Linear Predictability of Attention Heads in Large Language Models

Khalid Shaikh[1,†] **Asmit Kumar Singh**[1]
Rebecca Christopher Dsouza[2] **Shikhar Shiromani**[1]

[1]Georgia Institute of Technology [2]Santa Clara University

Abstract

Large language model (LLM) inference is increasingly bottlenecked by the Key–Value (KV) cache, yet the fine-grained structure of attention-head activations remains poorly understood. We show that pretrained Transformers exhibit a pervasive inter-head linear structure: for a given token, the Query, Key, and Value (QKV) vectors of an attention head can often be reconstructed as a linear combination of a small number of peer heads—typically within the same layer. Across Llama-3.1 8B, Falcon3-10B, OLMo-2-7B, and Qwen3-32B, just 2–5 reference heads recover many target heads with high fidelity (e.g., mean $R^2 \approx 0.76$ for Keys on C4 with five references, and frequently $R^2 > 0.85$ on GSM8K). This predictability is learned rather than architectural: it is largely absent at random initialization, rises rapidly during pretraining as we track through OLMo-2 checkpoints, and is supported by a theoretical lower bound showing high mean-squared error for linear prediction at initialization. We further connect emergence to increasing intra-layer alignment of Key projection subspaces. Finally, we exploit this redundancy for efficiency by caching only reference-head KV states and reconstructing the remaining heads on the fly via lightweight linear maps, achieving 2× KV-cache reduction with model-dependent accuracy trade-offs (4.5–5.5 pp average drop on Falcon3-10B/Qwen3-32B across five benchmarks, and larger drops on Llama-3.1 8B), and we find that reconstructing Keys is substantially less harmful than reconstructing Values. [1]

1 Introduction

A central question in understanding Large Language Models (LLMs) is what structural properties emerge in their attention mechanisms during pretraining. While prior interpretability work has studied *which* tokens heads attend to and *which* heads can be pruned, the fine-grained relationships among the internal Key, Value, and Query *activation vectors* of different heads remain poorly understood. In this paper, we show that these activations are not independent: in pretrained LLMs, individual heads are strongly *linearly predictable* from their peers, a property absent at initialization and arising as a consequence of pretraining. Beyond its theoretical interest, this structure has a direct practical implication: it enables compression of the KV cache—the primary memory bottleneck during autoregressive generation (Kwon et al., 2023)—without any architectural changes.

Previous interpretability work has shed light on attention, revealing functional specializations and significant redundancy between attention heads at a semantic level (Clark et al., 2019; Michel et al., 2019). These insights have led to effective techniques for pruning or sharing attention components to alleviate KV cache pressure (Agarwal et al., 2024; Ainslie et al., 2023; Shim et al., 2021). However, the finer-grained relationships among the *internal K, V, and Q vector activations* of different heads have remained largely unexplored; concurrent work (e.g., AQUA-KV; Shutova et al., 2025) has begun to probe this space in the context of quantization, but the broader structural phenomenon and its implications for head-elimination compression remain uncharacterized. In this paper, we ask: *do pretrained LLMs develop systematic linear dependencies among the activation vectors of different attention heads, and if so, when and how does this structure emerge during training?*

We give an overview of our contributions below.

Identification and Characterization of Linear Predictability (§2). We show that individual attention head K, V, and Q activations in pretrained LLMs are *linearly predictable* from a small set of peer heads. Across LLaMA-3.1 8B, Falcon3-10B,

[†]Corresponding author.

[1]Code: https://github.com/shaikhalid/Interpretability-and-Linear-Predictability-of-Attention-Heads

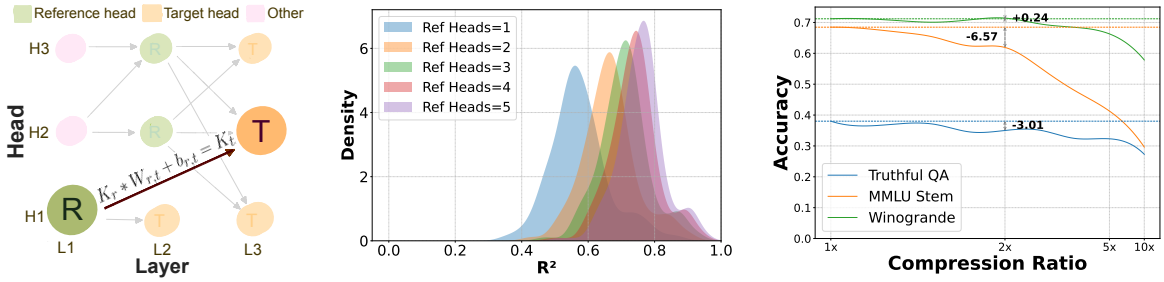


Figure 1: Linear predictability among attention heads enables significant KV-cache compression. (Left) Schematic of approximating a target head's (orange) Key activations using a learned linear projection from reference heads (green). (Middle) In a pretrained Llama-3.1 8B model, most heads' Key activations are highly predictable (high R^2) from a few peers on the C4 dataset (Raffel et al., 2023) , indicating a shared low-rank subspace. (Right) This predictability allows for substantial KV-cache compression with minimal performance degradation on benchmarks like TruthfulQA, MMLU-STEM, and Winogrande. The dashed line marks 2×compression.

OLMo-2-7B, and Qwen3-32B, Key states reach a mean R^2 of 0.76 on C4 with $N = 5$ reference heads, and $R^2 > 0.85$ on GSM8K (Fig. 1). Over 75% of pairwise head relationships exceed $R^2 = 0.5$, revealing a pervasive level of learned intra-layer structure not captured by prior attention-map redundancy analyses.

Emergence and Mechanistic Analysis (§3). Linear predictability is absent at initialization (avg $R^2 < 0.05$; fewer than 5% of head pairs exceed $R^2 = 0.10$) and develops during pretraining. Using OLMo-2 intermediate checkpoints (1B to ∼4T tokens), we trace a steady rise in intra-layer predictability while inter-layer links peak early and then decline. We connect this trajectory to growing *subspace overlap*: the overlap dimension of Key projection matrices within each layer increases monotonically during training, with Keys expanding fastest, Queries following, and Values remaining nearly orthogonal. A theoretical lower bound (Theorem 3.1) confirms that random initialization produces high MSE with overwhelming probability, establishing that this structure is learned during pretraining, not architectural.

KV Cache Compression via Head Elimination (§4). As a practical application, we compress the KV cache by storing only reference heads and reconstructing target heads on-the-fly with learned linear maps, achieving 2× memory reduction. Accuracy trade-offs are model-dependent: Qwen3-32B and Falcon3-10B incur average drops of 5.5 pp and 4.5 pp respectively across five benchmarks, while LLaMA-3.1 8B shows a larger 9.9 pp drop (Table 1). Predicting *Keys* is substantially safer than predicting *Values* (§4.3), consistent with the asymmetric subspace overlap growth in §3.

Relation to concurrent work. (Shutova et al., 2025) improve KV-cache quantization by learning compact layer-wise linear predictors—predicting each layer's keys from the previous layer's reconstructed keys, and values from the previous layer's reconstructed values together with current-layer keys—and then storing only quantized residuals. In contrast, we characterize head-level predictability (predominantly within-layer) and exploit it to eliminate KV storage for a subset of heads entirely: we cache a small set of full-precision reference heads and reconstruct the remaining heads on-the-fly using the linear maps identified in §2 and applied in §4.

2 Linear Predictability of Attention Heads

A core finding of this work is that pretrained Transformers develop striking linear dependencies among their attention head activations. In this section, we establish and quantify this phenomenon: the Key, Value, and Query activations of individual heads are linearly predictable from a small set of peer heads across diverse model families and tasks. We demonstrate that the activations of a target head can often be accurately reconstructed using a linear combination of activations from a small number of peer heads, a phenomenon that underpins the practical applications discussed later.

Notation. For an input $\boldsymbol{X} \in \mathbb{R}^{N \times m}$, each of the H heads in a layer projects $\boldsymbol{X}$ into Query, Key, and Value vectors via learned matrices $\boldsymbol{W}_{Q,h}, \boldsymbol{W}_{K,h}, \boldsymbol{W}_{V,h} \in \mathbb{R}^{m \times d_h}$ (e.g., $\boldsymbol{K}_h = \boldsymbol{X}\boldsymbol{W}_{K,h}$). Our analysis focuses on the interrelationships between these $\boldsymbol{K}_h, \boldsymbol{V}_h, \boldsymbol{Q}_h$ activation vectors across heads h and layers.

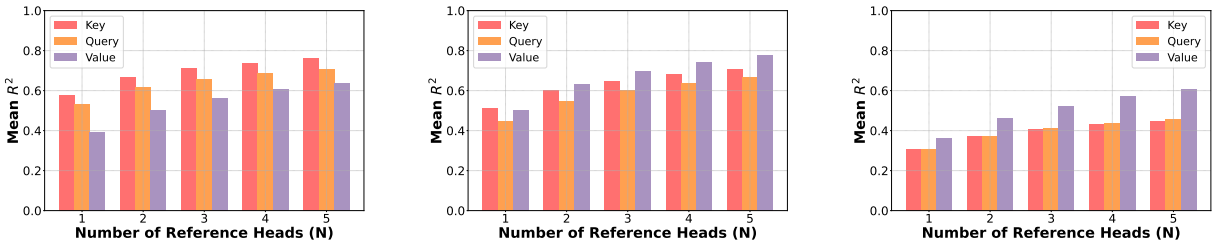


Figure 2: Linearity is pervasive—keys, queries, and values all become more predictable as the number of reference heads grows. Shown is the mean R^2 when reconstructing **key**, **query**, and **value** activations on a shared 150-sequence slice of C4 for three pretrained models: (Left) LLaMA-3.1 8B (Middle) Falcon-3 10B (Right) OLMo-2 7B. Across all panels, every stream (K, Q, V) shows a monotonic rise in R^2 as additional references are added, confirming that head-level representations in each model occupy a low-rank sub-space that can be captured with only a handful of heads.

2.1 Empirical Evidence of Linear Predictability

We fit OLS regressors to predict each head's K, Q, V activations from a subset of peer heads, on 150 sequences from C4 (≈30K token activations per model; full methodology in Appendix A). Two probes are run: *pairwise*, which fits each (H_t, H_r) pair under the constraint $\text{layer}(H_t) \geq \text{layer}(H_r)$, and *multi-predictor*, which combines the top-N reference heads from the pairwise stage in a multivariate OLS.

The results, shown in Figure 1 (middle panel) for Llama-3.1 8B Key states on C4, reveal remarkably strong linear predictability with a small number of peers. Pairwise: 75% of head pairs exceed $R^2 > 0.5$ (median 0.57). Multi-predictor: with $N = 3$ references the mean R^2 reaches 0.71 for Key states; with $N = 5$ over 75% of heads exceed $R^2 > 0.72$ and the overall mean reaches 0.76. Most of the gain comes from the first 2–3 well-chosen peers, with diminishing returns beyond.

Takeaway: *Individual head activations are systematically correlated with, and can be accurately approximated by simple linear transformations of a small set of peer heads.*

2.2 Discussion: Implications of Linear Predictability

Two questions naturally follow from the above observations. *(1) Why does the phenomenon matter?* Beyond revealing structural redundancy among attention heads, it shows we can drop many key-value vectors at decoding time and reconstruct them with linear projections, cutting memory usage (see Section 4). *(2) Why is this linearity surprising?* Predicting heads across layers is unexpected because the vectors pass through nonlinear transformations. Even within a layer, one head can linearly predict another only if their projection subspaces overlap. Concretely, let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times k}$ be projection matrices. For every input $\boldsymbol{x} \in \mathbb{R}^m$ we seek $\boldsymbol{C} \in \mathbb{R}^{k \times k}$ satisfying $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{C} = \boldsymbol{x}^\top \boldsymbol{B}$, which demands $\boldsymbol{A}\boldsymbol{C} = \boldsymbol{B}$. If the column spaces of $\boldsymbol{A}$ and $\boldsymbol{B}$ are disjoint, exact prediction is infeasible to achieve; low error can still be attained when the subspaces align or have small principal angles. As the next section demonstrates, this linear structure is absent at random initialization and emerges progressively during pretraining—a finding that transforms what could be dismissed as an engineering trick into a fundamental structural property of learned Transformers.

3 Insights on Linear Predictability of Heads

The observation of strong linear relationships among attention head activations (Section 2) prompts further investigation into their origin and development. This section examines how and when this structure arises, addressing two key questions: (1) Is linear predictability an intrinsic property of Transformers, or is it learned? (2) How does this predictability evolve throughout the pretraining process? Our findings indicate that linear predictability is negligible at initialization and is indeed a learned byproduct of pretraining. We also provide a theoretical argument underscoring why such predictability is unlikely in randomly initialized networks. To our knowledge, no prior work—including concurrent methods that exploit inter-head or inter-layer linearity for cache compression (Shutova et al., 2025)—has studied when or why these linear dependencies arise; the analysis in this section addresses this gap.

Evolution of Linearity: An Emergent Property of Pretraining. We first contrast the linear predictability observed in fully pretrained language models with their randomly initialized (untrained) counterparts. By analyzing Llama-3.1 8B on the C4 dataset, we find that prior to any pretraining, the average R^2 for predicting one head's Key activations

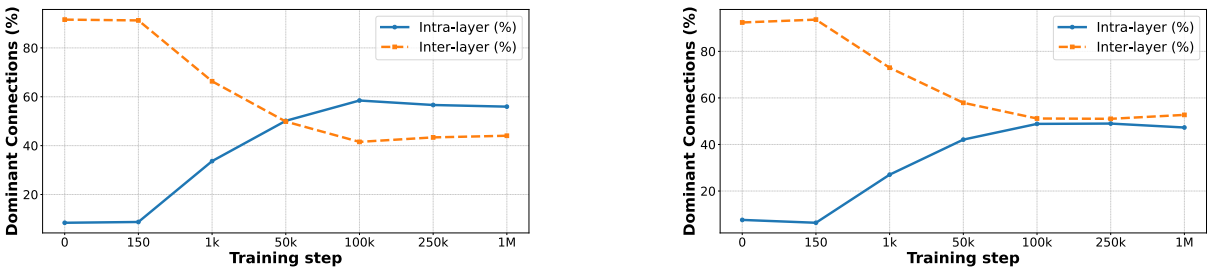


Figure 3: Dominant predictors migrate from cross-layer to within-layer. For OLMo-2 7B we track, across checkpoints, where a head's strongest linear predictors reside. (Left): percentage of heads whose *single* best (R^2-max) predictor lies in the same layer versus a different layer. (Right): the same percentages after aggregating each head's *five* best predictors. At initialization almost all dominant links are inter-layer; by 50 k–100 k steps intra-layer links overtake and remain prevalent, indicating that pre-training progressively concentrates shared computation inside individual layers.

from another single head is exceptionally low (with average $R^2 < 0.05$). This starkly contrasts with the significantly higher R^2 values we observe in the fully pretrained version of the same model. This comparison strongly suggests that the observed linear structure and inter-head redundancy are not inherent to the architecture but rather emerge and are refined during the pretraining phase.

As Figure 4 shows, pre-training transforms the head-to-head relationships in LLaMA–3.1 8B from almost independent to strongly linearly predictable. In the randomly initialized network (orange), fewer than 5% of head pairs exceed $R^2 = 0.10$; nearly the entire mass is concentrated at the origin. After optimization (blue), the distribution shifts dramatically rightward: the median climbs to $R^2 \approx 0.50$, roughly half of all pairs now surpass 0.50, and the top decile approaches 0.80. This broad peak around 0.55 confirms that the training process induces a shared low-rank sub-space in which one head's activations can be reconstructed from a small set of others.

Furthermore, analyzing the nature of these connections, we observe distinct trends for intra-layer versus inter-layer predictability as training progresses. Figure 3 shows the fraction of the strong connections that are intra-layer versus inter-layer at each OLMo-2 checkpoint. Initially, inter-layer is relatively higher, but as pretraining advances, intra-layer predictability steadily strengthens and becomes dominant, while strong inter-layer predictive links become less prevalent.

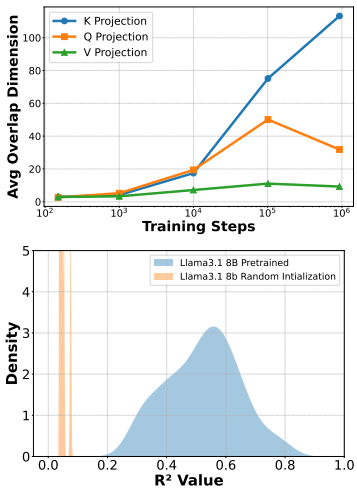


Figure 4: Left: Pretraining aligns head subspaces. Overlap dimension averaged across layers at different points during OLMo-2 training, indicating increasing intra-layer alignment. **Right: Training induces linear predictability.** Kernel-density estimate of R^2 for LLaMA-3.1 8B shows a broad moderate–high peak after training versus near-zero values at random initialization.

To gain finer-grained insight into this emergence, we use the publicly released intermediate checkpoints of the OLMo-2 model. This allows us to track the evolution of linear predictability from random initialization through various stages of pretraining (1B, 10B, 100B, and 1T tokens, up to the final model trained on $\approx$ 4T tokens). Concurrently, to analyze the structural origin of this linearity, we examine the overlap of the attention activation subspaces within layers for OLMo-2. Let $\boldsymbol{W}_K^{(\ell)} = [\boldsymbol{W}_{K,1}^{(\ell)}, \ldots, \boldsymbol{W}_{K,H}^{(\ell)}] \in \mathbb{R}^{m \times H d_h}$ be the concatenation of all key projection matrices in a specific layer ℓ. We measure the shared subspace using the *overlap dimension*:

$$\mathrm{OD}_H(\boldsymbol{W}_K^{(\ell)}) = \sum_{i=1}^{H} \mathrm{rank}(\boldsymbol{W}_{K,i}^{(\ell)}) - \mathrm{rank}(\boldsymbol{W}_K^{(\ell)}).$$

A higher OD_H indicates greater linear dependence among the column spaces of the individual head

projection matrices within that layer. We measure the intra-layer subspace overlap for key, value, and query projection vectors at each checkpoint. Figure 4 illustrates this evolution. At the very first checkpoint (100 steps) all three projections are nearly orthogonal—each layer shares less than 5 basis directions among its heads. As training unfolds, the overlap dimension grows monotonically—but not uniformly—across the three projections. Key subspaces expand the fastest, showing a steep, almost exponential rise that dominates late-stage checkpoints. Query projections follow the same trajectory, albeit with a gentler slope, while Value projections remain largely flat, indicating near-orthogonality throughout. This growing subspace overlap within layers directly mirrors, and likely underpins, the observed rise in intra-layer linear predictability of head activations.

Theoretical Analysis: Why Predictability is Low at Random Initialization? To further understand why significant linear predictability is absent before training, we theoretically analyze the case of randomly initialized projection matrices. We show that, with high probability, the best linear map to predict one randomly projected vector from another (or a set of others) incurs a substantial mean-squared error. Specifically, for a single-layer Transformer where input token embeddings $\boldsymbol{x}$ are drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_m)$ and Key projection matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times d_h}$ have i.i.d. Gaussian entries, the expected squared error of the best linear predictor is lower bounded by $\Omega(d_h m)$. This bound also holds if $\boldsymbol{A}$ represents the concatenation of n reference head projections ($m \times nd_h$) and $\boldsymbol{B}$ is a single target head projection ($m \times d_h$). This theoretical result (Theorem 3.1) formally establishes that strong linear predictability between arbitrary head activations is not expected in a randomly initialized state, corroborating our empirical findings and highlighting that such structure must be learned.

Theorem 3.1. *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times k}$ be two random matrices where each entry is sampled independently from $\mathcal{N}(0,1)$ and $k \leq m/2$. Let $\boldsymbol{x} \in \mathbb{R}^m$ be an input vector drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_m)$ independently of $\boldsymbol{A}, \boldsymbol{B}$. Then, there is an absolute constant $c_1 > 0$ such that*

$$\Pr\Big\{ \inf_{\boldsymbol{C} \in \mathbb{R}^{k \times k}} \mathop{\mathbb{E}}_{\boldsymbol{x}} \big\| \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{C} - \boldsymbol{x}^\top \boldsymbol{B} \big\|_2^2 < \frac{1}{2} k(m-k) \Big\} \leq 2\exp(-c_1 m).$$

Equivalently, with probability at least $1 - 2e^{-c_1 m}$ we have

$$\inf_{\boldsymbol{C}} \Big[\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_m)} \| \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{C} - \boldsymbol{x}^\top \boldsymbol{B} \|_2^2 \Big] \;\geq\; \frac{1}{4} mk.$$

The proof uses Hanson-Wright (Vershynin, 2018) and is given in Appendix D.

Discussion of Theoretical Result. The theorem formalizes the intuition that the column spaces of two independently drawn random projection matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are unlikely to be well-aligned. Therefore, projecting an input vector $\boldsymbol{x}$ through them will typically yield vectors whose relationship is not well captured by a simple linear transformation, leading to high reconstruction error. This contrasts sharply with the low error (high R^2) observed in pretrained models, emphasizing that the alignment of these projection subspaces is a key outcome of the learning process. Since one of the key steps of the proof is the application of Hanson-Wright inequality, which holds for any subgaussian vectors, the proof can naturally be extended to other common initialization methods such as Xavier uniform or Xavier normal initialization (Glorot and Bengio, 2010).

The monotonic growth of Key subspace overlap during pretraining, coinciding with the rise in linear predictability, suggests that gradient updates across heads sharing the same input $\boldsymbol{X}$ may drive projection matrices toward aligned column spaces. While a full theoretical characterization of this process remains open, the empirical trajectory—from near-orthogonal initialization to substantially overlapping trained subspaces—indicates a systematic, not accidental, alignment. Note that while Theorem 3.1 formally applies to the first layer under Gaussian inputs, the empirical evidence ($R^2 < 0.05$ across *all* layers at initialization) suggests the qualitative conclusion extends throughout the network.

4 Exploiting Linear Predictability for KV Cache Compression

As established in Section 2, the internal Key (K) and Value (V) activations of attention heads in pretrained LLMs exhibit significant linear predictability. This finding opens a new avenue for optimizing LLM inference, particularly by addressing the memory bottleneck of the KV cache. During autoregressive decoding, LLMs generate tokens one by one. For each newly generated token, the model attends to all previously processed tokens. To avoid recomputing attention over the entire history at

each step, the Key and Value vectors computed for each token in the sequence are stored in a *KV cache* (Kwon et al., 2023; Agrawal et al., 2024; Zheng et al., 2024; Yu et al., 2022) As the sequence length grows, this cache can become very large, consuming a substantial portion of available GPU memory. This memory footprint can be a critical bottleneck, limiting the maximum batch size, context length, or even the feasibility of deploying very large models on consumer devices with limited GPU memory.

The linear predictability we observed suggests a direct strategy for KV cache compression: if the K and V states of a "target" head can be accurately reconstructed from a linear combination of K and V states from a few "reference" heads, we only need to store the KV states for the reference heads and a small set of learned projection weights. Our approach differs fundamentally from concurrent predictor+quantization methods such as AQUA-KV (Shutova et al., 2025): rather than storing quantized residuals for *all* heads after subtracting a layer-to-layer predicted baseline, we *eliminate* storage of target heads entirely, retaining full-precision states only for reference heads and reconstructing the rest on-the-fly via the linear maps learned in §2. This head-elimination strategy directly leverages the intra-layer linear structure characterized above. In the rest of this section, we outline the methodology for this strategy.

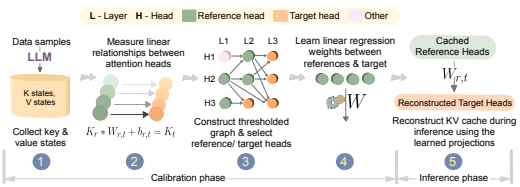


Figure 5: Pipeline for KV-cache compression via head-level redundancy. (1) We log key/value activations on a calibration set. (2) Pair-wise linear probes quantify how well one head predicts another. (3) A thresholded graph selects a minimal reference set that covers all targets. (4) Compact weight matrices $W_{r\to t}$ are trained for those links. (5) During inference, we cache only the reference heads and reconstruct the others on-the-fly, shrinking memory with minimal extra compute.

4.1 KV Cache Compression Methodology

Our compression strategy involves two phases (Figure 5). In the **calibration phase**, we collect K and V activations on a small sample (1–5% of C4 or a task-specific dataset), then partition all heads into reference heads $\mathcal{H}_{\text{ref}}$ (whose KV states will be stored explicitly) and target heads $\mathcal{H}_{\text{target}}$ (whose KV states will be predicted). Partitioning uses a directed graph $G = (V, E)$ where edge weights are pairwise R^2 scores between heads.

Algorithm 1 implements the partition: a binary search over an r^2 threshold τ prunes weak edges from G; a head is a valid prediction candidate only if it retains at least m incoming edges; the algorithm greedily selects target heads from this pool until a fraction f (e.g., $f = 0.5$ for $2\times$ compression) is covered. For each selected target, only its top-m predictors by R^2 are kept, capping each linear map's input size. We then fit multivariate OLS to learn the projection weights for each target.

Selection of Target heads to be Predicted

Input: Weighted, directed graph $G = (V, E, w)$.
Data: Fraction of target heads to predict f, minimum reference heads to be a valid target m.
Result: Target heads T, r^2 edge strength τ

low $\leftarrow 0$, **high** $\leftarrow 1$ `// binary-search bounds`
while *not converged* **do**
  `// 1. Prune the graph to significant edges`
  $\tau \leftarrow (\textbf{low} + \textbf{high})/2$
  $G^* \leftarrow$ subgraph of G with edges $\geq \tau$
  `// 2. Greedily select a set of target heads`
  $T \leftarrow \emptyset$
  Candidates $\leftarrow \{v \in V(G^*) \mid \text{indegree}(v) \geq m\}$
  while *a valid candidate can be added to* T **do**
    `/* A candidate` c `is valid if every node in` $T \cup \{c\}$ `*/`
    `/* is fed by at least` m `non-target nodes. */`
    $c \leftarrow$ Valid Candidate $\setminus T$
    $T \leftarrow T \cup \{c\}$
  end
  `// 3. Adjust binary-search window`
  if $|T| < f \cdot |V|$ **then**
    high $\leftarrow \tau$ `// too few targets`
  else
    low $\leftarrow \tau$ `// too many targets`
  end
end

In the **inference phase**, we cache only the reference heads $\mathcal{H}_{\text{ref}}$ and reconstruct $\mathcal{H}_{\text{target}}$ on-the-fly via the learned projections. Storing $|\mathcal{H}_{\text{ref}}|$ heads plus the small weight matrices $\boldsymbol{W}^*$ shrinks the overall KV cache footprint.

4.2 Experiments and Results

Next, we evaluate the efficacy of our KV cache compression method. We focus on two main aspects:

1. **Compression Efficacy**: What is the impact on downstream task accuracy when using our method to predict (e.g., 50% of) KV states, compared to baseline full-cache inference?

2. **Cross-Task Transferability**: Can linear regressors trained on one task generalize effectively to KV prediction on unseen tasks, both within and across domains?

Models and Benchmarks. We validate our KV cache compression approach on Llama3 (8B), Falcon3 (10B), and Qwen3 (32B) model families. Benchmarks include Hellaswag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), PIQA (Bisk et al., 2019), ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), MMLU Pro (Wang et al., 2024), Humaneval (Chen et al., 2021), and MATH (Lewkowycz et al., 2022). All experiments were conducted on H100-NVL GPUs with 80GB of HBM memory.

Compression vs. Accuracy. Table 1 reports the primary results, comparing baseline accuracy with our compression method implementing 2×KV cache compression (predicting 50% of heads). Across models, this approach achieves substantial memory reduction with accuracy trade-offs that vary by architecture. Qwen3-32B incurs an average accuracy drop of 5.5 pp; Falcon3-10B incurs 4.5 pp; LLaMA-3.1 8B shows a larger 9.9 pp drop (Table 1).

Transferability of Linear Relations Across Tasks. A key question for practicality is whether the learned linear projections are task-specific or capture more general inter-head relationships. Table 2 (Llama3.1 8B) shows considerable transferability *within* the same domain (e.g., GSM8K calibration generalizes well to other math tasks) and partial degradation across disparate domains (math $\rightarrow$ code, math $\rightarrow$ psychology). Cross-domain transfer is not always detrimental: Psychology calibration even improves MMLU Law accuracy from 22.25 to 24.88 and drops MMLU Health by only $\approx$ 1.5 pp, suggesting some learned relationships capture broadly applicable contextual reasoning. Per-task numbers are in Appendix B.

4.3 Disentangling Key and Value Compression

The subspace overlap analysis in §3 revealed that Key projection matrices develop far greater intra-layer alignment than Value matrices during pretraining—Key subspaces expand fastest while Value subspaces remain nearly orthogonal throughout training. This asymmetry predicts that Key states should be more amenable to linear reconstruction than Value states. We now test this prediction directly by disentangling the two streams.

Table 3 reports average accuracy across the math evaluation suite (GSM8k, MMLU-Pro Math, and the MATH subtasks) on Llama3.1 8b under three settings: Key-only prediction (half of Key heads reconstructed, Values stored normally), Value-only prediction, and joint K+V prediction (the setting evaluated in Table 1 and Table 2). A clear hierarchy emerges.

Key compression is the least disruptive: reconstructing half the Key heads drops average math accuracy by only 1.97 pp. Value compression is more harmful at 4.30 pp, and joint K+V compression is worst at 6.73 pp. Furthermore, even a projector calibrated on a semantically distant domain (Psychology) can compress Keys while retaining roughly 92% of baseline accuracy on math benchmarks; the same cross-domain projector applied to Values degrades performance far more sharply. Full per-task results for Key-only and Value-only compression are in Appendix B.

This asymmetry has a natural interpretation: because Key projection subspaces become highly aligned during pretraining, the linear maps between Key heads are well-conditioned and generalise across calibration domains. Value subspaces, remaining largely orthogonal, retain more task-specific structure that a single linear map cannot easily approximate.

5 Related Work

Our research builds upon two main areas: the interpretability of attention head redundancy and techniques for KV cache compression. Prior work in mechanistic interpretability has extensively studied attention head functions, revealing specialization and considerable redundancy (Clark et al., 2019; Voita et al., 2019; Olsson et al., 2022). This has led to efficiency methods like head pruning based on importance scores or attention map correlations (Michel et al., 2019; Agarwal et al., 2024), and ar-

Table 1: Evaluation accuracy (%) across multiple benchmarks with and without 50% KV prediction.

Model	Method	TruthfulQA	HellaSwag	MMLU Stem	GPQA	Winogrande	Average
Qwen3 32B	Baseline	39.16	82.70	82.39	40.62	73.16	63.61
	50% KV Pred	40.02	77.02	74.21	31.91	67.24	58.08
Falcon3 10B	Baseline	37.05	80.95	68.47	34.82	71.19	58.50
	50% KV Pred	35.00	74.23	61.90	27.67	71.42	54.04
Llama3.1 8B	Baseline	37.08	79.28	58.70	29.68	73.87	55.72
	50% KV Pred	30.35	66.53	44.53	26.34	61.40	45.83

Table 2: Model performance across different tasks for **50% key and value** predictions on Llama3.1 8b instruct

Evaluation Domain	Task/Metric	Baseline	Calibration Domain		
			Maths (GSM8K)	Coding (Humaneval)	Psychology (MMLU Psyc)
MATH	GSM8k	75.35	71.03	67.77	62.31
	Math Intermediate Algebra	10.74	6.97	7.41	7.64
	Math Prealgebra	41.90	36.28	32.60	34.32
	Math Precalculus	11.42	6.59	6.77	7.14
	Math Algebra	42.40	26.11	22.41	20.97
	MMLU Pro Math	38.00	33.30	30.94	30.71
	Average	**36.77**	**30.04**	**27.98**	**27.18**
CODING	Humaneval	58.35	20.12	39.63	12.80
	Average	**58.35**	**20.12**	**39.63**	**12.80**
OTHERS	MMLU Pro CS	39.02	38.78	34.63	37.07
	MMLU Pro Law	22.25	17.25	20.43	24.88
	MMLU Pro Health	47.92	36.18	37.04	46.45
	MMLU Pro Psychology	57.51	52.88	52.25	57.39
	Average	**41.68**	**36.27**	**36.09**	**41.44**

Table 3: Average accuracy across the math evaluation suite (GSM8k, MMLU-Pro Math, and the MATH subtasks) under three compression settings (50% of heads reconstructed in each case). Keys are substantially more resilient to linear approximation than Values, consistent with the asymmetric subspace overlap growth observed during pretraining (§3).

Compression setting	Avg. acc.	Δ vs. baseline
Baseline (no compression)	36.77%	—
Key-only (50% keys predicted)	34.80%	−1.97 pp
Value-only (50% values predicted)	32.47%	−4.30 pp
Joint K+V (50% of each predicted)	30.04%	−6.73 pp

chitectural modifications like GQA that share K/V projections (Ainslie et al., 2023). These approaches typically assess redundancy via downstream task impact or attention map similarity. Our work differs by focusing on a finer-grained phenomenon: the linear predictability of internal Key, Value, and Query vector activations themselves. We show that these raw activations often exhibit strong linear correlations, a distinct type of learned structure not directly captured by prior methods.

Complementary research on KV cache optimization aims to reduce its memory footprint during inference: low-rank projections (Wang et al., 2020), cross-layer factorization (XKV; Chang et al., 2025), and dictionary-based sparse coding (LEXICO; Kim et al., 2024). Most closely related to our work, AQUA-KV (Shutova et al., 2025) learns layer-wise linear predictors to improve KV-cache *quantization* by storing a quantized residual for *every* head. We instead exploit inter-head linearity to *eliminate* storing a subset of heads' KV caches entirely, using at most 5 reference heads with no quantization, and—unlike AQUA-KV—characterize the emergence and theoretical underpinnings of the underlying phenomenon.

6 Discussion and Final Remarks

We identify and characterize inter-head linear predictability as an emergent structural property of pretrained Transformers: across four model families, individual head K/V/Q activations can be reconstructed from 2–5 peer heads, a regularity absent at random initialization and developing steadily during pretraining. OLMo-2 checkpoints show intra-layer prediction strength and projection-subspace overlap growing monotonically, with Key subspaces aligning fastest—supported also by The-

orem 3.1. Eliminating storage for ∼50% of heads via learned linear maps yields 2× KV-cache reduction with model-dependent drops (5.5/4.5/9.9 pp on Qwen3/Falcon/LLaMA); Key compression is substantially more benign than Value compression, consistent with the subspace asymmetry.

Limitations

While we demonstrate the phenomenon, the exact causal mechanisms behind its emergence during pretraining are not well-understood and remain an open question. Our KV cache compression method, though effective, introduces a calibration step and a minor inference-time reconstruction latency; moreover, the degree of linear predictability and thus compression benefits can vary with the specific model, task, and data.

References

Saurabh Agarwal, Batuhan Acun, Benjamin Hosmer, Maged Elhoushi, Yuhuan Lee, Siva Venkataraman, Dimitris Papailiopoulos, and Clifford J. Wu. 2024. CHAI: Clustered head attention for efficient LLM inference. *Preprint*, arXiv:2403.08058.

Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. 2024. Taming throughput-latency tradeoff in llm inference with sarathi-serve. *Preprint*, arXiv:2403.02310.

Joshua Ainslie, Santiago Ontanon, Yi Tay, Chris Alberti, Philip Pham, Jean-Baptiste Cordonnier, Philip Wang, Xuezhi Wang, Zhen Qin, and Slav Petrov. 2023. Grouped-query attention. *Preprint*, arXiv:2305.13245.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Kaiyu Chang, Yutong He, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2025. xkv: Cross-layer key-value compression for efficient transformers. *Preprint*, arXiv:2503.18893.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. *Preprint*, arXiv:1906.04341.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Junhyuck Kim, Jongho Park, Jaewoong Cho, and Dimitris Papailiopoulos. 2024. Lexico: Extreme KV cache compression via sparse coding over universal dictionaries. *Preprint*, arXiv:2412.08890.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Kyuhong Shim, Iksoo Choi, Wonyong Sung, and Jungwook Choi. 2021. Layer-wise pruning of transformer attention heads for efficient language modeling. *Preprint*, arXiv:2110.03252.

Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, Denis Mazur, Nikita Surkov, Ivan Ermakov, and Dan Alistarh. 2025. Cache me if you must: Adaptive key-value quantization for large language models. *arXiv preprint arXiv:2501.19392*.

Roman Vershynin. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Lena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

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

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 521–538, Carlsbad, CA. USENIX Association.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. Sglang: Efficient execution of structured language model programs. *Preprint*, arXiv:2312.07104.

A Methodology Details for Linear Predictability Analysis

This appendix provides the full OLS-regression methodology summarized in §2.1.

Activation collection. For our primary analysis, we collected Key, Value, and Query activations for every head in every layer by processing 150 randomly sampled sequences (average 200 tokens each) from the C4 dataset. This provided approximately 30,000 token activation samples per model for fitting our regression models. Additional datasets like GSM8k were used for further validation.

Pairwise head predictability (Stage 1). Our initial probe assessed direct linear relationships between pairs of heads. For every ordered pair of heads (H_t, H_r) within a model (target H_t, reference H_r, with the constraint layer$(H_t) \geq$ layer(H_r) to avoid using future information), we fit an Ordinary Least Squares (OLS) model to predict the activations of H_t using those of H_r. The coefficient of determination (R^2) from these regressions quantifies the strength of this pairwise linear dependency. As visualized in Figure 1 (middle panel, showing results for Key states in Llama-3.1 8B on C4), a considerable number of head pairs exhibit non-trivial R^2 values, suggesting that many individual heads can, to some extent, be linearly predicted by at least one other single head. For instance, 75% of pairs have $R^2 > 0.5$, and the median R^2 is 0.57 for Key states.

Enhanced predictability with multiple peer heads (Stage 2). While pairwise analysis revealed individual predictive links, we further explored the potential of using multiple peer heads to enhance the predictability of a target head's activations. For each target head H_t, we selected the top N reference heads that yielded the highest R^2 scores in the pairwise analysis (these could be from the same or preceding layers). We then assessed their combined predictive strength by fitting a multivariate OLS model using their concatenated state vectors. The results, depicted in Figure 1 (middle panel, for Llama-3.1 8B Key states on C4), reveal a remarkably strong linear predictability with a small number of peer heads. While adding more reference heads generally increases the average R^2, the most significant gains are often achieved with just a few (e.g., 2–3) well-chosen peers. For instance, using $N = 3$ reference heads, a substantial fraction of target heads can already be predicted with high fidelity (a mean R^2 of 0.71 is achieved for Key states). Increasing to $N = 5$ reference heads pushes this predictability even further, with over 75% of heads becoming predictable with an $R^2 > 0.72$, and the overall mean R^2 reaching a

high of 0.76 for Key states.

B Results for Transferability of Linear Relations Across Tasks

We investigate how replacing half of the *key* (K) or *value* (V) states with linear predictions impacts downstream accuracy. In particular, we ask: *Are keys or values more resilient to approximation, and does the calibration domain of the linear projector matter?*

Tables 2, 4, and 5 disentangle three settings: joint prediction of both streams (Table 3), prediction of only *keys* (Table 4), and prediction of only *values* (Table 5). A clear hierarchy emerges: **key compression is less harmful than value compression, and the joint setting is the most disruptive**. For the maths suite, the average score declines from $36.77\,\%$ to $34.80\,\%$ (-1.97 pp) when half of the keys are reconstructed, but drops to $32.47\,\%$ (-4.30 pp) for value reconstruction and to $30.04\,\%$ (-6.73 pp) when both keys and values are predicted.

Calibrating on a matching domain helps reduce the accuracy drop, but value states remain much more fragile than keys. Math-calibrated projectors, for instance, preserve GSM8K accuracy under key compression ($74.60\,\%$ vs. $75.35\,\%$ baseline) yet still lose nearly 3 pp when compressing values. Surprisingly, even a projector trained on a seemingly unrelated domain—Psychology—can compress keys while still preserving roughly 92% of the baseline accuracy on math benchmarks; when the same cross-domain projector is applied to the values, however, performance deteriorates far more sharply.

These results suggest that keys contain more linearly redundant or similar information than values across tasks from different domains. Predicting them offers significant memory savings with only modest accuracy degradation. Values are more task-specific; compressing them—even partially—risks larger performance drops, and simultaneous K + V compression compounds these errors. These findings advocate a hybrid strategy: aggressively compress keys while treating values conservatively, or calibrating them with domain-specific regressors, to balance memory efficiency with robust task accuracy.

C Additional Results on Linear Predictability of Heads

Trends in inter-layer linear connection. In this section, we quantify how the strength and prevalence of dominant linear links among attention heads vary with their inter-layer distance. To do so, we first construct a directed graph G whose nodes correspond to heads (parsed as (layer, head) pairs) and whose edge weights are the R^2 scores from pairwise OLS regressions of key–state activations. Edges are split into "close" and "far" windows using two distance thresholds (≤ 2 vs. ≥ 3 layers, and ≤ 4 vs. ≥ 5 layers). For each proximity category, we aggregate raw edge weights and normalize them to yield the percentage contribution of that category within its window ("% Conn") and compute the corresponding mean R^2. Table 6 then reports these two metrics for both 2- and 4-layer windows, showing that nearby links contribute over 53% of each window's normalized weight and consistently exhibit higher predictability than distant links. Far-layer connections remain substantial ($\approx 46\%$) yet are consistently weaker, underscoring a strong local alignment in the attention subspace despite the presence of non-local dependencies.

Distributions of linear predictability across different tasks and models. In this section, we further demonstrate that inter-head linear dependencies persist across both downstream benchmarks and pretraining corpora. We consider two representative settings: (i) LLaMA-3.1 8B evaluated on GSM8k, and (ii) Olmo-2 7B evaluated on its own pretraining data. Figures 6 and 7 respectively plot, for each projection type (Key left, Query center, Value right), (a) the cumulative distribution function (CDF) of R^2 and (b) the kernel-density estimate (KDE) of the same, as the number of reference heads N varies from 1 to 5.

Across both models and datasets, increasing N yields a clear rightward shift in the CDFs and a corresponding movement of KDE peaks toward higher R^2, with the largest marginal gain between $N = 1$ and $N = 2$ and diminishing returns beyond $N = 4$. Finally, GSM8k exhibits slightly stronger linear alignment (median $R^2 \approx 0.85$ for Keys at $N = 5$) compared to Olmo-2's raw training data, indicating that downstream tasks can amplify intrinsic linear structure in attention head activations.

Intensity of linear relations across layers. As shown in Figure 8 for LLaMA-3.1 8B on C4, the R^2 for each attention head—when linearly pre-

Table 4: Model performance across different tasks for **50% key** predictions on Llama3.1 8b instruct

Evaluation Domain	Task/Metric	Baseline	Calibration Domain		
			Maths (GSM8K)	Coding (Humaneval)	Psychology (MMLU Psyc)
MATH	GSM8k	75.35	74.60	74.60	74.22
	Math Intermediate Algebra	10.74	9.85	11.18	9.30
	Math Prealgebra	41.90	40.64	40.18	39.83
	Math Precalculus	11.42	9.70	10.99	10.07
	Math Algebra	42.40	37.23	37.07	34.50
	MMLU Pro Math	38.00	36.78	35.16	34.86
	Average	**36.77**	**34.80**	**34.86**	**33.79**
CODING	Humaneval	58.35	51.82	45.73	46.34
	Average	**58.35**	**51.82**	**45.73**	**46.34**
OTHERS	MMLU Pro CS	39.02	38.04	37.32	36.00
	MMLU Pro Law	22.25	20.16	22.43	20.79
	MMLU Pro Health	47.92	47.31	46.33	47.31
	MMLU Pro Psychology	57.51	56.89	56.01	56.01
	Average	**41.68**	**40.60**	**40.52**	**40.00**

Table 5: Model performance across different tasks for **50% value** predictions on Llama3.1 8b instruct

Evaluation Domain	Task/Metric	Baseline	Calibration Domain		
			Maths (GSM8K)	Coding (Humaneval)	Psychology (MMLU Psyc)
MATH	Gsm8k	75.35	72.78	69.21	68.61
	Math Intermediate Algebra	10.74	9.41	9.30	8.41
	Math Prealgebra	41.90	40.64	34.90	42.13
	Math Precalculus	11.42	8.24	7.69	7.87
	Math Algebra	42.40	30.41	27.96	28.47
	MMLU Pro Math	38.00	33.38	33.75	33.67
	Average	**36.77**	**32.47**	**30.46**	**31.53**
CODING	Humaneval	58.35	42.68	49.39	35.36
	Average	**58.35**	**42.68**	**49.39**	**35.36**
OTHERS	MMLU Pro CS	39.02	36.58	36.34	36.82
	MMLU Pro Law	22.25	19.61	18.25	22.25
	MMLU Pro Health	47.92	40.46	37.16	47.67
	MMLU Pro Psychology	57.51	52.38	51.87	57.89
	Average	**41.68**	**37.25**	**35.91**	**41.16**

Table 6: Local vs. distant dominant linear connections. Boldface and arrows highlight the higher density and strength of near-layer links.

Window	Proximity	% Conn.	Mean R^2
2-layer	Near (≤ 2)	**53.6%** $\uparrow$	**0.2946** $\uparrow$
	Far (≥ 3)	46.40% $\downarrow$	0.2550 $\downarrow$
4-layer	Near (≤ 4)	**53.59%** $\uparrow$	**0.2895** $\uparrow$
	Far (≥ 5)	46.41% $\downarrow$	0.2506 $\downarrow$

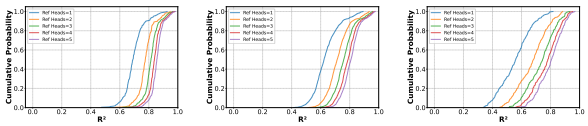
dicted from its top reference heads—varies in intensity across layers. While we observe higher linear redundancy in some early layers and lower redundancy in deeper layers for this specific model and dataset, these patterns should not be taken as universally representative. Different architectures and downstream tasks may exhibit distinct layer-wise profiles of predictability; the only consistent finding is that the density of linear predictability fluctuates across depth of the model.

D Proof of Theorem 3.1

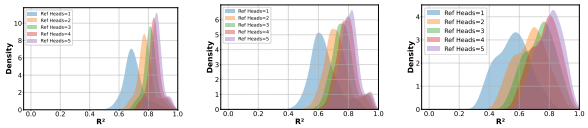
Theorem D.1 (Hanson-Wright inequality (Vershynin, 2018, Thm. 6.2.2))**.** *Let $A \in \mathbb{R}^{n\times n}$ be arbitrary matrix and let $X = (X_1, \ldots, X_n)^\top \in \mathbb{R}^n$ have independent, mean-zero, sub-Gaussian coordinates. Then, for every $t \geq 0$,*

$$\Pr\Big\{ \big| X^\top AX - \mathbb{E}\big[X^\top AX\big] \big| \geq t \Big\} \leq 2\exp\Big[-c\min\Big\{ \frac{t^2}{K^4\,\|A\|_F^2}, \frac{t}{K^2\,\|A\|_2} \Big\} \Big],$$

where $K = \max_i \|X_i\|_{\psi_2}$ and $c > 0$ is an absolute universal constant.

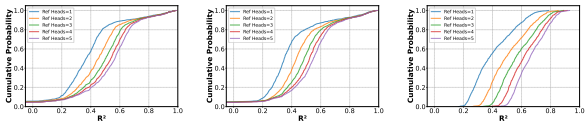


(a) Key (K) on GSM8k **(b)** Query (Q) on GSM8k **(c)** Value (V) on GSM8k

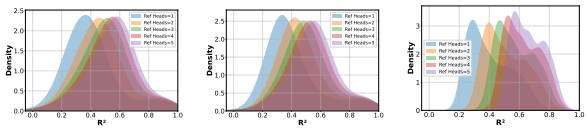


(d) Key (K) on GSM8k **(e)** Query (Q) on GSM8k **(f)** Value (V) on GSM8k

Figure 6: GSM8k – Distribution of linear predictability (R^2) for LLaMA-3.1 8B. Top row: cumulative distribution functions (CDFs); bottom row: kernel-density estimates (KDEs). Ordering (left→right): Key, Query, Value.



(a) Key (K) on Olmo Train **(b)** Query (Q) on Olmo Train **(c)** Value (V) on Olmo Train



(d) Key (K) on Olmo Train **(e)** Query (Q) on Olmo Train **(f)** Value (V) on Olmo Train

Figure 7: Olmo 2 7B – Distribution of linear predictability (R^2) on its pre-training corpus. Top row: cumulative distribution functions (CDFs). Bottom row: kernel-density estimates (KDEs). Left→right in each row: Key, Query, Value.

We will now discuss the proof for Theorem 3.1. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times k}$ have i.i.d. $\mathcal{N}(0,1)$ entries, and assume $k \le m/2$. Draw

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m),$$

independently of $\boldsymbol{A}, \boldsymbol{B}$. We will show that there exist absolute constants $c_1, c_2 > 0$ such that

$$\Pr\Big\{\inf_{\boldsymbol{C}\in\mathbb{R}^{k\times k}} \mathop{\mathbb{E}}_{\boldsymbol{x}} \big\|\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{C} - \boldsymbol{x}^\top \boldsymbol{B}\big\|_2^2 < c_1\, m\, k\Big\} \le 2\exp\big(-c_2\, m\, k\big).$$

See that, for any fixed $\boldsymbol{C} \in \mathbb{R}^{k\times k}$,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{x}} \big\|\boldsymbol{x}^\top(\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B})\big\|_2^2 \\
&= \mathrm{tr}\big((\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B})^\top(\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B})\, \mathop{\mathbb{E}}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top]\big) \\
&= \mathrm{tr}\big((\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B})^\top(\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B})\big) = \|\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B}\|_F^2.
\end{aligned}$$

Hence we have

$$\inf_{\boldsymbol{C}} \mathop{\mathbb{E}}_{\boldsymbol{x}} \|\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{C} - \boldsymbol{x}^\top \boldsymbol{B}\|_2^2 = \inf_{\boldsymbol{C}} \|\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B}\|_F^2.$$

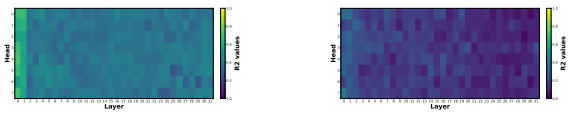


Figure 8: Heatmap of head-level linear predictability across transformer layers of Llama3.1 8B on C4. Each row corresponds to one attention head and each column to a model layer; the color intensity encodes the coefficient of determination R^2 when reconstructing that head's activations from its strongest reference heads. The gradient from warm to cool colors illustrates that linear redundancy is highest in early layers and gradually diminishes in deeper layers for Llama3.1 8B. **Left** key states, **right** value states.

.

Our goal now is to derive a lower bound for $\inf_{\boldsymbol{C}} \|\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B}\|_F^2$. We will use a standard fact that random Gaussian matrices have full column rank almost surely (with probability 1). Thus, the matrix A has full column rank and the concatenated matrix $[A, B] \in \mathbb{R}^{m\times 2k}$ has full rank as well, implying that the columns of A and B are linearly independent of each other.

The function $f(\boldsymbol{C}) = \|\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B}\|_F^2$ has a unique minimizer $\boldsymbol{C}^* = \boldsymbol{A}^\dagger \boldsymbol{B}$ given by the Moore-Penrose pseudoinverse $\boldsymbol{A}^\dagger = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top$ which exists since $\boldsymbol{A}$ has full column rank. The orthogonal projector onto its column space is given by

$$\boldsymbol{P}_{\boldsymbol{A}} = \boldsymbol{A}\,\boldsymbol{A}^\dagger, \quad \text{and let } \boldsymbol{R} = \boldsymbol{I}_m - \boldsymbol{P}_{\boldsymbol{A}}.$$

By expansion with $\boldsymbol{C} = \boldsymbol{A}^\dagger\boldsymbol{B}$, see that

$$\begin{aligned}
\inf_{\boldsymbol{C}} \|\boldsymbol{A}\boldsymbol{C} - \boldsymbol{B}\|_F^2 &= \|\boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{B} - \boldsymbol{B}\|_F^2 \\
&= \|(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{A}})\,\boldsymbol{B}\|_F^2 \\
&= \sum_{j=1}^{k} \big\|\boldsymbol{R}\,\boldsymbol{b}_j\big\|_2^2,
\end{aligned}$$

where $\boldsymbol{b}_j \in \mathbb{R}^m$ is the jth column of $\boldsymbol{B}$.

Conditional on $\boldsymbol{A}$, the matrices $\boldsymbol{P}_{\boldsymbol{A}}, \boldsymbol{R}$ are fixed symmetric matrices of rank k and $m-k$ respectively. Additionally,

$$\begin{aligned}
\boldsymbol{P}_{\boldsymbol{A}}^2 &= \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top \\
&= \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top = \boldsymbol{P}_{\boldsymbol{A}}.
\end{aligned}$$

This implies that all the eigenvalues of $\boldsymbol{P}_{\boldsymbol{A}}$ and consequently $\boldsymbol{R}$ are either 0 or 1. Since $\boldsymbol{R}$ has rank $m-k$,

$$\|\boldsymbol{R}\|_2 \le 1, \qquad \mathrm{tr}(\boldsymbol{R}) = \|\boldsymbol{R}\|_F^2 = m - k.$$

Each column $\boldsymbol{b}_j \sim N(\boldsymbol{0}, \boldsymbol{I}_m)$ is independent of $\boldsymbol{R}$, and

$$\mathbb{E}\,\|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 = \mathrm{tr}(R) = m - k.$$

By the Hanson-Wright inequality (Thm D.1) , for any $t > 0$,

$$\begin{aligned}
&\Pr\Big\{\big|\|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 - (m-k)\big| \ge t \;\Big|\; \boldsymbol{A}\Big\} \\
&\quad \le 2\exp\Big(-c\,\min\big\{\tfrac{t^2}{\|\boldsymbol{R}\|_F^2},\, \tfrac{t}{\|\boldsymbol{R}\|_2}\big\}\Big) \\
&\quad \le 2\exp\Big(-c\min\{t^2/(m-k),\, t\}\Big).
\end{aligned}$$

Setting $t = (m-k)/2$ and using $m - k \ge m/2$, we get

$$\Pr\big\{\|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 < \tfrac{1}{2}(m-k)\big|\boldsymbol{A}\big\} \le 2\exp\big(-c_3\, m\big).$$

Because the $\boldsymbol{b}_j$'s are independent, a union bound over $j = 1, \ldots, k$ yields

$$\Pr\Big\{\exists\, j:\ \|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 < \tfrac{1}{2}(m-k)\Big\} \le k{\cdot}2e^{-c_3 m} \le 2e^{-c_4\, m},$$

where in the last inequality we used $k \le m/2$ and $ne^{-cn} = e^{-O(n)}$.

Thus, with probability $\ge\ 1 - 2e^{-cm}$, each $\|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 \ge \frac{1}{2}(m-k)$, so

$$\inf_{\boldsymbol{C}} \|\boldsymbol{A}\boldsymbol{C}-\boldsymbol{B}\|_F^2 = \sum_{j=1}^{k} \|\boldsymbol{R}\,\boldsymbol{b}_j\|_2^2 \ge k{\cdot}\tfrac{1}{2}(m-k) \ge \frac{1}{4}mk,$$

where the last inequality uses $k \le m/2$ (so $m - k \ge m/2$).

E Resources Used

Compute hardware. All experiments were executed on NVIDIA **H100-NVL** GPUs.

- **GPU model & memory** : H100-NVL (80 GB HBM).
- **GPU count per run** : 4.
- **Interconnect** : NVLink-2 / NVSwitch / PCIe 5.0.
- **CPU host** : dual-socket *AMD EPYC 9334* (2 × 32 cores, 64 threads total).
- **System RAM** : **756 GB**.
- **Storage**: **1 TB**.

Software stack. Ubuntu 22.04.5 LTS, Linux kernel **5.15.0-135-generic**; **CUDA 12.6** (nvcc 12.6.68, build Wed Aug 14 2024); cuDNN **9.1.0**; PyTorch (Paszke et al., 2019) **2.4.0**; Python **3.12.3**.

Compute budget. Each full run consumed ≈ 80 GPU-hours (4 × H100-NVL × *N* hours).