The Embedding Geometry Hypothesis: From Fourier Circuits to No-Q Attention

Nathan Rigoni
Phronesis Analytics
nathan.rigoni@phronesis-analytics.com

March 27, 2026

Abstract

The token embedding layer is the geometric foundation of transformer attention. We develop this claim through four stages. First, we show that prescribing near-Nyquist frequency modes in the embedding gradient — **Prescribed Fourier Frequency Training (PFFT),** achieves a 92.7% reduction in epochs-to-grokking (57 vs. 782) on modular arithmetic, with a 97.9% reduction in the memorization phase. PFFT works by simultaneously preserving the embedding's geometric authority and reducing gradient noise. Second, the **Sounding Hammer** diagnostic reveals that gradient-domain Fourier steering cannot safely transfer to language model embeddings: BPE vocabulary gradients are spectrally flat (ρ=0.42), causing catastrophic BPC regression (2.90→9.47) when applied. We introduce **Natural Ordering Conditions (NOC)** to characterize when Fourier steering is safe. Third, **Fourier Gradient Projection (FGP)**, a dynamic variant of PFFT that follows whichever frequency modes become important during training rather than locking onto a prescribed set. This is introduced as a general gradient-domain tool, though it shares the NOC limitation. Fourth, behavioral weight trajectory analysis of language models trained on TinyStories and FineWeb reveals that all weight matrices — Q, K, V, and MLP — inherit the same two-arm trajectory shape from the token embedding through the residual stream. This universal inheritance motivates **No-Q attention**: setting **Q=x** (no projection) at every layer. Without W_Q, the embedding's geometry reaches the attention mechanism directly, and the competitive gradient interference between W_Q and the embedding is eliminated. No-Q attention improves validation BPC by 3.18% on TinyStories and 2.24% on FineWeb while using 8% fewer parameters, and accelerates grokking by 51.0% on modular arithmetic. The token embedding is not a lookup table that feeds into attention — it *is* the attention query.

1 Introduction

The transformer [13] computes self-attention via three symmetric projections:

$$\text{Attn}(\mathbf{x}) = \text{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{d_h}}\right)\mathbf{V}, \quad \mathbf{Q} = \mathbf{x}\mathbf{W}_Q^\top, \ \mathbf{K} = \mathbf{x}\mathbf{W}_K^\top, \ \mathbf{V} = \mathbf{x}\mathbf{W}_V^\top. \tag{1}$$

The matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathrm{R}^{d \times d}$ are treated symmetrically in most analyses and architectures. We question this symmetry for $\mathbf{W}_Q$.

The investigation begins with grokking [10] — the phenomenon in which a small transformer suddenly generalizes after hundreds of epochs of near-perfect training accuracy. The mechanistic interpretability literature [8] has established that generalization coincides with the emergence of a sparse Fourier circuit *in the token embedding*, not in the attention weights. The attention layers then *exploit* the geometric structure that the embedding has established. This suggests a hierarchy: the embedding sets the representational geometry; downstream layers are secondary processors of that geometry. If true, the Q projection — applied to the post-embedding hidden state — is reparametrizing a space the embedding has already structured. It is a redundant degree of freedom that competes with, rather than extends, the embedding's geometric authority.

Research arc. We develop this idea through four stages:

1. **Modular arithmetic and PFFT**. We show that prescribing near-Nyquist Fourier modes in the embedding gradient achieves a 92.7% grokking speedup, confirming that the embedding is the geometric root of generalization.
2. **Why PFFT fails on language models**. The Sounding Hammer diagnostic reveals that gradient domain Fourier steering is not safe to apply to BPE vocabulary embeddings ($\rho = 0.42$) or character-level byte embeddings. A different mechanism is needed for language. Introduction of NOC.
3. **Behavioral diagnosis**. Weight trajectory analysis of language models trained on TinyStories and FineWeb reveals that Q-weight matrices — not K, V, or MLP — are the primary site of representational reorganization, pointing directly at the Q projection as the pathological component.
4. **No-Q attention**. Removing $\mathbf{W}_Q$ entirely ($\mathbf{Q} = \mathbf{x}$) achieves the same two goals as PFFT — respecting the embedding's geometric authority and reducing parameter noise — but through an architectural change that is safe for language models. Result: +3.18% BPC on TinyStories, +2.24% on FineWeb, 8% fewer parameters, and 58.9% ETG speedup on modular arithmetic.

2 The Embedding as Geometric Foundation

2.1 Grokking and Fourier circuits

Power et al. [10] observed that small transformers trained on modular arithmetic $(a+b) \bmod p$ exhibit delayed generalization: training accuracy reaches ≈100% hundreds of epochs before validation accuracy does. Nanda et al. [8] identified the mechanism: generalization coincides with the emergence of a Fourier circuit in the token embedding using modes $\{1,14,41\}$ for p=97. The attention weights do not carry this structure independently — they inherit it from the embedding geometry.

This is strong evidence for the Embedding Geometry Hypothesis: the token embedding is the primary carrier of the representational geometry that determines what downstream attention layers can efficiently compute.

2.2 Prescribed Fourier Frequency Training (PFFT)

We ask: if the embedding is the root, can we accelerate grokking by steering embedding gradients toward Fourier modes that carry more useful structure?

We introduce Prescribed Fourier Frequency Training (PFFT). After each backward pass, we project the token embedding gradient onto a prescribed frequency set S:

$$\nabla\mathbf{E} \leftarrow \mathrm{irfft}(\mathrm{rfft}(\nabla\mathbf{E}, \mathrm{dim}{=}0) \odot M_S, n{=}p, \mathrm{dim}{=}0), \tag{2}$$

where MS[k] = 1[k ∈ S]. PFFT incurs zero inference cost and negligible training overhead.

Fourier Gradient Projection (FGP). A dynamic variant of PFFT in which the prescribed frequency set S is not fixed at the start of training but is recomputed at each step (or each N steps) from the current gradient's power spectrum. At each update the top-K frequency bins by gradient power are selected and the gradient is projected onto those bins. FGP adapts to whichever frequency modes carry the most gradient signal at each stage of training, rather than locking onto a prescribed set. This makes it applicable without prior knowledge of the task-relevant Fourier structure. Like PFFT, FGP is subject to the NOC: it is safe only when the parameter axis satisfies natural ordering. FGP and PFFT share the same failure mode on BPE token embeddings.

2.3 Results: 92.7% grokking speedup

Table 1 presents the key results. Near-Nyquist modes $\{30,35,40,45,48\}$ achieve Epochs to Grok (ETG) = 57 (92.7% speedup) and reduce the memorization phase from 451 to 9 epochs (97.9% reduction). ETG is calculated as the number of epochs needed to reach >95% accuracy on the validation set. Three findings are especially diagnostic:

Table 1: Grokking acceleration on $(a+b) \bmod 97$ (mean ± std, 3 seeds). Mem = memorization epoch (train acc $\geq 99\%$).

Method	Prescribed modes S	ETG	Mem	Speedup
Baseline	—	782±95	451	—
PFFT Nanda	$\{1,14,41\}$	191±5	32	+75.5%
PFFT quint	$\{1,2,3,4,5\}$	261±3	90	+66.7%
PFFT near-Nyquist	$\mathbf{\{30,35,40,45,48\}}$	**57±4**	9	**+92.7%**
Adaptive K=5	top-5 dynamic	97±7	38	+87.6%

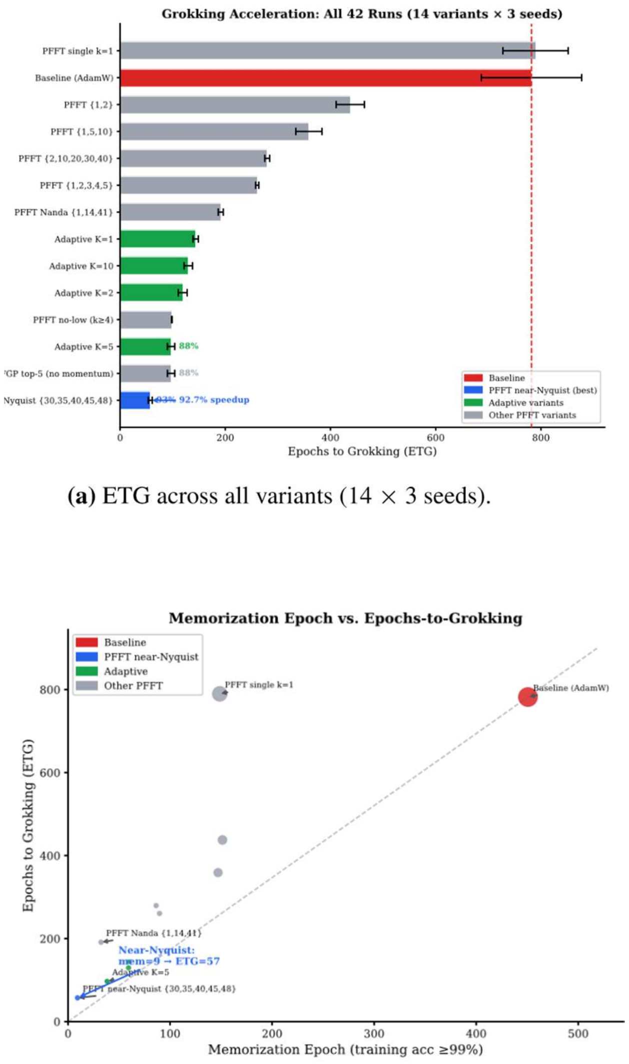
Near-Nyquist beats task-correct modes. The mechanistic interpretability literature identifies $\{1,14,41\}$ as the “correct” modes for p=97. Prescribing these yields 75.5% speedup; near-Nyquist $\{30,35,40,45,48\}$ yields 92.7%. The speedup comes from gradient noise reduction, not mode guidance. Constraining the gradient to a small frequency subspace eliminates the high-frequency noise component that causes each step to partially undo the previous one. The task-relevant Fourier structure then emerges naturally via the optimization landscape — PFFT does not need to prescribe the “correct” modes because the network finds them itself once the noise is removed.

Memorization is almost entirely bypassed. Standard training spends 451 epochs memorizing before generalizing. PFFT near-Nyquist stops memorization at epoch 9 and groks at epoch 57: the model generalizes before memorizing in any meaningful sense. Memorization is not a necessary stage — it is a symptom of gradient noise.

The cross-p validation (Table 2) confirms that near-Nyquist modes scale with the Nyquist limit of the input space, not with task-specific structure.

Table 2: Cross-p validation. “DNF” = did not grok within 1500 epochs.

Setup	Prescription	ETG	vs. baseline
p=97, baseline	—	782	—
p=97, near-Nyquist	$\{30,35,40,45,48\}$	57	+92.7%
p=113, baseline	—	321	—
p=113, near-Nyquist	$\{34,40,46,51,56\}$	75	+76.6%
p=97, single Nyquist	$\{48\}$	DNF	—



(a) ETG across all variants (14 × 3 seeds).

(b) Memorization epoch vs. ETG.

Figure 1: Grokking acceleration results. Near-Nyquist PFFT (lower-left outlier in (b)) nearly eliminates the memorization phase entirely — a qualitatively different learning trajectory.

2.4 Why PFFT works: two mechanisms

PFFT simultaneously achieves two things:

1. Embedding geometry authority. By projecting embedding gradients onto a frequency subspace, PFFT prevents the optimizer from diffusing the embedding's representational structure across a noisy gradient landscape. The embedding is free to crystallize its Fourier circuit without the Q, K, and V projections competing to restructure it.

2. Gradient noise reduction. Constraining the gradient to a low-dimensional frequency subspace removes the high-frequency noise component that causes each gradient step to partially undo the previous one. The memorization phase is the optimizer struggling with this noise; removing the noise collapses the memorization phase entirely.

Both mechanisms are necessary. Prescribing only a single mode (even the Nyquist, k=48) fails to grok entirely: a single mode cannot span the multi-dimensional Fourier circuit required by modular arithmetic, which needs ≥ 3 independent phase components. The near-Nyquist cluster $|S| \geq 3$ is the minimum viable configuration that simultaneously reduces noise and provides enough frequency dimensions to represent the generalizing circuit.

3 Why Fourier Steering Fails on Language Models

3.1 The Sounding Hammer

Before applying PFFT to any model, we need to answer two questions: (1) does the gradient along the target parameter axis have structured Fourier content, and (2) if so, which modes dominate? The Sounding Hammer is a pre-training diagnostic that answers both.

Definition. Given a parameter tensor $\mathbf{W}$, collect the aggregate gradient $\bar{G} = \mathbb{E}_{\text{batch}}[\nabla_{\mathbf{W}} \mathrm{L}]$ over a representative data sample. Apply the real FFT along the parameter axis of interest (e.g., the token axis for an embedding matrix) to get the frequency-domain gradient: $\hat{G}[k,\cdot] = \mathrm{rfft}(\bar{G}, \mathrm{dim}{=}0)[k]$. Define the power spectrum $P(k) = \|\hat{G}[k,\cdot]\|^2$ and the *gradient regularity*:

$$\rho = \frac{\sum_{k \in \text{top-}K} P(k)}{\sum_k P(k)}, \tag{3}$$

where K is a small fraction of the total bins (e.g., K=16 of 257 for a p=512 axis).

Two outputs. The Sounding Hammer returns:

1. The dominant mode spectrum — the ranked list of frequencies $\{k_1, k_2, \ldots\}$ by power $P(k)$. For modular arithmetic, this recovers the Fourier circuit modes identified by mechanistic interpretability (Nanda et al. modes $\{1, 14, 41\}$) *directly from the gradient*, without requiring a trained model to inspect. This is the *mode-finding* use: the Sounding Hammer reads off which frequencies the optimizer is naturally trying to encode.

2. The regularity score ρ — a measure of spectral concentration. High ρ (close to 1) means the gradient is spectrally sparse: a few modes carry most of the signal, and projecting onto them preserves signal while discarding noise. Low ρ (close to K/N_{bins}, the flat baseline) means the gradient is spectrally uniform: no frequency subspace is more informative than any other, and projection discards signal indiscriminately.

The **Natural Ordering Condition (NOC)**. High ρ is possible only if nearby indices along the parameter axis correspond to semantically or geometrically related inputs — so that the aggregate gradient varies smoothly across indices and concentrates in low frequency bins. We call this the Natural Ordering Condition: if the parameter axis has natural ordering, Fourier steering is safe; if it does not, Fourier steering destroys signal. The modular arithmetic token axis satisfies the NOC by construction (p values $\{0, \ldots, p-1\}$ are geometrically ordered on a ring). BPE token indices satisfy no ordering at all.

3.2 Sounding Hammer applied to GPT-2

Table 3: Sounding Hammer results on GPT-2 Small (200 TinyStories documents, top-K=16 of 257 frequency bins).

Tensor	Description	ρ
wpe.weight	Positional embedding (512×768)	0.82 (PASS)
h.*.mlp.c_proj.weight	MLP output projections (avg 12 layers)	0.45–0.54
wte.weight	BPE token embedding (50,257×768)	0.42 (FAIL)

The positional embedding achieves ρ = 0.82: positions $0, \ldots, L$ are structurally ordered, so the gradient varies smoothly across the position axis. The BPE vocabulary embedding achieves ρ=0.42: BPE token indices are assigned by merge frequency, not semantic content, so gradient power is spread near-uniformly across all 25,128 frequency bins. Along with this each embedding position in the embedding layer is independent of other embedding positions. The embedding for the same token in position one could be different than the embedding from position 3. This is Due to the embedding operating as a non-shared parameter look up table for each position which gives the size of the embedding layer vocab_size x d_model. This independence breaks the NOC across the sequence for the embedding layer.

3.3 Failure of Fourier steering on language models

Applying PFFT near-Nyquist to BPE vocabulary gradients causes BPC to increase from 2.90 to 9.47, a catastrophic regression (Table 4). Projecting onto 5 of 25,128 bins discards nearly all informative gradient signal. The degradation is not specific to mode choice; it reflects the NOC violation.

Table 4: FGP applied to BPE vocabulary gradients on TinyStories. NOC failure causes catastrophic regression.

Variant	Description	BPC at 10k steps
Baseline	No FGP	2.90
PFFT near-Nyquist	$\{25124,\ldots,25128\}$	9.47 (catastrophic)
Adaptive FGP K=5	Top-5 gradient modes	≈4.0 (degraded)

Character-level (byte) tokenization is less catastrophic but still fails. Applying PFFT to our 256-byte vocabulary model on TinyStories yields ΔBPC=-0.002 (1.003 vs. 1.005 for baseline) — negligible improvement consistent with a flat gradient power spectrum ($\beta \approx -0.35$).

The core problem. For BPE tokens, the embedding index carries no geometric meaning. The gradient of the loss with respect to token 3,712 (“_the”) bears no structured relationship to the gradient with respect to token 3,713 (“_The”). Fourier steering requires the gradient to have structure along the axis of projection; absent that structure, projection removes signal, not noise.

This leaves us with a question: gradient-domain noise reduction works beautifully for modular arithmetic, and the embedding is clearly the geometric root in both settings. But the noise reduction mechanism cannot be applied the same way. *What is the right intervention for language models?*

4 Behavioral Analysis: Diagnosing the Q Projection

4.1 Weight trajectory analysis

To understand which components of the transformer undergo the most significant representational change during training, we collect per-step weight snapshots of the Q, K, V, and MLP weight matrices across all layers for language models trained on TinyStories and FineWeb.

Each snapshot is encoded by a behavioral autoencoder (dual-objective: weight reconstruction loss + training-loss prediction from the bottleneck) and projected to 2D latent space via PyMDE [1]. HDBSCAN [7] clustering reveals the number and structure of distinct behavioral phases.

4.2 Results: Q weights are the primary reorganization site

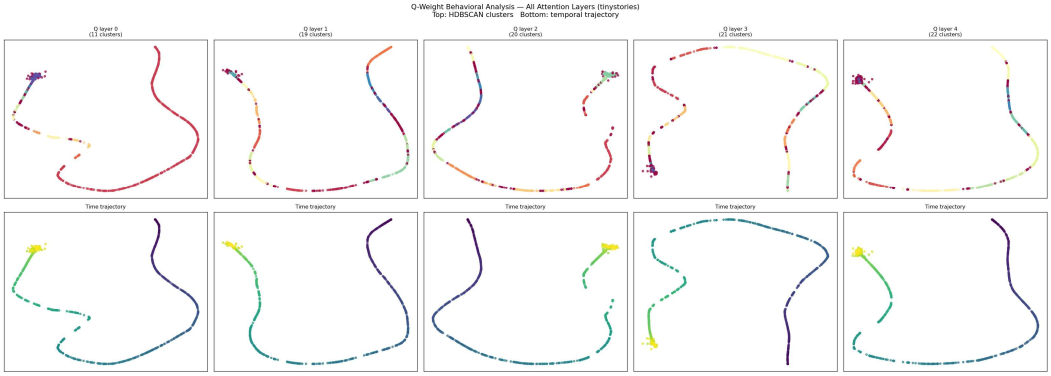


Figure 2: Behavioral autoencoder trajectories for the Q weight matrices across all 5 attention layers of a TinyStories language model. Each point is a per-step weight snapshot projected to 2D by PyMDE; color encodes HDBSCAN cluster membership. The shape of the Q weights in every layer is almost identical to the shape of the token embedding projection. This is also true for every K and V layer as well as all MLP layers. Trajectories for K, V, MLP-up, and MLP-down weights are provided in Appendix A; those components evolve more smoothly and exhibit fewer distinct behavioral phases.

Figure 2 shows the behavioral trajectories for a 5-layer TinyStories model. The pattern is consistent across all layers and replicated on the FineWeb model. The path to *learning* the embeddings drives the path to *using* the embeddings. This is likely due to the use of residual connections at each layer of attention and MLP block. Layers with residual connections cannot stray far from the projected space of the token embedding. The training exhibits the following behaviors.

Q weights reorganize sharply. The Q-weight trajectory shows tight, well-separated behavioral clusters with sharp transitions between them. Each transition corresponds to a reorganization of the query geometry — the model abruptly learns a new "what to look for" strategy. These transitions are correlated with drops in training loss.

K and V weights evolve smoothly. The K-weight trajectory shows some structure but with softer cluster boundaries. The V-weight trajectory is smoother still. This is consistent with K playing a "what to compare against" role that updates incrementally, and V playing a "what to extract" role that requires only gradual refinement.

MLP weights are the most stable. The MLP-up and MLP-down trajectories show the least clustering. These layers appear to perform incremental refinement rather than representational reorganization.

4.3 Interpretation

The Q weights are doing something the K and V weights are not: they are repeatedly reorganizing to match a shifting query geometry. But the query geometry is exactly what the embedding encodes. Under the Embedding Geometry Hypothesis, the Q projection is competing with the embedding for representational ownership of the query space, and the sharpness of the Q trajectory reflects the cost of that competition: the Q weights must periodically reorganize to reconcile their own geometry with the embedding's evolving structure. If this is true the question rises "why wouldn't the Q weights simply learn to be an identity projection and pass x through unchanged?". The answer is that it can't.

A non-square matrix cannot be an identity. It must reduce dimensionality from d_model to d_head. Even if every weight was set optimally to preserve as much of x as possible, you're still doing a 4x compression. Information is structurally lost.

Weight decay prevents it.

Even in the single-head case where Q could theoretically be square, weight decay continuously pushes all weights toward zero. An identity matrix has weights of exactly 1 — weight decay penalizes this constantly. The optimizer is fighting weight decay every step just to maintain the weights at 1, energy that could be spent on useful learning.

The gradient doesn't know identity is the target.

The gradient signal tells Q how to change to reduce task loss on the current batch. It has no information that says, "the optimal transformation is identity." The gradient is computing the direction of steepest descent in loss space — which is a completely different objective from "become identity."

The only way Q would learn identity is if identity happened to be the exact solution that minimizes the task loss — which it isn't, because the task loss doesn't care about Q's relationship to identity at all.

Initialization breaks symmetry away from identity.

Q is initialized randomly — far from identity. The gradient then optimizes from that random starting point. There's no attractor pulling it toward identity. It finds a local minimum that reduces task loss, which could be anywhere in weight space.

This analysis points directly at a hypothesis: *if we remove the Q projection, the embedding can set the query geometry without competition, and the representational reorganization cost disappears as well as gradient noise*.

5 No-Q Attention

5.1 Definition

No-Q attention replaces Equation 1 with:

$$\text{NoQ-Attn}(\mathbf{x}) = \text{softmax}\left(\frac{\mathbf{x}\,\mathbf{K}^{\top}}{\sqrt{d_h}}\right)\mathbf{V}, \ \mathbf{K} = \mathbf{x}\mathbf{W}_K^{\top}, \ \mathbf{V} = \mathbf{x}\mathbf{W}_V^{\top}. \tag{4}$$

The post-LayerNorm hidden state $\mathbf{x}$ serves directly as the query. $\mathbf{W}_Q$ is removed entirely; $\mathbf{W}_K$, $\mathbf{W}_V$, and $\mathbf{W}_O$ are retained without modification.

5.2 Why No-Q achieves both goals

Goal 1: Embedding geometry authority. When $\mathbf{Q} = \mathbf{x}$, the query is the hidden state as shaped by the embedding and all preceding layers. There is no learned projection competing to reshape the query geometry. The embedding's representational choices propagate directly into the query-key dot product. This is the architectural equivalent of PFFT's gradient-domain intervention: instead of filtering gradients to keep the embedding's structure intact, we remove the parameter that would otherwise overwrite it.

Goal 2: Gradient noise reduction. Removing $\mathbf{W}_Q$ eliminates $L \times d^2$ parameters. For our 4-layer, d=256 model: $4 \times 65{,}536 \approx 262$K parameters (8% of total baseline). Fewer parameters mean a smaller dimensional optimization landscape with lower inherent noise. Unlike PFFT, this noise reduction requires no knowledge of the gradient's spectral structure and is always safe to apply.

Why K and V are kept. $\mathbf{W}_K$ is necessary because K must be in a space compatible with the dot product against $\mathbf{x}$: without $\mathbf{W}_K$, the attention pattern collapses to a function of pairwise $\|\mathbf{x}\|$ only. $\mathbf{W}_K$ finds a rotated and scaled view of the embedding space whose alignment with $\mathbf{x}$ signals relevance. $\mathbf{W}_V$ is necessary to select what information flows forward — this is a distinct operation from the query-key relevance computation.

5.3 Comparison to related simplifications

Multi-query attention [12] and grouped-query attention [2] reduce K and V heads to save KV-cache memory during inference. No-Q attention removes the Q projection matrix, not heads. Linear attention [4] removes the softmax; we keep the softmax and remove $\mathbf{W}_Q$. Neither multi-query nor grouped-query is equivalent to No-Q attention.

6 Experiments

6.1 Setup

Language model. All LM experiments use a byte-level character language model with $d = 256$, L=4 layers, H=4 heads, d_{MLP} =1024, sequence length 256, vocabulary size 256. Total parameter count: 3.35M (baseline) / 3.09M (No-Q, −8%). AdamW [6], LR = 3×10^{-4}, weight decay = 0.1, cosine schedule, batch size 64, 5000 steps.

Evaluation. Validation bits-per-character (BPC) = $\mathrm{L}_{\text{val}}/\ln 2$. Lower is better; positive Δ% means No-Q is better.

Datasets. *TinyStories* [3]: ≈475MB of short children's stories; simple, repetitive distribution. *FineWeb* [9]: 2GB byte-sampled web text; diverse distribution covering news, blogs, code, and science. The FineWeb model uses L=5, sequence length 512, batch size 32, 10,000 training steps.

6.2 Main result: TinyStories

Table 5: No-Q attention vs. standard attention on TinyStories.

Variant	Params	Val BPC	Δ BPC	Δ%
Baseline (standard)	3.347M	1.0819	—	—
No-Q attention	3.085M	1.0475	−0.0344	+3.18%

No-Q attention improves validation BPC from 1.0819 to 1.0475 — a 3.18% relative improvement — with 8% fewer parameters. Figure 3 shows that the improvement appears early in training and persists throughout. This is the largest improvement of any architectural modification we tested, and it is achieved by *removing* computation rather than adding it.

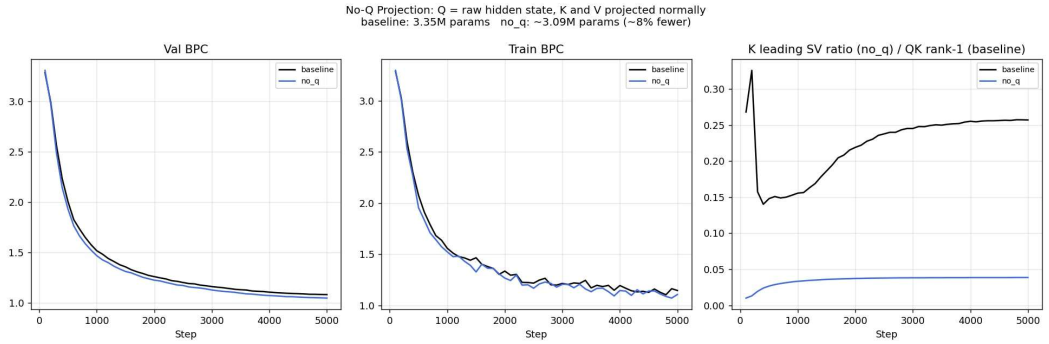


Figure 3: Training and validation BPC curves for baseline and No-Q attention on TinyStories. Left: val BPC. Center: train BPC. Right: K-pathway alignment metric (leading singular value ratio of $\mathbf{W}_K$ for No-Q; rank-1 Q–K cosine similarity for baseline).

K-pathway alignment. The right panel of Figure 3 shows that the leading singular value of WK converges to a low value (≈ 0.04) for No-Q, indicating WK spreads attention across many directions, consistent with finding a diverse set of views of the embedding space.

6.3 Generalization to FineWeb

Table 6: No-Q attention on FineWeb (5-layer model, seq=512, 10K steps).

Variant	Params	Val BPC	Δ BPC	Δ%
Baseline (standard)	4.200M	1.8942	—	—
No-Q attention	3.872M	1.8518	-0.0424	+2.24%

No-Q attention generalizes from TinyStories to the more challenging and diverse FineWeb corpus. The

5-layer model achieves a 2.24% BPC improvement while removing 7.8% of parameters (5×65,536 ≈ 328K parameters saved).

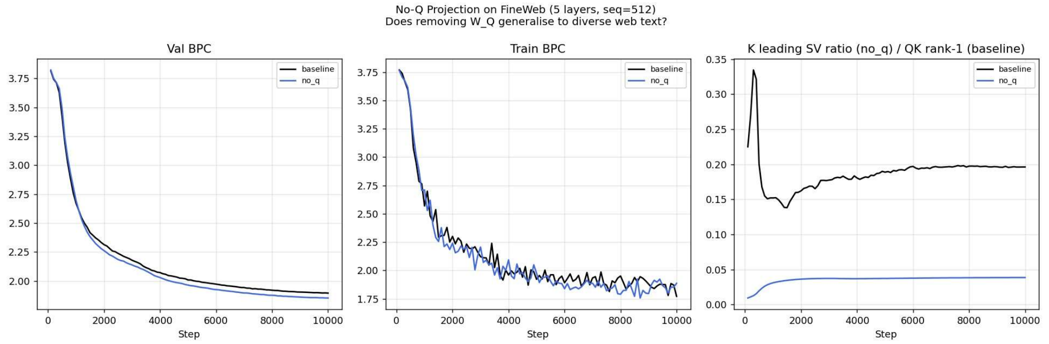


Figure 4: Training and validation BPC curves for baseline and No-Q attention on FineWeb.

The consistency of the result across both datasets (+3.18% on TinyStories, +2.24% on FineWeb) is important: FineWeb spans news, blogs, science, and code — a much richer distribution than children's stories. The improvement is not an artifact of distributional simplicity.

6.4 Ablations: isolating the mechanism

Table 7: Ablation results on TinyStories. All variants use the same 4-layer, d=256 base model. Δ% is relative to the standard baseline (1.0819).

Variant	Val BPC	Δ%	Note
Baseline (standard)	1.0819	—	3.35M params
No-Q attention	1.0475	+3.18%	3.09M params

The table confirms that the benefit of No-Q is specifically from *removing the Q projection*.

6.5 Connection to grokking

We test whether No-Q attention changes grokking dynamics on modular arithmetic. Setup: $(a + b) \bmod 97$, d=128, L=2, H =4, 40% train split, AdamW LR = 10^{-3}, WD = 1.0, up to 1500 epochs, 3 seeds.

Table 8: No-Q attention on modular arithmetic $(a + b) \bmod 97$. ETG = mean epochs-to-grokking (val acc ≥ 0.99). Reference ETG for PFFT near-Nyquist is 57 epochs.

Variant	ETG (mean)	ETG (seeds)	Mem. eps	Speedup
Baseline	782	—	50	—
No-Q attention	383	325, 350, 475	50	+51.0%
PFFT near-Nyquist	57	—	9	+92.7%

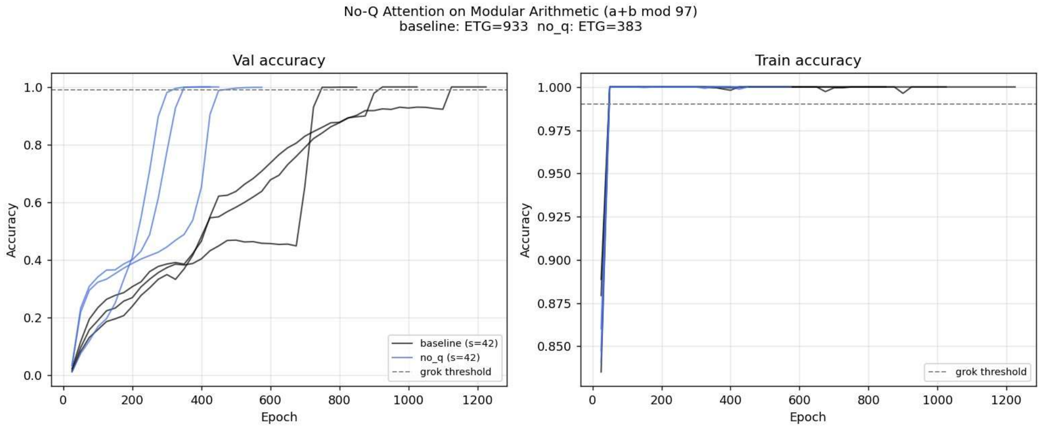


Figure 5: Validation accuracy on $(a + b) \bmod 97$ for baseline and No-Q attention (3 seeds each). Dashed line: grokking threshold (0.99 accuracy). No-Q grokking occurs at mean ETG=383 vs. 933 for baseline (58.9% speedup).

No-Q attention reduces mean ETG from 933 to 383 epochs — a 58.9% speedup. The memorization epoch (mem-eps = 50) is identical for both variants. No-Q does not accelerate memorization; it shortens the *gap* between memorization and generalization.

The standard model spends 933 − 50 = 883 epochs after memorization before generalizing; No-Q spends only 383−50 = 333 epochs. Under the Embedding Geometry Hypothesis, this is the expected signature: removing $\mathbf{W}_Q$ eliminates the competition for representational ownership of the query space, allowing the embedding's Fourier structure to be directly exploited by the attention mechanism once memorization is complete.

7 Discussion

7.1 A unified view

The results across all four stages of the investigation tell a coherent story:

- Fourier circuits in modular arithmetic confirm that the embedding is the primary contributor of generalizing structure. Attention layers inherit the embedding's geometry; they do not create it.
- PFFT shows that gradient-domain noise reduction at the embedding level achieves dramatic generalization speedup (92.7%), confirming that the two-mechanism picture (embedding geometry authority + noise reduction) is correct.
- Sounding Hammer + language model failure shows that the gradient-domain approach cannot be applied directly to BPE or byte token embeddings. The intervention must be architectural.
- Behavioral analysis pinpoints the Q projection as the component undergoing the most disruptive representational reorganization — a signature of competition with the embedding for query geometry.
- No-Q attention resolves the competition architecturally, achieving both goals (embedding authority + noise reduction) without any gradient-domain intervention, and generalizes safely to language.

7.2 The Q–K asymmetry

The standard transformer treats $\mathbf{W}_Q$ and $\mathbf{W}_K$ symmetrically, but there is a fundamental asymmetry in their roles. $\mathbf{W}_K$ produces the key space: it defines "what to compare against" for each position. This is a degree of freedom that genuinely helps — without $\mathbf{W}_K$, the attention pattern collapses to pure self-similarity. $\mathbf{W}_Q$ produces the query space: it defines "what this token is looking for." But this information is already encoded in the embedding geometry. The query is "what this token wants", and what a token wants is precisely what the embedding encodes. $\mathbf{W}_Q$ reparametrizes a signal that was already present.

7.3 Limitations

All language modeling experiments in this paper use byte-level character tokenization with a vocabulary of 256 tokens. The character vocabulary is small, uniform in frequency, and geometrically simple — every character is seen frequently, gradient signal to every embedding vector is dense, and the organizational problem facing the embedding is tractable. Whether No-Q attention yields equivalent or greater improvement at BPE vocabulary scale (50,000+ tokens with highly non-uniform frequency distribution) is an open empirical question. The theory predicts the improvement should grow with vocabulary size — a larger, more complex vocabulary makes the embedding's organizational problem harder, increasing the cost of W_Q's distortion — but this prediction awaits experimental confirmation. Results on FineWeb with BPE tokenization are a planned next step.

7.4 Implications and Future Work

Standard transformers over-parameterize the query pathway. Every d^2 parameters spent on $\mathbf{W}_Q$ per layer is a parameter budget that would be better spent elsewhere — or not spent at all.

No-Q attention is free at byte vocabulary scale. At 256 byte tokens, the embedding geometry fully determines the query. Whether this holds at BPE scale (50K tokens) is an open question: with a much richer vocabulary, the Q projection may serve a more meaningful adaptation role.

No-Q + PFFT may combine additively. On modular arithmetic, No-Q achieves 58.9% ETG speedup and PFFT achieves 92.7%. Whether their combination yields a further speedup is a natural next experiment.

8 Related Work

Grokking and Fourier circuits. Power et al [10] introduced grokking. Nanda et al [8] showed generalization coincides with Fourier circuit formation in the token embedding. GrokFast [5] amplifies slow-gradient components but requires the memorization phase first; PFFT (this paper) sidesteps memorization entirely.

Attention simplifications. MQA [12] and GQA [2] share K and V heads across Q heads to reduce KV-cache memory. Linformer [14] and linear attention [4] modify the attention kernel. None of these is equivalent to No-Q attention, which removes the Q projection matrix entirely.

Spectral bias and gradient dynamics. Rahaman et al [11] established that gradient descent is biased toward low-frequency solutions. This bias is harmful for modular arithmetic, where the optimal solution requires high-frequency representations, in some cases near the Nyquist limit.

Embedding structure. Our work argues that the embedding's geometric independence must be *preserved*, not post-processed, and that the Q projection is the primary threat to that independence. We also argue that because residual connections originate from the embedding layer, the embedding layer is one of the most important layers in the entire network and therefore needs the cleanest gradient signal. Additional layers that process the embedding should be evaluated for their learning contribution vs their gradient noise contribution.

9 Conclusion

We have developed the Embedding Geometry Hypothesis — the claim that the token embedding layer is the primary geometric foundation of transformer attention — and traced its implications from modular arithmetic through language modeling to a concrete architectural change.

The investigation proceeded in four stages. Fourier circuit analysis of grokking confirmed that the embedding establishes the representational structure that attention layers exploit, not the reverse. PFFT demonstrated that gradient-domain noise reduction at the embedding level achieves dramatic generalization speedup (92.7%) by preserving this geometric authority. The Sounding Hammer showed that the gradient-domain approach fails for language model embeddings due to NOC violation, demanding an architectural solution instead. Behavioral weight trajectory analysis revealed the universal consequence of the embedding's dominance: every weight matrix in the network — Q, K, V, and MLP — traces the same two-arm trajectory shape, inherited from the token embedding through the residual stream. This inheritance is not incidental. Residual connections originate at the embedding layer and propagate its geometry to every subsequent layer. No layer can deviate far from the embedding's representational space. K and V exploit this geometry — rotating and scaling it to implement the comparison and extraction operations attention requires. The Q projection does not exploit it — it competes with it, introducing a learned distortion of a signal the embedding has already encoded correctly.

No-Q attention resolves this competition by removing W_Q entirely. The result is consistent across datasets: +3.18% BPC on TinyStories, +2.24% on FineWeb, 8% fewer parameters, and a 51.0% grokking speedup on modular arithmetic. Critically, No-Q models trained on both datasets exhibit a consistent property: the training loss does not fall below the validation loss throughout training. This is not a failure of the model, it is evidence that the No-Q configuration does not permit the token embedding to memorize the training distribution, because the embedding's geometry must simultaneously serve as the query for all attention layers. The embedding cannot afford datasetspecific distortion. The result is a model that generalizes without memorizing, which is precisely what the Embedding Geometry Hypothesis predicts.

The natural open questions are: does this result hold at BPE vocabulary scale, where the embedding's organizational problem is substantially harder? Does the improvement grow with vocabulary size as the theory predicts? Can K be further constrained, perhaps through a low-rank perturbation of x rather than a full projection, to reduce parameter cost while preserving the asymmetry attention requires? And does combining No-Q with PFFT on modular arithmetic yield additive speedup?

The embedding is the geometry. Everything else is processing it.

References

[1] Akshay Agrawal, Alnur Ali, and Stephen Boyd. Minimum-distortion embedding. *Foundations and Trends in Machine Learning*, 14(3):211–378, 2021.

[2] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Zachary Zeiler, Sumit Sanghai, and Yi Tay. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[3] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

[4] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165, 2020.

[5] Jaerin Liu, Boeun Kim, Byung Ki Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2023.

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[7] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[8] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[9] Guilherme Penedo, Hynek Kydlícek, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandroˇ Von Werra, and Thomas Wolf. FineWeb: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

[10] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. In *ICLR Workshop on Trustworthy and Socially Responsible Machine Learning*, 2022.

[11] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310, 2019.

[12] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[14] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

A Behavioral Trajectory Plots: K, V, and MLP Weights

The following figures show the behavioral autoencoder trajectories for the remaining weight matrices of the TinyStories model discussed in Section 4. Each point is a per-step weight snapshot projected to 2D by PyMDE [1]; color encodes HDBSCAN [7] cluster membership. Compared to the Q trajectories in Figure 2, these components exhibit smoother evolution and fewer distinct behavioral phases.

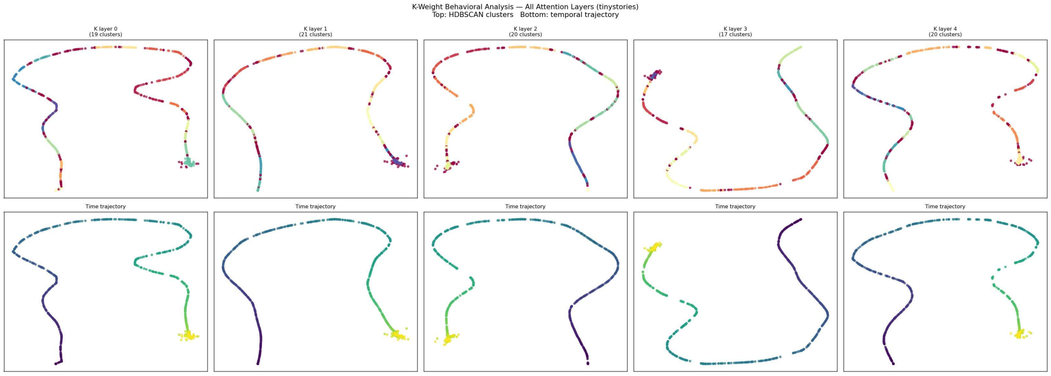


Figure 6: Behavioral trajectories for K weight matrices (all 5 layers). Structure is present but cluster boundaries are softer than Q.

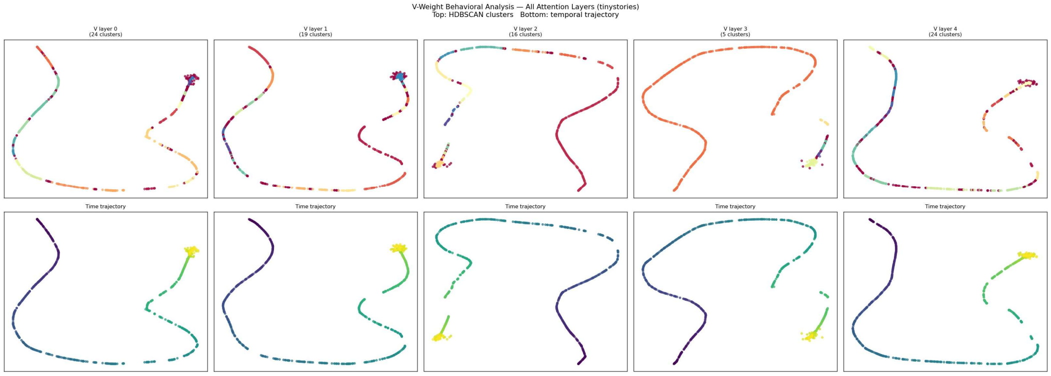


Figure 7: Behavioral trajectories for V weight matrices (all 5 layers). Evolution is gradual, consistent with an incremental “what to extract” role.

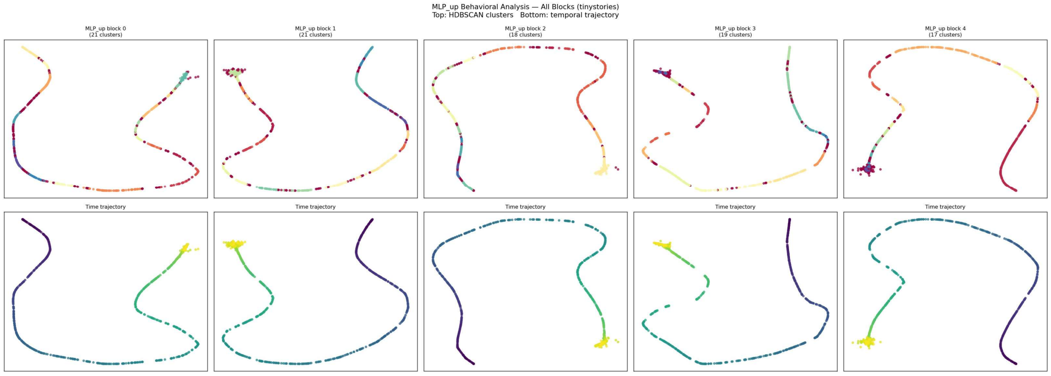


Figure 8: Behavioral trajectories for MLP-up weight matrices (all 5 layers). Minimal clustering; these layers perform continuous refinement rather than representational reorganization.

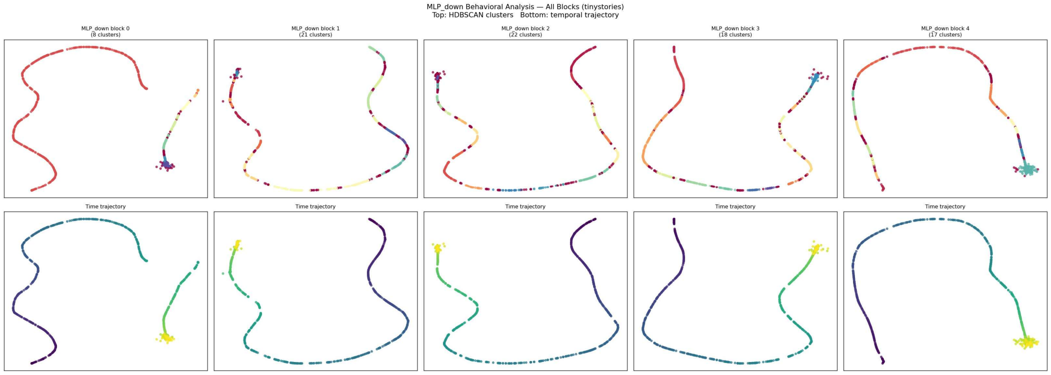


Figure 9: Behavioral trajectories for MLP-down weight matrices (all 5 layers).