# MAPPING ENTROPY: A DISTRIBUTIONAL INFORMATION-THEORETIC PROBE FOR LLM MANIFOLD STRUCTURE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We propose mapping entropy, a distributional, information-theoretic probe for comparing LLM representations. Each model's manifold is partitioned by a Vector Quantization (VQ) bottleneck; aligning two partitions via a shared probing corpus yields per-code conditional entropies whose distribution exposes where representations align and diverge. Scalar similarity metrics average away this per-code structure.

Experiments across Qwen3-14B (sixteen depths), the Qwen3 family (0.6B–30B), and Pythia-6.9B yield three findings. First, per-code entropy distributions are highly heterogeneous: for distant layer pairs, 69% of codes exceed 6 bits, yet a zero-entropy shared base persists. Shuffled-token, frequency, and partitioner controls rule out artifactual explanations. Second, synthetic calibration shows Miller-Madow bias is substantial but systematic, and multi-seed validation establishes that VQ training noise dominates corpus sampling variance. Finally, mean conditional entropy predicts discrete-partition transfer accuracy ($r{=}-0.96$), whereas CKA does not ($r{=}0.02$) and Procrustes shows only a weak effect ($r{=}0.46$). A continuous-space transfer task gives the complementary picture: CKA predicts it ($r{=}0.77$), mapping entropy does not ($r{=}0.25$).

## 1 INTRODUCTION

Manifold probes like Sparse Autoencoders (SAEs) (Bricken et al., 2023; Huben et al., 2024) are intra-model: they cannot compare representations across models or layers. Cross-model probes like CKA (Kornblith et al., 2019) and SVCCA (Morcos et al., 2018; Raghu et al., 2017) are symmetric and scalar: they return one number that averages over per-code heterogeneity. A single pair may simultaneously contain zero-entropy codes (near-perfect alignment) and 9+ bit codes (severe divergence).

We introduce **mapping entropy**: each model's manifold is partitioned by a VQ bottleneck (Van den Oord et al., 2017) trained on frozen representations. Passing a shared probing corpus through two VQ bottlenecks yields a co-occurrence matrix, from which we derive per-code conditional entropies. The *distribution* of these entropies, rather than their mean alone, is the core diagnostic object.

Our contributions:

- A distributional diagnostic (§4–§6). Per-code entropy exposes heterogeneity that CKA's scalar average flattens: a 0-bit shared base coexists with 8+ bit blind spots. Controls rule out frequency, position, partitioner, and corpus-statistic artifacts, and the trajectory replicates across 6 models in 2 architecture families.
- Estimator calibration and error bars (§3.3–§3.4). Synthetic calibration at the paper's sampling ratio quantifies Miller-Madow bias; it is substantial ($-0.8$ to $-2.5$ bits) but systematic, so relative comparisons hold. NSB (Nemenman et al., 2002) cuts the bias but inflates variance. Bootstrap CIs show corpus sampling variance is negligible next to VQ training variance, and multi-seed $t$-tests with conservative $n{=}3$ per group still yield Cohen's $d > 5$ for cross-arch gaps (§5).
- Discrete-partition transfer prediction (§7). Across 50 model-depth pairs, mean conditional entropy predicts transfer accuracy at $r{=}-0.96$, stable under leave-one-out CV. CKA on continuous hidden



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

states shows no correlation. Under the same (Fano-consistent) definition of conditional entropy, $k$-means partitions also predict transfer ($r{=}-0.93$), consistent with Fano's inequality applying to any partitioner; VQ's advantage over generic clustering appears instead in per-code distributional heterogeneity (Table 6), where VQ's cross-layer maps are consistently sharper than $k$-means', rather than in the scalar transfer correlation.

## 2 Related Work

**Intra-model probes.** Sparse Autoencoders (SAEs) disentangle superposition within a single model but cannot compare representations across models. Crosscoders (Lindsey et al., 2024) extend SAEs to jointly analyze features across layers, but remain intra-model and require expensive joint training. Mapping entropy provides cross-model comparison without joint training, using independent VQ bottlenecks aligned via a shared corpus. The tuned lens (Belrose et al., 2023) probes intermediate layers by training a probe per layer; it is model-internal and does not compare across models.

**Cross-model similarity.** CKA and SVCCA produce symmetric scalar similarities between representation spaces. SVCCA applies CCA after dimensionality reduction, trading spatial resolution for robustness. Neither captures per-code heterogeneity: both return a single number that averages over the distribution of alignments. Mapping entropy retains the full per-code distribution, so one can see which codes align and which diverge. We compare against CKA, SVCCA, and Procrustes alignment (Schönemann, 1966) (§4.1), and also evaluate continuous-space transfer to check whether the metrics capture different aspects of alignment (§7). A survey of representational alignment (Sucholutsky et al., 2023) covers additional metrics (e.g., RSA); mapping entropy differs by operating on discrete partitions rather than continuous geometry.

**Information-theoretic foundations.** The Variation of Information (Meilă, 2007) decomposes into $H(A|B) + H(B|A)$; we verify the identity $H(A|B) - H(B|A) = H(A) - H(B)$ holds exactly for plug-in estimators on the same co-occurrence matrix (§3.3). The Miller-Madow correction (Miller, 1955; Paninski, 2003) eliminates arbitrary frequency thresholds. Fano's inequality (Cover & Thomas, 2006) relates conditional entropy to classification accuracy; we discuss this relationship in §7.

**Discrete representation learning.** VQ-VAE introduced vector quantization for representation learning; prior work (Roy et al., 2018; Kaiser et al., 2018) explored VQ for sequence models. Our application of VQ as a cross-model alignment layer appears to be novel: we train independent bottlenecks and align them via co-occurrence, without joint training.

**Model stitching and alignment.** Model stitching (Bansal et al., 2021) tests whether representations transfer by training linear projections between layers. The Platonic Representation Hypothesis (Huh et al., 2024) posits convergence of representations across models; our cross-architecture observations (§5) give a distributional view of where convergence holds and where it breaks.

## 3 Mapping Entropy: A Distributional Manifold Probe

Our framework has two stages: (1) train VQ bottlenecks that partition each model's manifold into a discrete codebook, and (2) align two partitions via a shared probing corpus and compute per-code conditional entropies.

### 3.1 VQ as a Rate-Distortion Manifold Partitioner

For each model/layer, we train a VQ bottleneck on frozen representations. Hidden states $h \in \mathbb{R}^{d_{\text{model}}}$ are projected to $z_e \in \mathbb{R}^{d_{vq}}$ ($d_{vq}{=}256$ by default) and quantized against a codebook $\mathcal{C} = \{e_1, \ldots, e_K\}$ ($K{=}16{,}384$) via cosine similarity. The codebook is $k$-means initialized and EMA-updated (decay $0.95$). The VQ reconstruction error exhibits an inverted-U profile across depth: shallow and final layers are easy ($\sim$0.08), mid-depth layers harder ($\sim$0.25). During training, per-batch code utilization at the final step is around 43–88%, varying with model and depth. Details in Appendix A.

### 3.2 Co-occurrence and Conditional Entropy

We pass a shared probing dataset $\mathcal{D}$ ($N \approx 32$M tokens, subsampled from a 10B-token corpus; see Appendix B) through both frozen VQ bottlenecks. Both models share a tokenizer, guaranteeing token-position alignment. The co-occurrence matrix $M \in \mathbb{R}^{K \times K}$ yields per-code conditional entropies $H(B|A{=}i)$ and $H(A|B{=}j)$. We write $A \to B$ to denote that $A$ is the anchor and $B$ is the target; the reported quantity is $H(B|A)$.

### 3.3 Miller-Madow Bias Correction and Estimator Calibration

We apply the Miller-Madow correction $H_{\text{MM}} = H_{\text{plug}} + (K_i{-}1)/(2n_i)$, where $K_i$ is the number of distinct targets observed for code $i$ and $n_i$ is its count. This eliminates any frequency threshold: all observed codes (count $\geq 1$) are included. The frequency-weighted mean $\sum_i P(A{=}i) \cdot H_{\text{MM}}(B|A{=}i)$ gives the conditional entropy estimate. We verify correctness with the identity $H(A|B) - H(B|A) = H(A) - H(B)$, which holds exactly for plug-in estimators on the same co-occurrence matrix.

To quantify estimator bias under the paper's sampling regime, we construct synthetic co-occurrence matrices with known entropy from a diagonal-plus-uniform mixture (target $B{=}A$ with probability $p_{\text{diag}}$, else $B{\sim}\text{Uniform}(K)$), varying $p_{\text{diag}}$ to sweep true entropy from 2.6 to 12.6 bits. For each condition we sample $n{=}K{\times}2000$ observations (50 trials) and compare each estimator against the known true entropy. At this ratio ($n/K{\approx}2000$, $K{=}16{,}384$), Miller-Madow retains substantial negative bias (Table 1): MM underestimates true entropy by 0.77–2.47 bits, while NSB reduces bias to 0.04–0.39 bits. However, because (1) the bias is systematic (always negative), so *relative* comparisons between pairs remain valid; (2) all reported findings rely on relative differences or distributions, not absolute entropy values; and (3) NSB has $\sim$10$\times$ higher variance per code, we retain Miller-Madow as the primary estimator. Bootstrap confidence intervals on real data (100 resamples, 80% subsampling) show corpus sampling variance is negligible ($\sigma < 0.001$ bits), one to two orders of magnitude smaller than VQ training variance (§3.4).

Table 1: Estimator bias at $n/K{=}2000$ ($K{=}16{,}384$). Miller-Madow retains substantial negative bias; NSB reduces it by 2–50$\times$.

| True H | Plug-in bias | MM bias | NSB bias |
|---|---|---|---|
| 2.59 | $-0.837$ | $-0.766$ | $-0.389$ |
| 4.86 | $-1.393$ | $-1.253$ | $-0.330$ |
| 6.98 | $-1.846$ | $-1.638$ | $-0.166$ |
| 8.98 | $-2.234$ | $-1.958$ | $-0.039$ |
| 10.86 | $-2.575$ | $-2.233$ | $-0.041$ |
| 12.59 | $-2.879$ | $-2.472$ | $-0.249$ |

### 3.4 Reference Baselines

To establish a reference for genuine representational drift, we define an adjacent-depth baseline by computing the mapping entropy across adjacent layers in our depth grid for Qwen3-14B (15 steps, e.g., L2$\to$L3, L3$\to$L5). This baseline averages $H = 3.44$ bits, and all 15 transitions (even the lowest, L2$\to$L3 at 2.47 bits) exceed the VQ noise floor reported below, confirming that it captures true inter-layer drift rather than training noise.

To isolate VQ training noise, we train VQ on the same layer with different random seeds (3 seeds $\times$ 3 depths $\times$ 3 models) and compute VI between same-layer partitions. The floor averages $H{=}2.19$ bits (SD$\approx$0.01–0.03); Table 2 shows 14B alone (2.30 bits). The cross-arch per-pair SD ($\sim$0.007–0.016 bits) is comparable to the floor SD, indicating VQ training noise is the dominant variance source. The adjacent-depth baseline (3.44 bits) is 1.6$\times$ the expanded floor (2.19 bits), and even its tightest step (2.47 bits) stays above the floor, placing the whole baseline above the noise level. Full tables in Appendix C.

Table 2: VQ-stochasticity floor (same-layer, different seeds, Qwen3-14B). Mean $H$=2.30 bits. JS= Jensen-Shannon divergence between per-code entropy spectra.

| Layer | Mean $H$ | JS | Frac>6b |
|---|---|---|---|
| L2 | 1.75 | 0.001 | 0.020 |
| L20 | 2.81 | 0.003 | 0.037 |
| L38 | 2.35 | 0.002 | 0.008 |

## 4 Distributional Resolution and Scalar Metric Comparison

The per-code entropy distribution spans from 0 bits (shared statistical base) to 9+ bits (severe divergence). For L2→L39, 69% of codes exceed 6 bits on the observed scale, while zero-entropy codes persist (Figure 1). These are observed values; Miller-Madow underestimates true entropy by 0.8–2.5 bits (Table 1), so the true distribution is shifted higher but its shape is preserved. The shuffled-token control below verifies this shape is genuine.

To test whether the zero-entropy "shared base" reflects genuine alignment rather than corpus statistics (e.g., position-invariant function tokens), we shuffle the target model's token order (destroying position alignment) and recompute co-occurrence. Under shuffling, mean $H(B|A)$ increases from 4.4–6.7 bits (aligned) to 11.5–11.6 bits (shuffled), a 1.7–2.7× increase (Table 15 in Appendix H). The fraction of codes exceeding 6 bits jumps from 0.22–0.69 (aligned) to 1.00 (shuffled): every active code becomes high-entropy when alignment is destroyed. The zero-entropy codes that persist under alignment vanish entirely. We take this as evidence that the shared base reflects genuine representational alignment, not corpus statistics.

We characterize low-entropy ($H \leq 1$) and high-entropy ($H \geq 6$) codes across four pairs (Table 3). Cross-arch pairs use depth-matched layers; d$X$ denotes $X\%$ of model depth (e.g., d50=50%). Low-entropy codes consistently exist (42–1,282 codes, 0.2–7.2% coverage). High-entropy codes dominate for distant pairs: L2→L39 has 65% coverage at $H \geq 6$, vs. 24% for L2→L10. Low- and high-entropy codes have similar frequencies ($\sim 5\text{–}7 \times 10^{-5}$) and approximately uniform position distributions, suggesting entropy level is not simply a proxy for token frequency or sequence position.

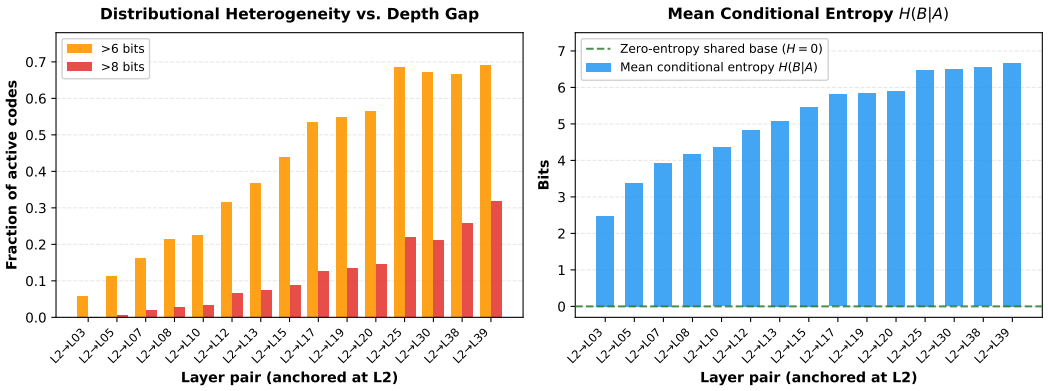


Figure 1: Distributional heterogeneity grows with depth gap in Qwen3-14B. Left: Fraction of active codes exceeding 6 bits (orange) and 8 bits (red), anchored at L2 and paired with progressively later layers. Right: Mean conditional entropy $H(B|A)$ for the same pairs. The dashed green line marks the zero-entropy shared base that persists across all depth gaps. Entropy grows from ∼2.5 bits (nearest sampled depth) to 6.67 bits (L2→L39), while the fraction of high-entropy codes grows from near-zero to 69%.

### 4.1 Comparison with Scalar Similarity Metrics

We compute three scalar similarity metrics on depth-matched hidden states: linear CKA, SVCCA, and Procrustes alignment. Table 4 shows the results for the key 8B↔30B and 14B↔30B pairs.

Table 3: Code characterization: low-entropy ($H \leq 1$) vs. high-entropy ($H \geq 6$) codes.

| | Low-H ($\leq$1 bit) | | High-H ($\geq$6 bits) | |
|---|---|---|---|---|
| **Pair** | **Codes** | **Coverage** | **Codes** | **Coverage** |
| L2→L39 (intra, max gap) | 42 | 0.2% | 10,573 | 64.9% |
| L2→L10 (intra, small gap) | 571 | 3.9% | 3,424 | 23.5% |
| 8B→30B d50 (cross-arch) | 1,282 | 7.2% | 4,230 | 28.8% |
| 14B→30B d75 (cross-arch) | 725 | 4.2% | 5,930 | 39.6% |

SVCCA and Procrustes are computed on a 50K-token subsample, owing to memory constraints; CKA is computed on the full probing corpus.

Table 4: Scalar similarity metrics for cross-architecture pairs at depth-matched layers. Higher=more similar. SVCCA and Procrustes are computed on a 50K-token subsample; CKA on the full probing corpus.

| | CKA | | SVCCA | | Procrustes | |
|---|---|---|---|---|---|---|
| **Depth** | 8B | 14B | 8B | 14B | 8B | 14B |
| d05 | 0.531 | 0.429 | 0.623 | 0.707 | 0.597 | 0.666 |
| d25 | 0.402 | 0.445 | 0.695 | 0.697 | 0.586 | 0.615 |
| d50 | 0.577 | 0.579 | 0.709 | 0.714 | 0.595 | 0.616 |
| d75 | 0.630 | 0.678 | 0.681 | 0.676 | 0.562 | 0.580 |
| d95 | 0.835 | 0.685 | 0.787 | 0.690 | 0.669 | 0.655 |

At d05 and d95, CKA detects the non-monotonicity (14B less similar than 8B), consistent with mapping entropy. At d25 and d75, CKA disagrees with mapping entropy: it shows 14B more similar to 30B than 8B, while mapping entropy shows the opposite. At d50 the two metrics agree and both place 14B closer to 30B than 8B. The two anchors also differ in topology from the target (8B/14B Dense vs. 30B MoE); we return to this in §5.

## 5 Depth-Wise Trajectory and Cross-Architecture Observations

We train VQ at 16 depths in Qwen3-14B Dense (Qwen Team, 2025) across 120 ordered pairs (Figure 2). Mapping entropy increases monotonically with depth gap, ranging from $\approx 2.5$ bits (L2→L3) to 6.67 bits (L2→L39, $1.94\times$ the baseline in §3.4). Crucially, code-level distributions reveal that early-to-mid pairs undergo codebook expansion (negative marginal entropy gap; target uses more codes), while late-to-final pairs exhibit codebook contraction (positive gap; vocabulary consolidation). This expansion-then-contraction pattern generalizes across the Qwen3 suite (0.6B–30B, Dense and MoE) and Pythia-6.9B (Biderman et al., 2023) (Figure 3, Appendix D).

Table 5 shows cross-architecture results with multi-seed error bars. The 14B Dense shows higher friction against 30B MoE than 8B at d05 ($d$=9.5) and d95 ($d$=14.9). Notably, the d50 result *reverses* sign ($d$=$-5.4$): the 14B appears *less* divergent than 8B at mid-depth, contrary to the d05/d95 pattern ($p<10^{-3}$, conservative $t$-test). This sign reversal shows that VQ-based probes can produce depth-dependent effects, and motivates multi-seed validation. d25 and d75 use a single seed, so Table 5 reports their $\Delta$ in bits only; no effect size or test is available at those depths. We report these as observations: the differences may reflect training data or optimization, not solely topology.

We recompute the d05 friction (14B→30B − 8B→30B) across 3 perturbed configurations: $K$=8192 (halved codebook), $d_{vq}$=128 (halved dimension), and $d_{vq}$=512 (doubled dimension). The friction gap ($\Delta$) is consistently positive across all configurations including default (mean $+0.21 \pm 0.02$ bits, range $[+0.17, +0.23]$). Seed variance is $\pm$0.01–0.02 bits while configuration variance is $\pm$0.05 bits; the multi-seed mean friction at d05 ($\Delta = +0.14$) is smaller than any single-seed perturbation, indicating it is the most conservative estimate.

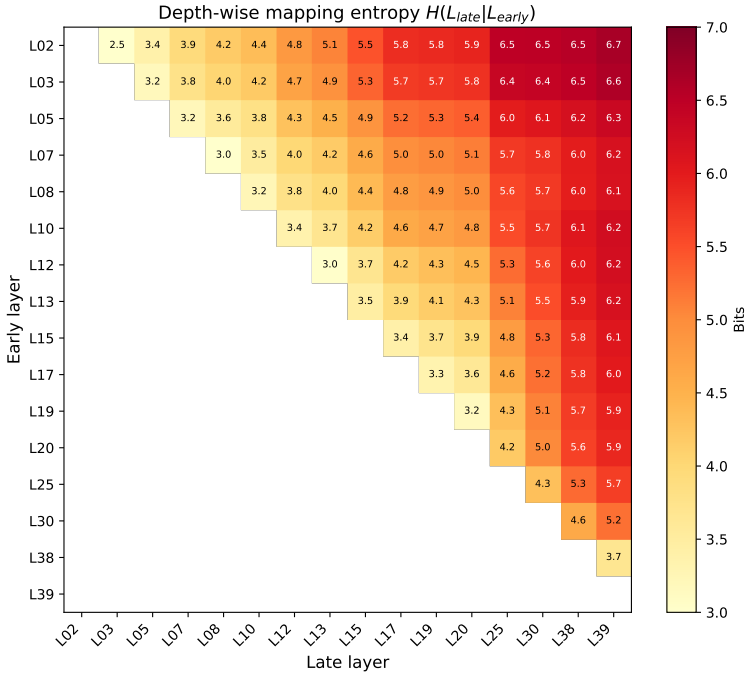


Figure 2: Depth-wise mapping entropy heatmap for Qwen3-14B. Each cell shows the mean conditional entropy $H(L_{\text{late}} \mid L_{\text{early}})$ in bits for all 120 ordered layer pairs across 16 depths. Entropy grows monotonically with depth gap, from $\sim$2.5 bits (L2$\rightarrow$L3) to 6.67 bits (L2$\rightarrow$L39).

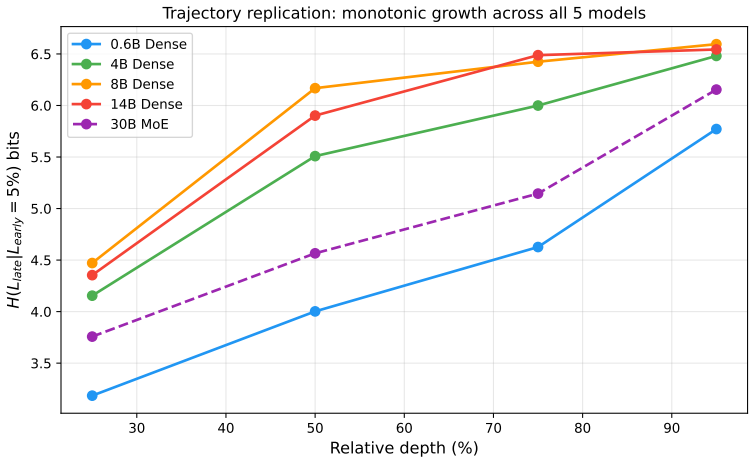


Figure 3: Trajectory replication across the Qwen3 family. Each line shows the mean conditional entropy $H(L_{\text{late}} \mid L_{\text{early}})$ for one model, where the early layer is fixed at 5% relative depth (d05) and the late layer varies from d25 to d95. All five Qwen3 models (0.6B–30B, Dense and MoE) exhibit monotonic entropy growth, despite differences in parameter count and architecture. The 30B MoE (dashed) shows slightly lower entropy at shallow depths but converges to the same range by d95.

Table 5: Cross-architecture friction. Entries are mean conditional entropy $H(B|A)$ (bits). Depths d05/d50/d95 show mean $\pm$ SD across 9 anchor$\times$target seed pairs; d25/d75 use a single seed. $\Delta$ is the mean difference (14B$-$8B). Cohen's $d$ and significance tests rely on seed replicates and are computed only at d05/d50/d95 ($^{**}p < 10^{-3}$, conservative $t$-test, $n = 3$ anchor-averaged replicates, $\mathrm{df} = 4$). Values of $d$ are pooled over 9 seed pairs; conservative $n = 3$ effect sizes are larger ($+16.9$, $-9.5$, $+37.9$).

| **Pair** | **d05** | **d25** | **d50** | **d75** | **d95** |
|---|---|---|---|---|---|
| 8B $\rightarrow$ 30B | 2.98$\pm$.02 | 3.47 | 4.47$\pm$.01 | 4.74 | 3.56$\pm$.01 |
| 14B $\rightarrow$ 30B | 3.12$\pm$.01 | 3.60 | 4.42$\pm$.01 | 5.01 | 3.71$\pm$.01 |
| $\Delta$ (bits) | $+0.139$ | $+0.127$ | $-0.057$ | $+0.273$ | $+0.157$ |
| Cohen's $d$ | $+9.5^{**}$ | — | $-5.4^{**}$ | — | $+14.9^{**}$ |

## 6 Partitioner Baseline: VQ vs. $k$-means vs. Random

A key question is whether the heterogeneity observed in §4 depends on VQ's learned codebook, or whether any discretization would reveal the same structure. We compare three partitioners on the same projected representations and the same depths: VQ (learned codebook), $k$-means (MiniBatchKMeans, $K$=16,384), and random (uniform assignment). The $k$-means centroids are placed using a random 200K-token subsample, which defines the partition but contributes no observations of its own; every statistic we report is then computed from labels assigned to all $\approx$ 32M tokens, exactly as for VQ and random. Each anchor code therefore receives $\approx$ 1,978 observations, giving a plug-in per-code entropy ceiling of $\log_2(1{,}978) \approx 10.95$ bits; the Miller-Madow term adds up to $\approx$0.5 bits at this sampling ratio, so the attainable maximum for our estimator is $\approx$11.4 bits, which is where the random baseline sits. Codebook utilization is consistently high: VQ and $k$-means maintain around 16K active codes, while random uses all 16,384 by construction.

Table 6: Partitioner baseline: all partitioners assign $\approx$ 32M tokens ($K$=16,384). $H$ is weighted conditional entropy $H(B|A)$ in bits (↓); MI $= H(B) - H(B|A)$ in bits; Frac>6b is the fraction of active anchor codes with entropy $> 6$ bits; NMI $= 1 - \mathrm{VI}/H(A, B)$ (↑). The random partitioner is constant across all pairs ($H = 11.30$, MI $= 2.70$, Frac>6b $= 1.000$, NMI $= 0.107$), acting as a maximum-diffusion *upper* reference; its residual MI is a finite-sample artifact (16,384 $\times$ 16,384 table, 0.12 tokens/cell) cleared by both learned methods. Since marginal entropy differs between $k$-means and VQ, NMI serves as the scale-invariant metric.

| | **VQ** | | | | **$k$-means** | | | |
|---|---|---|---|---|---|---|---|---|
| **Pair** | $H$ | MI | Frac>6b | NMI | $H$ | MI | Frac>6b | NMI |
| L2→L10 | 4.35 | 9.47 | 0.224 | 0.528 | 5.65 | 6.16 | 0.212 | 0.338 |
| L2→L20 | 5.90 | 7.98 | 0.565 | 0.410 | 6.26 | 5.36 | 0.334 | 0.284 |
| L2→L39 | 6.67 | 7.02 | 0.692 | 0.347 | 7.01 | 5.59 | 0.434 | 0.285 |
| L10→L20 | 4.82 | 9.06 | 0.315 | 0.486 | 6.03 | 5.59 | 0.119 | 0.313 |
| L10→L39 | 6.25 | 7.44 | 0.578 | 0.371 | 7.51 | 5.09 | 0.252 | 0.264 |
| L20→L39 | 5.87 | 7.82 | 0.518 | 0.396 | 7.25 | 5.36 | 0.146 | 0.284 |

Across all six layer pairs, VQ consistently outperforms $k$-means on every alignment metric: weighted $H$ is lower (by 0.34–1.37 bits), while NMI (by 0.062–0.190) and MI (by 1.43–3.47 bits) are higher. Rather than measuring alignment directly, Frac>6b reflects code-level divergence exposure. Diagnostically, it faithfully tracks the mean for VQ, showing perfect rank-correlation with weighted $H$ ($\rho = +1.00$), whereas $k$-means shows little correlation ($\rho = +0.26$). For example, $k$-means assigns L10→L20 a *higher* weighted $H$ than L2→L10 (6.03 vs. 5.65 bits) despite a *lower* Frac>6b (0.119 vs. 0.212). This reveals that $k$-means concentrates divergence into a few diffuse cells rather than across codes, causing its unweighted per-code mean to collapse as a transfer predictor (Appendix G). Unlike generic clustering, learned codebooks both improve alignment and expose fine-grained divergence at the code level.

Entropy growth is of the same order for both partitioners, but not equal: from L10 the slopes match ($+1.48$ bits for $k$-means vs. $+1.43$ for VQ), while from L2 VQ grows $1.7\times$ faster ($+2.32$ vs. $+1.36$). Trajectory slope therefore separates the two partitioners only weakly and inconsistently; absolute level separates them on all six pairs, with VQ uniformly lower in conditional entropy.

## 7 Discrete-Partition Transfer Prediction

We test whether mapping entropy predicts discrete-partition transfer accuracy, and whether CKA does.

For each of 50 model-depth pairs (10 cross-architecture pairs $\times$ 5 depths), we compute mapping entropy features, CKA similarity, SVCCA, Procrustes, and a transfer score. CKA is computed on the full $\sim$32M-token probing set; SVCCA and Procrustes are computed on a 50K-token subsample. The transfer score uses five random 80/20 train/test splits: on each training split, we build a maximum-likelihood mapping from anchor codes to target codes; on the held-out split, we predict target codes from anchor codes and measure accuracy.

Table 7 shows that mean $H$ strongly predicts transfer accuracy ($r{=}-0.96$, $p{<}10^{-27}$), while CKA does not ($r{=}0.02$). Leave-one-model-pair-out CV (10 folds) is stable: mean $H$ achieves $r{=}-0.96{\pm}0.005$ across folds, versus $r{=}0.01{\pm}0.08$ for CKA.

None of the three scalar metrics predicts discrete-partition transfer at $n{=}50$: CKA yields $r{=}0.02$ ($p{=}0.91$), SVCCA $r{=}0.06$ ($p{=}0.68$), and Procrustes $r{=}0.46$ ($p{=}0.001$). The first two are established nulls given the detection floor of $|r|{\approx}0.28$ at $\alpha{=}0.05$; Procrustes is significant but weak.

To check whether the CKA null on discrete transfer is simply a space mismatch, we also evaluate a continuous-space transfer task: linear probe transfer (logistic regression on anchor hidden states for next-token prediction, evaluated on target states aligned via least squares; 5 seeds, 50K tokens, 2 pairs $\times$ 5 depths). CKA predicts continuous transfer at $r{=}0.77$ ($p{=}0.009$), but mapping entropy does not ($r{=}0.25$, $p{=}0.49$). This $n{=}10$ comparison is underpowered for the mapping entropy null ($|r|{\approx}0.63$ required), but CKA's effect is strong enough to detect.

Table 7: Correlation with discrete-partition transfer accuracy ($n{=}50$).

| **Feature** | **Pearson** $r$ | $p$**-value** | **Spearman** $\rho$ | $n$ |
|---|---|---|---|---|
| Mean $H$ (unweighted) | $-0.960$ | $< 10^{-27}$ | $-0.962$ | 50 |
| Frac >6 bits | $-0.844$ | $< 10^{-13}$ | $-0.864$ | 50 |
| Skewness | $+0.837$ | $< 10^{-13}$ | $+0.876$ | 50 |
| Kurtosis | $+0.755$ | $< 10^{-9}$ | $+0.787$ | 50 |
| Std $H$ | $-0.688$ | $< 10^{-7}$ | $-0.576$ | 50 |
| **CKA** | $+0.016$ | 0.91 | $-0.126$ | 50 |

The entropy-transfer correlation is partly expected from Fano's inequality, linking conditional entropy to prediction error. However, partial correlations reveal that this predictive power resides entirely in the scalar mean: after controlling for mean $H$, higher moments lose significance (skewness $p{=}0.22$, kurtosis $p{=}0.07$, frac>6b $p{=}0.52$; Appendix F). Although the per-code distribution does not improve transfer prediction over the scalar mean, it uniquely provides code-level resolution on which features align or diverge.

To test whether VQ's learned codebook is necessary, we evaluate $k$-means partitions on the same 50 cross-architecture pairs. Under the frequency-weighted mean conditional entropy (§3.3), $k$-means predicts transfer nearly as strongly as VQ ($r{=}-0.93$ vs. $r{=}-0.96$, $p{<}10^{-22}$; Table 14). Because Fano's inequality applies universally, the scalar entropy-transfer correlation cannot distinguish learned from generic discretizations. Instead, VQ's true advantage lies in its per-code distributional structure (§6, Table 6) and interpretable semantic content (§4, Appendix I), which generic clustering fails to recover despite matching scalar predictive performance.

## 8 LIMITATIONS

- Corpus composition variance is not measured.
- 14B-vs-30B differences are not causally isolated from training data, optimization, or active-parameter regime.
- Miller-Madow retains substantial negative bias ($-0.8$ to $-2.5$ bits); relative comparisons hold, but absolute bit values are underestimated.
- VQ codes are partition cells, not semantically interpretable units; systematic token-level semantic analysis remains future work.

## 9 CONCLUSION

We proposed mapping entropy, a distributional, information-theoretic probe that exposes per-code entropy heterogeneity (0-bit shared base coexisting with 8+ bit blind spots) at a resolution scalar metrics cannot provide. Controls rule out partitioner, frequency, and corpus-statistic artifacts, and the trajectory replicates across 6 models in 2 architecture families. A calibration methodology quantifies estimator bias and variance sources. Mean conditional entropy predicts discrete-partition transfer accuracy while CKA does not; CKA instead predicts continuous-space transfer, so the two metrics capture complementary aspects of alignment. A $k$-means crossover shows the scalar transfer correlation itself is a Fano artifacts common to any partitioner; VQ's contribution is in the per-code distribution it exposes.

## 10 AI USE STATEMENT

We used generative AI tools for the following tasks. We used an LLM to classify the semantic content of VQ code activation contexts (Appendix I); all classifications were manually reviewed against the source contexts. We also used LLMs to assist with proofreading and editing the manuscript, and to assist in writing and debugging experimental code. All AI-assisted code was verified through execution and validation against known sanity checks. We take responsibility for the final content of this work.

## 11 REPRODUCIBILITY STATEMENT

Pseudocode for the main algorithms and detailed hyperparameters are provided in Appendix A. Probing corpus construction, including per-source sampling factors and token counts, is described in Appendix B. Estimator calibration methodology (Miller-Madow bias correction, NSB comparison, and bootstrap confidence intervals) is detailed in §3.3, with calibration results in Table 1. Multi-seed validation protocols and conservative statistical tests are documented in §3.4 and Appendix E. The discrete-partition transfer prediction experiment uses five random 80/20 train/test splits with leave-one-model-pair-out stability checks (Appendix F). Reference baselines (adjacent-depth and VQ-stochasticity floor) are reported in Appendix C, and trajectory replication across six models in two architecture families is shown in Appendix D.

## REFERENCES

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 225–236. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/01ded4259d101feb739b06c399e9cd9c-Paper.pdf.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2390–2399. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kaiser18a.html.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Crosscoders for cross-layer features. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

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

Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. doi: 10.1016/j.jmva.2006.11.013.

G. Miller. Note on the bias of information estimates. *Information theory in psychology : Problems and methods*, 1955.

Sewon Min, Suchin Gururangan, Eric Wallace, Weijia Shi, Hannaneh Hajishirzi, Noah Smith, and Luke Zettlemoyer. Silo language models: Isolating legal risk in a nonparametric datastore. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 25132–25159, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6cad062f78120689e8a678b8af9c6d92-Paper-Conference.pdf.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a7a3d70c6d17a73140918996d03c014f-Paper.pdf.

Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *International Conference on High Performance Computing, Networking, Storage and Analysis*, 2020.

Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. Theory and experiments on vector quantized autoencoders, 2018. URL https://arxiv.org/abs/1805.11063.

Peter H Schönemann. A generalized solution to the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Christopher J Cueva, Erin Grant, Iris Groen, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023.

Aaron Van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Phil Wang. vector-quantize-pytorch. https://github.com/lucidrains/vector-quantize-pytorch, 2020.

## A Algorithm Summary and Hyperparameters

This section presents the pseudocode for the main algorithms and summarizes the hyperparameters. Specifically: Algorithm 1 trains a VQ bottleneck for each (model, layer) pair. Algorithm 2 aligns two trained bottlenecks using the shared probing corpus to compute the per-code conditional entropy distribution. Table 8 lists major hyperparameter values.

**Algorithm 1** VQ bottleneck training. Run once per (model, layer, seed).

**Require:** frozen base model $f$, layer index $\ell$, training corpus $\mathcal{D}_{\text{train}}$, codebook size $K$, VQ dimension $d_{vq}$, steps $T$, cosine-loss weight $\lambda_{\cos}$, EMA decay $\gamma$, seed $s$
**Ensure:** down-projection $W_{\text{down}}$, codebook $\mathcal{C} = \{e_1, \ldots, e_K\}$, decoder $g$
seed all RNGs with $s$; freeze every parameter of $f$; discard layers above $\ell$
$W_{\text{down}} \leftarrow \textsc{OrthogonalInit}(d_{\text{model}} \to d_{vq})$
$\mathcal{C} \leftarrow \textsc{KMeansInit}(\{W_{\text{down}}\, f_\ell(x)\},\, K)$
zero-initialize the down-projection of every decoder block
**for** $t = 1$ **to** $T$ **do**
  $h \leftarrow f_\ell(x_t)$ for batch $x_t \sim \mathcal{D}_{\text{train}}$
  $z_e \leftarrow W_{\text{down}}\, h$
  $z_q,\ i,\ \mathcal{L}_{\text{commit}} \leftarrow \textsc{Quantize}(z_e, \mathcal{C})$
  $\hat{h} \leftarrow g(z_q)$
  $\mathcal{L} \leftarrow \textsc{SmoothL1}(\hat{h}, h) + \lambda_{\cos}\big(1 - \cos(\hat{h}, h)\big) + \mathcal{L}_{\text{commit}}$
  update $W_{\text{down}}, g$ with AdamW on $\nabla\mathcal{L}$; advance the cosine LR schedule
  update $\mathcal{C}$ by EMA with decay $\gamma$
**end for**
**return** $W_{\text{down}}, \mathcal{C}, g$

**Algorithm 2** Mapping entropy between two trained partitions.

**Require:** anchor bottleneck $A$, target bottleneck $B$, shared probing corpus $\mathcal{D}$ of $N$ tokens, codebook size $K$

**Ensure:** per-code entropies $\{H_{\text{MM}}(B|A{=}i)\}$, weighted mean $H(B|A)$, distributional summaries

$c^A \leftarrow [\,\textsc{CodeIndex}_A(t)\,]_{t\in\mathcal{D}}$; $\quad c^B \leftarrow [\,\textsc{CodeIndex}_B(t)\,]_{t\in\mathcal{D}}$

$M \leftarrow \mathbf{0}^{K\times K}$; $\quad$ **for** each token $t$ **do** $M[c_t^A,\, c_t^B] \mathrel{+}= 1$

$n_i \leftarrow \sum_j M[i,j]$; $\quad \mathcal{A} \leftarrow \{i : n_i > 0\}$

**for** $i \in \mathcal{A}$ **do**

  $p_{j|i} \leftarrow M[i,j] \,/\, n_i$; $\quad H_{\text{plug}}(B|A{=}i) \leftarrow -\sum_j p_{j|i} \log_2 p_{j|i}$

  $K_i \leftarrow |\{j : M[i,j] > 0\}|$

  $H_{\text{MM}}(B|A{=}i) \leftarrow H_{\text{plug}}(B|A{=}i) + (K_i - 1) \,/\, (2n_i)$

**end for**

$H(B|A) \leftarrow \sum_{i\in\mathcal{A}} (n_i/N)\; H_{\text{MM}}(B|A{=}i)$

repeat the loop with the roles of $A$ and $B$ exchanged (columns of $M$) to obtain $H_{\text{MM}}(A|B{=}j)$ and $H(A|B)$

**return** $\{H_{\text{MM}}(B|A{=}i)\}_{i\in\mathcal{A}}$

Table 8: Hyperparameters of the VQ bottleneck (Alg. 1). Values are the defaults used for every reported experiment unless a section states otherwise; parenthesized alternatives are the perturbation configurations of §5.

| Group | Hyperparameter | Symbol | Value |
|---|---|---|---|
| *Architecture* | | | |
| | Projection dimension | $d_{vq}$ | 256 (perturbed: 128, 512) |
| | Down-projection init | — | orthogonal |
| | Codebook size | $K$ | 16,384 (perturbed: 8192) |
| | Code assignment | — | cosine similarity |
| | Codebook init | — | $k$-means, 10 iterations |
| | Codebook update / EMA decay | $\gamma$ | EMA, 0.95 |
| | Commitment weight | — | 1.0 |
| | Decoder | — | 2 SwiGLU blocks, pre-LayerNorm, residual |
| | Decoder hidden / intermediate | — | $d_{\text{model}}$ / $2d_{\text{model}}$ |
| | Decoder down-projection init | — | zeros |
| | Output normalization | — | none |
| *Optimization* | | | |
| | Optimizer | — | AdamW |
| | Peak / minimum learning rate | — | $2{\times}10^{-4}$ / $1{\times}10^{-5}$ |
| | LR schedule, warmup | — | cosine with min-LR, 200 steps |
| | Update steps | $T$ | 2000 |
| | Effective batch size | — | 1024 sequences |
| | Max sequence length | — | 1024 tokens |
| | Cosine-loss weight | $\lambda_{\cos}$ | 10.0 |

### A.1 VQ Implementation Notes

The quantizer uses `vector-quantize-pytorch` (Wang, 2020); optimization uses AdamW (Loshchilov & Hutter, 2019) under DeepSpeed ZeRO-1 (Rajbhandari et al., 2020); and the frozen base model runs FlashAttention-2 (Dao, 2024) in bfloat16. Only the bottleneck (down-projection, codebook, decoder) is trained; all base-model parameters remain frozen throughout.

## B Probing Corpus Construction

The probing corpus is a 10B-token subset of the Open License Corpus (Min et al., 2024). Documents are tokenized and packed into 1025-token sequences using sentence-aware chunking (scanning backward for sentence boundaries to avoid truncating mid-sentence). A two-stage sampling mechanism controls the domain mixture: high-volume sources are downsampled with factor $p < 1.0$ (each document retained with probability $p$), and high-quality sources are upsampled with factor $f > 1.0$ ($\lfloor f \rfloor$ copies plus one additional with probability $f - \lfloor f \rfloor$). After sampling, the corpus is shuffled. Table 9 shows the sampling factors and final token counts per source.

Table 9: Corpus composition: sampling factors and final token counts per source.

| Source | Factor | Tokens (after sampling) |
|---|---|---|
| sw_github | 0.48 | 2,115,614,053 |
| ccby_s2orc | 0.335 | 2,302,326,294 |
| ccby_wikipedia | 0.34 | 1,099,858,221 |
| ccby_stackexchange | 0.60 | 1,001,642,267 |
| pd_books | 5.50 | 1,406,233,151 |
| sw_dm_math | 4.00 | 1,588,270,980 |
| ccby_stackoverflow | 2.00 | 166,386,808 |
| sw_amps_math | 4.00 | 4,743,040 |
| pd_law | 0.03 | 78,047,686 |
| ccby_law | 0.10 | 42,098,390 |
| sw_ubuntu_irc | 0.20 | 19,678,106 |
| pd_s2orc | 1.00 | 68,343,722 |
| sw_hackernews | 1.00 | 91,927,070 |
| pd_news | 1.00 | 17,644,380 |
| ccby_wikinews | 1.00 | 758,483 |
| **Total** | | **10,003,572,651** |

The corpus is tokenized with the Qwen3 tokenizer (shared across all Qwen3 models, guaranteeing token-position alignment). For Pythia experiments, we decode the Qwen3-tokenized corpus back to text and re-tokenize with the Pythia tokenizer (without special tokens, truncated to 1024 tokens). This preserves text-level alignment across model families, though token-level positions differ due to tokenizer mismatch. From the 10B-token corpus, we subsample a probing set of $N \approx 32$M tokens, which is shared across all experiments except for those explicitly noted.

## C Reference Baselines

Table 10 details the adjacent-depth baseline across our 16-depth grid in Qwen3-14B. Since the grid is non-uniformly spaced, these 15 transitions span 1 to 8 layers, averaging $H = 3.44$ bits to quantify inter-depth drift. Minimal JS divergence across all steps ($< 0.03$) indicates preserved representational structure between nearby layers. Furthermore, the 14B VQ-stochasticity floor (Table 2, mean $H = 2.30$ bits) is strictly lower than even the tightest adjacent step (L2→L3 at 2.47 bits), demonstrating that training stochasticity does not confound true representational drift.

Table 10: Adjacent-depth baseline: mapping entropy between neighboring layers in Qwen3-14B. Mean $H$=3.44 bits.

| Pair | Mean $H$ | Frac >6b | Frac >8b | JS |
|---|---|---|---|---|
| L2→L3 (1) | 2.47 | 0.059 | 0.001 | 0.005 |
| L3→L5 (2) | 3.19 | 0.107 | 0.003 | 0.028 |
| L5→L7 (2) | 3.22 | 0.088 | 0.004 | 0.006 |
| L7→L8 (1) | 2.99 | 0.074 | 0.001 | 0.006 |
| L8→L10 (2) | 3.24 | 0.083 | 0.007 | 0.003 |
| L10→L12 (2) | 3.36 | 0.092 | 0.004 | 0.004 |
| L12→L13 (1) | 3.03 | 0.052 | 0.000 | 0.003 |
| L13→L15 (2) | 3.46 | 0.073 | 0.001 | 0.013 |
| L15→L17 (2) | 3.39 | 0.073 | 0.001 | 0.006 |
| L17→L19 (2) | 3.34 | 0.058 | 0.001 | 0.005 |
| L19→L20 (1) | 3.15 | 0.058 | 0.001 | 0.001 |
| L20→L25 (5) | 4.19 | 0.166 | 0.007 | 0.005 |
| L25→L30 (5) | 4.32 | 0.202 | 0.012 | 0.004 |
| L30→L38 (8) | 4.62 | 0.263 | 0.023 | 0.004 |
| L38→L39 (1) | 3.68 | 0.090 | 0.004 | 0.008 |

## D TRAJECTORY REPLICATION

Table 11 shows the trajectory replication across the Qwen3 family (0.6B–30B, Dense and MoE) and Pythia-6.9B. All models exhibit: (1) monotonic entropy growth from d05 to d95, (2) early codebook expansion (negative marginal entropy gap), (3) late codebook contraction, and (4) heterogeneity (per-code entropy distribution spanning 0 to 8+ bits). The replication spans 2 architecture families (Qwen3 and Pythia) and covers both Dense and MoE topologies, supporting the generality of the trajectory pattern.

Table 11: Trajectory replication across the Qwen3 family and Pythia-6.9B. All models show monotonic growth, early expansion, and late contraction.

| Model | H(d05→d95) | Growth | Early exp. | Late contr. | Hetero. |
|---|---|---|---|---|---|
| 0.6B Dense | 3.18 → 5.77 | 2.59 | ✓ | ✓ | ✓ |
| 4B Dense | 4.15 → 6.48 | 2.33 | ✓ | ✓ | ✓ |
| 8B Dense | 4.47 → 6.59 | 2.12 | ✓ | ✓ | ✓ |
| 14B Dense | 4.35 → 6.54 | 2.19 | ✓ | ✓ | ∼ |
| 30B MoE | 3.76 → 6.15 | 2.40 | ✓ | ✓ | ✓ |
| Pythia-6.9B | 4.37 → 6.24 | 1.87 | ✓ | ✓ | ✓ |

## E MULTI-SEED STATISTICS

To verify that the cross-arch friction observed in §5 ($\Delta$ at d05/d95) is not an artifact of VQ training stochasticity, Table 12 compares 8B→30B and 14B→30B across multiple seeds. Because the 9 seed pairs per cell are non-independent (each seed appears in 3 pairs), we report conservative $t$-tests ($n$=3 per group, $df$=4). Cohen's $d$ exceeds 5 at all depths, so seed variance is negligible relative to the cross-arch gap. A VQ hyperparameter perturbation across 3 configurations shows the gap is consistently positive ($\Delta = +0.21 \pm 0.02$ bits).

At d95, the conservative $t$ (46.4) exceeds the naive $t$ (31.5). This occurs because the conservative procedure averages the 3 partner-seed values per anchor seed before computing the SD, which strips within-group variance (the denominator) faster than it reduces the mean difference (the numerator). This is a known property of averaging correlated observations: when within-group correlation is high, the averaged SD shrinks more than the effect size, inflating $t$. The conservative test is there-

fore conservative with respect to Type I error (it correctly accounts for non-independence) but can produce larger $t$-values than the naive test when within-group correlation is high.

Table 12: Pairwise comparisons (8B→30B vs 14B→30B). $\Delta$ = mean difference (14B−8B). Naive $t$ assumes 9 independent pairs; conservative $t$ uses $n$=3 per group.

| **Depth** | **$\Delta$** | **Cohen's $d$** | **$t$ (naive)** | **$t$ (conserv.)** | **$p$ (conserv.)** |
|---|---|---|---|---|---|
| d05 | $+0.139$ | $+9.5$ | 20.2 | 20.7 | $3\times10^{-5}$ |
| d50 | $-0.057$ | $-5.4$ | $-11.5$ | $-11.7$ | $3\times10^{-4}$ |
| d95 | $+0.157$ | $+14.9$ | 31.5 | 46.4 | $1\times10^{-6}$ |

## F Transfer Prediction Details

To test whether distribution shape carries predictive power beyond the scalar mean, we compute partial correlations after controlling for mean $H$. After controlling for mean $H$, the distributional shape features lose significance: skewness partial $r$=−0.18 ($p$=0.22), kurtosis partial $r$=−0.26 ($p$=0.07), std $H$ partial $r$=−0.09 ($p$=0.51), frac>6b partial $r$=−0.09 ($p$=0.52). The predictive signal is thus carried primarily by the mean entropy, not by higher moments of the distribution.

To check that the $r$=−0.96 correlation is not driven by a single outlier pair, we perform leave-one-model-pair-out CV, holding out each of the 10 model pairs in turn (10 folds, 45 points per fold). Table 13 shows the resulting stability: mean $H$ is highly stable; CKA fluctuates around zero.

Table 13: Leave-one-model-pair-out CV (10 folds). Mean $H$ is stable; CKA fluctuates around zero.

| **Feature** | **Mean $r$** | **Std $r$** | **Min $r$** | **Max $r$** |
|---|---|---|---|---|
| Mean $H$ | $-0.960$ | 0.005 | $-0.967$ | $-0.952$ |
| Skewness | $+0.838$ | 0.012 | $+0.816$ | $+0.860$ |
| Kurtosis | $+0.756$ | 0.020 | $+0.731$ | $+0.791$ |
| CKA | $+0.009$ | 0.076 | $-0.206$ | $+0.064$ |

## G Partitioner Crossover Details

To test whether the entropy–transfer correlation (§7) depends on VQ's learned codebook or holds for any discretization, we re-run the transfer prediction experiment with $k$-means partitions ($K$=16,384) on the same projected representations across 50 cross-architecture pairs. Table 14 reports the correlation between mean $H$ and transfer accuracy under two definitions of "mean $H$": (i) the frequency-weighted conditional entropy $\sum_i P(A{=}i)\, H(B|A{=}i)$ defined in §3.3 (Fano-consistent), and (ii) the unweighted per-code mean $(1/K_{\text{active}}) \sum_i H(B|A{=}i)$, which we report for transparency.

Under the weighted definition (upper block), $k$-means predicts transfer at $r$=−0.93, matching VQ ($r$=−0.96). This parity is expected given Fano's inequality.

Under the unweighted definition (lower block), the $k$-means correlation falls to $r$=+0.04. This divergence stems from heavily skewed cluster occupancy in $k$-means (entropy of 12.44 bits vs. 13.65 bits for VQ, out of 14.00), where ∼41% of clusters have zero entropy due to token sparsity. VQ's near-uniform usage keeps both definitions aligned.

The takeaway from this appendix is therefore not that VQ is uniquely required for transfer correlation, but rather that the scalar mean-$H$ / transfer correlation is an artifact of Fano's inequality that any reasonable partitioner will exhibit. The true distinction between learned (VQ) and generic ($k$-means) discretizations manifests not in Table 14, but in Table 6 (where VQ aligns better than $k$-means on weighted $H$, mutual information, and NMI across all 6 layer pairs, and where only VQ's Frac>6b tracks its mean), Table 3 (interpretable code entropy profiles), and Appendix I (semantic content of learned codes).

Table 14: Partitioner crossover: correlation of mean $H$ with transfer accuracy for two partitioners ($n$=50 cross-arch pairs), under both the weighted (Fano-consistent, primary) and unweighted (per-code, secondary) definitions of "mean $H$". VQ and $k$-means agree closely under the weighted definition; $k$-means only appears uncorrelated under the unweighted definition, whose skewed cluster occupancy makes it an unreliable estimand.

| **Definition** | **Partitioner** | **Pearson** $r$ | $p$**-value** | **Spearman** $\rho$ | $n$ |
|---|---|---|---|---|---|
| Weighted (primary) | VQ | $-0.958$ | $< 10^{-26}$ | $-0.952$ | 50 |
| Weighted (primary) | $k$-means | $-0.932$ | $< 10^{-22}$ | $-0.915$ | 50 |
| Unweighted (secondary) | VQ | $-0.960$ | $< 10^{-27}$ | $-0.962$ | 50 |
| Unweighted (secondary) | $k$-means | $+0.038$ | 0.79 | $+0.095$ | 50 |

## H Shuffled-Token Baseline

To test whether the zero-entropy "shared base" (§4) reflects genuine representational alignment or simply corpus statistics (e.g., position-invariant function tokens), we shuffle the target model's token order (destroying position alignment) while preserving marginal code frequencies, and recompute co-occurrence. Table 15 shows the results (5 shuffles, mean±std). Under shuffling, mean $H(B|A)$ increases 1.7–2.7× and all active codes exceed 6 bits (fraction >6b = 1.00), while the zero-entropy codes that persist under alignment vanish entirely.

Table 15: Shuffled-token baseline (5 shuffles). Aligned vs. shuffled $H(B|A)$ and fraction >6 bits.

| | **Aligned** | | **Shuffled** | | |
|---|---|---|---|---|---|
| **Pair** | **Mean** $H$ | **Frac >6b** | **Mean** $H$ | **Frac >6b** | **Ratio** |
| L2→L10 | 4.35 | 0.224 | 11.58±.00 | 1.000 | 2.7× |
| L2→L20 | 5.90 | 0.565 | 11.60±.00 | 1.000 | 2.0× |
| L2→L39 | 6.67 | 0.692 | 11.55±.00 | 1.000 | 1.7× |

## I Semantic Content of VQ Codes

To check whether the per-code entropy distribution reflects interpretable structure (not just statistical noise), we analyze the semantic content of the lowest- and highest-entropy codes. For each of 10 low-$H$ and 10 high-$H$ codes per pair, we sample 20 token positions where the code activates, extract 10-token context windows (5 random resamples per code), and use an LLM to classify what the contexts have in common.

Table 16 shows representative examples. Low-$H$ codes activate on specific, structured content: mathematical expressions, algebraic equations, XML/translation markup, DOI prefixes, and Wikipedia category links. These represent narrow syntactic patterns that both models process identically, forming the "shared statistical base." High-$H$ codes activate on diverse, noisy content: multilingual text, mixed code/prose, corrupted fragments, and URLs. These are genuine representational blind spots where models diverge.

The classification diversity (number of unique categories across 5 resamples) is consistently lower for low-$H$ codes than for high-$H$ codes (cross-arch: 2.70±0.78 vs. 3.70±1.10; intra-model: 2.40±0.92 vs. 3.50±1.02; pooled $n$=20 per group, Mann-Whitney $p$=0.001, Cohen's $d$=1.08), supporting that low-$H$ codes represent more stable, narrow patterns.

High-$H$ codes could reflect genuine cross-model divergence, or simply intrinsically heterogeneous input content (e.g., multilingual text) that both models struggle with. To distinguish these, we compare per-code cross-model entropy against per-code within-model entropy (VI between same-layer partitions trained with different seeds, where cross-model divergence is zero by construction). Across both pairs, the correlation is near zero (L2→L10: $r$=0.009, $p$=0.30; L2→L39: $r$=0.013, $p$=0.13; $n$=14,206 codes per pair). Within-model entropy stays flat at ≈1.72 bits across all cross-

Table 16: Semantic content of representative VQ codes (cross-arch 8B→30B d50). Low-$H$ codes fire on structured patterns; high-$H$ codes on diverse, noisy content.

| Code | $H$ | LLM Classification | Example Context |
|---|---|---|---|
| *Low-entropy (shared base)* | | | |
| 12790 | 0.01 | Wikipedia category links | `English male television actors` |
| 9491 | 0.01 | Academic DOI prefixes | `0.1016/j.diagmicrobio.` |
| 12959 | 0.01 | Arithmetic division | `8134 divided by -1 18` |
| 16080 | 0.01 | Polynomial expressions | `5 + 67/60*t**6` |
| 9847 | 0.01 | XML/Translation markup | `<location line="+102"/>` |
| *High-entropy (blind spots)* | | | |
| 585 | 9.75 | Multilingual text | `partijen. Relaties In een rel` |
| 6365 | 9.75 | URLs and file paths | `.org/display/DOCS/Admin+UIs` |
| 8924 | 9.82 | Mixed code and text | `public function optionsProvider()` |
| 7573 | 10.08 | Corrupted text | `not following correct procedure?` |
| 9048 | 10.52 | Mixed text and code | `<source>%n hour(s)</source>` |

model entropy bins, so high cross-model $H$ reflects genuine representational divergence rather than input difficulty.