definition 1 (Gradient sink)**.** Fix a layer ℓ, a candidate sink position s, and an early-token reference set $\{0, \ldots, T_e\}$ where $0 \le s \le T_e$. Let $\|\nabla_{\tilde{h}_t^\ell}\mathcal{L}\|_2$ denote the token-wise gradient norm at the post-RMSNorm attention input. The gradient-sink ratio of token s at layer ℓ, relative to $\{0, \ldots, T_e\}$, is defined as

$$R_{GS}^\ell(s; T_e) := \|\nabla_{\tilde{h}_s^\ell}\mathcal{L}\|_2 / \big(1/T_e \sum_{0 \le t \le T_e, t \ne s} \|\nabla_{\tilde{h}_t^\ell}\mathcal{L}\|_2\big). \tag{3}$$

For a prescribed threshold $\tau > 1$, token s is called a τ-gradient sink at layer ℓ if $R_{GS}^\ell(s; T_e) \ge \tau$.

In this work, we primarily report the continuous ratio $R_{GS}^\ell(s; T_e)$ rather than fixing a universal threshold τ, since the meaningful scale of gradient concentration can depend on model size, layer, and measurement protocol. In the experiments below, we take $s = 0$ and $T_e = 31$. Gradients are first aggregated over the full batch as in one optimization step and then converted to token-wise ℓ_2 norms, making the comparison reflect the effective training signal seen by the optimizer. For scratch-trained baselines, we use the corresponding batch size and context length specified in Appendix B; for pretrained models, we use a diagnostic batch of 512 packed sequences with context length 2048.

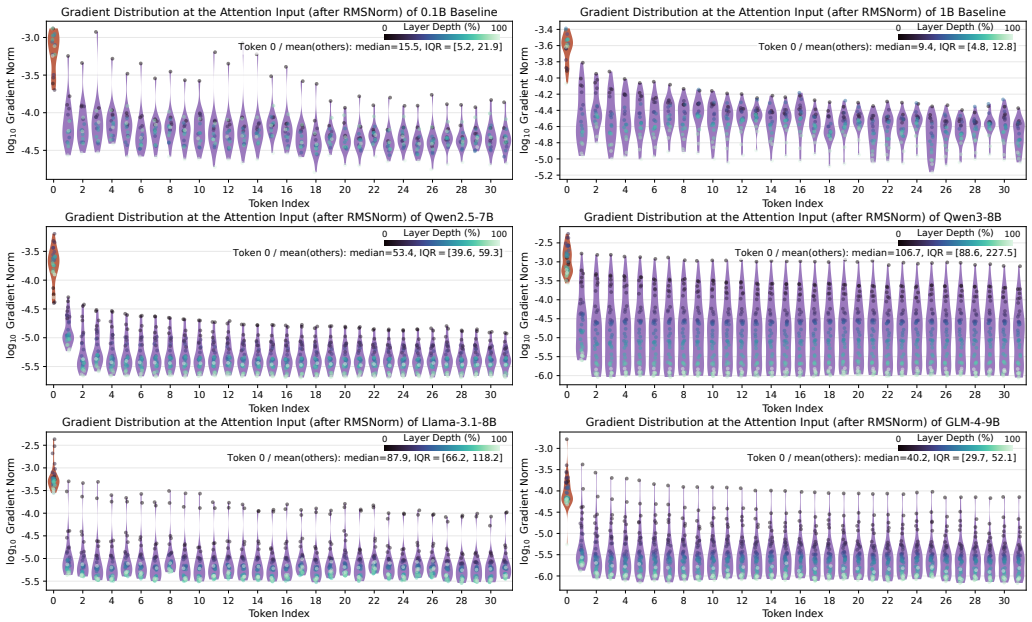


Figure 2: Gradient sinks across models. Each panel plots the token-wise distribution of $\log_{10} \|\nabla_{\widetilde{h}_t^\ell} \mathcal{L}\|_2$ at the post-RMSNorm attention input. Token 0 is highlighted. The reported median and interquartile range (IQR) of $R_{GS}^\ell(0)$ summarize how much larger the gradient on token 0 is than the mean over positions 1–31. Across both scratch-trained baselines and pretrained models, the first token consistently exhibits the strongest gradient concentration. Models include 0.1B and 1B baselines together with Qwen2.5-7B, Qwen3-8B, Llama-3.1-8B, and GLM-4-9B.

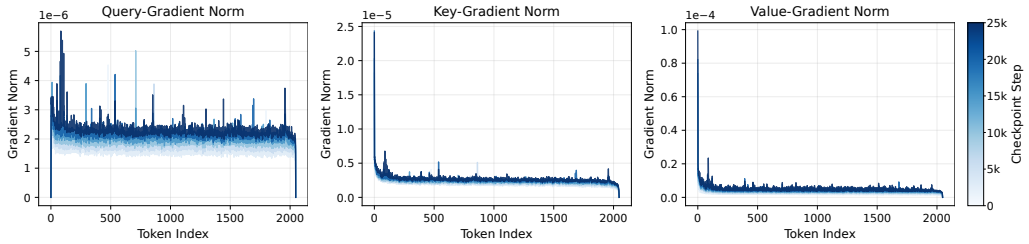


Figure 3: Token-wise gradient norms of query, key, and value across training checkpoints for the 1B model trained from scratch. Statistics are averaged over layers. Key and especially value gradients exhibit a pronounced spike at token 0, while query gradients remain comparatively flat.

Note that for pretrained models, these gradients should be interpreted as post-hoc diagnostic probes rather than the exact gradients from their official training runs.

Figure 2 shows that the sink token consistently exhibits substantially larger gradients than other early tokens, with the reported medians of $R_{GS}^\ell(0)$ typically lying in the 10–100 range, i.e., roughly one to two orders of magnitude above the early-token average. Gradient sinks therefore appear as a common empirical pattern during the training of Transformer language models.

To understand how this concentration is structured inside the attention block, we further decompose the gradients into query, key, and value pathways and track their token-wise norms across training checkpoints. For each token s, the incoming gradient splits as $\nabla_{\widetilde{h}_s^\ell} \mathcal{L} = W_Q^\top \nabla_{q_s^\ell} \mathcal{L} + W_K^\top \nabla_{k_s^\ell} \mathcal{L} + W_V^\top \nabla_{v_s^\ell} \mathcal{L}$, so the observed token-wise concentration can be examined pathway by pathway.

Figure 3 reveals a clear and highly structured asymmetry. On the key and value pathways, the gradient norm exhibits a pronounced spike at the first token across checkpoints, after which it drops rapidly and remains relatively flat over most of the sequence. Moreover, the value-path spike is consistently larger than the key-side one. By contrast, the query pathway looks qualitatively different:

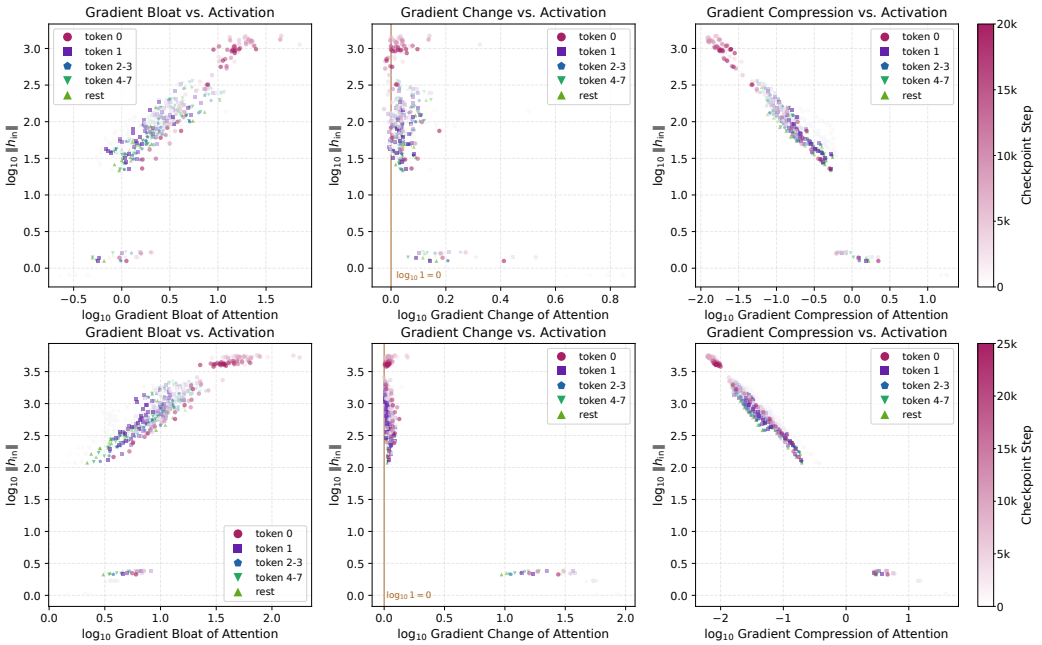


Figure 4: Scatter plots relating local gradient reshaping to input activation norms of the attention block. Top row: 0.1B model; bottom row: 1B model. From left to right, the panels show $\mathrm{Bloat}^{\mathrm{attn}}$, $\mathrm{Change}^{\mathrm{attn}}$, and $\mathrm{Compress}^{\mathrm{attn}}$ against $\|h_{\mathrm{in}}\|$ on a $\log_{10}$ scale. Each point is colored by token group (token 0, token 1, tokens 2–3, tokens 4–7, and the rest early tokens 8–15). Large-activation points from token 0 populate the high-bloat regime, yet their residual-stream gradient norms change only mildly for most layers. The $\mathrm{Compress}$ ratio indicates stronger attenuation at massive activation sites.

its token-wise gradient profile is much flatter, and the first-token query gradient is exactly zero due to causal masking. For each pathway, the statistics are averaged over layers.

These observations sharpen the empirical picture. Gradient sinks have a structured pathway profile, and the excess at the first token is carried mainly by the key and especially value gradients.

3.2 Massive activations align with gradient reshaping

We next ask how the pre-norm blocks transmit local gradients and how this reshaping relates to activation norms. Figure 1 (b) marks the gradient locations used in the following measurements. For a fixed layer and token, let $g_1 = \nabla_{h_t^{\ell+1/2}}\mathcal{L}$ be the gradient after the residual addition and $g_2 = \nabla_{\tilde{h}_t^{\ell}}\mathcal{L}$ the gradient at the normalized attention input. Backpropagation through RMSNorm produces the branch contribution $g_3 = J_{\mathrm{RMSNorm}}(h_t^{\ell})^{\top} g_2$. The skip path carries g_1 unchanged, so the total input gradient is $\nabla_{h_t^{\ell}}\mathcal{L} = g_1 + g_3$.

Three stages of this backward flow are considered. *Bloat* measures amplification through the attention branch from g_1 to g_2. *Change* measures the net change in residual-stream gradient norm after the branch contribution is added. *Compress* measures the norm gain through RMSNorm from g_2 to g_3. That is,

$$\mathrm{Bloat}_t^{\mathrm{attn},\ell} := \frac{\|g_2\|_2}{\|g_1\|_2}, \quad \mathrm{Change}_t^{\mathrm{attn},\ell} := \frac{\|g_1 + g_3\|_2}{\|g_1\|_2}, \quad \mathrm{Compress}_t^{\mathrm{attn},\ell} := \frac{\|g_3\|_2}{\|g_2\|_2}. \tag{4}$$

The analogous definitions and measurements for the MLP branch appear in Appendix C.2.

Figure 4 compares each gradient-reshaping ratio with the corresponding input activation norm averaged over evaluation batches. The left panels show that large-activation points, almost all from token 0, are precisely where branch-local amplification becomes extreme. Such large $\mathrm{Bloat}^{\mathrm{attn}}$ naturally raise the concern about distortions on residual-stream transport during backpropagation that could harm training stability [19, 44]. The middle panels show that this is largely not the case. The $\mathrm{Change}^{\mathrm{attn}}$ values remain tightly concentrated near 1 and therefore near $\log_{10} 1 = 0$ on the

plotted scale, aside from a small number of points outside the massive-activation regime. Thus, the residual-stream gradient norm usually does not change dramatically. The right panels suggest why this remains possible. Larger activation norms are consistently associated with smaller $\mathrm{Compress}^{\mathrm{attn}}$, revealing a strong negative relationship between activation scale and local gradient attenuation. That is, the most amplified gradients are also the most strongly compressed before being passed back into the residual stream, avoiding uncontrolled gradient growth.

Taken together, these measurements give a backward-pass view of massive activations on the first token. They suggest that these large norms are not merely forward numerical outliers, but mark sites where gradient concentrations are locally reshaped.

4 Theoretical Mechanism

This section explains the mechanism behind the empirical patterns above, making it clear how attention sinks, gradient sinks, and massive activations are connected in pre-norm Transformers. The key ingredients are attention routing and the scale sensitivity of RMSNorm. We present the main identities and bounds here, with the full theory and proofs deferred to Appendix D.

4.1 Attention weights route gradients backward

We first formalize the basic intuition given in Figure 5 that attention weights route not only values forward but also value-path gradients backward. Fix one layer and one attention head, and suppress their superscripts for simplicity. Let $A = (a_{tj}) \in \mathbb{R}^{T\times T}$ be the causal attention matrix, with $a_{tj} = 0$ for $j > t$. Stack the value states and head outputs row-wise as $V, Y \in \mathbb{R}^{T\times d_{\mathrm{head}}}$, so that $Y = AV$. Let $G := \nabla_Y \mathcal{L}$ be the upstream gradient at Y, and write its t-th row as $g_t := \nabla_{y_t}\mathcal{L}$. We now record the exact value-path identity induced by attention aggregation.

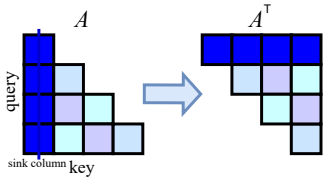


Figure 5: Attention weights as two-way routers.

Proposition 1 (Exact value-path aggregation)**.** *For one attention head, the value-path backward signal satisfies* $\nabla_V \mathcal{L} = A^\top G$*, or equivalently* $\nabla_{v_s}\mathcal{L} = \sum_{t=s}^{T-1} a_{ts} g_t$ *for any token index* s*.*

The identity shows the basic structural routing mechanism. A forward attention sink means that many later rows place substantial mass on one early column s of A. On the value path, backpropagation uses the corresponding row s of $A^\top$ and therefore aggregates the gradients of later tokens into the same early token. In other words, attention weights are two-way routers.

The full key and query identities are given in Appendix D.1, and they clarify the empirical asymmetry observed in Section 3. Key gradients still depend on sink columns, while query gradients are row-local. In particular, under causal masking, $\nabla_{q_0}\mathcal{L} = 0$ exactly.

4.2 Sink statistics control localized gradient pressure

We relate the value-path gradient to the attention-column statistics M_s and S_s defined in (2).

Theorem 1 (Value-path gradient control by sink statistics)**.** *For fixed model parameters, write* $\mathbb{E}_A[\cdot] = \mathbb{E}[\cdot \mid A]$ *given attention matrix* A*. Suppose that the upstream gradients on positions* $t \geq s$ *admit the decomposition* $g_t = \mu + \varepsilon_t$ *with* $\mu = \mathbb{E}_A[g_t] \in \mathbb{R}^{d_{\mathrm{head}}}$ *for all* $t \geq s$*. The noise terms satisfy* $\mathbb{E}_A[\varepsilon_t] = 0$*,* $\mathbb{E}_A\|\varepsilon_t\|_2^2 \leq \sigma^2$ *for all* t*, and* $|\mathbb{E}_A\langle \varepsilon_t, \varepsilon_{t'}\rangle| \leq \rho$ *for all* $t \neq t'$*. Then,* $M_s^2\|\mu\|_2^2 \leq \mathbb{E}_A\left[\|\nabla_{v_s}\mathcal{L}\|_2^2\right] \leq M_s^2\|\mu\|_2^2 + \sigma^2 S_s + \rho(M_s^2 - S_s)$*.*

Theorem 1 shows that sink structure provides a natural and quantitatively explicit route to value-path gradient concentration. The term $M_s^2\|\mu\|_2^2$ is the coherent accumulation along the sink column, while S_s and $\rho(M_s^2 - S_s)$ bound the accumulated variance and cross-token covariance. For a nonzero coherent component, larger attention-column mass raises the lower bound on the conditional second moment of the value-path gradient.

The dominant value-path spike seen in Section 3 is thus not accidental. The value path is the pathway where sink statistics most transparently govern gradient aggregation. Appendix D.3 further provides supporting results for the key and query pathways. The key bound has a similar dependence on M_s, whereas the query bound depends on within-row key dispersion.

4.3 RMSNorm turns activation scale into gradient regulation

It remains to connect this localized pressure to massive activations. In a pre-norm block, the relevant map is the RMSNorm applied before the attention or MLP branch.

Theorem 2 (Activation-dependent compression under RMSNorm). *Denote* $y = \mathrm{RMSNorm}(x) = \gamma \odot \frac{x}{\mathrm{rms}(x)}$ *with* $\mathrm{rms}(x) = \sqrt{\frac{1}{d}\|x\|_2^2 + \epsilon_{\mathrm{rms}}}$, *where* $x \in \mathbb{R}^d$ *is the input and* $\gamma \in \mathbb{R}^d$ *is the scale parameter. Let* $g_y = \nabla_y \mathcal{L}$ *be the upstream gradient at the RMSNorm output. Then we have* $\nabla_x \mathcal{L} = J_{\mathrm{rms}}(x)^\top g_y$, *and the Jacobian satisfies* $\|J_{\mathrm{rms}}(x)\|_{\mathrm{op}} \le \frac{\|\gamma\|_\infty}{\mathrm{rms}(x)}$.

Theorem 2 formalizes that a larger token-wise activation norm directly lowers the worst-case gain from upstream gradients to earlier layers. Consequently, for any target gradient scale $\tau > 0$, the condition $\mathrm{rms}(x) \ge \frac{\|\gamma\|_\infty}{\tau}\|g_y\|_2$ is sufficient to ensure $\|\nabla_x \mathcal{L}\|_2 \le \tau$. This is the mathematical sense in which large activations can act as gradient regulators rather than merely as pathological outliers.

Combining this with Proposition 1 and Theorem 1, attention sinks create localized value-path pressure, and RMSNorm makes activation scale an available local attenuation factor. Massive activations are therefore well positioned to emerge in response to sink-induced pressure, helping preserve stable residual-stream gradient transport by absorbing the pressure created by gradient sinks.

5 Mechanistic Intervention and Validation

The proposed mechanism predicts that relieving localized gradient pressure should weaken MA while retaining AS. We test this prediction with V-scale and a backward-only control in matched pretraining runs, then evaluate language modeling and long-context retrieval. Figure 6 summarizes the mechanism and intervention rationale.

The value path is the natural target channel to regulate gradients for three reasons. First, modifying only value states does not affect the attention weights that depend only on Q and K. Second, our observations in Section 3 confirm that value-path gradient norms of the sink token are several times larger than key-path ones. Third, prior studies have repeatedly reported that sink tokens tend to have unusually small value norms [6, 15, 34]. We observe the same pattern in our models, motivating a radial transform that uses $\|v\|_2$ to target sink tokens.

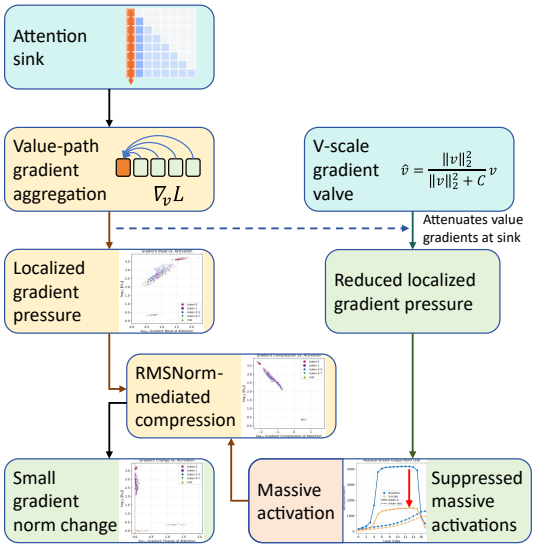


Figure 6: Mechanism and intervention. MA attenuates sink-induced gradient pressure through RMSNorm. V-scale provides an additional value-path gradient valve, reducing reliance on MA.

5.1 V-scale: a value-path gradient valve

We now introduce the intervention used to test the mechanism. For each attention head, after the standard value projection and before value aggregation, we replace $y_t = \sum_{j \le t} a_{tj} v_j$ by

$$\hat{v}_j = \phi(\|v_j\|_2^2)\, v_j, \quad \phi(r) = \frac{r}{r + C}, \quad \hat{y}_t = \sum\nolimits_{j \le t} a_{tj} \hat{v}_j,$$

where $C > 0$ is a scale parameter. We call this modification *V-scale*, shown as value path 2 in Figure 1 (c). We parameterize $C_{\ell,h} = d_{\mathrm{model}} \cdot d_{\mathrm{head}} \cdot \sigma^2 \exp(\theta_{\ell,h})$, where $\sigma = 0.02$ is the value-projection initialization standard deviation and $\theta_{\ell,h}$ starts at 0. This adds one learned parameter per layer and head and matches the typical squared value norm at initialization (see Appendix D.5). The backward behavior of V-scale is explicit:

Proposition 2 (Jacobian spectrum of V-scale). *Let* $r = \|v\|_2^2$ *and* $\hat{v} = \phi(r)v$ *with* $\phi(r) = \frac{r}{r+C}$. *Then the Jacobian of the V-scale map* $v \mapsto \hat{v}$ *is* $J_\phi(v) = \phi(r)I + \frac{2C}{(r+C)^2} vv^\top$ *with eigenvalue*

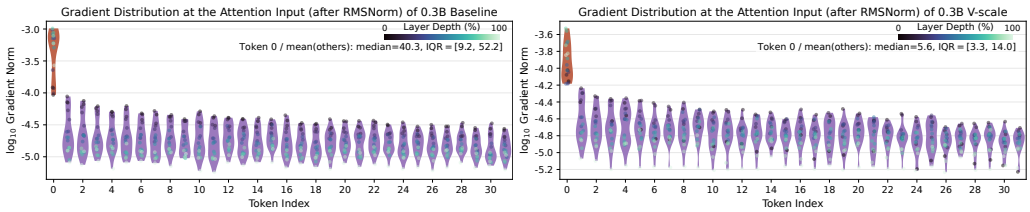


Figure 7: Gradient sinks in the 0.3B baseline and V-scale models. Each panel plots the token-wise distribution of $\log_{10} \|\nabla_{\tilde{h}_t^\ell} \mathcal{L}\|_2$ at the post-RMSNorm attention input. Token 0 is highlighted. V-scale still leaves a visible gradient sink but reduces the concentration relative to the baseline.

$\lambda_\perp(r) = \frac{r}{r+C}$ *on the* $(d_{\text{head}} - 1)$*-dimensional subspace orthogonal to* v *and* $\lambda_\parallel(r) = \frac{r^2+3Cr}{(r+C)^2}$ *along the radial direction* v.

Full derivations are given in Appendix D.5. Proposition 2 makes the backward effect of V-scale explicit. If $\|v\|_2^2 \ll C$ for the sink token, the value-path backward signal is strongly attenuated. Conversely, both eigenvalues are close to 1 when $\|v\|_2^2 \gg C$, so V-scale nearly preserves gradient transmission on large-norm value states. V-scale also contracts small-norm value states in the forward pass. We separate this effect from gradient modulation with the backward-only control below.

Several recent architectural variants mitigate AS or MA by changing the softmax function, inserting gates, or modifying normalization structure [6, 27, 37, 49]. Unlike these broader architectural changes, V-scale is intended as a mechanism test that adds a transparent value-path gradient valve.

Backward-only control Full V-scale changes both the forward values and their backward Jacobian. To isolate the latter, we also test a backward-only control of V-scale

$$\hat{v}_{\text{BO}} = \text{sg}(v) + \phi(\|v\|_2^2)v - \text{sg}\big(\phi(\|v\|_2^2)v\big)\,, \tag{5}$$

where sg stops gradients and is implemented with `detach`. The scaled terms cancel in the forward pass, giving $\hat{v}_{\text{BO}} = v$. During backpropagation, autodiff applies $J_\phi(v)$ from Proposition 2. We freeze $\theta_{\ell,h} = 0$ throughout this run to avoid updating a parameter that has no effect on the forward value.

5.2 Experiments

We train V-scale models using the same data, optimizer, and model configurations as the corresponding baselines. We first verify the mechanism using token-wise gradient and activation measurements, and then evaluate models on downstream tasks.

Mechanism verification We re-measure gradient sinks as defined in Definition 1. Figure 7 shows that in the 0.3B V-scale model, the gradient sink remains present but is substantially weaker, with the median $R_{GS}^\ell(0)$ defined in (3) dropping from roughly 40.3 in the baseline to 5.6. This is the expected outcome of a partial intervention on the value-path gradient.

Moreover, Figure 8 displays the layer-wise thresholded Sink Rate $\text{Sink}_s^{\epsilon,\ell}$ in (1) and output norms at several sites. The top row compares the 0.3B baseline, full V-scale, and backward-only models, while the bottom row compares the 1B baseline and full V-scale models. The same AS and MA protocol is used for all models, averaging over a batch of 512 inputs with 2048 tokens.

Across both model scales, V-scale preserves strong AS behavior and in several layers even slightly strengthens it. In contrast, on the MA metrics, the residual-stream and MLP output norms of token 0 are clearly reduced relative to the baselines. Attention output norms also decrease, consistent with the direct contraction of value states, but the much stronger effect appears in MLP outputs where V-scale has no direct forward scaling.

The backward-only control produces a similar reduction in MA while retaining substantial AS, as shown in the top row of Figure 8. These results show that intervening the value-path backward rule alone is sufficient to suppress MA during training.

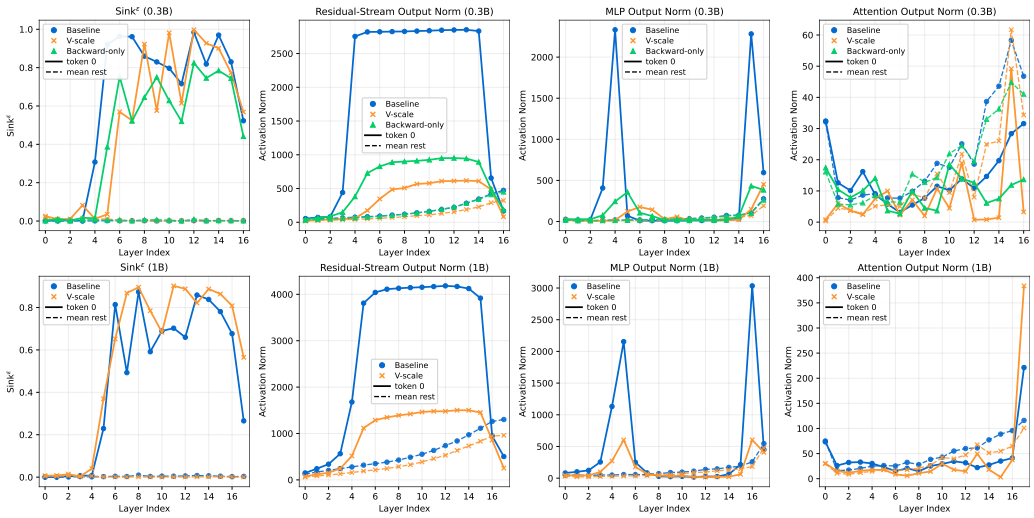


Figure 8: Forward phenomena in baseline, full V-scale, and backward-only models. Top row: 0.3B models; bottom row: the 1B baseline and full V-scale models. From left to right: thresholded sink rate, residual-stream output norm, MLP output norm, and attention output norm. We compare the first token with the mean over the rest early tokens (positions 1–15). Across both scales, V-scale largely preserves attention sinks while reducing the activation norms of token 0, with the larger reduction appearing in MLP outputs. The backward-only control shows the same qualitative pattern.

Table 1: Representative standard downstream evaluation of 1B baseline and V-scale models under `bfloat16`. Lower perplexity (ppl) is better, while higher accuracy (acc) is better. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Model	Wiki ppl↓	Lambada ppl↓	Lambada acc↑	ARC-C acc↑	BoolQ acc↑	COPA acc↑	HellaSwag acc↑	OBQA acc↑	PIQA acc↑	WinoGrande acc↑
Baseline	22.84	**15.27**	43.28	24.91	51.38	69.00	**51.59**	31.40	**72.52**	**54.22**
V-scale	**22.83**	15.91	**43.37**	**27.05**	**56.27**	**70.00**	51.29	**32.80**	71.49	53.28

Overall, these observations match the qualitative prediction of our mechanism. By providing an additional gradient valve on the dominant value path, V-scale reduces the pressure for the model to realize MA through large MLP outputs.

Downstream performance Beyond mechanism verification, we next examine how V-scale behaves on downstream evaluations. We evaluate the 1B baseline and full V-scale models on standard language-modeling and reasoning tasks using the LM Evaluation Harness [12]. This serves as a basic capability check, since suppressing massive activations should not come at the cost of ordinary downstream behavior. Table 1 shows comparable overall performance under `bfloat16`. V-scale substantially improves ARC-C and BoolQ, while some other tasks show small decreases.

As a further behavioral diagnostic, we evaluate Needle-in-a-Haystack (NIAH) retrieval under `bfloat16` and four post-training quantization methods: SmoothQuant (W8A8) [42], BitsAndBytes (BNB, W4A16) [8], GPTQ (W4A16) [9], and AWQ (W4A16) [17], where WxAy denotes x-bit weights and y-bit activations. Table 2 reports mean accuracy of multi-key NIAH over 5 evaluation seeds. V-scale outperforms the baseline in every tested combination of precision and context length, both within the native context and after YaRN extension [25]. At 8192 tokens, the advantages range from 8.68 to 15.04 percentage points. These consistent gains demonstrate improved retrieval when multiple key-value associations compete within a long context.

There are mixed results on the harder single-needle tasks shown in Appendix C.4. The performance drops in some quantization settings after YaRN extension, but the causes of these interactions remain unresolved. The gains on multi-key NIAH, BoolQ, and ARC-C may reflect their shared demand for selecting relevant information under competing cues. Learned V-scale parameters vary across

Table 2: Main multi-key Needle-in-a-Haystack result for 1B models. Entries are mean retrieval accuracy percentages over 5 random seeds; higher is better. We directly compare the baseline (B) and V-scale (V) models under native-context evaluation and YaRN extension with a factor of 4. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Quantization	Native				YaRN-extension							
	1024		2048		1024		2048		4096		8192	
	B	V	B	V	B	V	B	V	B	V	B	V
`bfloat16`	67.96	**73.88**	65.04	**71.64**	57.76	**59.28**	53.88	**59.72**	48.00	**56.16**	48.32	**62.00**
SmoothQuant	64.44	**71.64**	61.64	**69.60**	52.44	**57.92**	49.64	**57.92**	45.76	**52.88**	48.96	**57.64**
BNB	66.16	**74.04**	61.60	**70.56**	54.76	**58.00**	46.28	**58.40**	42.72	**55.88**	47.56	**62.60**
GPTQ	66.20	**67.88**	63.00	**65.56**	56.28	**56.68**	50.24	**57.08**	45.20	**51.16**	46.72	**58.96**
AWQ	63.96	**66.84**	61.68	**67.76**	54.04	**59.24**	49.32	**59.00**	42.52	**49.64**	47.04	**58.48**

layers and heads (see Figure 14 in Appendix C.3), but connecting this variation to performance gains requires further analysis of the affected head functions.

Sensitivity to activation quantization To examine sensitivity to activation range more directly, we compare symmetric W8A8 quantization with dynamic per-token and calibrated static per-tensor activation scales. The detailed results are deferred to Appendix C.5. Dynamic scaling keeps both models close to their `bfloat16` performance. However, under the more stressful static scaling, baseline retrieval falls to near-zero accuracy across all NIAH variants at 2048 tokens, whereas V-scale retains substantial retrieval capability. Quantizing only the MLP `down_proj` modules nearly reproduces this contrast. These results show that V-scale is less sensitive to a coarse, fixed activation range along the MLP pathway associated with its strongest MA reduction.

6 Conclusion

In this work, we revisit the relationship between attention sinks and massive activations from the perspective of backpropagation empirically and theoretically. Under causal masking, AS can induce concentrated gradient pressure on the sink token, forming gradient sinks, and MA emerges as an adaptive regulator of this localized pressure in pre-norm Transformers. To test this hypothesis, we introduce V-scale, a value-path intervention that modulates backpropagated gradients, and a backward-only control that removes its direct forward-value change. Across model sizes, we find that V-scale preserves AS but substantially suppresses MA, illustrating that the two phenomena can be separated once gradient sinks are controlled. The control retains substantial AS while suppressing MA, showing that backward modulation alone is sufficient for the predicted effect. On downstream tasks, V-scale remains comparable to the baseline on standard evaluations and improves multi-key retrieval across all tested context lengths and precision settings.

Our findings motivate studying how sink-induced gradient regulation interacts with numerical precision during training [26]. A broader direction is to examine whether interventions on this mechanism preserve in-context learning [41] and data-efficient agentic learning [40]. These connections call for direct evaluations of few-shot learning and agent experience reuse.

Limitations This work has some limitations. First, our study is restricted to Llama-like dense language models. It remains unclear whether and to what extent the same mechanism transfers to other settings, such as mixture-of-experts architectures or multimodal Transformers. Second, our analysis does not exclude other factors, such as data distribution, optimizer, or numerical precision, from shaping AS and MA during training. These limitations are left for future work.

Acknowledgements

Q. Yao is sponsored by Beijing Nova Program. Z. Lin was supported by the NSF China (No. 62276004), the Beijing Natural Science Foundation (No. L257007), and the Beijing Major Science and Technology Project (No. Z251100008425006).

References

[1] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, 2023.

[2] Y. An, X. Zhao, T. Yu, M. Tang, and J. Wang. Systematic outliers in large language models. In *ICLR*, 2025.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[4] F. Barbero, A. Arroyo, X. Gu, C. Perivolaropoulos, P. Velickovic, R. Pascanu, and M. M. Bronstein. Why do llms attend to the first token? In *COLM*, 2025.

[5] Y. Bondarenko, M. Nagel, and T. Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *NeurIPS*, 2023.

[6] R. Bu, H. Zhong, W. Chen, and Y. Li. Value-state gated attention for mitigating extreme-token phenomena in transformers. *arXiv:2510.09017*, 2025.

[7] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *NeurIPS*, 2022.

[8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023.

[9] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *ICLR*, 2023.

[10] Z. Fu, W. Zeng, R. Wang, and M. Li. Attention sink forges native moe in attention layers: Sink-aware training to address head collapse. In *ICML*, 2026.

[11] J. Gallego-Feliciano, S. A. McClendon, J. Morinelli, S. Zervoudakis, and A. Saravanos. Hidden dynamics of massive activations in transformer training. *arXiv:2508.03616*, 2025.

[12] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, et al. A framework for few-shot language model evaluation, 2023.

[13] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024.

[14] X. Gu, T. Pang, C. Du, Q. Liu, F. Zhang, C. Du, Y. Wang, and M. Lin. When attention sink emerges in language models: An empirical view. In *ICLR*, 2025.

[15] T. Guo, D. Pai, Y. Bai, J. Jiao, M. I. Jordan, and S. Mei. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. In *CPAL*, 2025.

[16] P. Kaul, C. Ma, I. Elezi, and J. Deng. From attention to activation: Unraveling the enigmas of large language models. In *ICLR*, 2025.

[17] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *MLSys*, 2024.

[18] G. Liu, W. Lin, T. Huang, R. Mo, Q. Mu, X. Wang, and L. Shen. Surgery: Mitigating harmful fine-tuning for large language models via attention sink. In *ICML*, 2026.

[19] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han. Understanding the difficulty of training transformers. In *EMNLP*, 2020.

[20] X. Liu, G. Chen, and W. Wang. Sinktrack: Attention sink based context anchoring for large language models. In *ICLR*, 2026.

[21] Y. Liu, J. C. Liang, H. Fan, W. Yang, Y. Cui, X. Han, L. Huang, D. Liu, Q. Wang, and C. Han. All you need is one: Capsule prompt tuning with a single vector. In *NeurIPS*, 2025.

[22] S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate bayesian inference. *JMLR*, 2017.

[23] S. McCandlish, J. Kaplan, D. Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv:1812.06162*, 2018.

[24] E. Miller. Attention is off by one, 2023. URL `https://www.evanmiller.org/attention-is-off-by-one.html`.

[25] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. YaRN: Efficient context window extension of large language models. In *ICLR*, 2024.

[26] H. Qiu and Q. Yao. Why low-precision transformer training fails: An analysis on Flash Attention. In *ICLR*, 2026.

[27] Z. Qiu, Z. Wang, B. Zheng, Z. Huang, K. Wen, S. Yang, R. Men, L. Yu, F. Huang, S. Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *NeurIPS*, 2025.

[28] Z. Qiu, Z. Huang, K. Wen, P. Jin, B. Zheng, Y. Zhou, H. Huang, Z. Wang, X. Li, H. Zhang, et al. A unified view of attention and residual sinks: Outlier-driven rescaling is essential for transformer training. *arXiv:2601.22966*, 2026.

[29] E. Queipo-de Llano, A. Arroyo, F. Barbero, X. Dong, M. M. Bronstein, Y. LeCun, and R. Shwartz-Ziv. Attention sinks and compression valleys in llms are two sides of the same coin. In *ICLR*, 2026.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[32] N. Shazeer. Glu variants improve transformer. *arXiv:2002.05202*, 2020.

[33] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[34] Z. Su and K. Yuan. Kvsink: Understanding and enhancing the preservation of attention sinks in kv cache quantization for llms. In *COLM*, 2025.

[35] Z. Su, Q. Li, H. Zhang, W. Ye, Q. Xue, Y. Qian, N. Wong, and K. Yuan. Unveiling super experts in mixture-of-experts large language models. In *ICLR*, 2026.

[36] M. Sun, X. Chen, J. Z. Kolter, and Z. Liu. Massive activations in large language models. In *COLM*, 2024.

[37] S. Sun, A. Canziani, Y. LeCun, and J. Zhu. Anatomy of massive activations and attention sinks. In *ICML*, 2026.

[38] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[40] Y. Wang, Z. Luo, P. Zhao, Y. Cai, and Q. Yao. Beyond scaling: A survey on data-efficient agentic learning. In *IJCAI*, 2026.

[41] S. Wu, Y. Wang, and Q. Yao. Why in-context learning models are good few-shot learners? In *ICLR*, 2025.

[42] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

[43] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

[44] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *ICML*, 2020.

[45] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

[46] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv:2505.09388*, 2025.

[47] A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Zhang, D. Rojas, G. Feng, H. Zhao, H. Lai, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv:2406.12793*, 2024.

[48] B. Zhang and R. Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019.

[49] Z. M. K. Zuhri, E. H. Fuadi, and A. F. Aji. Softpick: No attention sink, no massive activations with rectified softmax. In *Findings of ACL*, 2026.

A Related work

Phenomena and utilization Attention sinks (AS) refer to the tendency of Transformer-based LLMs to route disproportionate attention mass to a small set of early or otherwise uninformative tokens [14, 43]. Their practical importance was first made especially clear by StreamingLLM, which showed that preserving sink tokens in the KV cache enables stable streaming inference beyond the training context window [43]. Subsequent work has used or analyzed sink tokens for KV-cache compression, long-context inference, context anchoring, prompt tuning, and fine-tuning robustness [10, 18, 20, 21, 34]. Massive activations (MA) are sparse but extremely large activation values that recur in LLMs and often appear at specific token positions or features [2, 36]. They are closely related to activation outliers that dominate numerical ranges and complicate low-bit inference [5, 7, 42]. AS and MA are extreme-token phenomena whose functional roles and training origins have become active mechanistic questions [15, 28]. Our work identifies gradient sinks as a missing backward-pass phenomenon.

Mechanistic interpretations The study of extreme-token phenomena has produced several complementary explanations. One influential view interprets attention sinks as a way for attention heads to implement "no-op" behavior under the sum-to-one constraint of softmax attention. When a head has little useful information to mix into the current representation, it can place most of its attention on low-information tokens whose value vectors contribute little to the residual update [5, 14]. Other work argues that first-token attention can serve useful information-propagation roles, such as reducing over-mixing in deep Transformers [4]. Closely related work studies value-state drains, where sink tokens receive high attention but have unusually small value norms, and active-dormant head dynamics and value-state gating both emphasize this attention-value coupling [6, 15]. Another line connects attention sinks to massive activations, compression valleys, and outlier-driven rescaling, explaining how extreme residual or activation structures can shape forward attention behavior [16, 28, 29, 36, 37]. These accounts clarify why AS and MA often co-occur and why they can be useful or stable forward structures. Our work complements these explanations by treating attention sinks as sources of backward gradient concentration.

Mitigation and intervention Many studies regard attention sinks or activation outliers as harmful to LLMs and seek to mitigate them. One family changes the softmax mechanism itself, using clipped softmax, rectified alternatives, or related normalizers to reduce forced attention allocation and activation outliers [5, 16, 24, 49]. Another family introduces gates after the attention output or on the value states, giving heads an explicit route for suppressing no-op updates without relying on extreme attention logits [6, 27]. Recent work further studies the anatomy of AS and MA across architectures, showing that architectural choices can decouple the two phenomena [37]. Our proposed V-scale has a narrower purpose. It is designed as a mechanism test that keeps query-key attention scores unchanged while attenuating value-path gradient pressure, allowing us to ask whether MA can be suppressed while AS is largely preserved.

Normalization and pre-norm Transformers As a central component in modern Transformers, layer normalization was introduced to stabilize hidden-state dynamics, and subsequent work showed that normalization placement in Transformer residual blocks strongly affects optimization and gradient transport [3, 44]. RMSNorm removes mean-centering while preserving rescaling invariance, and it has become standard in Llama-like decoder-only models [13, 38, 45, 46, 47, 48]. In modern pre-norm RMSNorm Transformers, attention and MLP branches operate on normalized inputs, so very large residual-stream norms have a limited direct role in the forward branch computation. This naturally motivates us to ask how normalization shapes backward signal propagation during training. Qiu and Yao [26] study how numerical errors in low-precision FlashAttention accumulate through training dynamics and lead to instability. Our work examines how attention-induced gradient concentration is regulated through activation scale in pre-norm Transformers.

B Model and training configuration

Table 3 summarizes the configurations for the scratch-trained baselines and their matched V-scale variants. Unless otherwise specified, all models are decoder-only Transformers with RMSNorm, RoPE positional encoding, SwiGLU feed-forward blocks, and causal self-attention. We use the GPT-2 tokenizer [30]. All training runs were implemented in PyTorch using NVIDIA A100 GPUs (80GB). During preprocessing, documents are packed by inserting `<eos>` as the separator, without introducing dedicated `<bos>`. Weight decay is applied only to 2-D weight matrices. V-scale runs use the same data, optimizer, and schedule as their corresponding baselines.

Table 3: Model and training configuration.

Item	0.1B runs	0.3B runs	1B runs
Model			
total parameters	$\sim$101M	$\sim$304M	$\sim$1.04B
vocab size	50257	50257	50257
hidden layers	16	17	18
hidden size	512	1024	2048
attention heads	8	8	16
key/value heads	8	4	8
head dimensions	64	128	128
intermediate size	1408	2816	5504
RMSNorm ϵ	1e-5	1e-5	1e-5
RoPE base	10000	10000	10000
Optimizer			
optimizer	AdamW	AdamW	AdamW
AdamW β_1	0.9	0.9	0.9
AdamW β_2	0.95	0.95	0.95
AdamW ϵ	1e-8	1e-8	1e-8
weight decay	0.1	0.1	0.1
gradient clipping norm	1.0	1.0	1.0
max learning rate	2e-3	1e-3	1e-3
min learning rate	2e-4	1e-4	1e-4
Training			
learning rate schedule	cosine decay	cosine decay	cosine decay
total steps	20000	20000	25000
warmup steps	1000	1000	1000
context length	1024	2048	2048
global batch size	512	512	512
tokens per step	0.52M	1.05M	1.05M
total tokens	10.49B	20.97B	26.21B
training data	C4	C4	C4

C Additional empirical results

C.1 Supplementary results for QKV gradients

This subsection supplements the QKV-path decomposition in Figure 3. Figure 9 extends the diagnostic to smaller scratch-trained models, using the same layout as the 1B result in the main text. Figure 10 reports the corresponding measurements for pretrained LLMs. Across these settings, the query pathway remains comparatively flat, whereas key and especially value gradients show a pronounced spike at token 0.

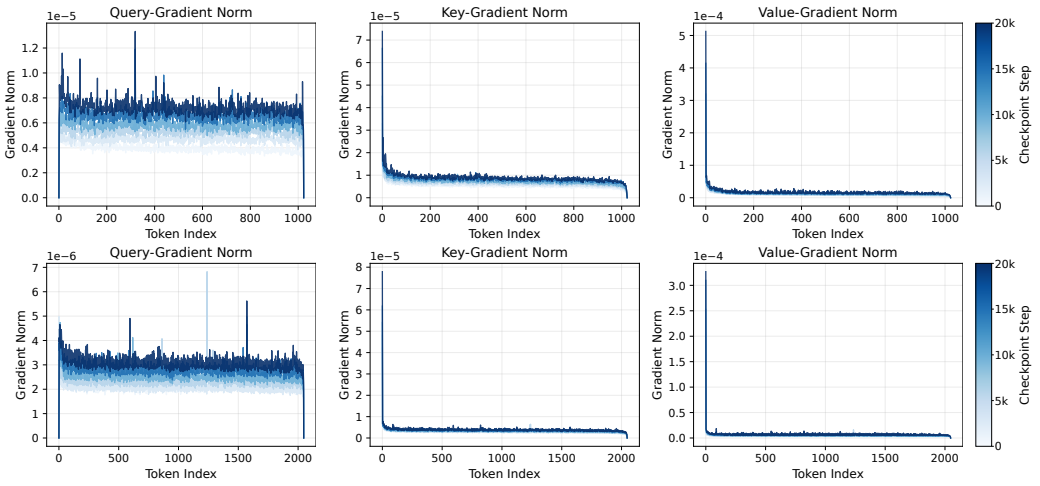


Figure 9: Token-wise QKV gradient norms across training checkpoints for smaller scratch-trained models, averaged over layers. Top row: 0.1B model; bottom row: 0.3B model. From left to right: query, key, and value gradient norms as functions of token position. Both models reproduce the pathway asymmetry observed in the 1B model: key and especially value gradients exhibit a strong spike at token 0, while query gradients remain comparatively flat.

C.2 Supplementary results for gradient reshaping

This subsection complements the gradient-reshaping measurements in Section 3.2 with the analogous MLP-branch diagnostics. Unlike the attention-branch results in Figure 4, the left panel of Figure 11 shows that $\mathrm{Bloat}^{\mathrm{mlp}}$ is much less dominated by token 0. Thus, the MLP branch does not appear to be the primary location where localized pressure is generated. The middle panel shows that $\mathrm{Change}^{\mathrm{mlp}}$ remains concentrated near zero ($\log_{10} 1$), and the right panel again shows a tight alignment between activation norms and compression, as observed in attention blocks. Figure 12 gives the corresponding attention-branch and MLP-branch measurements for the 0.3B baseline and pretrained LLMs. The same qualitative contrast persists: bloat at token 0 is sharper in the attention branch than in the MLP branch, whereas compression remains strongly tied to activation scale in both branches. Qwen3-8B is somewhat less aligned with the other pretrained models, which may reflect its post-training procedure.

C.3 Supporting observations for V-scale design

This subsection provides additional diagnostics for the design of V-scale. This intervention is intended to act as a value-path gradient valve. It should be strongest when the sink token has a small value-state norm, and the learned scale parameter should adapt across layers and heads. Figure 13 checks the first condition in baseline models. Across training checkpoints, the value state of token 0 is usually much smaller than the mean and maximum over other early tokens, especially in middle and later layers. Since the V-scale Jacobian attenuates gradients most strongly when $\|v\|_2^2 \ll C_{\ell,h}$, this pattern makes the intervention selective for the sink token rather than a uniform shrinkage of all value states.

Figure 14 reports the final learned V-scale parameters. Since $\lambda^{\ell,h} = \exp(\theta_{\ell,h})$ is initialized to one, positive $\theta_{\ell,h}$ increases the value-state scale parameter $C_{\ell,h}$ and strengthens attenuation for a fixed small value norm, while negative $\theta_{\ell,h}$ decreases $C_{\ell,h}$. Both scales show a clear layer-dependent

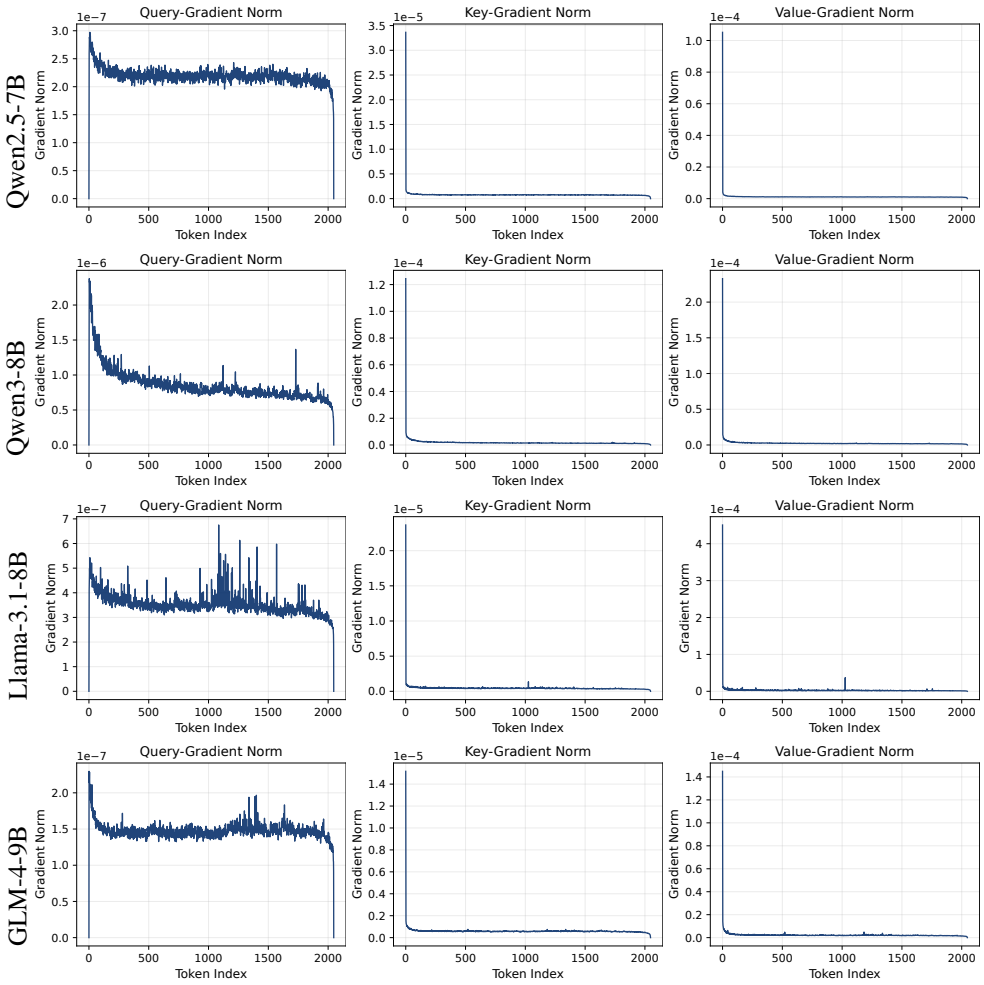


Figure 10: Token-wise gradient norms of QKV for pretrained LLMs. Each row corresponds to one model. From left to right: gradient norms of query, key, and value as functions of token position, averaged over layers. Across models, key and especially value gradients consistently exhibit a pronounced spike at token 0.

pattern, with larger positive values concentrated in earlier layers and mostly negative values in later layers. This indicates that V-scale is not uniform but instead learns layer- and head-specific strength for the value-path valve. Such heterogeneity may interact with specialized pathways for selecting relevant information under competing cues, potentially connecting to the gains on multi-key NIAH, BoolQ, and ARC-C reported in Appendix C.4. This hypothesis requires further evidence.

Finally, Figure 15 complements the 0.3B comparison in Figure 7 by showing the distributions of gradient sinks for additional V-scale models. Token 0 remains the largest-gradient position but the remaining concentration is moderate compared to the matched baselines in Figure 2.

C.4 Additional downstream tasks

This subsection reports additional downstream evaluations, supplementing the compact summary in Section 5. We focus on 1B models because this is the largest scratch-trained setting in our study. We pair each baseline with its V-scale counterpart and evaluate language modeling, reasoning, and retrieval in `bfloat16` and under post-training quantization. These measurements assess how the intervention affects downstream capability across precision settings.

Table 4 shows that V-scale maintains similar perplexity and improves ARC-C, BoolQ, COPA, and OpenBookQA under every listed precision setting. Several other tasks show small decreases. Together

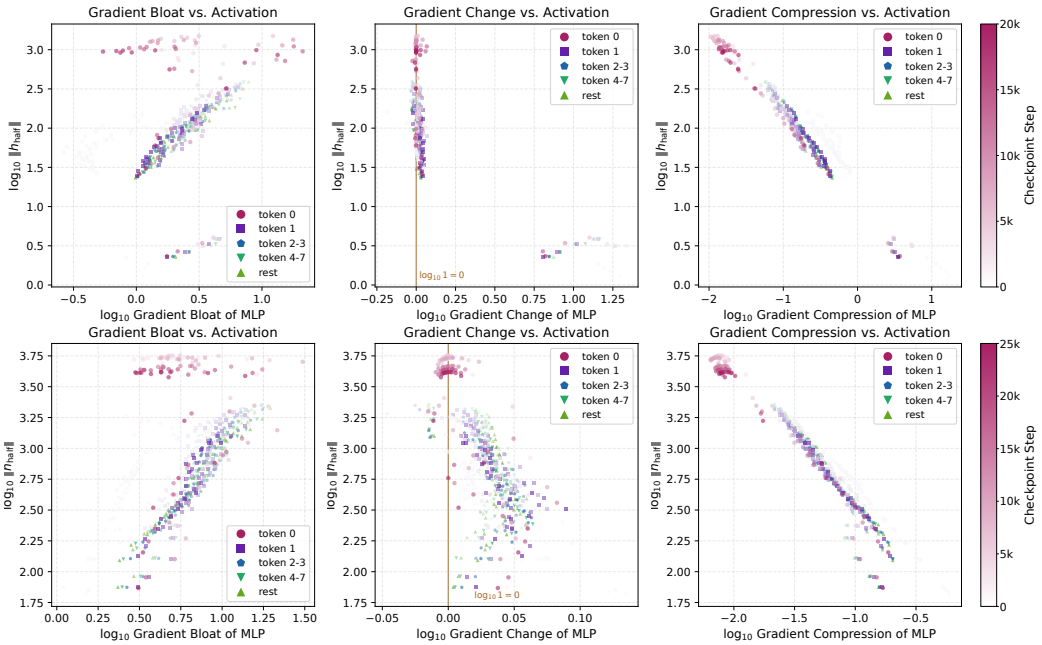


Figure 11: Scatter plots relating gradient reshaping to input activation norms of the MLP block. Top row: 0.1B model; bottom row: 1B model. From left to right: $\log_{10}\mathrm{Bloat}^{\mathrm{mlp}}$ vs. $\log_{10}\|h_{\mathrm{half}}\|$, $\log_{10}\mathrm{Change}^{\mathrm{mlp}}$ vs. $\log_{10}\|h_{\mathrm{half}}\|$, and $\log_{10}\mathrm{Compress}^{\mathrm{mlp}}$ vs. $\log_{10}\|h_{\mathrm{half}}\|$. Each point is colored by token group (token 0, token 1, tokens 2–3, tokens 4–7, and the rest early tokens 8–15). Compared with the attention branch, token 0 is less dominant in MLP bloat, while compression again aligns tightly with activation scale and the Change ratio remains close to one.

with the consistent multi-key retrieval gains below, these results show that suppressing MA preserves overall language-model capability and benefits several tasks requiring selective information use.

We next evaluate long-context retrieval using three Needle-in-a-Haystack (NIAH) variants. The native 1B models are evaluated up to 2048 tokens. We apply YaRN with an extension factor of 4 to evaluate contexts up to 8192 tokens. Tables 5–7 report means and standard deviations over 5 evaluation seeds. V-scale consistently improves multi-key retrieval across every listed context length and precision setting in Table 7. The single-needle template in Table 5 is close to saturated at shorter lengths and also improves under every precision setting at 8192 tokens.

The harder single-needle template in Table 6 gives mixed results. V-scale improves native 2048-token accuracy under `bfloat16`, SmoothQuant, BNB, and GPTQ, while AWQ accuracy falls. After YaRN extension, GPTQ and AWQ accuracy at 4096 tokens falls. These degradations show that the benefit observed in multi-key retrieval depends on the retrieval task. The interaction with W4A16 quantization and context extension remains to be explained.

C.5 Controlled activation quantization

We evaluate the 1B baseline and V-scale models under three symmetric INT8 configurations to examine sensitivity to activation range. In the *all-Linear dynamic-token* setting, every Linear module except `lm_head` uses static per-channel INT8 weights and dynamic per-token INT8 input activations. The *all-Linear static-tensor* setting keeps the same target modules and weight format, and replaces dynamic per-token activation scaling with a static per-tensor scale. The *down-projection-only static-tensor* setting applies the same static weight and activation formats only to modules matching `mlp.down_proj`. Dynamic per-token scaling adapts to the activation range of each token. Static per-tensor scaling shares a calibrated range across tokens, exposing sensitivity to activation outliers. Restricting this quantization to the MLP down-projection tests the pathway associated with the strongest MA reduction in Figure 8. These settings serve as controlled diagnostics of sensitivity to a fixed activation range.

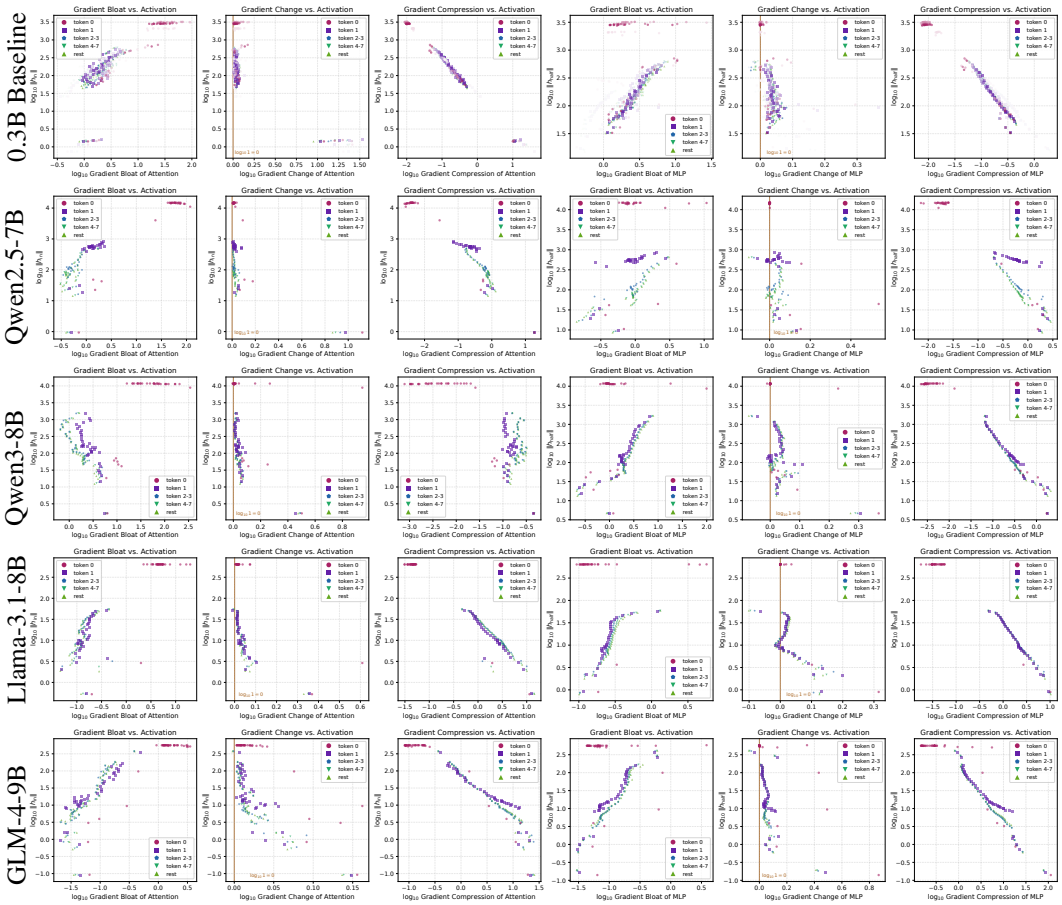


Figure 12: Supplementary gradient-reshaping measurements for the 0.3B baseline and pretrained models. From left to right, each row shows attention-branch Bloat, Change, and Compress, followed by MLP-branch Bloat, Change, and Compress.

We use the same downstream task configurations as in Appendix C.4. For the ten-task accuracy average, we use length-normalized accuracy for ARC-C, ARC-E, HellaSwag, OpenBookQA, PIQA, and SCIQ, and accuracy for BoolQ, COPA, LAMBADA, and WinoGrande. NIAH uses the native 2048-token context and evaluation seeds 0–4, matching the corresponding experiments in Appendix C.4. We report the mean and sample standard deviation across these seeds.

Table 8 shows that both models remain close to their `bfloat16` performance under dynamic per-token scaling. Static per-tensor scaling yields a much larger separation. With all Linear modules quantized, V-scale has substantially lower perplexity and higher mean accuracy than the baseline. Quantizing only the MLP down-projection produces similar perplexities and accuracies, locating most of the observed sensitivity in this pathway.

The retrieval results in Table 9 show the same dependence on activation scaling. Dynamic per-token scaling preserves retrieval in both models. With static per-tensor scaling, the baseline falls to near-zero accuracy on all three tasks, whereas V-scale retains substantial retrieval capability. Quantizing only `down_proj` nearly reproduces the all-Linear result. The contrast supports lower sensitivity to a fixed activation range in the MLP pathway.

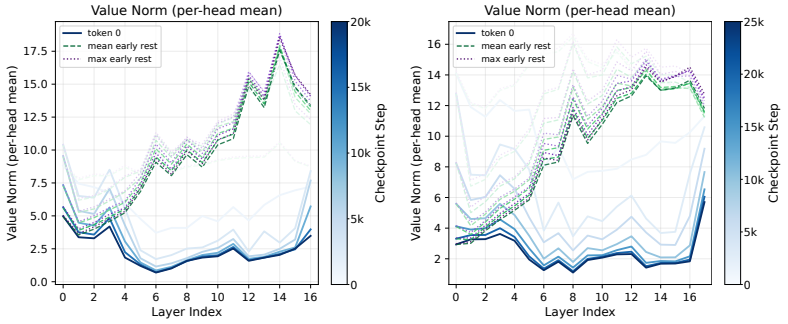


Figure 13: Value-state norms in baseline models, averaged over heads and shown across training checkpoints. Left: 0.3B baseline; right: 1B baseline. The solid curve is token 0, the dashed curve is the mean over other early tokens, and the dotted curve is their maximum, where "early" refers to positions 1–15. Token 0 usually has a much smaller value-state norm, placing it in the regime where V-scale can induce stronger value-path gradient attenuation.

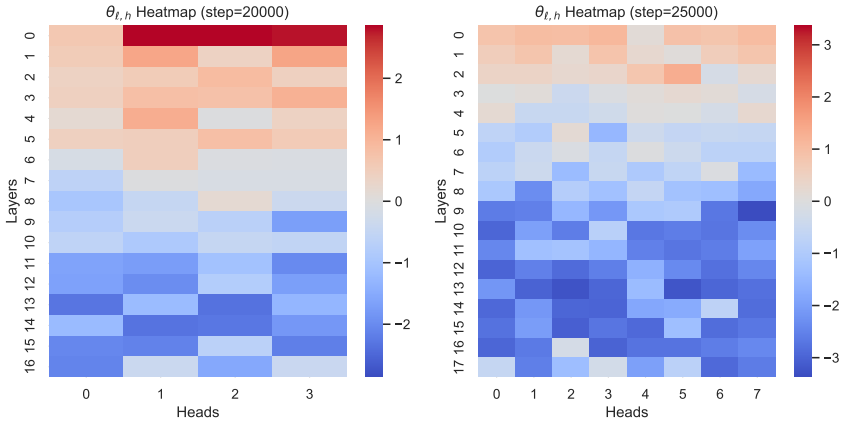


Figure 14: Final learned V-scale parameters $\theta_{\ell,h}$. Left: 0.3B V-scale model; right: 1B V-scale model. Each heatmap indexes layers by rows and attention heads by columns, with the color scale centered at the initialization value $\theta = 0$. Positive values correspond to larger $C_{\ell,h}$ and stronger attenuation for fixed small-norm value states, while negative values correspond to smaller $C_{\ell,h}$.

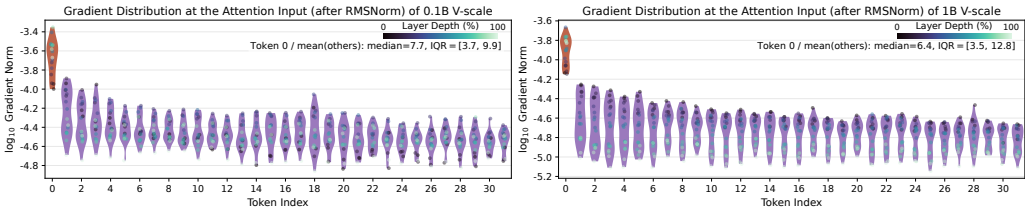


Figure 15: Gradient sinks in additional V-scale models. Left: 0.1B V-scale model; right: 1B V-scale model. Each panel plots the token-wise distribution of $\log_{10} \|\nabla_{\tilde{h}_t^\ell} \mathcal{L}\|_2$ at the post-RMSNorm attention input, with token 0 highlighted. The sink token remains visible but is more moderate.

Table 4: Standard downstream evaluation of 1B baseline and V-scale models with unquantized `bfloat16` and post-training quantization. `bfloat16` denotes unquantized evaluation. SmoothQuant is W8A8 quantization, and BNB, GPTQ, and AWQ are W4A16 quantization methods. Wiki ppl and Lambada ppl are perplexities, where lower is better. Lambada acc, ARC-C/E (AI2 Reasoning Challenge and Easy splits), BoolQ, COPA, HellaSwag, OBQA (OpenBookQA), PIQA, SCIQ, and WinoGrande are zero-shot accuracies in percent, where higher is better. V-scale keeps perplexity nearly unchanged and remains broadly comparable to the baseline across reasoning tasks, with gains on ARC-C, BoolQ, COPA, and OBQA and small drops on several other tasks. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Panel A: language modeling and core reasoning

Model	Quantization	Wiki ppl↓	Lambada ppl↓	Lambada acc↑	ARC-C acc↑	ARC-E acc↑	BoolQ acc↑	COPA acc↑
Baseline	bfloat16	22.84	**15.27**	43.28	24.91	**47.98**	51.38	69.00
V-scale	bfloat16	**22.83**	15.91	**43.37**	**27.05**	47.47	**56.27**	**70.00**
Baseline	SmoothQuant	23.17	16.61	43.02	25.26	**47.47**	50.92	67.00
V-scale	SmoothQuant	**23.01**	**16.58**	**43.14**	**26.11**	46.76	**55.57**	**69.00**
Baseline	BNB	23.49	16.40	42.73	25.00	**47.43**	54.19	72.00
V-scale	BNB	**23.40**	**15.86**	**43.55**	**27.56**	47.05	**55.96**	**74.00**
Baseline	GPTQ	23.21	**15.70**	43.26	25.77	**48.48**	52.54	68.00
V-scale	GPTQ	**23.17**	16.48	**43.76**	**26.79**	46.80	**56.45**	**70.00**
Baseline	AWQ	**23.32**	**16.34**	42.44	25.26	**48.06**	53.64	68.00
V-scale	AWQ	23.36	16.95	**43.06**	**26.71**	47.10	**55.96**	**69.00**

Panel B: additional commonsense and science QA tasks

Model	Quantization	HellaSwag	OBQA	PIQA	SCIQ	WinoGrande
Baseline	bfloat16	**51.59**	31.40	**72.52**	**69.90**	**54.22**
V-scale	bfloat16	51.29	**32.80**	71.49	69.30	53.28
Baseline	SmoothQuant	**51.49**	30.80	**72.47**	**69.00**	**55.17**
V-scale	SmoothQuant	51.13	**32.40**	71.00	68.30	54.46
Baseline	BNB	**51.59**	31.40	**73.01**	69.20	**55.56**
V-scale	BNB	51.29	**32.80**	71.27	**69.50**	54.14
Baseline	GPTQ	**51.61**	31.60	**72.85**	**69.20**	**54.70**
V-scale	GPTQ	51.31	**32.40**	71.44	68.40	**54.70**
Baseline	AWQ	**51.23**	30.60	**72.47**	**69.40**	**54.30**
V-scale	AWQ	50.88	**33.20**	71.65	69.10	53.51

Table 5: Needle-in-a-Haystack retrieval accuracy on the `niah_single_2` single-needle template for 1B models. Columns denote context length in tokens, and entries are mean accuracy percentages with standard deviations over 5 random seeds; higher is better. The first block evaluates native-context 1B models, for which only 1024 and 2048 tokens are tested. Blank cells therefore indicate untested longer contexts rather than failed runs. The second block evaluates the same models after YaRN extension with a factor of 4, enabling 4096- and 8192-token tests. This template is nearly saturated at shorter lengths, but V-scale gives consistently higher accuracy at 8192 tokens after YaRN extension. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Model	Quantization	1024	2048	4096	8192
1B Models					
Baseline	bfloat16	$99.96_{\pm 0.09}$	$\mathbf{98.00}_{\pm 0.69}$		
V-scale	bfloat16	$\mathbf{100.00}_{\pm 0.00}$	$97.80_{\pm 0.42}$		
Baseline	SmoothQuant	$99.96_{\pm 0.09}$	$\mathbf{95.92}_{\pm 2.09}$		
V-scale	SmoothQuant	$\mathbf{100.00}_{\pm 0.00}$	$95.64_{\pm 0.99}$		
Baseline	BNB	$99.88_{\pm 0.18}$	$\mathbf{98.36}_{\pm 0.86}$		
V-scale	BNB	$\mathbf{100.00}_{\pm 0.00}$	$95.76_{\pm 0.84}$		
Baseline	GPTQ	$99.92_{\pm 0.11}$	$94.64_{\pm 1.56}$		
V-scale	GPTQ	$\mathbf{100.00}_{\pm 0.00}$	$\mathbf{96.68}_{\pm 0.46}$		
Baseline	AWQ	$99.92_{\pm 0.11}$	$96.56_{\pm 1.24}$		
V-scale	AWQ	$\mathbf{100.00}_{\pm 0.00}$	$\mathbf{97.20}_{\pm 0.71}$		
1B Models with YaRN Extension (yarn_factor= 4)					
Baseline	bfloat16	$\mathbf{99.84}_{\pm 0.22}$	$98.64_{\pm 0.26}$	$98.20_{\pm 0.66}$	$94.20_{\pm 0.75}$
V-scale	bfloat16	$99.40_{\pm 0.40}$	$\mathbf{99.56}_{\pm 0.38}$	$\mathbf{98.32}_{\pm 0.41}$	$\mathbf{99.04}_{\pm 0.33}$
Baseline	SmoothQuant	$\mathbf{99.72}_{\pm 0.23}$	$97.48_{\pm 0.84}$	$93.24_{\pm 1.21}$	$88.40_{\pm 1.67}$
V-scale	SmoothQuant	$98.52_{\pm 0.30}$	$\mathbf{97.96}_{\pm 0.17}$	$\mathbf{93.72}_{\pm 0.63}$	$\mathbf{96.16}_{\pm 0.70}$
Baseline	BNB	$99.60_{\pm 0.37}$	$99.20_{\pm 0.37}$	$98.12_{\pm 0.87}$	$93.32_{\pm 0.77}$
V-scale	BNB	$\mathbf{99.76}_{\pm 0.17}$	$\mathbf{99.96}_{\pm 0.09}$	$\mathbf{99.40}_{\pm 0.35}$	$\mathbf{98.52}_{\pm 0.50}$
Baseline	GPTQ	$\mathbf{99.80}_{\pm 0.20}$	$97.32_{\pm 0.61}$	$\mathbf{96.92}_{\pm 0.44}$	$93.08_{\pm 1.18}$
V-scale	GPTQ	$99.64_{\pm 0.22}$	$\mathbf{98.96}_{\pm 0.43}$	$93.52_{\pm 1.14}$	$\mathbf{97.72}_{\pm 0.27}$
Baseline	AWQ	$\mathbf{99.72}_{\pm 0.23}$	$\mathbf{97.44}_{\pm 0.54}$	$\mathbf{94.60}_{\pm 1.02}$	$90.84_{\pm 1.07}$
V-scale	AWQ	$96.96_{\pm 0.77}$	$94.56_{\pm 0.52}$	$92.36_{\pm 1.40}$	$\mathbf{98.32}_{\pm 0.84}$

Table 6: Needle-in-a-Haystack retrieval accuracy on the harder `niah_single_3` single-needle template for 1B models. Columns denote context length in tokens, and entries are mean accuracy percentages with standard deviations over 5 random seeds; higher is better. The first block reports native-context evaluation up to 2048 tokens, while the second block uses YaRN extension with a factor of 4 to evaluate up to 8192 tokens. On native 2048-token evaluation, V-scale improves over the baseline for `bfloat16`, SmoothQuant, BNB, and GPTQ, with AWQ as the exception. After YaRN extension, the result is mixed, with gains in several 2048-token and BNB settings but degradation under some 4096- and 8192-token quantized settings. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Model	Quantization	1024	2048	4096	8192
1B Models					
Baseline	bfloat16	$\mathbf{99.68}_{\pm 0.11}$	$85.48_{\pm 1.24}$		
V-scale	bfloat16	$98.60_{\pm 0.79}$	$\mathbf{95.60}_{\pm 0.95}$		
Baseline	SmoothQuant	$\mathbf{99.00}_{\pm 0.28}$	$79.28_{\pm 0.93}$		
V-scale	SmoothQuant	$96.00_{\pm 0.84}$	$\mathbf{88.48}_{\pm 0.46}$		
Baseline	BNB	$99.32_{\pm 0.18}$	$79.80_{\pm 1.29}$		
V-scale	BNB	$\mathbf{99.44}_{\pm 0.48}$	$\mathbf{96.40}_{\pm 0.51}$		
Baseline	GPTQ	$\mathbf{99.44}_{\pm 0.26}$	$85.84_{\pm 1.88}$		
V-scale	GPTQ	$98.84_{\pm 0.75}$	$\mathbf{95.84}_{\pm 1.13}$		
Baseline	AWQ	$\mathbf{99.44}_{\pm 0.26}$	$\mathbf{96.16}_{\pm 1.25}$		
V-scale	AWQ	$91.96_{\pm 1.01}$	$85.20_{\pm 1.54}$		
1B Models with YaRN Extension (yarn_factor= 4)					
Baseline	bfloat16	$98.96_{\pm 0.26}$	$91.56_{\pm 2.03}$	$\mathbf{79.48}_{\pm 1.08}$	$\mathbf{75.68}_{\pm 1.51}$
V-scale	bfloat16	$\mathbf{99.08}_{\pm 0.70}$	$\mathbf{94.36}_{\pm 1.01}$	$77.16_{\pm 1.42}$	$73.52_{\pm 1.10}$
Baseline	SmoothQuant	$97.00_{\pm 0.82}$	$80.12_{\pm 2.11}$	$\mathbf{71.96}_{\pm 2.07}$	$64.00_{\pm 1.38}$
V-scale	SmoothQuant	$\mathbf{98.56}_{\pm 0.84}$	$\mathbf{84.84}_{\pm 1.15}$	$64.72_{\pm 1.57}$	$\mathbf{66.92}_{\pm 1.20}$
Baseline	BNB	$98.60_{\pm 0.63}$	$89.64_{\pm 1.11}$	$75.68_{\pm 1.14}$	$\mathbf{75.52}_{\pm 2.39}$
V-scale	BNB	$\mathbf{99.16}_{\pm 0.48}$	$\mathbf{98.64}_{\pm 0.46}$	$\mathbf{82.52}_{\pm 0.94}$	$71.72_{\pm 1.98}$
Baseline	GPTQ	$98.92_{\pm 0.23}$	$90.68_{\pm 1.30}$	$\mathbf{75.72}_{\pm 1.32}$	$\mathbf{73.68}_{\pm 0.70}$
V-scale	GPTQ	$\mathbf{99.36}_{\pm 0.57}$	$\mathbf{94.36}_{\pm 0.99}$	$61.04_{\pm 1.58}$	$69.20_{\pm 1.97}$
Baseline	AWQ	$\mathbf{98.76}_{\pm 0.52}$	$\mathbf{96.56}_{\pm 0.50}$	$\mathbf{78.24}_{\pm 0.74}$	$\mathbf{74.20}_{\pm 1.83}$
V-scale	AWQ	$98.24_{\pm 0.75}$	$87.84_{\pm 1.11}$	$49.80_{\pm 1.73}$	$70.56_{\pm 1.98}$

Table 7: Needle-in-a-Haystack retrieval accuracy on the `niah_multikey_1` multi-key template for 1B models. This setting requires retrieving information when multiple key-value associations are present, making it more sensitive to long-context attention allocation than the saturated single-needle cases. Columns denote context length in tokens, and entries are mean accuracy percentages with standard deviations over 5 random seeds; higher is better. Native-context models are evaluated at 1024 and 2048 tokens, and YaRN-extended models with a factor of 4 are evaluated up to 8192 tokens. V-scale improves every listed precision/quantization and context-length combination, with especially large gains at 8192 tokens after YaRN extension. Bold marks the better result within each baseline and V-scale pair under the same evaluation setting.

Model	Quantization	1024	2048	4096	8192
1B Models					
Baseline	bfloat16	$67.96_{\pm 1.34}$	$65.04_{\pm 1.09}$		
V-scale	bfloat16	$\mathbf{73.88}_{\pm 1.06}$	$\mathbf{71.64}_{\pm 3.46}$		
Baseline	SmoothQuant	$64.44_{\pm 1.43}$	$61.64_{\pm 2.91}$		
V-scale	SmoothQuant	$\mathbf{71.64}_{\pm 1.44}$	$\mathbf{69.60}_{\pm 2.19}$		
Baseline	BNB	$66.16_{\pm 1.68}$	$61.60_{\pm 1.83}$		
V-scale	BNB	$\mathbf{74.04}_{\pm 1.65}$	$\mathbf{70.56}_{\pm 2.19}$		
Baseline	GPTQ	$66.20_{\pm 1.12}$	$63.00_{\pm 1.54}$		
V-scale	GPTQ	$\mathbf{67.88}_{\pm 1.06}$	$\mathbf{65.56}_{\pm 2.25}$		
Baseline	AWQ	$63.96_{\pm 2.10}$	$61.68_{\pm 1.52}$		
V-scale	AWQ	$\mathbf{66.84}_{\pm 1.75}$	$\mathbf{67.76}_{\pm 3.10}$		
1B Models with YaRN Extension (yarn_factor= 4)					
Baseline	bfloat16	$57.76_{\pm 2.29}$	$53.88_{\pm 1.42}$	$48.00_{\pm 2.58}$	$48.32_{\pm 1.85}$
V-scale	bfloat16	$\mathbf{59.28}_{\pm 1.77}$	$\mathbf{59.72}_{\pm 2.33}$	$\mathbf{56.16}_{\pm 2.39}$	$\mathbf{62.00}_{\pm 1.88}$
Baseline	SmoothQuant	$52.44_{\pm 1.84}$	$49.64_{\pm 1.73}$	$45.76_{\pm 2.17}$	$48.96_{\pm 1.89}$
V-scale	SmoothQuant	$\mathbf{57.92}_{\pm 2.26}$	$\mathbf{57.92}_{\pm 1.98}$	$\mathbf{52.88}_{\pm 1.18}$	$\mathbf{57.64}_{\pm 1.28}$
Baseline	BNB	$54.76_{\pm 2.60}$	$46.28_{\pm 1.38}$	$42.72_{\pm 2.96}$	$47.56_{\pm 2.22}$
V-scale	BNB	$\mathbf{58.00}_{\pm 2.40}$	$\mathbf{58.40}_{\pm 2.20}$	$\mathbf{55.88}_{\pm 1.43}$	$\mathbf{62.60}_{\pm 1.06}$
Baseline	GPTQ	$56.28_{\pm 1.64}$	$50.24_{\pm 0.84}$	$45.20_{\pm 2.81}$	$46.72_{\pm 1.56}$
V-scale	GPTQ	$\mathbf{56.68}_{\pm 1.36}$	$\mathbf{57.08}_{\pm 0.91}$	$\mathbf{51.16}_{\pm 1.61}$	$\mathbf{58.96}_{\pm 2.20}$
Baseline	AWQ	$54.04_{\pm 1.91}$	$49.32_{\pm 2.02}$	$42.52_{\pm 2.98}$	$47.04_{\pm 2.04}$
V-scale	AWQ	$\mathbf{59.24}_{\pm 1.31}$	$\mathbf{59.00}_{\pm 1.46}$	$\mathbf{49.64}_{\pm 1.95}$	$\mathbf{58.48}_{\pm 2.02}$

Table 8: Standard downstream results under controlled activation quantization for 1B models. Each metric compares the baseline (B) and V-scale (V). Mean accuracy is the unweighted average of the ten accuracy metrics in Table 4. Bold marks the better result within each pair.

	WikiText PPL↓		LAMBADA PPL↓		Mean acc. (%)↑	
Setting	B	V	B	V	B	V
`bfloat16`	22.84	**22.83**	**15.27**	15.91	51.62	**52.23**
All Linear, dynamic-token W8A8	23.16	**23.04**	**16.35**	16.90	51.77	**52.21**
All Linear, static-tensor W8A8	33.55	**25.46**	46.85	**23.65**	48.54	**50.55**
`down_proj` only, static-tensor W8A8	32.94	**25.21**	43.63	**23.07**	48.37	**50.67**

Table 9: NIAH retrieval under controlled activation quantization at the native 2048-token context length. Single-2, Single-3, and Multi-key denote the same templates as Tables 5–7. Entries are mean accuracy percentages with sample standard deviations over five evaluation seeds. Higher is better, and bold marks the better result within each baseline (B) and V-scale (V) pair.

	Single-2		Single-3		Multi-key	
Setting	B	V	B	V	B	V
`bfloat16`	$\mathbf{98.00}_{\pm 0.69}$	$97.80_{\pm 0.42}$	$85.48_{\pm 1.24}$	$\mathbf{95.60}_{\pm 0.95}$	$65.04_{\pm 1.09}$	$\mathbf{71.64}_{\pm 3.46}$
All / dynamic	$96.04_{\pm 1.11}$	$\mathbf{96.28}_{\pm 0.54}$	$77.00_{\pm 1.26}$	$\mathbf{91.40}_{\pm 1.22}$	$63.44_{\pm 1.98}$	$\mathbf{66.88}_{\pm 2.17}$
All / static	$1.52_{\pm 0.39}$	$\mathbf{79.40}_{\pm 2.48}$	$0.00_{\pm 0.00}$	$\mathbf{75.32}_{\pm 2.29}$	$2.76_{\pm 0.86}$	$\mathbf{56.04}_{\pm 2.08}$
`down_proj` / static	$2.16_{\pm 0.36}$	$\mathbf{79.28}_{\pm 1.87}$	$0.04_{\pm 0.09}$	$\mathbf{79.08}_{\pm 1.95}$	$3.40_{\pm 1.09}$	$\mathbf{56.84}_{\pm 2.67}$

D Additional theory and proofs

This appendix consolidates the supplementary theoretical material used in the main text. Throughout, we fix one layer and one attention head, and use the notation introduced in Section 2.

D.1 Exact backpropagation identities

Lemma 1 (Softmax Jacobian identity)**.** *Fix a row t and consider the causal softmax vector $a_t = \mathrm{softmax}(z_t)$ on the support $\{j \le t\}$. Then for any $j, s \le t$,*

$$\frac{\partial a_{tj}}{\partial z_{ts}} = a_{tj}\big(\mathbf{1}_{j=s} - a_{ts}\big).$$

Proof. By definition, we have $a_{tj} = \frac{e^{z_{tj}}}{\sum_{\ell \le t} e^{z_{t\ell}}}$. Differentiating with respect to z_{ts} gives

$$\frac{\partial a_{tj}}{\partial z_{ts}} = \frac{e^{z_{tj}}\mathbf{1}_{j=s}\left(\sum_{\ell \le t} e^{z_{t\ell}}\right) - e^{z_{tj}}e^{z_{ts}}}{\left(\sum_{\ell \le t} e^{z_{t\ell}}\right)^2} = a_{tj}\big(\mathbf{1}_{j=s} - a_{ts}\big).$$

□

Lemma 2 (Sensitivity of y_t to a single logit)**.** *For fixed t and $s \le t$, we have*

$$\frac{\partial y_t}{\partial z_{ts}} = a_{ts}(v_s - y_t).$$

Proof. Using $y_t = \sum_{j \le t} a_{tj} v_j$ and Lemma 1, we have

$$\frac{\partial y_t}{\partial z_{ts}} = \sum\nolimits_{j \le t} \frac{\partial a_{tj}}{\partial z_{ts}} v_j = \sum\nolimits_{j \le t} a_{tj}(\mathbf{1}_{j=s} - a_{ts})v_j = a_{ts}v_s - a_{ts}\sum\nolimits_{j \le t} a_{tj}v_j = a_{ts}(v_s - y_t).$$

□

Proof of Proposition 1. For fixed t, the Jacobian of y_t with respect to v_s is $\frac{\partial y_t}{\partial v_s} = a_{ts} I\mathbb{I}_{s \le t}$. Applying the chain rule from $v_s \to y_t \to \mathcal{L}$ yields

$$\nabla_{v_s}\mathcal{L} = \sum\nolimits_{t=s}^{T-1} \left(\frac{\partial y_t}{\partial v_s}\right)^\top \nabla_{y_t}\mathcal{L} = \sum\nolimits_{t=s}^{T-1} (a_{ts} I)^\top g_t = \sum\nolimits_{t=s}^{T-1} a_{ts} g_t.$$

□

Proposition 3 (Exact key-side gradient)**.** *Define*

$$\beta_{ts} := \langle g_t, v_s - y_t \rangle.$$

Then we have

$$\nabla_{k_s}\mathcal{L} = \sum\nolimits_{t=s}^{T-1} a_{ts}\beta_{ts}\frac{q_t}{\sqrt{d_{\mathrm{head}}}}.$$

Proof. The key k_s affects the logits $z_{ts} = \langle q_t, k_s \rangle / \sqrt{d_{\mathrm{head}}}$ for all $t \ge s$. The chain rule gives

$$\nabla_{k_s}\mathcal{L} = \sum\nolimits_{t=s}^{T-1} \left(\frac{\partial z_{ts}}{\partial k_s}\right)^\top \frac{\partial \mathcal{L}}{\partial z_{ts}}.$$

Using $\frac{\partial z_{ts}}{\partial k_s} = \frac{q_t}{\sqrt{d_{\mathrm{head}}}}$ and Lemma 2, we have

$$\frac{\partial \mathcal{L}}{\partial z_{ts}} = \left\langle \nabla_{y_t}\mathcal{L}, \frac{\partial y_t}{\partial z_{ts}} \right\rangle = \langle g_t, a_{ts}(v_s - y_t) \rangle = a_{ts}\beta_{ts}.$$

Substituting these two identities gives the claim. □

Proposition 4 (Exact query-side gradient and first-token corollary)**.** *For the query pathway,*

$$\nabla_{q_s}\mathcal{L} = \sum\nolimits_{j\le s} a_{sj}\beta_{sj}\frac{k_j}{\sqrt{d_{\text{head}}}}.$$

Moreover, under causal masking, we have $\nabla_{q_0}\mathcal{L} = 0$.

Proof. The query q_s affects only the logits $\{z_{sj}\}_{j\le s}$. Hence

$$\nabla_{q_s}\mathcal{L} = \sum\nolimits_{j\le s}\left(\frac{\partial z_{sj}}{\partial q_s}\right)^{\top}\frac{\partial \mathcal{L}}{\partial z_{sj}} = \sum\nolimits_{j\le s}\frac{k_j}{\sqrt{d_{\text{head}}}}\left\langle g_s, \frac{\partial y_s}{\partial z_{sj}}\right\rangle.$$

Applying Lemma 2 with row s and column j gives $\frac{\partial y_s}{\partial z_{sj}} = a_{sj}(v_j - y_s)$, so we have

$$\frac{\partial \mathcal{L}}{\partial z_{sj}} = a_{sj}\langle g_s, v_j - y_s\rangle = a_{sj}\beta_{sj}.$$

This proves the general formula. For $s = 0$, causality implies $a_{00} = 1$ and $y_0 = v_0$. Thus we have $\beta_{00} = 0$ and $\nabla_{q_0}\mathcal{L} = 0$. □

D.2 Value-path theorem and deterministic bounds

We prove the main value-path theorem used in Section 4.

Fix the model parameters and condition on the attention matrix $A = (a_{tj})$, writing $\mathbb{E}_A[\cdot] = \mathbb{E}[\cdot \mid A]$. Then a_{ts}, M_s, and S_s are fixed, while the assumptions below describe the remaining variability of the upstream gradients.

Assumption 1 (Mean-plus-noise model)**.** For every position $t \ge s$, the upstream gradient admits the decomposition

$$g_t = \mu + \varepsilon_t \in \mathbb{R}^{d_{\text{head}}},$$

where $\mu = \mathbb{E}_A[g_t]$ is shared across these positions, so $\mathbb{E}_A[\varepsilon_t] = 0$. The shared conditional mean is a modeling assumption on the coherent component of the upstream gradients.

Assumption 2 (Bounded conditional noise moments)**.** There exist $\sigma^2, \rho \ge 0$ such that for all $t \ne t'$,

$$\mathbb{E}_A\|\varepsilon_t\|_2^2 \le \sigma^2, \quad \left|\mathbb{E}_A\langle \varepsilon_t, \varepsilon_{t'}\rangle\right| \le \rho.$$

Assumption 1 uses a mean-plus-noise description of stochastic gradients [22, 23], with a shared conditional mean across positions. Assumption 2 permits correlated noise; μ, σ^2, and ρ may depend on A. Under the zero-mean condition, its bounds equivalently constrain covariance traces:

$$\operatorname{Tr}\operatorname{Cov}_A(\varepsilon_t) = \mathbb{E}_A\|\varepsilon_t\|_2^2, \quad \operatorname{Tr}\operatorname{Cov}_A(\varepsilon_t, \varepsilon_{t'}) = \mathbb{E}_A\langle \varepsilon_t, \varepsilon_{t'}\rangle.$$

Here Cov_A denotes conditional covariance given A. Both covariance matrices are $d_{\text{head}} \times d_{\text{head}}$. Independence across tokens or coordinates is not required.

Proof of Theorem 1. By Proposition 1 and Assumption 1,

$$\nabla_{v_s}\mathcal{L} = \sum\nolimits_{t=s}^{T-1} a_{ts}(\mu + \varepsilon_t) = \left(\sum\nolimits_{t=s}^{T-1} a_{ts}\right)\mu + \sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t = M_s\mu + \sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t.$$

Therefore,

$$\begin{aligned}\mathbb{E}_A\left[\|\nabla_{v_s}\mathcal{L}\|_2^2\right] &= \mathbb{E}_A\left[\left\|M_s\mu + \sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t\right\|_2^2\right] \\ &= M_s^2\|\mu\|_2^2 + 2\,\mathbb{E}_A\left\langle M_s\mu, \sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t\right\rangle + \mathbb{E}_A\left\|\sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t\right\|_2^2.\end{aligned}$$

Under the conditioning, a_{ts}, M_s, and μ are fixed. Together with $\mathbb{E}_A[\varepsilon_t] = 0$, this makes the cross term vanish. Expanding the final term gives

$$\mathbb{E}_A\left\|\sum\nolimits_{t=s}^{T-1} a_{ts}\varepsilon_t\right\|_2^2 = \sum\nolimits_{t=s}^{T-1}\sum\nolimits_{t'=s}^{T-1} a_{ts}a_{t's}\,\mathbb{E}_A\langle \varepsilon_t, \varepsilon_{t'}\rangle.$$

Splitting diagonal and off-diagonal terms yields

$$\mathbb{E}_A\left[\|\nabla_{v_s}\mathcal{L}\|_2^2\right] = M_s^2\|\mu\|_2^2 + \sum_{t=s}^{T-1} a_{ts}^2\,\mathbb{E}_A\|\varepsilon_t\|_2^2 + \sum_{\substack{s\le t,t'\le T-1\\ t\neq t'}} a_{ts}a_{t's}\,\mathbb{E}_A\langle\varepsilon_t,\varepsilon_{t'}\rangle.$$

Finally, Assumption 2 implies

$$\sum_{t=s}^{T-1} a_{ts}^2\,\mathbb{E}_A\|\varepsilon_t\|_2^2 \le \sigma^2\sum_{t=s}^{T-1} a_{ts}^2 = \sigma^2 S_s,$$

and

$$\left|\sum_{\substack{s\le t,t'\le T-1\\ t\neq t'}} a_{ts}a_{t's}\,\mathbb{E}_A\langle\varepsilon_t,\varepsilon_{t'}\rangle\right| \le \rho\sum_{\substack{s\le t,t'\le T-1\\ t\neq t'}} a_{ts}a_{t's} = \rho(M_s^2 - S_s).$$

These estimates give the upper bound. The lower bound follows from the nonnegativity of $\mathbb{E}_A\|\sum_{t=s}^{T-1} a_{ts}\varepsilon_t\|_2^2$. □

Corollary 1 (Deterministic value-side upper bounds)**.** *For any realization of $\{g_t\}_{t=s}^{T-1}$, we have*

$$\|\nabla_{v_s}\mathcal{L}\|_2 \le \sum_{t=s}^{T-1} a_{ts}\|g_t\|_2 \le M_s\max_{t\ge s}\|g_t\|_2,$$

and also

$$\|\nabla_{v_s}\mathcal{L}\|_2 \le \sqrt{S_s}\left(\sum_{t=s}^{T-1}\|g_t\|_2^2\right)^{1/2}.$$

Proof. The first bound follows from Proposition 1 and the triangle inequality. The second follows from Cauchy-Schwarz:

$$\left\|\sum_{t=s}^{T-1} a_{ts}g_t\right\|_2 \le \left(\sum_{t=s}^{T-1} a_{ts}^2\right)^{1/2}\left(\sum_{t=s}^{T-1}\|g_t\|_2^2\right)^{1/2}.$$

□

D.3 Deterministic bounds for key and query paths

The main theorem concerns the value path. This subsection provides supporting bounds for the key and query pathways and clarifies the asymmetry.

Assumption 3 (Uniform boundedness for the key-side bound)**.** There exist constants $Q_{\max}, B_{\max} > 0$ such that for all t,

$$\|q_t\|_2 \le Q_{\max}, \quad |\beta_{ts}| \le B_{\max}.$$

Theorem 3 (Deterministic key-side upper bound)**.** *Under Assumption 3, we have*

$$\|\nabla_{k_s}\mathcal{L}\|_2 \le \frac{Q_{\max}B_{\max}}{\sqrt{d_{\text{head}}}}M_s.$$

Proof. Starting from Proposition 3 and applying the triangle inequality,

$$\|\nabla_{k_s}\mathcal{L}\|_2 = \left\|\sum_{t=s}^{T-1} a_{ts}\beta_{ts}\frac{q_t}{\sqrt{d_{\text{head}}}}\right\|_2 \le \sum_{t=s}^{T-1} a_{ts}|\beta_{ts}|\frac{\|q_t\|_2}{\sqrt{d_{\text{head}}}}.$$

Using $a_{ts}\ge 0$, $|\beta_{ts}|\le B_{\max}$, and $\|q_t\|_2\le Q_{\max}$ gives the result. □

Lemma 3 (Row-sum-zero of logits gradients)**.** *For a fixed row s, we have*

$$\sum_{j\le s}\frac{\partial\mathcal{L}}{\partial z_{sj}} = 0.$$

Proof. The derivation above gives $\frac{\partial\mathcal{L}}{\partial z_{sj}} = a_{sj}\langle g_s, v_j - y_s\rangle$. Therefore, we have

$$\begin{aligned}\sum_{j\le s}\frac{\partial\mathcal{L}}{\partial z_{sj}} &= \Big\langle g_s, \sum_{j\le s} a_{sj}(v_j - y_s)\Big\rangle\\ &= \Big\langle g_s, \sum_{j\le s} a_{sj}v_j - y_s\sum_{j\le s} a_{sj}\Big\rangle\\ &= \langle g_s, y_s - y_s\rangle = 0.\end{aligned}$$

□

Assumption 4 (Bounded advantage and bounded key dispersion)**.** Let

$$\bar{k}_s := \sum\nolimits_{j\le s} a_{sj} k_j.$$

There exist constants $B_{\max}^{(Q)}, \Delta K_{\max} > 0$ such that for all $j \le s$,

$$|\beta_{sj}| \le B_{\max}^{(Q)}, \quad \|k_j - \bar{k}_s\|_2 \le \Delta K_{\max}.$$

Theorem 4 (Deterministic query-side upper bound in dispersion form)**.** *Under Assumption 4,*

$$\|\nabla_{q_s}\mathcal{L}\|_2 \le \frac{B_{\max}^{(Q)} \Delta K_{\max}}{\sqrt{d_{\text{head}}}}.$$

Moreover, under causal attention, $\nabla_{q_0}\mathcal{L} = 0$ *exactly.*

Proof. Using Lemma 3, for any vector $\bar{k}_s$ we have

$$\sum\nolimits_{j\le s} \frac{\partial \mathcal{L}}{\partial z_{sj}} \bar{k}_s = \bar{k}_s \sum\nolimits_{j\le s} \frac{\partial \mathcal{L}}{\partial z_{sj}} = 0.$$

Hence the exact query-side gradient can be rewritten as

$$\nabla_{q_s}\mathcal{L} = \frac{1}{\sqrt{d_{\text{head}}}} \sum\nolimits_{j\le s} \frac{\partial \mathcal{L}}{\partial z_{sj}} (k_j - \bar{k}_s) = \frac{1}{\sqrt{d_{\text{head}}}} \sum\nolimits_{j\le s} a_{sj}\beta_{sj}(k_j - \bar{k}_s).$$

Taking norms and applying the triangle inequality yields

$$\|\nabla_{q_s}\mathcal{L}\|_2 \le \frac{1}{\sqrt{d_{\text{head}}}} \sum\nolimits_{j\le s} a_{sj}|\beta_{sj}| \, \|k_j - \bar{k}_s\|_2.$$

Using $\sum_{j\le s} a_{sj} = 1$, together with Assumption 4, proves the bound. The statement for $s = 0$ was already established in Proposition 4. □

D.4 RMSNorm as a gradient-compression mechanism

This subsection formalizes the RMSNorm mechanism used in Section 4 and measured empirically in Section 3.2.

For $x \in \mathbb{R}^d$, gain vector $\gamma \in \mathbb{R}^d$, and stabilizer $\epsilon_{\text{rms}} > 0$, define

$$\text{rms}(x) := \sqrt{\frac{1}{d}\sum\nolimits_{i=1}^{d} x_i^2 + \epsilon_{\text{rms}}}, \quad \text{RMSNorm}(x) := \gamma \odot \frac{x}{\text{rms}(x)}.$$

Denote $r := \text{rms}(x)$, $D := \text{diag}(\gamma)$, and $y := \text{RMSNorm}(x)$.

Proposition 5 (Exact Jacobian of RMSNorm)**.** *The Jacobian* $J_{\text{rms}}(x) := \partial y/\partial x \in \mathbb{R}^{d\times d}$ *is*

$$J_{\text{rms}}(x) = \frac{1}{r} D - \frac{1}{r^3 d} D x x^\top.$$

Proof. Since $y = D(x/r)$, we have

$$\frac{\partial y}{\partial x} = D\left(\frac{1}{r} I + x \frac{\partial}{\partial x}\left(\frac{1}{r}\right)\right).$$

Then $\frac{\partial}{\partial x}\left(\frac{1}{r}\right) = -\frac{1}{r^2}\frac{\partial r}{\partial x} = -\frac{1}{r^3 d} x^\top$ gives the stated formula. □

Now we prove Theorem 2 in Section 4.

Proof of Theorem 2. The chain rule gives $\nabla_x \mathcal{L} = J_{\text{rms}}(x)^\top \nabla_y \mathcal{L}$. Proposition 5 implies

$$J_{\text{rms}}(x) = \frac{1}{r} D\left(I - \frac{x x^\top}{d r^2}\right).$$

The matrix $I - \frac{x x^\top}{d r^2}$ has eigenvalues $\epsilon_{\text{rms}}/r^2$ and 1. Thus we have

$$\|J_{\text{rms}}(x)\|_{\text{op}} \le \frac{1}{r}\|D\|_{\text{op}} = \frac{\|\gamma\|_\infty}{r}.$$

□

Proposition 6 (Activation norm as a sufficient condition for bounded backpropagated gradients)**.** *Fix a target gradient scale $\tau > 0$. A sufficient condition for $\|\nabla_x \mathcal{L}\|_2 \le \tau$ is*

$$\mathrm{rms}(x) \ge \frac{\|\gamma\|_\infty}{\tau} \|\nabla_y \mathcal{L}\|_2.$$

Thus, for fixed τ and gain γ, the sufficient RMS level scales with the upstream gradient norm.

Proof. By Theorem 2, we have

$$\|\nabla_x \mathcal{L}\|_2 \le \frac{\|\gamma\|_\infty}{\mathrm{rms}(x)} \|\nabla_y \mathcal{L}\|_2.$$

To make the right-hand side at most τ, it suffices that $\mathrm{rms}(x) \ge \frac{\|\gamma\|_\infty}{\tau} \|\nabla_y \mathcal{L}\|_2$. □

Proposition 6 means that in a pre-norm Transformer, if a token receives unusually large gradient pressure, increasing its token-wise activation norm provides a direct channel for reducing the gradient magnitude passed to earlier layers. This is the mathematical sense in which massive activations can act as a gradient-compression response.

D.5 V-scale Jacobian and gradient attenuation

We first justify the initialization scale used in Section 5, then derive the V-scale Jacobian and its gradient attenuation properties. Recall that

$$C_{\ell,h} = d_{\mathrm{model}} d_{\mathrm{head}} \sigma^2 \exp(\theta_{\ell,h}), \qquad \sigma = 0.02, \quad \theta_{\ell,h} = 0 \text{ at initialization.}$$

RMSNorm with unit gain at initialization gives $\|\widetilde{h}\|_2^2 \approx d_{\mathrm{model}}$. For zero-mean independent entries of W_V with variance σ^2, the value state $v = W_V \widetilde{h}$ therefore satisfies

$$\mathbb{E}_{W_V}\left[\|v\|_2^2 \mid \widetilde{h} \right] = d_{\mathrm{head}} \sigma^2 \|\widetilde{h}\|_2^2 \approx d_{\mathrm{model}} d_{\mathrm{head}} \sigma^2.$$

Thus, $C_{\ell,h}$ starts at the typical squared value-state norm, while the learned factor $\exp(\theta_{\ell,h})$ allows this scale to adapt across layers and heads.

We now analyze the V-scale map used in Section 5. On the pre-aggregation value state v, define

$$\hat{v} = \phi(r) v, \quad r := \|v\|_2^2, \quad \phi(r) := \frac{r}{r + C}, \quad C > 0.$$

Proposition 7 (Exact Jacobian of V-scale)**.** *The Jacobian $J_\phi(v) := \partial \hat{v} / \partial v$ is*

$$J_\phi(v) = \phi(r) I + \frac{2C}{(r + C)^2} v v^\top. \tag{6}$$

Proof. Since $\hat{v} = \phi(r) v$ and $r = v^\top v$, we have

$$\frac{\partial \hat{v}}{\partial v} = \phi(r) I + v \left(\frac{\partial \phi(r)}{\partial v} \right)^\top.$$

Since $\phi'(r) = \frac{C}{(r+C)^2}$ and $\frac{\partial r}{\partial v} = 2v$, substituting

$$\frac{\partial \phi(r)}{\partial v} = \phi'(r) \frac{\partial r}{\partial v} = \frac{2C}{(r + C)^2} v$$

yields the claim. □

Proposition 8 (Jacobian spectrum of V-scale)**.** *Let $r = \|v\|_2^2$. Then the Jacobian in* (6) *has eigenvalue $\lambda_\perp(r) = \frac{r}{r+C}$ on the $(d_{\mathrm{head}} - 1)$-dimensional subspace orthogonal to v, and eigenvalue $\lambda_\parallel(r) = \frac{r^2 + 3Cr}{(r+C)^2}$ along the radial direction v itself. Moreover, $\max_{r \ge 0} \lambda_\parallel(r) = 9/8$, attained at $r = 3C$.*

Proof. Because $vv^\top$ is rank one, for any $u \perp v$ we have $vv^\top u = 0$, so Proposition 7 gives $J_\phi(v)u = \phi(r)u$. Along the radial direction,

$$J_\phi(v)v = \phi(r)v + \frac{2C}{(r+C)^2}vv^\top v = \left(\phi(r) + \frac{2Cr}{(r+C)^2}\right)v.$$

This proves the eigenvalue formulas.

For the maximum, define $f(r) := \lambda_\parallel(r) = \frac{r^2+3Cr}{(r+C)^2}$. Direct differentiation shows that $f'(r) = 0$ at $r = 3C$, and substituting this value gives $f(3C) = 9/8$. □

Corollary 2 (V-scale is a value-path gradient valve)**.** *Let $g_{\hat{v}} = \nabla_{\hat{v}}\mathcal{L}$ be the upstream gradient at the scaled value state. Then we have $\nabla_v\mathcal{L} = J_\phi(v)^\top g_{\hat{v}} = J_\phi(v)g_{\hat{v}}$, and therefore $\|\nabla_v\mathcal{L}\|_2 \le \lambda_\parallel(\|v\|_2^2)\|g_{\hat{v}}\|_2$. In particular, if $\|v\|_2^2 \ll C$, then the value-path backward signal is strongly attenuated as*

$$\|\nabla_v\mathcal{L}\|_2 \lesssim \frac{3\|v\|_2^2}{C}\,\|g_{\hat{v}}\|_2.$$

Proof. The first identity is the chain rule. Since $J_\phi(v)$ is symmetric, its operator norm equals its largest eigenvalue, which is $\lambda_\parallel(r)$ by Proposition 8. The small-r expansion follows directly from the expression for $\lambda_\parallel(r)$. □

These results show that V-scale introduces an additional value-path gradient valve. In the regime where sink value states have relatively small norm, the attenuation factor in Corollary 2 can be substantially below one, which is precisely the regime relevant to the observed intervention effect.

E Licenses for existing assets

Our experiments rely on several publicly available datasets, models, tokenizers, and evaluation or compression toolkits. We acknowledge the contributors and respect their licenses. The main existing assets are summarized in Table 10.

Table 10: Main existing assets used in this work.

Asset	Usage in this work	License
C4	Pretraining data	ODC-BY
GPT-2 tokenizer	Tokenization	MIT License
Qwen2.5-7B	Diagnostic measurements	Apache License 2.0
Qwen3-8B	Diagnostic measurements	Apache License 2.0
Llama-3.1-8B	Diagnostic measurements	Llama 3.1 Community License
GLM-4-9B	Diagnostic measurements	GLM-4-9B License
LM Evaluation Harness v0.4.11	Evaluation	MIT License
Bitsandbytes v0.49.0	Quantization	MIT License
LLM Compressor v0.9.0	Quantization	Apache License 2.0

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly describe our central claim.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated "Limitations" paragraph in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix D provides the assumptions, supporting propositions, and full proofs for the theoretical results used in Sections 4–5.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 2, 3, and 5 define the main observables, intervention, and evaluation setting, and Appendix B gives the detailed training hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a public code repository link in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Experimental settings are given in Sections 2, 3, and 5. Detailed configurations are listed in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not repeat full pre-training runs because of their computational cost. However, our Needle-in-a-Haystack evaluations report means and standard deviations over 5 evaluation seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B specifies that runs use NVIDIA A100 GPUs with 80GB memory and reports token budgets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and do not foresee any ethical concerns in this work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This work is a foundational mechanistic study and thus does not have direct societal impact. Section 6 discusses possible technical influence on future Transformer design, but the paper does not separately analyze positive and negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release new pretrained model weights, scraped datasets, or other assets that would create a new high-risk access pathway.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We discuss this in Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing, user studies, or other research with human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development does not involve LLMs as any important, original, or non-standard components. LLMs are primarily used to help us polish paper writeups.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.