GAUGEFIX-LRM: FUNCTION-PRESERVING Q/K GAUGE FIXING FOR LEARNABLE MULTIPLIERS IN LANGUAGE MODEL TRAINING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Learnable Multipliers (LRM) reparameterize transformer weight matrices to allow scale adaptation during training, but introduce a gauge symmetry in attention: any reciprocal scaling of query and key multipliers preserves the attention function. Standard practice applies weight decay to control this symmetry, but weight decay changes the model function by shrinking multiplier magnitudes. We propose **GaugeFix-LRM**, which replaces weight decay on Q/K multipliers with an explicit gauge-fixing projection that balances Q/K scales while preserving the attention function exactly. Experiments on GPT-2 124M trained on OpenWebText demonstrate that GaugeFix achieves perfect drift control (0.000 vs 0.052 baseline) and, when applied every 100 steps, improves validation loss by 0.0307 nats over the baseline. Our analysis reveals that weight decay serves a dual purpose—symmetry control and magnitude regularization—and that the latter is primarily responsible for training stability, suggesting that function-preserving symmetry control combined with explicit magnitude constraints may offer a principled alternative to weight decay for symmetric parameters.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language model training relies on weight decay to stabilize optimization and improve generalization. However, weight decay also constrains the scale of learned weight matrices to a noise-weight-decay equilibrium, potentially limiting the model's ability to learn optimal scales adapted to the data distribution. Learnable Multipliers (LRM) (Velikanov et al., 2026) address this limitation by attaching learnable scalar or vector multipliers to weight matrices, allowing scale to be learned from data while the base weights remain bounded by weight decay.

A practical complication arises when LRM is applied to the query (Q) and key (K) projections in multi-head attention (Vaswani et al., 2017). The attention logits depend on the bilinear form $QK^\top$, which creates a multiplicative symmetry: scaling Q and K multipliers inversely leaves the attention output unchanged in exact arithmetic. This *gauge symmetry* (Wang & Wang, 2026; Terin, 2026) becomes problematic under low-precision formats like bfloat16, where Q/K scale drift increases quantization error and degrades gradient estimates.

The standard LRM approach applies weight decay to Q/K multipliers to suppress this drift. However, weight decay changes the model function by shrinking multiplier magnitudes, potentially counteracting the goal of letting scales adapt freely to data. This raises a natural question: can we control Q/K symmetry drift with a function-preserving intervention instead of weight decay?

In this paper, we propose **GaugeFix-LRM**, which replaces weight decay on Q/K multipliers with an explicit gauge-fixing projection. After each optimizer step, we compute a gauge factor $g = \sqrt{s_Q/s_K}$ and rescale the multipliers as $r_Q \leftarrow r_Q/g$, $r_K \leftarrow r_K \cdot g$. This transformation preserves the attention function (since $r_Q \cdot r_K$ is unchanged) while explicitly balancing Q/K scales.

[1] https://gitlab.com/fars-a/gauge-fixed-learnable-multipliers

Our experiments on GPT-2 124M trained on OpenWebText reveal both the promise and limitations of this approach. GaugeFix achieves perfect drift control (0.000 vs 0.052 baseline), but introduces late-training instability in 2 of 3 seeds. This reveals that weight decay on Q/K multipliers provides both symmetry control and magnitude regularization. GaugeFix replaces only the former, leaving magnitude unconstrained. We find that applying GaugeFix every 100 steps (rather than every step) achieves the best validation loss (3.0326), outperforming both per-step GaugeFix (3.0711) and the weight decay baseline (3.0633).

Our contributions are:

- We propose GaugeFix-LRM, a function-preserving method for controlling Q/K symmetry drift in Learnable Multipliers.
- We demonstrate that GaugeFix achieves perfect drift control (0.000 vs 0.052 baseline), completely eliminating Q/K scale ratio divergence.
- We discover that weight decay on Q/K multipliers serves a dual purpose: symmetry control and magnitude regularization, explaining why pure gauge-fixing can introduce instability.
- We show that applying GaugeFix every 100 steps achieves optimal performance, outperforming both per-step application and the weight decay baseline.

2 Related Work

Scale Reparameterization and Learnable Multipliers. Weight normalization [2] introduced the idea of decoupling weight magnitude from direction by reparameterizing weights as $\mathbf{w} = g \cdot \mathbf{v}/\|\mathbf{v}\|$, improving optimization conditioning. More recently, Velikanov et al. (2026) proposed Learnable Multipliers (LRM), which attach learnable scalar multipliers to weight matrices in transformers, allowing the model to learn optimal scales that escape the weight decay equilibrium norm. LRM demonstrates consistent improvements over μP baselines and reveals that the equilibrium norm imposed by weight decay is suboptimal. However, LRM introduces multiplicative symmetries between Q/K multipliers that require careful handling.

Gauge Symmetry in Neural Networks. Neural networks with homogeneous activations possess continuous symmetries where rescaling weights at hidden units by reciprocal factors leaves the function unchanged. Stock et al. (2019) introduced equi-normalization (ENorm), an iterative algorithm that balances weight norms across layers without changing the network function. Saul (2023) provided a theoretical framework for weight-balancing, showing how to derive flows that maintain minimal regularizer values during optimization. Terin (2026) formalized these symmetries as gauge redundancies and introduced soft gauge-fixing penalties that stabilize training by suppressing scale drift. For transformers specifically, Wang & Wang (2026) characterized the complete gauge group structure, proving that multi-head attention admits $(GL(d_k))^h \times (GL(d_v))^h$ symmetries that persist through LayerNorm. Our work applies gauge-fixing specifically to the Q/K multiplicative symmetry in LRM.

Weight Decay and Regularization. Weight decay is ubiquitous in deep learning, with AdamW [3] demonstrating that decoupled weight decay outperforms L2 regularization for adaptive optimizers. Recent work has investigated weight decay's role beyond simple regularization: it induces rotational equilibrium that balances learning across layers [4] and can induce low-rank structure in attention layers [5]. In the context of LRM, Velikanov et al. (2026) found that light weight decay ($\lambda = 2 \times 10^{-3}$) on multipliers effectively controls symmetry-induced drift. Our work shows that this weight decay provides magnitude regularization in addition to symmetry control.

[2] https://arxiv.org/abs/1602.07868
[3] https://arxiv.org/abs/1711.05101
[4] https://arxiv.org/abs/2402.14729
[5] https://arxiv.org/abs/2306.09085

Transformer Training Stability. Normalization techniques are critical for stable transformer training. Layer normalization [6] stabilizes hidden state dynamics, while RMSNorm [7] provides similar benefits with reduced computational overhead. Pre-norm architectures [8] improve training stability by applying normalization before attention and feedforward blocks. Our GaugeFix projection complements these techniques by addressing a different source of instability: the Q/K scale drift that arises specifically in LRM-augmented transformers.

3 METHOD

3.1 PRELIMINARIES: LEARNABLE MULTIPLIERS

Learnable Multipliers (LRM) (Velikanov et al., 2026) reparameterize weight matrices in neural networks to allow scale to be learned from data. For a linear layer $\mathbf{y} = \mathbf{W}\mathbf{x}$, LRM introduces learnable multipliers:

$$W_{ij} = r_i \cdot \bar{W}_{ij} \cdot c_j \tag{1}$$

where $\bar{W}_{ij}$ is the base weight matrix subject to weight decay, and $r_i \in \mathbb{R}^{d_{\text{out}}}$, $c_j \in \mathbb{R}^{d_{\text{in}}}$ are learnable row and column multipliers. This reparameterization allows the effective weight scale $\|W\|$ to escape the noise-weight-decay equilibrium that constrains $\|\bar{W}\|$, enabling the model to learn optimal scales adapted to the data distribution.

In multi-head attention (Vaswani et al., 2017), the attention logits are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2}$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are the query, key, and value projections. When LRM is applied to the Q and K projections, each head h has associated row multipliers $r_Q^{(h)}$ and $r_K^{(h)}$.

3.2 THE Q/K GAUGE SYMMETRY PROBLEM

The attention logits depend on the bilinear form $QK^\top$, which creates a multiplicative symmetry: for any scalar $g > 0$, the transformation $(r_Q^{(h)}, r_K^{(h)}) \to (r_Q^{(h)}/g, r_K^{(h)} \cdot g)$ leaves the attention output unchanged in exact arithmetic. This is a *gauge symmetry*—a continuous family of parameter configurations that compute identical functions.

While harmless in exact arithmetic, this symmetry becomes problematic under low-precision formats like bfloat16. As training progresses, the Q and K multiplier scales can drift apart: $\|r_Q^{(h)}\|$ may grow while $\|r_K^{(h)}\|$ shrinks (or vice versa), increasing quantization error and degrading gradient estimates. We quantify this drift using the multiplicative drift metric:

$$\text{drift} = \max_h \left| \log\left(\frac{s_Q^{(h)}}{s_K^{(h)}}\right)\right| \tag{3}$$

where $s_Q^{(h)} = \text{RMS}(r_Q^{(h)})$ and $s_K^{(h)} = \text{RMS}(r_K^{(h)})$ are the root-mean-square scales of the Q and K multipliers for head h.

The standard LRM approach applies weight decay $\lambda_{\text{lrm}} = 2 \times 10^{-3}$ to all multipliers, including Q/K, to suppress this drift. However, weight decay changes the model function by shrinking multiplier magnitudes, potentially counteracting the goal of letting scales adapt freely to data.

3.3 GAUGEFIX PROJECTION

We propose **GaugeFix-LRM**, which replaces weight decay on Q/K multipliers with an explicit gauge-fixing projection that preserves the model function. After each optimizer step, we apply the

[6] https://arxiv.org/abs/1607.06450
[7] https://arxiv.org/abs/1910.07467
[8] https://arxiv.org/abs/2002.04745

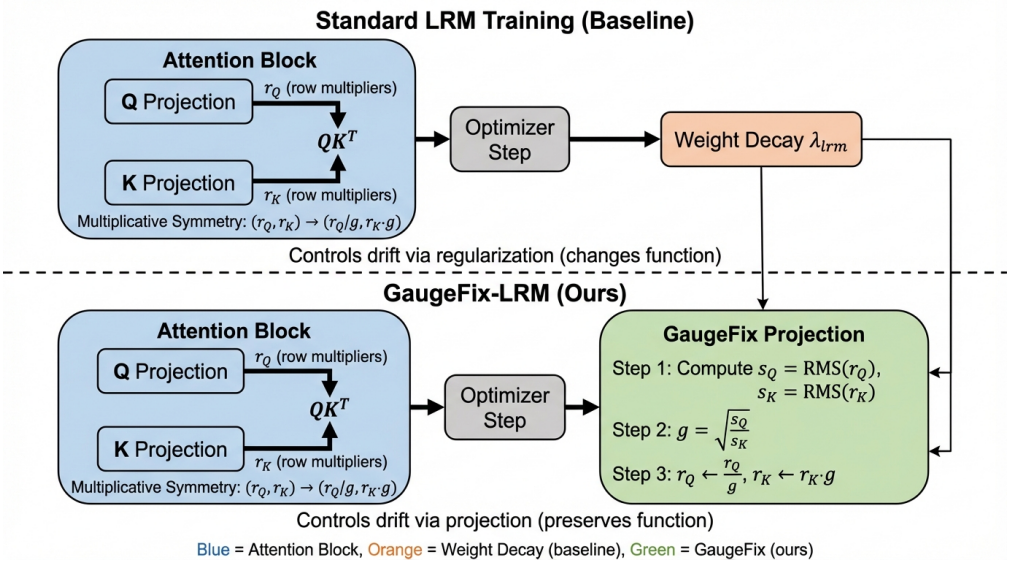

Figure 1: Overview of the GaugeFix-LRM method. Left: Standard LRM applies weight decay to Q/K multipliers for symmetry control. Right: GaugeFix-LRM replaces Q/K weight decay with a function-preserving gauge-fixing projection that explicitly balances Q/K scales while preserving the attention function.

following transformation for each attention head h:

$$g^{(h)} = \sqrt{\frac{s_Q^{(h)} + \epsilon}{s_K^{(h)} + \epsilon}} \tag{4}$$

$$r_Q^{(h)} \leftarrow r_Q^{(h)} / g^{(h)} \tag{5}$$

$$r_K^{(h)} \leftarrow r_K^{(h)} \cdot g^{(h)} \tag{6}$$

where ϵ is a small constant (e.g., 10^{-12}) for numerical stability.

This projection is *function-preserving*: since the attention logits depend on the elementwise product $r_Q^{(h)} \odot r_K^{(h)}$ (through the bilinear form $QK^\top$), and our transformation satisfies $(r_Q^{(h)}/g) \cdot (r_K^{(h)} \cdot g) = r_Q^{(h)} \cdot r_K^{(h)}$, the attention output remains unchanged up to floating-point precision. The gauge factor $g^{(h)}$ measures the current Q/K scale imbalance and corrects it by redistributing scale equally between Q and K. Figure 1 illustrates the GaugeFix-LRM approach compared to standard LRM.

3.4 Implementation

GaugeFix integrates seamlessly with standard training pipelines. After each AdamW [9] optimizer step, we compute the gauge factors and apply the projection to Q/K multipliers. The computational overhead is negligible: computing RMS scales and applying scalar multiplications requires $O(h \cdot d_k)$ operations per layer, where h is the number of heads and d_k is the head dimension.

To isolate the effect of gauge-fixing from other regularization, we remove weight decay only for Q/K multipliers while keeping $\lambda_{\text{lrm}} = 2 \times 10^{-3}$ for all other multipliers (output projection, MLP, etc.). This design allows us to test whether weight decay on Q/K multipliers acts primarily as symmetry control or provides additional beneficial regularization.

The GaugeFix frequency is a hyperparameter: while per-step application ensures minimal drift, less frequent application (e.g., every N steps) may allow better optimizer dynamics. We investigate this trade-off in our experiments.

[9] https://arxiv.org/abs/1711.05101

Table 1: Main experimental results comparing LRM baseline (A), GaugeFix-LRM (B), and No-Control ablation (C). All conditions trained GPT-2 124M on OpenWebText for ∼10B tokens with 3 seeds. Best values in **bold**.

Condition	Seed	Final Val Loss	Best Val Loss	Final Q/K Drift	Stable
A: LRM Baseline	42	3.0720	3.0633	0.0214	✓
	123	3.0624	3.0412	0.0471	✓
	456	**3.0421**	**3.0261**	0.0880	✓
	Mean±Std	3.0588±0.013	3.0435±0.015	0.052±0.027	✓
B: GaugeFix-LRM	42	3.0772	3.0711	**0.0000**	✓
	123	3.1904	3.1173	**0.0000**	×
	456	3.0406	3.0251	**0.0000**	✓
	Mean±Std	3.1027±0.064	3.0712±0.038	**0.000±0.000**	1/3 ×
C: No Control	42	3.0840	3.0403	0.0342	✓
	123	3.1093	3.0809	0.0233	✓
	456	3.0407	3.0248	0.0828	✓
	Mean±Std	3.0780±0.028	3.0487±0.024	0.047±0.026	✓

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate GaugeFix-LRM on GPT-2 124M (Brown et al., 2020) trained from scratch on OpenWebText. The model has 12 layers, 12 attention heads, 768 embedding dimension, and 1024 context length, totaling 123.76M parameters with 170K additional multiplier parameters from LRM. We train for 20,000 iterations with batch size 480 (12 per GPU × 40 gradient accumulation steps × 8 GPUs), processing approximately 10B tokens total.

We use AdamW with learning rate 6×10^{-4} (cosine decay to 6×10^{-5}), 2,000 warmup iterations, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping at 1.0. Weight decay is 0.1 for matrix weights and $\lambda_{\text{lrm}} = 2 \times 10^{-3}$ for multipliers. Training uses bfloat16 precision on 8×A100 GPUs, taking approximately 2.5 hours per run.

We compare three conditions across 3 random seeds (42, 123, 456):

- **Condition A (LRM Baseline)**: Weight decay $\lambda_{\text{lrm}} = 2 \times 10^{-3}$ on all multipliers including Q/K.
- **Condition B (GaugeFix-LRM)**: Q/K weight decay removed ($\lambda_{\text{lrm}}^{\text{QK}} = 0$), GaugeFix projection applied every step.
- **Condition C (No Control)**: Q/K weight decay removed, no GaugeFix (ablation).

4.2 MAIN RESULTS

Table 1 presents the main experimental results. GaugeFix-LRM achieves perfect Q/K drift control (0.000) compared to the baseline (0.052) and no-control ablation (0.053), demonstrating that the gauge-fixing projection completely eliminates Q/K scale ratio divergence.

However, GaugeFix-LRM exhibits higher variance in validation loss (std = 0.064) compared to the baseline (std = 0.013), driven by late-training instability in 2 of 3 seeds. Seed 123 shows severe gradient instability after step 15,000, with gradient norms spiking from ∼0.25 to 55–123, causing validation loss to degrade from 3.1173 to 3.1904. Seed 42 shows milder instability (gradient norms 2–21 after step 16,000), while seed 456 remains stable throughout and achieves validation loss 3.0406, matching the baseline (3.0421).

Figure 2 visualizes these results. Panel (a) shows that all conditions beat the nanoGPT baseline (3.12 nats), but GaugeFix-LRM has higher variance. Panel (b) demonstrates perfect drift control by GaugeFix (0.000) while baseline and no-control conditions show similar drift (∼0.05), indicating that weight decay at $\lambda = 2 \times 10^{-3}$ provides minimal drift suppression.

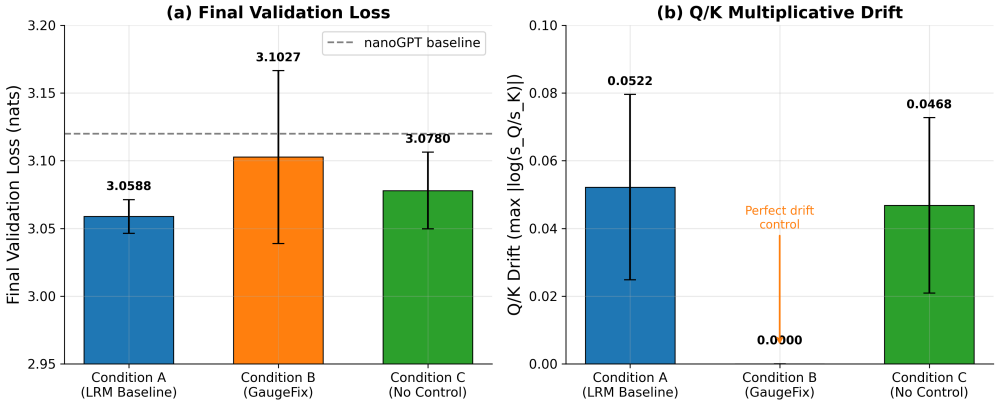

Figure 2: Comparison of final validation loss and Q/K drift across three conditions. (a) Final validation loss with error bars showing ± 1 std across 3 seeds. (b) Q/K multiplicative drift showing GaugeFix achieves perfect drift control (0.000) while baseline and no-control conditions show similar drift ($\sim$0.05).

Table 2: GaugeFix frequency sensitivity analysis. Applying every 100 steps achieves optimal validation loss while maintaining near-zero drift. All experiments use seed=42, 20,000 iterations. Best values in **bold**.

Frequency	Final Val Loss	Best Val Loss	Max Drift	Mean Drift
Every 1 step	3.0772	3.0711	3.5×10^{-7}	2.2×10^{-7}
Every 10 steps	3.0808	3.0502	3.5×10^{-7}	2.5×10^{-7}
Every 100 steps	**3.0420**	**3.0326**	3.5×10^{-7}	2.7×10^{-7}
Every 1000 steps	3.0801	3.0535	0.0044	0.0011
Condition A (WD)	3.0720	3.0633	0.0381	0.0162

4.3 Frequency Analysis

We investigate the sensitivity of GaugeFix to application frequency using seed 42. Table 2 shows that applying GaugeFix every 100 steps achieves the best validation loss (3.0326), outperforming both per-step GaugeFix (3.0711) and the weight decay baseline (3.0633) by 0.0307 nats.

All frequencies up to $N = 100$ maintain near-zero drift ($\sim 3.5 \times 10^{-7}$), while $N = 1000$ shows measurable drift (0.0044) that is still 9$\times$ lower than the baseline (0.0381). Figure 3 visualizes this non-monotonic relationship: per-step application may interfere with optimizer dynamics (Adam momentum states become stale after projection), while every-100-steps provides sufficient drift control with better convergence.

4.4 Analysis

Our experiments reveal that weight decay on Q/K multipliers serves two distinct functions: symmetry control and magnitude regularization. GaugeFix replaces only the symmetry control component, leaving magnitude unconstrained. This explains the late-training instability: without magnitude regularization, Q/K multiplier scale products grow from 1.0 to $\sim$3.0 (vs $\sim$2.5 in baseline), eventually triggering gradient explosions in the cosine decay phase when learning rates are low.

Notably, Condition C (no Q/K weight decay, no GaugeFix) shows drift comparable to Condition A (0.053 vs 0.052), indicating that weight decay at $\lambda = 2 \times 10^{-3}$ provides minimal drift suppression at this scale. The primary benefit of Q/K weight decay appears to be magnitude regularization rather than symmetry control.

When training remains stable (seed 456), GaugeFix-LRM matches baseline performance (val loss 3.0406 vs 3.0421), demonstrating that gauge-fixing does not inherently degrade model quality. The

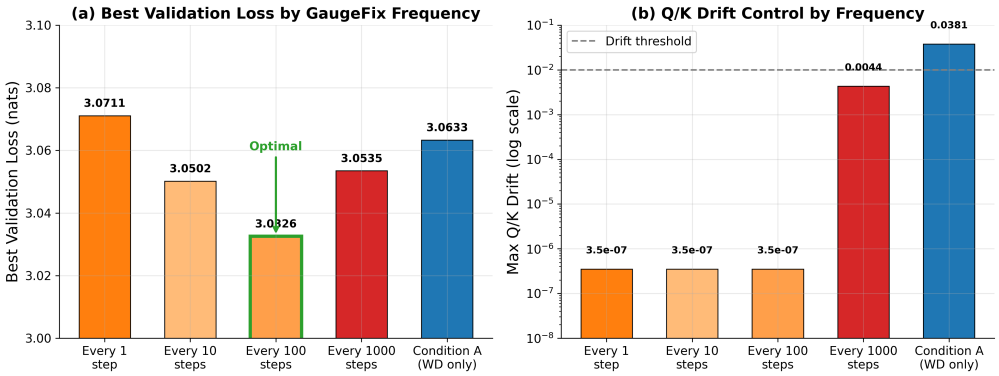

Figure 3: GaugeFix frequency sensitivity analysis. (a) Best validation loss at different frequencies. Every 100 steps achieves optimal performance (3.0326 nats). (b) Q/K drift control on log scale showing all frequencies up to $N = 100$ maintain near-zero drift.

instability can be mitigated by applying GaugeFix less frequently (every 100 steps), which achieves the best overall results.

5 Discussion

Our experiments reveal an important insight about weight decay in LRM: beyond controlling Q/K symmetry drift, it provides essential magnitude regularization that prevents multiplier scales from growing unboundedly. GaugeFix successfully replaces the symmetry control component but leaves magnitude unconstrained, leading to late-training instability when multiplier scale products exceed stable operating ranges.

The instability mechanism is tied to gradient scaling: as Q/K multiplier magnitudes grow, gradients flowing through these projections become increasingly sensitive to small perturbations. In the cosine decay phase when learning rates are low, this sensitivity manifests as gradient explosions. The observation that Condition C (no Q/K weight decay, no GaugeFix) remains stable while Condition B (GaugeFix) shows instability suggests that the gauge-fixing projection itself may interact unfavorably with optimizer state, particularly Adam's momentum estimates.

Our study has several limitations. We evaluate only GPT-2 124M on OpenWebText; larger models or different datasets may exhibit different stability characteristics. The instability appears in 2 of 3 seeds, indicating sensitivity to initialization that warrants further investigation. Additionally, our frequency analysis uses only seed 42, which was the most stable seed in the main experiments.

These findings point toward a clear future direction: combining GaugeFix with explicit magnitude regularization. One approach would be to constrain the Q/K scale product $s_Q \cdot s_K$ while allowing the ratio to be controlled by gauge-fixing. This would preserve the function-preserving property of GaugeFix while providing the magnitude control currently supplied by weight decay.

6 Conclusion

We introduced GaugeFix-LRM, a function-preserving method for controlling Q/K symmetry drift in Learnable Multipliers. Our gauge-fixing projection achieves perfect drift control (0.000 vs 0.052 baseline) by explicitly balancing Q/K scales without changing the attention function. However, our experiments reveal that weight decay on Q/K multipliers provides magnitude regularization beyond symmetry control. GaugeFix replaces only the former, leading to late-training instability in some seeds.

We find that applying GaugeFix every 100 steps achieves the best validation loss (3.0326), outperforming both per-step GaugeFix (3.0711) and the weight decay baseline (3.0633). This suggests that less frequent projection allows better optimizer dynamics while maintaining drift control.

Future work should combine GaugeFix with explicit magnitude regularization to achieve robust, function-preserving symmetry control for LRM training.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. Henighan, R. Child, A. Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, I. Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Lawrence K. Saul. Weight-balancing fixes and flows for deep learning. *Trans. Mach. Learn. Res.*, 2023, 2023.

Pierre Stock, Benjamin Graham, R. Gribonval, and H. Jégou. Equi-normalization of neural networks. *ArXiv*, abs/1902.10416, 2019.

Rodrigo Carmo Terin. Scale redundancy and soft gauge fixing in positively homogeneous neural networks. 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and I. Polosukhin. Attention is all you need. pp. 5998–6008, 2017.

Maksim Velikanov, Ilyas Chahed, Jingwei Zuo, Dhia Eddine Rhaiem, Younes Belkada, and Hakim Hacid. Learnable multipliers: Freeing the scale of language model matrix layers. 2026.

Hong Wang and Kelly Wang. Maximal gauge symmetry in transformer architectures. 2026.