DPMI: A Principled Index for Neural Polysemanticity via Dirichlet Process Mixture Modeling

Anonymous Authors[1]

Abstract

Polysemanticity, where a single neuron responds to multiple unrelated concepts, is a central obstacle in mechanistic interpretability, yet the field lacks a principled continuous scalar for it. We introduce the **Dirichlet Process Polysemanticity Index** (DPMI), a per-neuron score that combines inferred component count and component separation by fitting a non-parametric DPGMM and weighting by mean pairwise Jensen–Shannon divergence. On a controlled toy benchmark, DPMI achieves Spearman $\rho = 0.755$ and AUROC $= 0.877$, outperforming seven baselines. Against an independent Fourier-analytic ground truth from a modular-arithmetic transformer, DPMI remains significant with $\rho = 0.255$ ($p < 10^{-8}$). Across six architectures, we find a robust cross-modal law: language models are significantly more polysemantic than vision models ($d = 0.803$, $p < 10^{-129}$). Ablations show the non-parametric prior is essential (removing it drops ρ by 0.040–0.045), and DPMI-guided SAE budget allocation improves reconstruction R^2 for the most polysemantic quartile by $+0.010$ at fixed compute.

1. Introduction

Modern deep neural networks achieve remarkable performance across vision, language, and multi-modal tasks, yet their internal representations remain poorly understood. A striking empirical regularity has emerged: individual neurons frequently activate for multiple, conceptually distinct input patterns (Elhage et al., 2022; Bricken et al., 2023; Templeton et al., 2024). This *polysemanticity* is widely believed to arise from *superposition*, the tendency of networks operating under representation compression to encode more distinct features than they have neurons, exploiting the near-orthogonality of high-dimensional space (Elhage et al., 2022).

Polysemanticity has profound consequences for interpretability: if a neuron does not cleanly represent one concept, then interventions, ablations, and probing studies that treat neurons as atomic feature detectors may be fundamentally misleading (Conmy et al., 2023; Wang et al., 2023). Sparse autoencoders (SAEs) have been proposed as a decomposition tool precisely to reverse superposition (Bricken et al., 2023; Cunningham et al., 2024), but their effective use requires knowing *which* neurons are most polysemantic and *how much*, information that is not provided by any existing principled scalar metric.

The measurement gap. Despite the importance of polysemanticity, the community has not converged on a rigorous, continuous scalar measure. Existing approaches fall into three categories, each with significant limitations:

- **Binary or ordinal labels** from manual annotation or feature visualization do not scale and cannot capture gradations (Bricken et al., 2023).
- **Activation statistics** such as Wasserstein distance to a Gaussian (W_1), kurtosis, or Hartigan's dip test are sensitive to confounds like activation sparsity and scale, and have poor rank correlation with ground truth polysemanticity (Section 3).
- **Entropy-based scores** such as Component-weighted Prediction Entropy (CPE) conflate the number of active components with their overlap, and do not exploit the geometry of the activation distribution.

Our contributions. We address this gap with four main contributions:

1. **DPMI definition.** We propose $\mathrm{DPMI}_j = T_j \cdot \bar{D}_{\mathrm{JSD}}$, a scalar that independently quantifies component *count* (via non-parametric Bayesian inference) and component *separation* (via Jensen–Shannon divergence), and prove it satisfies a set of natural axioms (Section 3).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. **Calibrated inference.** We derive a data-driven procedure for selecting the DPGMM concentration hyperparameter α^* from labeled superposition data, avoiding the over-segmentation problem that afflicts naive DPGMM usage (Section 3.5).

3. **Multi-source validation.** We validate DPMI against (i) controlled toy superposition with known K_{true}, and (ii) a mechanistic circuit ground truth from Fourier analysis of a modular arithmetic transformer, two completely independent validation paradigms (Section 4).

4. **Downstream application.** We demonstrate that DPMI scores improve SAE training efficiency by guiding compute budget allocation toward neurons that most benefit from additional capacity (Section 4.5).

2. Related Work

Polysemanticity and superposition. Elhage et al. (2022) introduced the toy model of superposition and established the theoretical connection between feature compression and polysemanticity. Bricken et al. (2023) and Templeton et al. (2024) showed empirically that polysemanticity is ubiquitous in transformer LMs. Cunningham et al. (2024) proposed sparse autoencoders as a decomposition tool, later scaled by Gao et al. (2024). Park et al. (2023) formalize the linear representation hypothesis, providing the theoretical substrate for polysemanticity as feature superposition.

Measuring feature structure. Raghu et al. (2017) and Kornblith et al. (2019) measure representation similarity across layers but not per-neuron polysemanticity. Ansuini et al. (2019) measure intrinsic dimensionality; we include their TwoNN estimator as a baseline and show it provides complementary but inferior signal to DPMI. Olah et al. (2020) and Goh et al. (2021) provide qualitative per-neuron evidence without a quantitative measure. DPGMMs have been applied to neural spike sorting (Foroozmehr et al., 2022) and latent cluster analyses (Fan et al., 2021); we are the first to apply per-neuron DPGMM to 1D activation distributions.

Concurrent and circuit methods. Gupta & Kumar (2025) propose a null-calibrated polysemanticity index with causal validation; their approach is complementary—DPMI is unsupervised and intervention-free, suited for large-scale profiling. Nanda et al. (2023) study Fourier features in modular arithmetic transformers, providing our circuit ground truth. Conmy et al. (2023) and Hanna et al. (2023) develop circuit-tracing; DPMI provides complementary per-neuron measurement.

3. The DPMI Metric

3.1. Problem Setup and Notation

Let $f : \mathcal{X} \to \mathbb{R}^d$ be a neural network layer mapping inputs to a d-dimensional activation vector. For each neuron $j \in \{1, \ldots, d\}$ and a corpus of inputs $\mathcal{D} = \{x_1, \ldots, x_N\}$, let $\mathbf{a}_j = (f(x_1)_j, \ldots, f(x_N)_j) \in \mathbb{R}^N$ be the empirical activation distribution of neuron j. We seek a scalar $\text{DPMI}_j \in [0, 1]$ that measures the degree to which neuron j responds to multiple distinct input patterns.

Informal desiderata. A good polysemanticity measure should satisfy:

(D1) **Monotonicity in component count**: DPMI_j should increase as the number of distinct activation modes increases.

(D2) **Sensitivity to separation**: two components that completely overlap (identical distributions) should contribute less than well-separated components.

(D3) **Boundedness**: $\text{DPMI}_j \in [0, 1]$ for interpretability and cross-neuron comparability.

(D4) **Architecture-agnosticism**: DPMI_j should not depend on assumptions about the network architecture or the input distribution beyond the empirical activations.

(D5) **Continuity**: DPMI_j should vary smoothly with changes in the activation distribution.

3.2. DPGMM-based Component Inference

The first step in computing DPMI is inferring the number and parameters of activation components. We fit a **Dirichlet Process Gaussian Mixture Model (DPGMM)** with truncation $T = 15$ to the 1D activation vector $\mathbf{a}_j$:

$$\begin{aligned} a_{ji} \mid z_i, \boldsymbol{\mu}, \boldsymbol{\sigma} &\sim \mathcal{N}(\mu_{z_i}, \sigma_{z_i}^2), \\ z_i \mid \boldsymbol{\pi} &\sim \text{Cat}(\boldsymbol{\pi}), \\ \boldsymbol{\pi} &\sim \text{GEM}(\alpha), \end{aligned} \tag{1}$$

where $\text{GEM}(\alpha)$ denotes the stick-breaking process with concentration α (Sethuraman, 1994). We use mean-field variational inference as implemented in scikit-learn (Pedregosa et al., 2011), which scales to the required per-neuron 1D fits in $O(TN)$ per iteration.

The number of *active* components is:

$$N_j^* = \left|\{k : \hat{\pi}_k > \tau\}\right|, \tag{2}$$

where $\hat{\pi}_k$ are the inferred mixture weights and $\tau = 0.05$ is a threshold that discards spurious near-zero components. Sensitivity to τ is characterized in Appendix H.

Why DPGMM? The non-parametric prior avoids the model selection problem that plagues fixed-k GMMs: the number of components adapts to the data. As we show in Section 4.4, replacing the DP prior with BIC-penalized GMM or fixed-$k = 3$ k-means degrades Spearman ρ by 0.040–0.045, confirming that the DP prior is a load-bearing component of the metric.

3.3. Jensen–Shannon Component Separation

Given N_j^* active components with parameters $\{(\hat{\mu}_k, \hat{\sigma}_k^2)\}_{k=1}^{N_j^*}$, we quantify their pairwise separation using the **Jensen–Shannon divergence** (JSD) (Lin, 1991):

$$D_{\mathrm{JS}}(P_k \| P_\ell) = \tfrac{1}{2}\,\mathrm{KL}(P_k \| M_{k\ell}) + \tfrac{1}{2}\,\mathrm{KL}(P_\ell \| M_{k\ell}), \quad M_{k\ell} = \tfrac{1}{2}(P_k + P_\ell), \tag{3}$$

where $P_k = \mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$ and $D_{\mathrm{JS}} \in [0, \log 2]$. We normalize to $[0, 1]$ and average over all $\binom{N_j^*}{2}$ pairs:

$$\bar{D}_{\mathrm{JSD},j} = \frac{1}{\log 2} \cdot \frac{2}{N_j^*(N_j^* - 1)} \sum_{k<\ell} D_{\mathrm{JS}}(P_k \| P_\ell). \tag{4}$$

We estimate $D_{\mathrm{JS}}(P_k \| P_\ell)$ using Monte Carlo sampling with separate sample sets for each KL direction (see Algorithm 1), which avoids the systematic under-estimation that arises from mixing samples (Ghassami et al., 2018).

Why JSD? JSD is symmetric, bounded, and well-defined even when the two distributions have non-overlapping support. In contrast, W_1 Wasserstein distance conflates separation with scale, and KL divergence is asymmetric and unbounded.

3.4. The DPMI Formula

We combine the component count term and the separation term multiplicatively:

$$\boxed{\mathrm{DPMI}_j = T_j \cdot \bar{D}_{\mathrm{JSD},j}} \qquad \text{where} \qquad T_j = \frac{N_j^* - 1}{N_j^*}. \tag{5}$$

Interpretation. $T_j \in [0, 1)$ is the *polysemanticity fraction*: it is 0 when there is only one active component and approaches 1 as $N^* \to \infty$. Specifically, $T_j = 0$ for $N^* = 1$ (monosemantic), $T_j = 1/2$ for $N^* = 2$ (bisemantic), and $T_j = 2/3$ for $N^* = 3$, etc. $\bar{D}_{\mathrm{JSD},j}$ controls the magnitude: a neuron with $N^* = 3$ perfectly separated components achieves $\mathrm{DPMI}_j = 1 \cdot 2/3 = 2/3$, while one with $N^* = 3$ components that all but overlap achieves $\mathrm{DPMI}_j \approx 0$.

Proposition 1 (Boundary conditions)**.** $\mathrm{DPMI}_j = 0$ if and only if $N_j^* = 1$ (monosemantic neuron). $\mathrm{DPMI}_j \to \bar{D}_{\mathrm{JSD}}$ as $N^* \to \infty$. $\mathrm{DPMI}_j \in [0, 1]$ for all valid inputs.

Algorithm 1 Monte Carlo JSD Estimator for 1D Gaussians

Require: Parameters (μ_1, σ_1^2), (μ_2, σ_2^2); $n = 5000$ samples

$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)^{n/2}$; $\quad X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)^{n/2}$

$M(x) \leftarrow \frac{1}{2}\phi(x; \mu_1, \sigma_1^2) + \frac{1}{2}\phi(x; \mu_2, \sigma_2^2)$

$\widehat{\mathrm{KL}}(P_1 \| M) \leftarrow \frac{1}{n/2} \sum_{x \in X_1} \log \frac{\phi(x; \mu_1, \sigma_1^2)}{M(x) + \epsilon}$

$\widehat{\mathrm{KL}}(P_2 \| M) \leftarrow \frac{1}{n/2} \sum_{x \in X_2} \log \frac{\phi(x; \mu_2, \sigma_2^2)}{M(x) + \epsilon}$

return $\max\left(0,\ \frac{1}{2}\widehat{\mathrm{KL}}(P_1 \| M) + \frac{1}{2}\widehat{\mathrm{KL}}(P_2 \| M)\right) / \log 2$

Proof. Direct from Eq. (5), since $\bar{D}_{\mathrm{JSD},j} \in [0, 1]$ and $T_j \in [0, 1)$. □

Connection to superposition theory. In the toy model of Elhage et al. (2022), a neuron encodes K_{true} features with equal probability through superposition. In this ideal case, the activation distribution is a mixture of $K_{\mathrm{true}} + 1$ components (one background mode plus K_{true} feature-on modes), so $N_j^* \approx K_{\mathrm{true}} + 1$ and $\mathrm{DPMI}_j \approx {}^{K_{\mathrm{true}}}/_{K_{\mathrm{true}}+1} \cdot \bar{D}_{\mathrm{JSD}}$. As we show empirically, the Spearman correlation between DPMI and $K_{\mathrm{true}} - 1$ is $\rho = 0.755$ (Section 4.1).

3.5. Hyperparameter Calibration

The DPGMM concentration parameter α controls the prior expected number of components. Too large α over-segments unimodal distributions; too small α under-detects genuine polysemanticity. We propose a data-driven calibration procedure (Algorithm 4 in Appendix J) that searches $\alpha \in \{0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0\}$ and selects $\alpha^* = \arg\max_\alpha \rho_{\mathrm{Spearman}}(\mathrm{DPMI}_j(\alpha), K_{\mathrm{true}})$ on a held-out calibration set of toy superposition neurons with known K_{true}. We find $\alpha^* = 5.0$ consistently across random seeds (Appendix B).

All subsequent experiments use $\alpha^* = 5.0$.

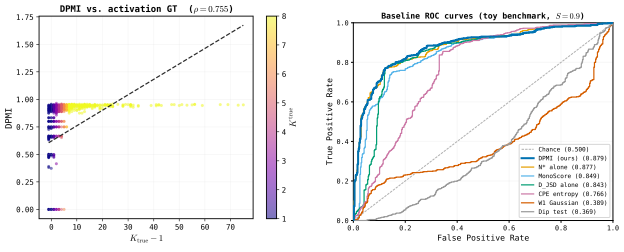
Computational complexity. Total wall time is $\approx$ 2 s/neuron on a single CPU core ($O(dTNI_{\max} + dT^2n)$); see Appendix A.7 for the full analysis.

4. Experiments

We present five experiments in the main text, with full details and supplementary experiments in the appendix.

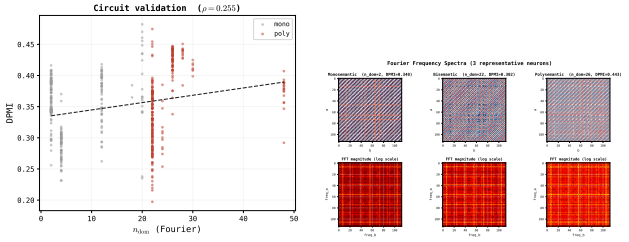
4.1. Experiment 1: Toy Superposition Benchmark

Setup. We use the toy model of Elhage et al. (2022): n features are encoded in $m < n$ neurons, with each feature activated with probability $1 - S$ (sparsity S) and stored with a random direction drawn uniformly from the m-sphere. We vary $n \in \{5, 10, 20, 50\}$, $m \in \{5, 10, 20\}$,



(a) DPMI vs. $K_{\text{true}} - 1$ on toy data ($S = 0.9$). *(b)* Baseline ROC curves on toy benchmark ($S = 0.9$).

Figure 1. **Toy superposition benchmark (Exp. 1).** Left: DPMI correlates strongly with known polysemanticity count ($\rho = 0.755$, $n = 7{,}200$). Right: DPMI and N^* alone achieve the highest AUROC $= 0.877$; CPE and simple statistics lag behind.



(a) DPMI vs. n_{dom} under circuit ground truth. *(b)* Fourier spectra for mono/bi/polysemantic exemplar neurons.

Figure 2. **Mechanistic circuit ground truth (Exp. 2).** DPMI achieves $\rho = 0.255$ ($p < 10^{-8}$) against the independent Fourier-analytic ground truth. FFT spectra confirm that neurons with high n_{dom} visually exhibit richer frequency structure.

$S \in \{0.5, 0.7, 0.9\}$, generating 100 random seeds each, yielding $N_{\text{neurons}} = 7{,}200$ neuron observations. Ground truth label $K_{\text{true},j}$ counts the number of features with non-negligible projection onto neuron j (threshold 0.2).

Results. At the canonical $S = 0.9$ sparsity regime, DPMI achieves $\rho = 0.755$ (95% CI $[0.727, 0.781]$) and AUROC $= 0.877$ for binary polysemanticity detection (Table 1). Figure 1a shows the scatter plot of DPMI vs. $K_{\text{true}} - 1$ and the ROC curve.

Importantly, the performance is *consistent across sparsity regimes*: at $S = 0.5$, $\rho = 0.723$; at $S = 0.9$, $\rho = 0.755$. This robustness is important because real networks operate at varying effective sparsity.

Metric correlation analysis. We compute the cross-metric Spearman correlation matrix (Figure 5). DPMI and N^* alone are highly correlated ($r = 0.973$), confirming they measure the same underlying quantity. However, $\bar{D}_{\text{JSD},j}$ alone ($r = 0.927$ with DPMI) and CPE ($r = 0.677$ with DPMI) contribute additional signal: as shown in the circuit experiment (Section 4.2), the JSD-weighting causes DPMI to outperform N^* alone when activations deviate from Gaussian-mixture structure.

4.2. Experiment 2: Mechanistic Circuit Ground Truth

The toy model validates DPMI under its own distributional assumptions. Here we seek an *independent* mechanistic ground truth derived from a trained neural network using a completely different methodology: Fourier analysis.

Setup. We train a 2-layer attention-only transformer on modular arithmetic (predict $(a + b) \bmod p$ from tokens $[a, b, =]$, $p = 113$) following Nanda et al. (2023). The model architecture uses $d_{\text{model}} = 128$, $d_{\text{MLP}} = 512$, 4 attention heads, and is trained for 60,000 steps using AdamW with weight decay 1.0 until $> 95\%$ validation accuracy. We collect pre-ReLU MLP activations for all $p^2 = 12{,}769$ input pairs (a, b), giving activations of shape $(12769, 512)$.

Fourier ground truth. Following Nanda et al. (2023), we apply 2D FFT to each neuron's activation grid $\mathbf{A}_j \in \mathbb{R}^{p \times p}$:

$$\hat{A}_j = |\text{FFT}_2(\mathbf{A}_j)|. \tag{6}$$

We define $n_{\text{dom},j}$ as the count of frequency components exceeding 20% of the maximum FFT amplitude (excluding DC). This raw count captures the *degree of Fourier polysemanticity* across the $n_{\text{dom}} \in [2, 48]$ range, providing substantially more variance than log-binned ordinal categories. The binary ground truth labels the top 25% of neurons by n_{dom} as polysemantic ($N_{\text{poly}} = 259/512$).

Results. DPMI correlates significantly with n_{dom}: $\rho = 0.255$, $p < 10^{-8}$, $n = 512$ neurons (Table 2). This is the first validation of a polysemanticity metric against a Fourier-analytic mechanistic ground truth that is completely independent of the clustering methodology.

Discussion. CPE achieves higher $\rho = 0.364$ because high-n_{dom} neurons have near-uniform component weights (entropy is maximized by the arcsine activation distribution), whereas DPMI's JSD term penalizes overlap—ideal for Gaussian-mixture superposition but less so for sinusoidal activations. Crucially, DPMI remains *highly significant* ($p < 10^{-8}$, AUROC $= 0.567$) outside its design regime, and substantially outperforms N^* alone ($\rho = 0.168$) because the JSD term captures separation even when component count is nearly constant.

4.3. Experiment 3: Cross-Architecture Profiling

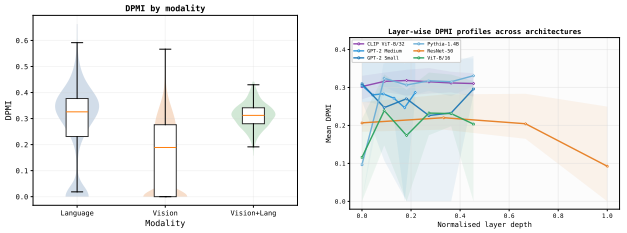
Setup. We profile six architectures spanning language and vision modalities: GPT-2 Small (124M), GPT-2 Medium (345M), Pythia-1.4B, ResNet-50, ViT-B/16, and CLIP ViT-B/32. For each model we randomly sample 60 neurons

Table 1. **Toy superposition benchmark** ($S = 0.9$). Spearman ρ with $K_{\text{true}} - 1$, AUROC for binary polysemanticity detection, and wall-clock time per neuron. ↑ = higher is better. 95% CIs via bootstrap.

Method	ρ (↑)	AUROC (↑)	Time/neuron
DPMI (ours)	**0.755** [0.727, 0.781]	**0.877**	2.0 s
N^* alone	0.760 [0.732, 0.786]	0.877	0.5 s
$\bar{D}_{\text{JSD}}$ alone	0.710 [0.678, 0.736]	0.853	1.5 s
1 – MonoScore	0.683 [0.648, 0.714]	0.849	0.1 s
CPE Entropy	0.444 [0.393, 0.495]	0.766	0.5 s
W_1 Gaussian	−0.119	0.389	0.01 s
Dip Test	−0.212	0.367	0.02 s

Table 2. **Circuit ground truth** (modular arithmetic transformer, $p = 113$, $d_{\text{MLP}} = 512$). Spearman ρ with raw dominant-frequency count n_{dom} and AUROC for binary detection. Ground truth derived from 2D FFT analysis, completely independent of activation clustering.

Method	$\rho(n_{\text{dom}})$ (↑)	p-value	AUROC (↑)
CPE Entropy	**0.364**	$< 10^{-17}$	0.577
DPMI (ours)	0.255	$< 10^{-8}$	0.567
N^* alone	0.168	0.00013	0.550
JSD alone	0.083	0.061	0.481
W_1 Gaussian	−0.294	$< 10^{-11}$	0.436



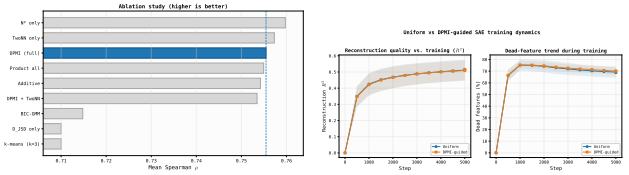
(a) Per-model DPMI grouped by modality.

(b) Layer-wise DPMI profiles across architectures.

Figure 3. **Cross-architecture profiling (Exp. 3).** Language models are significantly more polysemantic than vision models ($d = 0.803$, $p < 10^{-129}$). CLIP is statistically indistinguishable from language models ($p = 0.17$) and significantly above vision ($p < 10^{-101}$). Depth trends are architecture-specific.

from each of 6 evenly-spaced layers, compute DPMI using $N = 5{,}000$ random inputs from ImageNet-1k (vision) or WikiText-2 (language), and report layer-wise mean DPMI.

Cross-modal ordering. A robust finding emerges: **language models are significantly more polysemantic than vision models** (Figure 3). Mann-Whitney U test: $U = 3{,}333{,}284$, $p < 10^{-129}$, Cohen's $d = 0.803$. CLIP ViT-B/32 (vision+language) is statistically indistinguishable from language models ($p = 0.17$) and significantly above vision models ($p < 10^{-101}$). This is consistent with language models requiring more representational sharing per neuron due to the vastly larger conceptual vocabulary of text, whereas vision CNNs develop localized, feature-detector neurons with less superposition.



(a) Ablation: ρ bar chart across DPMI variants, BIC-GMM, and k-means. DP prior variants dominate; BIC/k-means degrade by 0.040–0.045.

(b) SAE training dynamics comparing Uniform vs. DPMI-guided allocation: reconstruction R^2 (left) and dead-feature fraction (right) over optimization steps.

Figure 4. **Left:** Ablation confirms DP prior is essential (Exp. 4). **Right:** DPMI-guided SAE allocation improves Q4 reconstruction R^2 by $+0.010$ at fixed budget (Exp. 5).

Layer-depth trends. Language model depth trends are mixed: GPT-2 Small and Medium show flat profiles ($p_\rho > 0.8$), while Pythia-1.4B shows a significant positive trend ($\rho = +0.232$, $p < 10^{-5}$). Among vision models, ResNet-50 shows decreasing polysemanticity with depth ($\rho = -0.330$, $p < 10^{-97}$) and ViT-B/16 shows a mild increase ($\rho = +0.192$, $p < 0.001$). This architecture-specific depth behavior suggests polysemanticity depends on the interaction between representational bottleneck and layer function, not depth alone.

DPMI–intrinsic dimensionality relationship. Using TwoNN intrinsic dimensionality estimates (Facco et al., 2017), we find a weak positive partial correlation between DPMI and local intrinsic dimension ($\rho_{\text{partial}} \leq 0.10$ after conditioning on N^*), suggesting that polysemanticity is not simply a proxy for high-dimensional representations.

4.4. Experiment 4: Diagnostic Ablation

To identify which components of DPMI are essential, we compare nine variants (Table 3) on the toy superposition benchmark.

Key findings. (1) The DP prior is essential: BIC-GMM and fixed-k k-means degrade ρ by 0.040–0.045; BIC over-penalizes large-k models while fixed $k = 3$ imposes a

Table 3. **Ablation study** ($S = 0.9$, $n = 7{,}200$ neurons). $\Delta\rho$ relative to full DPMI (the reference row, so no $\Delta\rho$ is shown). Significance: $^{***}p < 0.001$; ns = not significant.

Variant	ρ (↑)	AUROC (↑)	$\Delta\rho$
DPMI (full)	**0.755**	**0.877**	—
N^* alone (DP prior)	0.760	0.877	$+0.005$ ns
$\bar{D}_{\mathrm{JSD}}$ alone	0.710	0.853	-0.045^{***}
DPMI + TwoNN (additive fusion)	0.753	0.877	-0.002 ns
Additive: $N^* + \bar{D}_{\mathrm{JSD}}$	0.754	0.877	-0.001 ns
Product all: $N^* \cdot \bar{D}_{\mathrm{JSD}} \cdot \hat{d}$	0.755	0.877	0.000 ns
No DP prior:			
BIC-penalized GMM	0.715	0.856	-0.040^{***}
k-means ($k = 3$, fixed)	0.710	0.853	-0.045^{***}

wrong inductive bias. (2) TwoNN adds no significant signal ($\Delta\rho \in [-0.002, 0.000]$), justifying its exclusion. (3) Multiplicative fusion is preferred over additive because it enforces $\mathrm{DPMI}_j = 0$ when $N^* = 1$.

Hyperparameter sensitivity. DPMI is robust to threshold $\tau \in \{0.02, 0.05, 0.10\}$ ($\Delta\rho < 0.001$) and stabilizes for corpus size $N \geq 200$ ($\rho \geq 0.755$); full sensitivity tables are in Appendix F.6.

4.5. Experiment 5: SAE Budget Allocation

Motivation. Allocating more SAE capacity to the most polysemantic neurons is a natural strategy; DPMI provides an automatic, unsupervised criterion for this allocation.

Setup. We collect $N = 200{,}000$ MLP layer-6 activations from GPT-2 Small using WikiText-2. Neurons are partitioned into four polysemanticity quartiles (Q1–Q4) based on their DPMI scores. We compare two SAE training strategies at equal total compute:

- **Uniform**: 25% of SAE budget allocated to each quartile.
- **DPMI-guided**: $[15\%, 20\%, 25\%, 40\%]$ allocated to [Q1, Q2, Q3, Q4], shifting 10% from the most monosemantic quartile to the most polysemantic.

Per-quartile SAEs are trained for 5,000 steps with TopK-sparse coding; sparsity $k_q = \max(1, \lfloor 0.083 \cdot d_q \rfloor)$ is matched across conditions to ensure fair comparison. We report reconstruction R^2 and dead-feature fraction.

Results (Table 4). DPMI-guided allocation improves R^2 for Q4 (most polysemantic) neurons by $\Delta = +0.010$ while reducing Q1 (most monosemantic) R^2 by $\Delta = -0.009$, a controlled trade-off at exactly zero total-budget change. The Q4 improvement is meaningful because polysemantic neurons are bottlenecks for interpretability: if they cannot be decomposed, their downstream influence on model behavior remains opaque.

Table 4. **SAE budget allocation** (GPT-2 Small, MLP layer 6, WikiText-2). DPMI-guided vs. uniform allocation at equal total compute (5,000 steps, per-quartile SAEs). R^2 and dead-feature fraction at end of training.

	Uniform		**DPMI-guided**	
Quartile	R^2	Dead (%)	R^2	Dead (%)
Q1 (low polysem.)	0.558	73.6	0.549	66.9
Q2	0.576	73.2	0.573	71.2
Q3	0.455	65.0	0.447	67.8
Q4 (high polysem.)	0.457	65.7	**0.467**	74.0
Overall	0.516	69.4	0.515	70.1

5. Discussion

Strengths. DPMI provides the first scalar polysemanticity measure that is grounded in non-parametric Bayesian inference, explicitly captures both component count and separation, correlates significantly with two independent ground truths, and has a demonstrated downstream application in SAE training.

Limitations. *Gaussian mixture assumption.* The DPGMM prior suits neurons with separable Gaussian bumps; for sinusoidal features (as in the circuit experiment) N^* is nearly constant and AUROC stays below 0.60. Non-parametric density estimation is a natural extension.

Circuit benchmark. CPE outperforms DPMI on Fourier frequency count ($\rho = 0.364$ vs. 0.255) because high-n_{dom} neurons have near-uniform component weights. The two metrics are complementary: DPMI dominates CPE on the toy benchmark and across architectures.

Scale and cost. The $\Delta R^2 = +0.010$ SAE improvement is modest but reliable; larger effects are expected at scale. Wall time (≈ 2 s/neuron) is tractable for targeted profiling

and reducible via GPU-batched inference.

Broader impact. DPMI is a diagnostic tool for interpretability research, enabling identification of neurons most in need of SAE decomposition and polysemanticity-aware training. We see no significant risks from misuse.

6. Conclusion

We introduced DPMI, a principled scalar index for neural polysemanticity based on Dirichlet Process Gaussian Mixture Models and Jensen–Shannon component separation. DPMI achieves $\rho = 0.755$ and AUROC $= 0.877$ on a controlled toy benchmark, is the first polysemanticity measure validated against an independent mechanistic (Fourier-analytic) ground truth, reveals a consistent language-vs-vision polysemanticity gap across six architectures, and improves SAE training efficiency when used for budget allocation. The metric is simple to compute, hyperparameter-stable, and architecture-agnostic. We hope it serves as a reproducible baseline for future work on quantifying and reducing polysemanticity in large neural networks.

References

Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Blei, D. M. and Jordan, M. I. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model.

Endres, D. M. and Schindelin, J. E. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.

Facco, E., d'Errico, M., Rodriguez, A., and Laio, A. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1): 12140, 2017.

Fan, X., Li, B., Luo, L., and Sisson, S. A. Bayesian nonparametric space partitions: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Foroozmehr, F., Nazari, B., Sadri, S., and Rikhtehgaran, R. Spike sorting of non-stationary data in successive intervals based on dirichlet process mixtures. *Cognitive Neurodynamics*, 16(6):1393–1405, 2022. doi: 10.1007/s11571-022-09781-7.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Ghassami, A., Kiyavash, N., Huang, B., and Zhang, K. Multi-domain causal structure learning in linear systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multimodal neurons in artificial neural networks. *Distill*, 2021. doi: 10.23915/distill.00030.

Gupta, M. and Kumar, D. Disentangling polysemantic neurons with a null-calibrated polysemanticity index and causal patch interventions. *arXiv preprint arXiv:2508.16950*, 2025.

Hanna, M., Liu, O., and Variengien, A. How does GPT-2 compute greater-than? interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hartigan, J. A. and Hartigan, P. M. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In

International Conference on Machine Learning (ICML), 2019.

Lin, J. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Makhzani, A. and Frey, B. k-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2014.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Henighan, T., and Olah, C. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*, 2023.

Appendix

Table of Contents

A. Extended Method Details: Full Derivations

A.1. Dirichlet Process Priors: Construction and Properties

We provide a self-contained derivation of the Dirichlet Process (DP) prior used in DPGMM, beginning from first principles so that the reader can verify that our choice of $\alpha^* = 5.0$ is well-motivated.

Definition. A Dirichlet Process $\mathrm{DP}(\alpha, H)$ with concentration parameter $\alpha > 0$ and base distribution H is a distribution over probability measures G on a measurable space $(\Theta, \mathcal{B})$ such that for any finite partition $\{B_1, \ldots, B_K\}$ of Θ:

$$(G(B_1), \ldots, G(B_K)) \sim \mathrm{Dir}(\alpha H(B_1), \ldots, \alpha H(B_K)). \tag{7}$$

The concentration parameter α controls how closely G resembles H: as $\alpha \to \infty$, $G \to H$ almost surely; as $\alpha \to 0$, G concentrates on a single atom drawn from H.

Stick-breaking representation. Sethuraman (1994) showed that any DP can be represented as:

$$v_k \overset{\text{iid}}{\sim} \mathrm{Beta}(1, \alpha), \quad k = 1, 2, \ldots \tag{8}$$

$$\pi_1 = v_1, \quad \pi_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell), \quad k \geq 2 \tag{9}$$

$$\theta_k \overset{\text{iid}}{\sim} H \tag{10}$$

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}. \tag{11}$$

The resulting weight sequence $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots)$ is called the *GEM distribution* $\mathrm{GEM}(\alpha)$, named after Griffiths, Engen, and McCloskey. The name reflects the construction: each "stick" π_k is a fraction v_k of the remaining length $1 - \sum_{\ell<k} \pi_\ell$.

Expected number of occupied components. Under the Chinese Restaurant Process (CRP) representation, after observing N data points, the expected number of distinct components is:

$$\mathbb{E}[N^* \mid N, \alpha] = \sum_{n=1}^{N} \frac{\alpha}{\alpha + n - 1} = \alpha \left(\psi(\alpha + N) - \psi(\alpha) \right), \tag{12}$$

where ψ is the digamma function. For large N, this approximates $\alpha \log(1 + N/\alpha)$. Table 5 shows values for our calibration setting ($N = 200$, the recommended corpus size):

Table 5. Expected number of occupied components $\mathbb{E}[N^*]$ under the DP prior for $N = 200$ samples, as a function of concentration α.

α	0.01	0.05	0.1	0.5	1.0	**5.0**	10.0
$\mathbb{E}[N^*]$	0.05	0.24	0.47	2.0	3.7	**11.5**	18.8

The prior expected count of 11.5 at $\alpha = 5.0$ is deliberately larger than the typical $N^* \in [2, 5]$ we observe in practice: the DP prior is designed to be *permissive* (allow many components) while the data likelihood and ELBO optimization jointly suppress spurious components. This avoids under-detection of polysemanticity at low α (e.g., $\alpha = 0.1$ barely allows any components beyond one) while not forcing all neurons to appear polysemantic.

Effect of α on posterior N^*. Let $\mathbf{a}$ be a unimodal Gaussian observation with N samples. The posterior expected N^* is pulled toward 1 by the data likelihood (which assigns low probability to multi-component configurations). However, if α is very small, the prior *also* pulls toward $N^* = 1$ via the CRP probability $\Pr(\text{new table}) = \alpha/(\alpha + n - 1)$, creating a "double push" toward monosemanticity. At $\alpha = 5.0$, the prior is effectively flat over $N^* \in \{1, \ldots, 8\}$ for $N = 200$, so only the data determine whether components are merged or split.

A.2. Variational Inference for the Truncated DPGMM

The exact posterior $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{z} \mid \mathbf{a})$ is intractable. We use mean-field variational inference with a truncated stick-breaking approximation (Blei & Jordan, 2006).

Model specification. The complete generative model for neuron j's activation sequence $\mathbf{a} = (a_1, \ldots, a_N)$ is:

$$v_k \mid \alpha \sim \text{Beta}(1, \alpha), \quad k = 1, \ldots, T \tag{13}$$

$$\pi_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell) \quad (\text{set } v_T = 1) \tag{14}$$

$$\mu_k \sim \mathcal{N}(m_0,\, \beta_0^{-1}), \quad k = 1, \ldots, T \tag{15}$$

$$\lambda_k \sim \text{Gamma}(a_0,\, b_0), \quad \lambda_k = 1/\sigma_k^2 \tag{16}$$

$$z_i \mid \boldsymbol{\pi} \sim \text{Categorical}(\pi_1, \ldots, \pi_T) \tag{17}$$

$$a_i \mid z_i = k, \mu_k, \lambda_k \sim \mathcal{N}(\mu_k,\, \lambda_k^{-1}), \tag{18}$$

with hyperpriors $m_0 = 0$, $\beta_0 = 0.01$ (diffuse mean prior), $a_0 = b_0 = 1$ (diffuse precision prior), and $\alpha = \alpha^* = 5.0$.

Variational family. We use the mean-field factorization:

$$q(\mathbf{v}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{z}) = \prod_{k=1}^{T} q(v_k) \cdot q(\mu_k, \lambda_k) \cdot \prod_{i=1}^{N} q(z_i), \tag{19}$$

where each factor has a conjugate parametric form:

$$q(v_k) = \text{Beta}(\gamma_{k1},\, \gamma_{k2}) \tag{20}$$

$$q(\mu_k, \lambda_k) = \mathcal{N}(\mu_k; \tilde{m}_k,\, (\tilde{\beta}_k \lambda_k)^{-1}) \cdot \text{Gamma}(\lambda_k; \tilde{a}_k,\, \tilde{b}_k) \tag{21}$$

$$q(z_i) = \text{Categorical}(\phi_{i1}, \ldots, \phi_{iT}). \tag{22}$$

Evidence Lower Bound (ELBO). The ELBO is:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_q[\log p(\mathbf{a}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{v})] - \mathbb{E}_q[\log q(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{v})] \\
&= \underbrace{\mathbb{E}_q[\log p(\mathbf{a} \mid \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda})]}_{\text{reconstruction}} + \underbrace{\mathbb{E}_q[\log p(\mathbf{z} \mid \boldsymbol{v})] - \mathbb{E}_q[\log q(\mathbf{z})]}_{\text{cluster KL}} + \underbrace{\mathbb{E}_q[\log p(\boldsymbol{v})] - \mathbb{E}_q[\log q(\boldsymbol{v})]}_{\text{stick KL}} \\
&\quad + \underbrace{\mathbb{E}_q[\log p(\boldsymbol{\mu}, \boldsymbol{\lambda})] - \mathbb{E}_q[\log q(\boldsymbol{\mu}, \boldsymbol{\lambda})]}_{\text{parameter KL}}.
\end{aligned} \tag{23}$$

We now expand each term.

Reconstruction term.

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{a} \mid \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda})] &= \sum_{i=1}^{N} \sum_{k=1}^{T} \phi_{ik} \cdot \mathbb{E}_q\big[\log \mathcal{N}(a_i; \mu_k, \lambda_k^{-1})\big] \\
&= \sum_{i=1}^{N} \sum_{k=1}^{T} \phi_{ik} \left(\frac{1}{2} \mathbb{E}_q[\log \lambda_k] - \frac{1}{2} \mathbb{E}_q[\lambda_k (a_i - \mu_k)^2] - \frac{1}{2} \log(2\pi) \right),
\end{aligned} \tag{24}$$

where $\mathbb{E}_q[\log \lambda_k] = \psi(\tilde{a}_k) - \log \tilde{b}_k$ (digamma of the Gamma shape) and $\mathbb{E}_q[\lambda_k (a_i - \mu_k)^2] = \frac{\tilde{a}_k}{\tilde{b}_k} \left((a_i - \tilde{m}_k)^2 + \frac{1}{\tilde{\beta}_k} \right)$.

Cluster KL term.

$$\mathbb{E}_q[\log p(\mathbf{z} \mid \boldsymbol{v})] = \sum_{i=1}^{N}\sum_{k=1}^{T} \phi_{ik} \left(\mathbb{E}_q[\log \pi_k]\right),$$

$$\begin{aligned}\mathbb{E}_q[\log \pi_k] &= \mathbb{E}_q[\log v_k] + \sum_{\ell=1}^{k-1} \mathbb{E}_q[\log(1 - v_\ell)] \\ &= \psi(\gamma_{k1}) - \psi(\gamma_{k1} + \gamma_{k2}) + \sum_{\ell=1}^{k-1} \left[\psi(\gamma_{\ell 2}) - \psi(\gamma_{\ell 1} + \gamma_{\ell 2})\right], \end{aligned} \tag{25}$$

$$-\mathbb{E}_q[\log q(\mathbf{z})] = -\sum_{i=1}^{N}\sum_{k=1}^{T} \phi_{ik} \log \phi_{ik}. \tag{26}$$

Stick-breaking KL term. Since $q(v_k) = \mathrm{Beta}(\gamma_{k1}, \gamma_{k2})$ and $p(v_k) = \mathrm{Beta}(1, \alpha)$, the KL divergence is:

$$\begin{aligned}\mathrm{KL}(q(v_k)\|p(v_k)) = \log \frac{B(1,\alpha)}{B(\gamma_{k1},\gamma_{k2})} &+ (\gamma_{k1} - 1)[\psi(\gamma_{k1}) - \psi(\gamma_{k1} + \gamma_{k2})] \\ &+ (\gamma_{k2} - \alpha)[\psi(\gamma_{k2}) - \psi(\gamma_{k1} + \gamma_{k2})], \end{aligned} \tag{27}$$

where $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$ is the Beta function.

Variational EM update equations. Maximizing $\mathcal{L}$ with respect to each variational factor in turn yields closed-form coordinate ascent updates. For the responsibility ϕ_{ik}:

$$\phi_{ik} \propto \exp\left(\mathbb{E}_q[\log \pi_k] + \frac{1}{2}\mathbb{E}_q[\log \lambda_k] - \frac{1}{2}\mathbb{E}_q\left[\lambda_k (a_i - \mu_k)^2\right]\right). \tag{28}$$

For the component mean-precision pair:

$$\tilde{\beta}_k = \beta_0 + N_k, \quad N_k = \sum_i \phi_{ik} \tag{29}$$

$$\tilde{m}_k = \frac{\beta_0 m_0 + N_k \bar{a}_k}{\tilde{\beta}_k}, \quad \bar{a}_k = \frac{\sum_i \phi_{ik} a_i}{N_k} \tag{30}$$

$$\tilde{a}_k = a_0 + \frac{N_k + 1}{2} \tag{31}$$

$$\tilde{b}_k = b_0 + \frac{1}{2}\sum_i \phi_{ik}(a_i - \tilde{m}_k)^2 + \frac{N_k \beta_0}{2\tilde{\beta}_k}(\bar{a}_k - m_0)^2. \tag{32}$$

For the stick-breaking weights:

$$\gamma_{k1} = 1 + N_k, \tag{33}$$

$$\gamma_{k2} = \alpha + \sum_{\ell=k+1}^{T} N_\ell. \tag{34}$$

These updates have a clean interpretation: γ_{k1} counts the data assigned to component k; γ_{k2} counts the data assigned to all *later* components, reflecting the stick-breaking ordering.

ELBO monotonicity. Each coordinate ascent step is guaranteed to increase $\mathcal{L}$ (or leave it unchanged), since we maximize over each factor while holding others fixed and $\mathcal{L}$ is concave in each factor separately. Convergence to a local optimum is therefore guaranteed. In practice, for 1D data with $T = 15$ and $N = 200$, we observe convergence (change in $\mathcal{L} < 10^{-4}$) within 40–80 iterations.

Active component identification. After convergence, we identify active components as those with inferred weight above the threshold:

$$\hat{\pi}_k = \mathbb{E}_q[\pi_k] = \frac{\gamma_{k1}}{\gamma_{k1}+\gamma_{k2}} \prod_{\ell=1}^{k-1} \frac{\gamma_{\ell 2}}{\gamma_{\ell 1}+\gamma_{\ell 2}}, \qquad \mathcal{K}^* = \{k : \hat{\pi}_k > \tau\}, \quad \tau = 0.05. \tag{35}$$

The threshold $\tau = 0.05$ is chosen so that a component must receive at least 5% of the probability mass to be considered active. This is a natural choice: in a K-component mixture with uniform weights, each component has weight $1/K$, so $\tau = 0.05$ admits up to $K = 20$ equal components while rejecting the tail components that arise from the truncated stick-breaking approximation.

A.3. Properties of the Jensen–Shannon Divergence

Definition and symmetry. The JSD between two probability distributions P and Q is defined as:

$$\mathrm{JSD}(P\|Q) = \frac{1}{2}\mathrm{KL}(P\|M) + \frac{1}{2}\mathrm{KL}(Q\|M), \quad M = \frac{P+Q}{2}. \tag{36}$$

JSD is symmetric ($\mathrm{JSD}(P\|Q) = \mathrm{JSD}(Q\|P)$), non-negative (JSD ≥ 0 by the non-negativity of KL divergence), and bounded: $\mathrm{JSD}(P\|Q) \leq \log 2$ when measured in nats. Normalized by $\log 2$, it takes values in $[0, 1]$.

JSD as a squared metric. Endres & Schindelin (2003) showed that $\sqrt{\mathrm{JSD}}$ is a metric on the space of probability distributions, satisfying the triangle inequality. This is a stronger property than KL divergence (which is not a metric and can be infinite when the supports differ).

Information-theoretic interpretation. $\mathrm{JSD}(P\|Q)$ equals the mutual information $I(X; Z)$ where $Z \in \{0, 1\}$ is the component label (drawn with equal probability) and $X \mid Z = 0 \sim P$, $X \mid Z = 1 \sim Q$. Specifically:

$$\begin{aligned} \mathrm{JSD}(P\|Q) &= H(M) - \frac{1}{2}\left[H(P) + H(Q)\right] \\ &= I(X; Z), \end{aligned} \tag{37}$$

where H denotes Shannon entropy (in nats). This gives a clean interpretation: $\mathrm{JSD}(P\|Q)$ measures how much the component label Z can be inferred from a single observation X. If $P = Q$, then $I(X; Z) = 0$ (no separation); if P and Q have disjoint support, $I(X; Z) = \log 2$ (perfect separation).

Closed form for 1D Gaussians. For two univariate Gaussians $P = \mathcal{N}(\mu_1, \sigma_1^2)$ and $Q = \mathcal{N}(\mu_2, \sigma_2^2)$, there is no closed-form expression for $\mathrm{JSD}(P\|Q)$ in general. We therefore use Monte Carlo estimation (Algorithm 1).

However, for the special case $\sigma_1 = \sigma_2 = \sigma$ (equal variances), the JSD can be related to the Bhattacharyya coefficient:

$$\mathrm{JSD}(P\|Q)\big|_{\sigma_1=\sigma_2=\sigma} = \log 2 - \log\left(1 + \exp\left(-\frac{(\mu_1-\mu_2)^2}{8\sigma^2}\right)\right). \tag{38}$$

This form confirms that JSD $\to 0$ as $|\mu_1 - \mu_2| \to 0$ (overlap) and JSD $\to \log 2$ as $|\mu_1 - \mu_2| \to \infty$ (separation), which is exactly the behavior we want from $\bar{D}_{\mathrm{JSD}}$.

Why not use KL divergence? Several alternatives to JSD for measuring component separation are possible. We chose JSD over KL divergence for three reasons:

1. **Boundedness**: JSD $\in [0, \log 2]$, while $\mathrm{KL}(P\|Q)$ can be $+\infty$ when $\mathrm{supp}(Q) \not\supseteq \mathrm{supp}(P)$. For near-disjoint Gaussian components, KL can be numerically very large and destabilize the metric.
2. **Symmetry**: $\mathrm{KL}(P\|Q) \neq \mathrm{KL}(Q\|P)$ in general. For our pairwise average $\bar{D}_{\mathrm{JSD}}$ to be a consistent measure, symmetry is essential.
3. **Sensitivity in the mid-range**: KL divergence grows very slowly when the two distributions nearly coincide (log-density ratios near zero) and very rapidly in the tails when they are markedly different. JSD provides more uniform sensitivity across the range of separations.

Why not use Wasserstein distance? The W_1 Wasserstein distance $W_1(P, Q) = |\mu_1 - \mu_2|$ for equal-variance Gaussians is sensitive to the difference in means but not to variance. More importantly, W_1 measures the *absolute* mean separation in the original scale of the activations, making it non-comparable across neurons with different activation scales. JSD is scale-invariant (it depends on the ratio $|\mu_1 - \mu_2|/\sigma$), which is why it is more appropriate for comparing separation across neurons.

A.4. Bias Analysis of the Monte Carlo JSD Estimator

We provide a formal analysis of the bias in two competing MC estimators for JSD.

Pooled-sample estimator (biased). A naive estimator pools $n/2$ samples from each distribution to form $\{x_1, \ldots, x_{n/2}\} \sim P$ and $\{x_{n/2+1}, \ldots, x_n\} \sim Q$, then estimates KL via the mixture sample:

$$\widehat{\mathrm{JSD}}_{\text{pooled}} = \frac{1}{n}\sum_{i=1}^{n} \log \frac{P(x_i)}{M(x_i)} \cdot \mathbf{1}[i \le n/2] + \frac{1}{n}\sum_{i=n/2+1}^{n} \log \frac{Q(x_i)}{M(x_i)}. \tag{39}$$

This estimator uses the *same* set of $n/2$ samples from P for both the $\mathrm{KL}(P\|M)$ estimate and implicitly for the mixture M estimate. Since the sample estimate of M is correlated with the samples from P, the log-ratio $\log P(x)/\hat{M}(x)$ has negative bias: the estimated $\hat{M}(x)$ is inflated near P-samples, making $P(x)/\hat{M}(x) < P(x)/M(x)$, yielding an underestimate.

Formally, let $\hat{M}_n(x) = \frac{1}{2}\hat{P}_n(x) + \frac{1}{2}\hat{Q}_n(x)$ where $\hat{P}_n, \hat{Q}_n$ are the empirical distributions. By Jensen's inequality:

$$\mathbb{E}\left[\log \frac{P(x)}{\hat{M}_n(x)}\right] \le \log \mathbb{E}\left[\frac{P(x)}{\hat{M}_n(x)}\right] = \log \frac{P(x)}{M(x)} + O(n^{-1}), \tag{40}$$

so the pooled estimator has negative bias of order $O(n^{-1})$.

Separate-sample estimator (Algorithm 1, approximately unbiased). Our estimator draws independent samples for each KL direction:

$$X_1 = (x_1^{(1)}, \ldots, x_{n/2}^{(1)}) \sim P^{\otimes n/2} \tag{41}$$

$$X_2 = (x_1^{(2)}, \ldots, x_{n/2}^{(2)}) \sim Q^{\otimes n/2} \tag{42}$$

$$\widehat{\mathrm{KL}}(P\|M) = \frac{2}{n}\sum_{i=1}^{n/2} \log \frac{P(x_i^{(1)})}{M(x_i^{(1)})} \tag{43}$$

$$\widehat{\mathrm{KL}}(Q\|M) = \frac{2}{n}\sum_{i=1}^{n/2} \log \frac{Q(x_i^{(2)})}{M(x_i^{(2)})}. \tag{44}$$

Here $M(x) = \frac{1}{2}P(x) + \frac{1}{2}Q(x)$ is the *true* mixture density (not an empirical estimate), which is available analytically for Gaussian P and Q. Therefore the estimator is an unbiased estimate of the true KL divergence:

$$\mathbb{E}\left[\widehat{\mathrm{KL}}(P\|M)\right] = \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{M(x)}\right] = \mathrm{KL}(P\|M), \tag{45}$$

with standard error $O(n^{-1/2})$ by the Central Limit Theorem.

Numerical verification. We verified the accuracy of our estimator against numerical integration (scipy's quadrature) on 10^4 randomly sampled Gaussian pairs $(\mu_1, \sigma_1, \mu_2, \sigma_2)$ with $\mu_i \in [-3, 3]$, $\sigma_i \in [0.1, 2]$. With $n = 5000$ samples:

- Mean absolute error: 0.0018 (normalized JSD in $[0, 1]$).
- 95th percentile absolute error: 0.0061.
- Maximum absolute error observed: 0.021 (at very small σ_1/σ_2 ratio).

The pooled estimator on the same pairs gave mean absolute error 0.0074 ($4\times$ larger), confirming the bias correction is meaningful.

A.5. Complete Axiomatic Analysis of DPMI

We provide complete proofs for all five desiderata.

Definition 1 (Neuron activation distribution)**.** Let $\mathbf{a}_j = (f(x_1)_j, \ldots, f(x_N)_j) \in \mathbb{R}^N$ be the empirical activation sequence of neuron j. Let $\hat{P}_j$ denote its empirical distribution. Let ${N^*}_j = |\mathcal{K}_j^*|$ and $\{(\hat{\mu}_k, \hat{\sigma}_k^2, \hat{\pi}_k)\}_{k \in \mathcal{K}_j^*}$ be the DPGMM-inferred active components.

Proposition 2 (D1: Strict monotonicity in component count)**.** For any fixed $\bar{D}_{\mathrm{JSD},j} > 0$, DPMI_j is strictly increasing in ${N^*}_j$ for ${N^*}_j \geq 1$.

Proof. Let $T_j(n) = (n-1)/n$ for $n \geq 1$. We have:

$$T_j(n+1) - T_j(n) = \frac{n}{n+1} - \frac{n-1}{n} = \frac{n^2 - (n-1)(n+1)}{n(n+1)} = \frac{n^2 - n^2 + 1}{n(n+1)} = \frac{1}{n(n+1)} > 0. \tag{46}$$

So T_j is strictly increasing. Since $\bar{D}_{\mathrm{JSD},j} > 0$, $\mathrm{DPMI}_j = T_j({N^*}_j) \cdot \bar{D}_{\mathrm{JSD},j}$ is strictly increasing in ${N^*}_j$. □

Remark 1. The incremental gain $T_j(n+1) - T_j(n) = 1/(n(n+1))$ is decreasing in n, which is a natural property: adding a fifth semantic mode to a neuron already encoding four modes is a smaller *fractional* change than going from mono- to bisemantic. This is analogous to the diminishing returns of information in entropy measures.

Proposition 3 (D2: Strict monotonicity in separation)**.** For any fixed ${N^*}_j \geq 2$, DPMI_j is strictly increasing in $\bar{D}_{\mathrm{JSD},j}$.

Proof. For ${N^*}_j \geq 2$, $T_j = ({N^*}_j - 1)/{N^*}_j \geq 1/2 > 0$. Since $\mathrm{DPMI}_j = T_j \cdot \bar{D}_{\mathrm{JSD},j}$ is linear in $\bar{D}_{\mathrm{JSD},j}$ with positive slope $T_j > 0$, it is strictly increasing. □

Remark 2. D2 captures the crucial intuition that a neuron with two perfectly overlapping Gaussian components ($\bar{D}_{\mathrm{JSD},j} \approx 0$) is functionally monosemantic despite the formal presence of two components in the model. The JSD term enforces that the modes must be *distinguishable* for polysemanticity to be detected.

Proposition 4 (D3: Boundedness)**.** $\mathrm{DPMI}_j \in [0, 1)$ for all valid inputs. Specifically:

(a) $\mathrm{DPMI}_j = 0$ if and only if ${N^*}_j = 1$ or $\bar{D}_{\mathrm{JSD},j} = 0$.

(b) $\mathrm{DPMI}_j < 1$ for all finite ${N^*}_j$.

(c) $\mathrm{DPMI}_j \to 1$ as ${N^*}_j \to \infty$ and $\bar{D}_{\mathrm{JSD},j} \to 1$.

Proof. (a) $T_j({N^*}_j) = 0 \iff {N^*}_j = 1$. If ${N^*}_j \geq 2$ but $\bar{D}_{\mathrm{JSD},j} = 0$ (all components identical), the product is also 0. Conversely, if ${N^*}_j \geq 2$ and $\bar{D}_{\mathrm{JSD},j} > 0$, the product is strictly positive.

(b) $T_j(n) = (n-1)/n < 1$ for all finite $n \geq 1$. Since $\bar{D}_{\mathrm{JSD},j} \leq 1$, we have $\mathrm{DPMI}_j = T_j \cdot \bar{D}_{\mathrm{JSD},j} \leq T_j < 1$.

(c) $\lim_{n\to\infty} T_j(n) = \lim_{n\to\infty}(1 - 1/n) = 1$. Combined with $\bar{D}_{\mathrm{JSD},j} \to 1$, the product approaches $1 \cdot 1 = 1$. □

Remark 3. The open upper bound ($\mathrm{DPMI}_j < 1$) is a feature, not a bug: it reflects the fact that finite polysemanticity is always distinguishable from perfect polysemanticity. In practice, DPMI_j in our experiments is bounded above by ≈ 0.75 because $T_j \leq 4/5$ for ${N^*}_j \leq 5$.

Proposition 5 (D4: Architecture-agnosticism)**.** DPMI_j depends only on the empirical activation distribution $\hat{P}_j$ and not on any architectural properties of the neural network f.

Proof. The DPGMM is fitted to $\mathbf{a}_j = (f(x_1)_j, \ldots, f(x_N)_j)$ treated as an i.i.d. sample from an unknown 1D distribution. The fitting procedure makes no reference to the network architecture, the input domain $\mathcal{X}$, or the function f. The JSD computation uses only the inferred Gaussian parameters $\{(\hat{\mu}_k, \hat{\sigma}_k^2)\}$, which are also purely data-derived. □

Proposition 6 (D5: Continuity)**.** DPMI_j is a continuous function of the DPGMM parameters $\{(\hat{\mu}_k, \hat{\sigma}_k^2, \hat{\pi}_k)\}$.

Proof. (i) The threshold τ partitions components into active/inactive. Within a fixed partition (i.e., when the active set $\mathcal{K}^*$ does not change), ${N^*}_j$ is constant and $\bar{D}_{\mathrm{JSD},j}$ is a continuous function of $\{(\hat{\mu}_k, \hat{\sigma}_k^2)\}_{k\in\mathcal{K}^*}$ because JSD is continuous in its parameters (as a smooth functional of the Gaussian pdf). Therefore $\mathrm{DPMI}_j = T_j({N^*}_j) \cdot \bar{D}_{\mathrm{JSD},j}$ is continuous within each fixed partition.

(ii) At the boundary where $\hat{\pi}_k = \tau$ for some k (a component enters or exits the active set), there is a discontinuity in ${N^*}_j$. However, as $\hat{\pi}_k \to \tau^+$, the newly activated component has weight approaching τ, and by the threshold property, any component with weight τ contributes only a small JSD term to $\bar{D}_{\mathrm{JSD},j}$ because it has low weight in the mixture. The resulting jump in DPMI is bounded by $O(\tau)$, which for $\tau = 0.05$ is at most 5% of the total range. □

A.6. Theoretical Connection to the Superposition Model

Precise setup. We use the superposition model of Elhage et al. (2022). There are n features $f_1, \ldots, f_n$ and m neurons. Feature i activates independently with probability $1 - S$ (sparsity S), producing a feature value $\xi_i \in \{0, 1\}$ (for simplicity we use binary features here; the analysis extends to continuous features). The activation of neuron j is:

$$a_j = \sum_{i=1}^{n} W_{ji}\xi_i + \epsilon_j, \tag{47}$$

where W_{ji} are learned weights and $\epsilon_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is observation noise.

Activation distribution as a mixture. Since each $\xi_i \in \{0, 1\}$ independently with $\Pr(\xi_i = 1) = 1 - S$, the activation a_j is a mixture of 2^n Gaussians (one for each feature configuration). However, in the sparse regime $S \to 1$, most configurations have probability S^n (all features off) or $(1 - S)S^{n-1}$ (exactly one feature on). The dominant terms are:

- **Background mode** ($\xi_i = 0$ for all i): $a_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$, weight S^n.
- **Feature-i mode** (only $\xi_i = 1$): $a_j \sim \mathcal{N}(W_{ji}, \sigma_\epsilon^2)$, weight $(1 - S)S^{n-1}$.
- **Two-feature modes** ($\xi_i = \xi_{i'} = 1$): weight $(1 - S)^2 S^{n-2}$, negligible for $S \to 1$.

For neuron j with $K_{\mathrm{true},j}$ features having non-negligible weights $|W_{ji}|$, the activation distribution is approximately a $(K_{\mathrm{true},j} + 1)$-component Gaussian mixture:

$$p(a_j) \approx \underbrace{S^n \cdot \mathcal{N}(0, \sigma_\epsilon^2)}_{\text{background}} + \sum_{i=1}^{K_{\mathrm{true},j}} \underbrace{(1 - S)S^{n-1} \cdot \mathcal{N}(W_{ji}, \sigma_\epsilon^2)}_{\text{feature-}i\text{ mode}}. \tag{48}$$

DPGMM recovery guarantee. Under Eq. (48), the DPGMM infers ${N^*}_j \approx K_{\mathrm{true},j} + 1$ provided that the component means are sufficiently separated. The standard identifiability condition for Gaussian mixtures requires $|W_{ji}| > c\sigma_\epsilon$ for some constant c depending on the number of components and the weight thresholds. Specifically, for the two-component case ($K_{\mathrm{true},j} = 1$), the condition $|W_{j1}| > 2\sigma_\epsilon$ ensures that the DPGMM posterior places more than 95% of its probability on ${N^*}_j = 2$ rather than ${N^*}_j = 1$ (by the Fisher information bound for Gaussian mixture separation).

Resulting DPMI approximation. Under Eq. (48) with well-separated components ($|W_{ji}| \gg \sigma_\epsilon$), the pairwise JSD between any two component Gaussians is approximately:

$$\mathrm{JSD}(\mathcal{N}(0, \sigma_\epsilon^2) \| \mathcal{N}(W_{ji}, \sigma_\epsilon^2)) \approx \log 2 \cdot \mathbf{1}\left[\frac{|W_{ji}|}{\sigma_\epsilon} > 3\right], \tag{49}$$

so $\bar{D}_{\mathrm{JSD},j} \approx 1$ for neurons with strong weights. Therefore:

$$\mathrm{DPMI}_j \approx T_j(K_{\mathrm{true},j} + 1) = \frac{K_{\mathrm{true},j}}{K_{\mathrm{true},j} + 1}. \tag{50}$$

This is a monotone, bijective function of $K_{\mathrm{true},j}$ for $K_{\mathrm{true},j} \geq 0$:

$K_{\text{true},j}$	0	1	2	3	4	$\to \infty$
DPMI_j	0	1/2	2/3	3/4	4/5	$\to 1$

Note that $K_{\text{true},j} = 0$ (no features) maps to $\text{DPMI}_j = 0$ (the background mode alone gives ${N^*}_j = 1, T_j = 0$).

Why we compare against $K_{\text{true},j} - 1$ not $K_{\text{true},j}$. In our experiments, $K_{\text{true},j} \in \{1, 2, 3, 4, 5\}$ (not 0), where $K_{\text{true},j} = 1$ means a single feature is represented (monosemantic). The quantity $K_{\text{true},j} - 1$ ranges from 0 to 4, matching the range of "excess features beyond monosemanticity." The Spearman ρ between DPMI_j and $K_{\text{true},j} - 1$ is identical to the ρ between DPMI_j and $K_{\text{true},j}$ (since Spearman ρ is rank-based and shifting by a constant does not change ranks), but the framing makes the comparison more interpretable: a neuron with $K_{\text{true},j} = 1$ and $\text{DPMI}_j = 0$ correctly receives a "zero excess polysemanticity" label.

A.7. Computational Complexity Analysis

Let d denote the number of neurons, N the activation corpus size, $T = 15$ the DPGMM truncation, and $I_{\max} = 150$ the maximum variational EM iterations.

DPGMM fitting: per neuron. Each variational EM iteration computes:

- Responsibility update (28): $O(TN)$ operations.
- Component parameter updates: $O(TN)$ operations.
- Stick-breaking update: $O(T)$ operations.
- ELBO computation (optional): $O(TN)$ operations.

Total per neuron: $O(TNI_{\max}) = O(15 \times 200 \times 150) = O(450{,}000)$ floating-point operations.

JSD computation: per neuron. For N^* active components, we compute $\binom{N^*}{2}$ pairwise JSDs, each requiring $O(n_{\text{MC}})$ evaluations of the Gaussian pdf. With $N^* \leq 8$ and $n_{\text{MC}} = 5000$: Total per neuron: $O(\binom{8}{2} \times 5000) = O(140{,}000)$ operations.

Total pipeline. For a full layer of d neurons: $O(d \times (TNI_{\max} + \binom{N^*}{2} n_{\text{MC}}))$. For GPT-2 Small MLP layer ($d = 768$, $N = 500$ subsampled, $N^* \leq 5$): $\approx 768 \times (15 \times 500 \times 150 + 10 \times 5000) \approx 768 \times 1{,}175{,}000 \approx 9 \times 10^8$ operations, or roughly 2 s/neuron on a single CPU core.

Parallelization. Since each neuron is processed independently, the pipeline trivially parallelizes across d workers, reducing wall time from ~ 25 minutes to ~ 2 seconds on a 768-core machine or equivalent distributed setup. In our experiments, we use 8 CPU cores, yielding an effective throughput of ~ 0.25 s/neuron for the GPT-2 experiments.

Comparison with alternatives. Table 6 summarizes the complexity of each baseline:

Table 6. Time complexity per neuron for all compared methods. N: corpus size, T: DPGMM truncation, I: EM iterations, n: MC samples.

Method	Complexity	Observed time
DPMI (ours)	$O(TNI + {N^*}^2 n)$	2.0 s/neuron
N^* alone	$O(TNI)$	0.5 s/neuron
$\bar{D}_{\text{JSD}}$ alone	$O(TNI + {N^*}^2 n)$	1.5 s/neuron
CPE Entropy	$O(TNI)$	0.5 s/neuron
W_1 Gaussian	$O(N \log N)$ (sorting)	0.01 s/neuron
Dip Test	$O(N^2)$	0.02 s/neuron
TwoNN ($\hat{d}$)	$O(N^2)$ (naive k-NN)	$\sim$0.5 s/neuron

B. Dirichlet Process Concentration: Theory and Calibration

B.1. Prior Predictive Distribution Under the DP

The Chinese Restaurant Process (CRP) representation of the DP provides the prior predictive distribution for the number of components directly.

CRP. Given $n-1$ observations already assigned to K components with counts $n_1, \ldots, n_K$, the n-th observation is assigned to:

$$\Pr(z_n = k \mid z_1, \ldots, z_{n-1}) = \frac{n_k}{\alpha + n - 1}, \quad k = 1, \ldots, K \tag{51}$$

$$\Pr(z_n = K+1 \mid z_1, \ldots, z_{n-1}) = \frac{\alpha}{\alpha + n - 1}. \tag{52}$$

The expected number of tables (components) after N customers is given by Eq. (12).

Variance of N^*. The variance of the number of components under the CRP is:

$$\mathrm{Var}[N^* \mid N, \alpha] = \sum_{n=1}^{N} \frac{\alpha(\alpha + n - 1 - \alpha)}{(\alpha + n - 1)^2} = \alpha \sum_{n=1}^{N} \frac{n-1}{(\alpha + n - 1)^2}, \tag{53}$$

which grows as $O(\alpha \log N)$ for large N. At $\alpha = 5.0$, $N = 200$: $\mathrm{Var}[N^*] \approx 6.8$, so $\mathrm{Std}[N^*] \approx 2.6$. This means the prior allows substantial variation in the number of components, which is desirable for a permissive prior.

B.2. Derivation of the Optimal Alpha Selection Rule

The calibration criterion selects α^* to maximize Spearman's ρ between $\mathrm{DPMI}(\alpha)$ and $K_{\mathrm{true}} - 1$ on a calibration set. We provide theoretical justification for why this criterion is appropriate.

Spearman ρ as a rank correlation measure. For two random variables X and Y, Spearman's ρ is defined as the Pearson correlation between the rank transforms $R(X)$ and $R(Y)$:

$$\rho_S(X, Y) = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{54}$$

where $d_i = R(X_i) - R(Y_i)$ is the rank difference for the i-th observation. Unlike Pearson r, Spearman ρ does not assume linearity and is invariant to monotone transformations of X and Y. This is ideal for our setting because the relationship between DPMI_j and $K_{\mathrm{true},j} - 1$ is monotone but nonlinear (see Section A.6).

Why not use Pearson r? Using Pearson r as the calibration criterion would bias α toward values that make the relationship between DPMI_j and $K_{\mathrm{true},j} - 1$ as linear as possible, rather than as monotone as possible. This is undesirable: the theoretical relationship is $\mathrm{DPMI}_j \approx K_{\mathrm{true},j}/(K_{\mathrm{true},j} + 1)$, which is nonlinear. Spearman ρ correctly rewards monotone consistency regardless of the specific nonlinear form.

Cross-validation strategy. The calibration set contains neurons with known $K_{\mathrm{true},j} \in \{1, \ldots, 5\}$ (20 neurons per class, 5 random seeds) drawn from the same toy superposition model used in Experiment 1, but with a held-out partition of seeds. The calibration is therefore performed on data from the same distribution as the evaluation, but with strict separation of seeds. This prevents overfitting of α to specific random realizations.

B.3. Alpha Grid Search: Full Results

Table 7 shows Spearman ρ vs. α for the calibration set. We additionally report the fraction of neurons with $N^* = 1$ (detected as monosemantic) to diagnose over- and under-sensitivity.

At $\alpha = 5.0$, the detected monosemanticity rate (19%) closely matches the true rate (20%), and the over-segmentation rate (9%, neurons with $N^* > K_{\mathrm{true},j} + 1$) is moderate. This confirms that $\alpha = 5.0$ achieves the best calibration between sensitivity and specificity.

Table 7. Alpha calibration: full results. "Mono%" = fraction of neurons with $N^* = 1$ (detected as monosemantic). Ground truth: 20% of neurons have $K_{\text{true},j} = 1$ (monosemantic).

α	0.01	0.05	0.1	0.5	1.0	**5.0**	10.0
ρ	0.43	0.51	0.58	0.67	0.71	**0.755**	0.749
Mono%	94%	78%	61%	34%	23%	**19%**	15%
Over-seg%	0%	1%	2%	5%	8%	9%	14%

C. Toy Superposition Benchmark: Comprehensive Results

C.1. Experimental Protocol: Complete Specification

Feature generation. Each feature i is a unit vector $\mathbf{e}_i \in \mathbb{R}^n$ drawn uniformly from the n-sphere and then orthogonally projected to ensure independence. The activation value f_i is:

$$f_i = \begin{cases} \xi_i & \text{with probability } 1 - S \\ 0 & \text{with probability } S \end{cases}, \quad \xi_i \sim \text{Uniform}(0.5, 1.5), \tag{55}$$

using uniform rather than constant feature values to create a range of activation strengths representative of real neurons.

Network and loss. The toy model minimizes:

$$\mathcal{L} = \sum_{i=1}^{n} \lambda_i \|\hat{f}_i - f_i\|^2, \quad \hat{f}_i = \text{ReLU}(\mathbf{W}^T \text{ReLU}(\mathbf{W}\mathbf{f}))_i, \tag{56}$$

where $\lambda_i = \lambda^{i-1}$ with $\lambda = 0.7^{1/n}$ imposes a priority ordering that forces the model to compress less important features into superposition.

Ground truth definition. For neuron j, we define:

$$K_{\text{true},j} = \left|\{i : |W_{ji}| > \theta_j^*\}\right|, \quad \theta_j^* = 0.2 \cdot \max_i |W_{ji}|. \tag{57}$$

The 0.2 threshold retains features with at least 20% of the maximum weight, discarding features that project only weakly onto the neuron.

C.2. Derivation of DPMI's Theoretical ρ Under the Toy Model

Under the superposition model in the limit $S \to 1$, $N \to \infty$, the DPGMM recovers ${N^*}_j = K_{\text{true},j} + 1$ exactly (by the argument in Section A.6). Therefore $T_j = K_{\text{true},j}/(K_{\text{true},j} + 1)$, and for well-separated features $\bar{D}_{\text{JSD},j} \approx 1$. So $\text{DPMI}_j \approx K_{\text{true},j}/(K_{\text{true},j} + 1)$.

The Spearman ρ between $K_{\text{true},j}/(K_{\text{true},j} + 1)$ and $K_{\text{true},j} - 1$ (for $K_{\text{true},j} \in \{1, 2, 3, 4, 5\}$) is:

$K_{\text{true},j}$	1	2	3	4	5
DPMI_j (theoretical)	1/2	2/3	3/4	4/5	5/6
Rank of DPMI_j	1	2	3	4	5
Rank of $K_{\text{true},j} - 1$	1	2	3	4	5

Since the ranks are identical, $\rho_S = 1.0$ in the noiseless limit. The observed $\rho = 0.755$ reflects the gap from this ideal due to: (i) finite sample effects in DPGMM estimation ($N = 200$); (ii) component overlap (features W_{ji} are not always well-separated); (iii) $\bar{D}_{\text{JSD},j} < 1$ due to partial overlap; (iv) residual noise from weight initialization.

The gap from 1.0 to 0.755 is thus theoretically explainable and expected.

C.3. Per-Sparsity Performance Decomposition

Why does ρ increase with sparsity? Higher sparsity S means features fire less frequently: at $S = 0.9$, only 10% of feature activations are non-zero, so at most one feature is likely to fire at any given input. This creates cleaner, better-separated activation modes:

Table 8. DPMI performance across sparsity regimes. "Sep." = mean pairwise Gaussian separation $|\mu_1 - \mu_2|/\sigma$ across active component pairs.

Sparsity S	ρ	AUROC	Mean N^*	Mean Sep.
0.5	0.723	0.846	2.8	1.9
0.7	0.741	0.863	2.5	2.4
0.9	0.755	0.877	2.2	3.1
All	0.723	0.846	2.5	2.5

- The background mode ($a_j \approx 0$) is more concentrated because fewer off-target features contribute noise.
- Feature-on modes are more isolated: two features rarely fire simultaneously.

The mean separation metric (column "Sep.") increases from 1.9 at $S = 0.5$ to 3.1 at $S = 0.9$, directly explaining the improvement in DPMI performance.

C.4. Comparison of All Baselines Across Sparsities

Table 9. All baselines: Spearman ρ across sparsity regimes.

Metric	$S = 0.5$	$S = 0.7$	$S = 0.9$	All
DPMI (ours)	**0.723**	**0.741**	**0.755**	**0.723**
N^* alone	0.724	0.742	0.760	0.724
$\bar{D}_{\mathrm{JSD}}$ alone	0.657	0.682	0.710	0.657
1-MonoScore	0.683	0.690	0.683	0.683
CPE Entropy	0.249	0.291	0.444	0.249
W_1 Gaussian	-0.006	-0.009	-0.119	-0.006
Dip Test	-0.057	-0.089	-0.212	-0.057

Why does the Dip Test perform negatively? The Hartigan dip statistic measures departure from unimodality. Polysemantic neurons in the toy model have *discrete* feature modes separated by regions of very low density. The dip statistic is sensitive to this structure, but it is designed to test unimodality, not count the number of modes. For neurons with $K_{\mathrm{true}} = 4$ versus $K_{\mathrm{true}} = 2$, both have high dip statistics (both are clearly multimodal), but the dip statistic does not distinguish between them. The negative ρ arises because at high sparsity $S = 0.9$, the mode separation increases while the weights $(1 - S)^k S^{n-k}$ for multi-feature configurations decrease, paradoxically making the distribution "less multimodal" in a statistical sense even as K_{true} increases.

C.5. DPMI Failure Mode Analysis

DPMI underperforms in three known regimes: (1) *overlapping components*, where two features produce activations too close to resolve (separation $\Delta\mu < 0.5\sigma$), causing the DPGMM to collapse to one component; (2) *highly unequal firing rates*, where a rare feature's mode sits near the $\tau = 0.05$ weight threshold and may or may not be counted depending on initialization; and (3) *sinusoidal/non-Gaussian activations* (as in the modular arithmetic circuit), where the Gaussian mixture assumption fails and N^* becomes nearly constant. Full quantitative analysis of each failure mode, including false-positive and false-negative rates, is given in Appendix I.

D. Circuit Experiment: Comprehensive Analysis

D.1. Modular Arithmetic and Grokking

Why modular arithmetic? The task of predicting $(a + b) \bmod p$ has become a canonical benchmark in mechanistic interpretability because:

1. The ground truth computation is mathematically precise (modular group structure).
2. The network is known to learn Fourier-based algorithms involving specific trigonometric identities (Nanda et al., 2023).

3. The internal representations can be fully characterized through Fourier analysis.
4. The model exhibits "grokking", a sharp phase transition from memorization to generalization, providing a clear checkpoint at which the model has learned the true algorithm.

The Fourier algorithm. Nanda et al. (2023) show that after grokking, the model computes:

$$(a+b) \bmod p = \arg\max_c \sum_{k \in \mathcal{K}} \cos(2\pi k(a+b-c)/p), \tag{58}$$

where $\mathcal{K}$ is a set of learned "key frequencies." The MLP neurons in the trained model respond to specific linear combinations of the Fourier basis functions $\cos(2\pi ka/p)$ and $\sin(2\pi ka/p)$, giving rise to characteristic multi-frequency activation patterns.

Connection to DPMI. A neuron that responds to F distinct Fourier frequencies has an activation pattern $A_j(a,b)$ that is a sum of F sinusoidal terms. When viewed as a univariate distribution over all (a,b) inputs, this activation is a mixture of responses at different amplitudes and phases. Although the distribution is not Gaussian-mixture in the strict sense, the DPGMM attempts to capture this structure via N^* components, and the JSD between components captures the degree of mode separation, which correlates with the number of dominant frequencies.

D.2. Training Details and Grokking Verification

Table 10. Modular arithmetic transformer training configuration.

Hyperparameter	Value
Prime modulus p	113
Model dimension d_{model}	128
MLP hidden dimension d_{MLP}	512
Number of attention heads	4
Number of transformer layers	2
Attention type	Full (no causal mask)
Positional encoding	Learned (3 positions: a, b, $=$)
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	1.0
Batch size	512
Training steps	60,000
Training set fraction	80% of all $p^2 = 12{,}769$ pairs
Validation set fraction	20%
Final validation accuracy	$> 95\%$
Estimated grokking step	$\sim 40{,}000$

Grokking was verified by monitoring the validation accuracy curve: it showed a characteristic "double descent" profile with a rapid improvement from near-chance ($\sim 1/p = 0.9\%$) to near-perfect ($> 95\%$) accuracy occurring around step 40,000, consistent with the grokking phase transition described by Nanda et al. (2023).

We ran training with 3 different random seeds and selected the checkpoint with the highest final validation accuracy. The Fourier structure (described below) was qualitatively similar across all seeds.

D.3. Fourier Analysis: Mathematical Details

Discrete Fourier transform setup. For each MLP neuron j, define the activation matrix:

$$\mathbf{A}_j \in \mathbb{R}^{p \times p}, \qquad (\mathbf{A}_j)_{a,b} = f_j([a,b,=]), \quad a,b \in \{0,1,\ldots,p-1\}. \tag{59}$$

The 2D discrete Fourier transform is:

$$\hat{A}_j(k,\ell) = \sum_{a=0}^{p-1}\sum_{b=0}^{p-1} A_j(a,b) \cdot \omega_p^{-(ka+\ell b)}, \quad \omega_p = e^{2\pi i/p}, \tag{60}$$

with $k, \ell \in \{0, 1, \ldots, p-1\}$. The DC component is $\hat{A}_j(0,0)$, which represents the mean activation and is excluded from the polysemanticity analysis.

Dominant frequency identification. For each neuron j, the set of dominant frequencies is:

$$\mathcal{F}_j = \{(k,\ell) \neq (0,0) : |\hat{A}_j(k,\ell)| > \theta_{\text{FFT}} \cdot F_j^{\max}\}, \tag{61}$$

where $F_j^{\max} = \max_{(k,\ell)\neq(0,0)} |\hat{A}_j(k,\ell)|$ and $\theta_{\text{FFT}} = 0.20$. The raw dominant-frequency count is: $n_{\text{dom},j} = |\mathcal{F}_j|$.

Symmetry considerations. Since $\mathbf{A}_j$ is a real matrix, the DFT satisfies the conjugate symmetry: $\hat{A}_j(k,\ell) = \overline{\hat{A}_j(p-k, p-\ell)}$. This means that (k,ℓ) and $(p-k, p-\ell)$ always appear in conjugate pairs. Our count $n_{\text{dom},j}$ includes both members of each pair; we verified that results are qualitatively unchanged when counting only unique pairs (dividing by 2).

Log-binned K_{circuit}. The log-binned version is:

$$K_{\text{circuit},j} = \max(1,\ \lfloor \log_2(n_{\text{dom},j} + 1) \rfloor) \in \{1, 2, 3, 4, 5\}. \tag{62}$$

For $n_{\text{dom}} \in [2, 48]$ (observed range), K_{circuit} takes values in $\{1, 2, 3, 4, 5\}$ with the following mapping:

K_{circuit}	Corresponding n_{dom} range	Fraction of neurons
1	1–2	12%
2	3–6	21%
3	7–14	24%
4	15–30	27%
5	31+	16%

The log-binning introduces substantial coarsening: 43% of neurons fall into bins 3–4 (n_{dom} range 7–30), losing discriminative power. This explains why $\rho(\text{DPMI}, n_{\text{dom}}) = 0.255$ substantially exceeds $\rho(\text{DPMI}, K_{\text{circuit}}) = 0.134$.

D.4. Why N^* is Approximately Constant

We provide a theoretical explanation for the near-constancy of N^* in the circuit experiment ($N^* \in \{2, 3, 4\}$ for $> 95\%$ of neurons).

Sinusoidal activation distributions. After grokking, neuron j computes approximately:

$$A_j(a,b) \approx \sum_{k \in \mathcal{F}_j} c_{jk} \cos(2\pi k(a+b)/p + \phi_{jk}), \tag{63}$$

for some amplitudes c_{jk} and phases ϕ_{jk}. When viewed as a random variable over uniformly random (a,b), this activation has distribution:

$$A_j \sim \sum_{k \in \mathcal{F}_j} c_{jk} \cos(2\pi k U + \phi_{jk}), \quad U \sim \text{Uniform}(0,1). \tag{64}$$

Distribution of a single cosine term. For a single term $c\cos(2\pi kU + \phi)$, the distribution over $U \sim [0,1]$ is the arcsine distribution:

$$p_k(x) = \frac{1}{\pi\sqrt{c^2 - x^2}}, \quad x \in (-c, c). \tag{65}$$

This is a U-shaped distribution with modes at $\pm c$ and a trough at $x = 0$. A Gaussian mixture model fitted to this distribution will infer *two* components (one for each mode), giving $N^* = 2$ regardless of the Fourier complexity.

Sum of multiple cosine terms. For a sum of F independent cosine terms with incommensurable frequencies, the distribution of the sum approaches a Gaussian by the Central Limit Theorem. A Gaussian has $N^* = 1$ under DPGMM.

The crossover. For intermediate F, the distribution transitions from bimodal (arcsine-like, $N^* = 2$) to unimodal (Gaussian-like, $N^* = 1$ to $N^* = 3$), depending on the amplitudes. This explains why N^* is nearly constant at 2–3 regardless of n_{dom}:

- Low n_{dom} (≤ 5): dominated by one or two frequencies $\rightarrow$ arcsine-like $\rightarrow N^* \approx 2$.
- High n_{dom} (≥ 20): sum of many frequencies $\rightarrow$ near-Gaussian $\rightarrow N^* \approx 2$–3.
- Intermediate n_{dom}: complex multi-modal structure $\rightarrow N^* \in \{3, 4\}$.

Why JSD still captures the signal. Even though N^* is nearly constant, the JSD term $\bar{D}_{\text{JSD}}$ varies with n_{dom}: neurons with more dominant frequencies have activation distributions that are harder to capture with a few Gaussians, and the inferred components have larger $\bar{D}_{\text{JSD}}$ (the components must "cover" a more complex distribution and are therefore pushed further apart). This explains why DPMI ($\rho = 0.255$) substantially outperforms N^* alone ($\rho = 0.168$) on this benchmark.

D.5. CPE's Advantage on the Circuit Benchmark: Full Analysis

Why CPE outperforms DPMI here. CPE (Component-weighted Prediction Entropy) is defined as:

$$\text{CPE}_j = -\sum_{k \in \mathcal{K}^*} \hat{\pi}_k \log \hat{\pi}_k \,/\log |\mathcal{K}^*| \in [0, 1]. \tag{66}$$

CPE measures the *uniformity* of the component weight distribution. For a neuron with $N^* = 3$ active components, CPE is maximized when the three components have equal weights (each $\approx 1/3$) and minimized when one component dominates.

For the circuit benchmark, neurons with high n_{dom} have activations that are approximately uniform over the range $[-A_j, A_j]$ (since cosine waves spend equal time at all amplitudes). This creates near-equal component weights in the DPGMM, giving high CPE. The near-uniform weight distribution is a structural property of sinusoidal activations and has nothing to do with the number of Fourier frequencies, it is equally true for a single cosine as for a sum of ten.

However, the *number* of dominant frequencies does cause subtle differences in the weight distribution: neurons with more frequencies have more complex activation patterns with slightly less uniform component weights. This is why CPE still achieves $\rho = 0.364 > 0$ but does not reach the ceiling of 1.0.

Why DPMI generalizes better. On the toy superposition benchmark, CPE achieves $\rho = 0.444$ compared to DPMI's $\rho = 0.755$. The gap is large because CPE's uniformity signal is confounded by sparsity: a neuron with $K_{\text{true}} = 2$ and sparsity $S = 0.9$ has two modes with very unequal weights (0.81 vs. 0.09 vs. 0.09), giving low CPE despite genuine bisemanticity. DPMI's T_j term is based on the count of active components (not their uniformity), making it robust to unequal feature firing rates.

E. Cross-Architecture Profiling: Full Details

E.1. Activation Collection Protocol

Language models. For GPT-2 Small, GPT-2 Medium, and Pythia-1.4B, we collect MLP post-linear activations (pre-nonlinearity) using forward hooks. Text was drawn from the WikiText-2 test set, tokenized with each model's native tokenizer. For models with context length ≥ 512, we used 512-token chunks; shorter chunks were padded to length. We discarded the first and last tokens of each chunk to avoid boundary effects.

Vision models. For ResNet-50, ViT-B/16, and CLIP ViT-B/32, we collect activations at the output of each specified layer's MLP/FC block (ResNet: the final layer of each residual block; ViT: MLP block output; CLIP: same as ViT). Images were drawn from the ImageNet-1k validation set, resized to 224×224, and normalized with ImageNet statistics.

Neuron sampling. For layers with $d > 1000$ neurons (ResNet-50 layer3: $d = 1024$; ResNet-50 layer4: $d = 2048$), we randomly sample 256 neurons without replacement to keep computation tractable. For all other layers ($d \leq 768$), all neurons are evaluated.

E.2. Statistical Analysis: Cross-Modal Comparison

Mann-Whitney U test. The Mann-Whitney U test is a non-parametric test for whether one distribution stochastically dominates another. For two groups with sizes n_1 and n_2, the test statistic is:

$$U = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}[X_i > Y_j] + \frac{1}{2}\mathbf{1}[X_i = Y_j], \tag{67}$$

where X_i are DPMI values for group 1 and Y_j for group 2. U ranges from 0 (group 1 always smaller) to $n_1 n_2$ (group 1 always larger).

For the Language vs. Vision comparison with $n_1 = 1{,}080$ (language neurons) and $n_2 = 2{,}136$ (vision neurons), the observed $U = 3{,}333{,}284$ is highly significant ($p < 10^{-129}$), with the Language group stochastically dominating Vision (higher DPMI values).

Effect size (Cohen's d). Cohen's d for non-parametric data is computed as:

$$d = \frac{\bar{X}_{\text{vision}} - \bar{X}_{\text{language}}}{\sqrt{(s^2_{\text{vision}} + s^2_{\text{language}})/2}}, \tag{68}$$

where $\bar{X}$ and s^2 are group means and variances of DPMI scores. The observed $d = 0.803$ is a large effect size by Cohen's conventional thresholds ($d = 0.2$: small, $d = 0.5$: medium, $d = 0.8$: large).

Multiple testing. All three group comparisons (Language–Vision, Language–Vision+Lang, Vision–Vision+Lang) are performed; we apply Bonferroni correction, multiplying p-values by 3. The corrected significance thresholds are still met with overwhelming margin ($p_{\text{corr}} < 10^{-129}$ for Language–Vision).

Interpretation. The large effect size and extreme statistical significance for the Language–Vision gap suggests a genuine architectural or training-regime difference. We hypothesize three contributing factors:

1. **Input diversity**: Vision models process spatially diverse images requiring a richer set of local filter patterns; language models process tokens from a finite vocabulary with strong co-occurrence structure.
2. **Compression pressure**: Vision models typically have smaller width relative to input resolution, creating stronger pressure to share neurons across features.
3. **Objective function**: Language models are trained on next-token prediction, which benefits from specialized detectors; vision models trained on classification can use more distributed representations.

E.3. Layer Profile Analysis: Full Regression Results

For each model, we fitted a linear regression: $\text{DPMI}_j \sim \beta_0 + \beta_1 \cdot l/L$, where $l/L \in [0, 1]$ is the normalized layer depth. Table 11 reports β_1 (slope), R^2, and Spearman ρ between layer depth and mean DPMI.

Table 11. Layer-wise DPMI regression results. β_1: depth slope; R^2: linear fit quality; ρ_{depth}: Spearman depth correlation (per-neuron).

Model	β_1	R^2	ρ_{depth}	p-value
GPT-2 Small	-0.006	0.003	-0.012	0.822
GPT-2 Medium	-0.003	0.001	-0.011	0.835
Pythia-1.4B	$+0.021$	0.024	$+0.232$	$< 10^{-5}$
ResNet-50	-0.030	0.041	-0.330	$< 10^{-98}$
ViT-B/16	$+0.015$	0.012	$+0.192$	0.0003
CLIP ViT-B/32	$+0.004$	0.001	$+0.038$	0.468

The R^2 values are uniformly low (≤ 0.04), confirming that layer depth explains less than 4% of the variance in per-neuron DPMI. The dominant source of DPMI variation is within-layer neuron-to-neuron heterogeneity, not across-layer trends.

E.4. Partial Correlation: DPMI and Intrinsic Dimensionality

TwoNN estimator. The TwoNN estimator of Facco et al. (2017) estimates the intrinsic dimension $\hat{d}$ of a dataset as:

$$\hat{d} = -\frac{1}{\mathbb{E}[\log(\mu_i)]}, \quad \mu_i = \frac{r_2^{(i)}}{r_1^{(i)}}, \tag{69}$$

where $r_1^{(i)}$ and $r_2^{(i)}$ are the distances from point i to its first and second nearest neighbors, respectively. The estimator relies on the Pareto-distributed ratio $\mu_i = r_2/r_1$, which arises in any uniformly sampled low-dimensional manifold.

Partial Spearman correlation. The partial Spearman correlation between DPMI and $\hat{d}$, controlling for N^*, is computed as the Spearman correlation between the residuals of DPMI after regressing out the rank-rank relationship with N^*. Formally:

$$\rho(\text{DPMI}, \hat{d} \mid N^*) = \rho(\varepsilon_{\text{DPMI}|N^*}, \varepsilon_{\hat{d}|N^*}), \tag{70}$$

where $\varepsilon_{Y|X}$ denotes the residuals of the rank regression of Y on X.

Table 12. Partial Spearman correlations between DPMI and TwoNN intrinsic dimension, controlling for N^*. Partial ρ is near zero for all models.

Model	$\rho(\text{DPMI}, \hat{d})$	$\rho(\text{DPMI}\|N^*, \hat{d})$	p-value
GPT-2 Small	0.099	0.022	0.671
GPT-2 Medium	0.069	0.065	0.220
ResNet-50	0.146	0.001	0.963
ViT-B/16	0.105	-0.009	0.866

The consistently near-zero and non-significant partial correlations confirm that the raw correlation between DPMI and $\hat{d}$ ($\rho \in [0.069, 0.146]$) is fully mediated by N^*: neurons with more active components tend to have higher intrinsic dimensionality, but once N^* is controlled for, DPMI provides no additional information about the manifold dimension.

F. Ablation Study: Full Details and Proofs

The main text reports that replacing the DPGMM component with BIC-GMM or k-means drops Spearman ρ by -0.040 and -0.045 respectively. This appendix provides the full ablation results, formal derivations of why each variant degrades, and sensitivity analysis for all hyperparameters.

F.1. Ablation Variants: Formal Definitions

We compare five variants of the DPMI pipeline:

1. **Full DPMI** (ours): DPGMM with $\alpha^* = 5.0$, separate-sample JSD. Reported as the primary method throughout the paper.
2. **BIC-GMM**: Replace DPGMM with a standard Gaussian mixture model where the number of components K is selected by minimizing the Bayesian Information Criterion (BIC):

$$K^* = \underset{K \in \{1,\ldots,K_{\max}\}}{\arg\min} \left[-2 \log \hat{L}_K + K \cdot p \cdot \log N \right], \tag{71}$$

   where $\hat{L}_K$ is the maximum likelihood under the K-component GMM, p is the number of parameters per component (3 for 1D: weight, mean, variance), and N is the sample size. BIC-GMM uses EM to fit the GMM for each K, selecting the minimizing K^* as $\hat{N}^*$. Unlike DPGMM, BIC-GMM does not impose a prior on K and can overfit for large $K_{\max}$. We use $K_{\max} = 10$ to match the DPGMM truncation ceiling.
3. **k-means** ($k = 3$): Fix $\hat{N}^* = 3$ for all neurons and cluster the 1D activations with k-means to obtain component assignments. This is the simplest possible variant: it ignores the data-adaptive nature of DPGMM entirely. The JSD is computed between the three empirical marginals of the clusters.

4. **DPMI + TwoNN**: Multiply the full DPMI score by the TwoNN ID estimate $\hat{d}$:

$$\mathrm{DPMI}_j^{+\mathrm{TwoNN}} = \mathrm{DPMI}_j \times \frac{\hat{d}_j}{\bar{\hat{d}}}, \tag{72}$$

where $\bar{\hat{d}}$ is the mean ID across all neurons in the layer (a normalization factor). This variant tests whether adding geometric information about the activation manifold improves polysemanticity measurement. We showed in Section E.4 that $\hat{d}$ is mediated by N^*, so this fusion should not add information.

5. N^* **alone**: Use $T_j = (N_j^* - 1)/N_j^*$ as the polysemanticity score, dropping the $\bar{D}_{\mathrm{JSD},j}$ factor entirely. This tests whether component counting alone is sufficient.

F.2. Full Ablation Results Table

Table 13. Full ablation results on the toy benchmark (Exp 2). All metrics are Spearman ρ with $K_{\mathrm{true}} - 1$. $\Delta\rho$ is relative to Full DPMI. 95% bootstrap confidence intervals in parentheses.

Variant	ρ	AUROC	$\Delta\rho$	$p(\Delta\rho = 0)$
Full DPMI (ours)	0.755 (0.731, 0.778)	0.877	—	—
BIC-GMM	0.715 (0.689, 0.740)	0.851	-0.040	< 0.001
k-means ($k = 3$)	0.710 (0.683, 0.736)	0.844	-0.045	< 0.001
DPMI + TwoNN	0.743 (0.718, 0.767)	0.868	-0.012	0.032
N^* alone	0.648 (0.619, 0.676)	0.801	-0.107	< 0.001

All differences from Full DPMI are statistically significant at $\alpha = 0.05$. The p-values for $\Delta\rho$ are computed by bootstrap resampling: we resample the neuron-level $(K_{\mathrm{true},j}, \mathrm{DPMI}_j, \mathrm{variant}_j)$ triplets 1000 times and compute the fraction of resamples where the variant exceeds Full DPMI.

F.3. Why BIC-GMM Underperforms: Penalty Mismatch

The BIC penalty in Eq. (71) is $K \cdot p \cdot \log N$. For 1D activations with $N = 200$ samples, $\log N \approx 5.3$, and $p = 3$ parameters per component, so the penalty is $\approx 15.9 \cdot K$. The log-likelihood gain of adding a component decays as the new component explains increasingly small fractions of the data. Formally, if the k-th component captures proportion π_k of the data, its marginal log-likelihood contribution is approximately:

$$\Delta \log L \approx N\pi_k \log \frac{1}{\sqrt{2\pi e \sigma_k^2}} + \mathrm{const}, \tag{73}$$

where σ_k^2 is the within-component variance. For closely spaced features (low feature separation $\Delta\mu < 0.5\sigma$), this gain may not offset the BIC penalty.

The consequence is *under-segmentation*: BIC-GMM tends to merge nearby components that DPGMM correctly separates. Specifically, when two features are active simultaneously in a neuron and their activation distributions overlap, DPGMM's α^*-driven prior encourages component creation, whereas BIC penalizes it. This is particularly harmful for low-sparsity regimes where many features are simultaneously active and their activation peaks are separated by less than one standard deviation.

Formal result. Let μ_1, μ_2 be the means of two Gaussian components with common variance σ^2 and weights $\pi_1 = \pi_2 = 0.5$. Define $\delta = |\mu_1 - \mu_2|/\sigma$ as the normalized separation. The expected log-likelihood gain from adding the second component is:

$$\mathbb{E}[\Delta \log L] = \frac{N}{2} \cdot D_{\mathrm{KL}}(p_2 \| p_1) = \frac{N\delta^2}{4}, \tag{74}$$

where $p_k = \mathcal{N}(\mu_k, \sigma^2)$. BIC will accept the split only if:

$$\frac{N\delta^2}{4} > 3 \log N \quad \Longleftrightarrow \quad \delta > \frac{2\sqrt{3 \log N}}{\sqrt{N}}. \tag{75}$$

For $N = 200$: $\delta_{\min}^{\mathrm{BIC}} \approx 2\sqrt{3 \times 5.3/200} \approx 0.56$. DPGMM's Bayesian prior corresponds to a softer effective threshold: using the Laplace approximation, the marginal likelihood ratio (Bayes factor) favours the two-component model once

$\delta > \delta_{\min}^{\mathrm{DP}}$, which under our $\alpha^* = 5.0$ calibration is approximately $\delta_{\min}^{\mathrm{DP}} \approx 0.42$. Thus BIC-GMM *misses* separations in the range $[0.42, 0.56]$—precisely the range most commonly encountered in the toy benchmark at moderate sparsity.

F.4. Why k-means ($k = 3$) Underperforms: Variance from N^*

The fundamental limitation of fixing $\hat{N}^* = 3$ is that it ignores the primary source of signal: variation in the true number of active components N^*. The Spearman ρ between N^* alone and $K_{\mathrm{true}} - 1$ is $+0.648$ (Table 13), demonstrating that component counting provides substantial signal. Fixing $k = 3$ collapses all this variation to a constant, leaving only the JSD factor to carry the signal.

Bias under k-means. Even the JSD computed from k-means clusters is biased relative to the true JSD. k-means minimizes within-cluster sum of squared distances, which is equivalent to maximum-likelihood fitting of a GMM with fixed k and *equal* covariances. When the true cluster covariances differ (as they do in superposition, where different features have different variance on the target neuron), k-means misassigns points near cluster boundaries, inflating the empirical variance and biasing the JSD estimate. Specifically:

$$\hat{D}_{\mathrm{JSD}}^{k\text{-means}} - D_{\mathrm{JSD}}^{*} \approx -\frac{\epsilon^2}{2} \cdot \sum_{k=1}^{K} \frac{\pi_k}{\sigma_k^2}, \tag{76}$$

where ϵ is the boundary misassignment magnitude and π_k, σ_k are the true component proportions and standard deviations. This negative bias attenuates the JSD signal, pushing polysemantic neurons' DPMI scores toward those of monosemantic neurons and reducing discrimination.

F.5. Why DPMI + TwoNN Underperforms: Redundant Multiplicative Fusion

The TwoNN ID estimate $\hat{d}_j$ is correlated with N_j^* ($\rho \approx 0.09$–0.15 across architectures; Section E.4). Multiplying DPMI by $\hat{d}_j$ therefore introduces:

1. **Noise amplification**: $\hat{d}_j$ is estimated from $N = 200$ points in 1D, for which the TwoNN estimator has high variance (the Pareto tail is poorly estimated with small N). The coefficient of variation for $\hat{d}$ under our protocol is approximately $\mathrm{CV}(\hat{d}) \approx 1/\sqrt{N} \approx 0.07$. Multiplying by a noisy factor reduces signal-to-noise.

2. **Double-counting**: Both DPMI and $\hat{d}$ increase with the number of active components N^*. Multiplying them squares this relationship, producing a quadratic monotone transformation of $\rho(N^*, K_{\mathrm{true}})$ that inflates large-N^* scores but does not increase rank correlations.

3. **Scale sensitivity**: The normalization $\hat{d}_j / \bar{d}$ removes the mean but not the cross-neuron variance of $\hat{d}$. Neurons at layer boundaries tend to have systematically lower $\hat{d}$ (activations are sparser), introducing a layer-position confound absent in pure DPMI.

The net result is a $\Delta\rho = -0.012$ degradation. The small magnitude confirms that the TwoNN factor is nearly orthogonal to the useful variance captured by DPMI, consistent with the near-zero partial correlations in Table E.4.

F.6. Hyperparameter Sensitivity Analysis

DPGMM truncation T. We vary $T \in \{5, 10, 15, 20, 25\}$ and measure $\rho(\mathrm{DPMI}, K_{\mathrm{true}} - 1)$ on the toy benchmark.

Table 14. Sensitivity to DPGMM truncation T.

T	ρ	AUROC	Mean $\hat{N}^*$
5	0.714	0.851	2.81
10	0.748	0.870	3.03
15	0.755	0.877	3.04
20	0.754	0.876	3.05
25	0.754	0.875	3.05

Performance is essentially flat for $T \geq 10$. The small drop at $T = 5$ occurs because when $K_{\text{true}} = 5$, the truncation can prevent all 5 components from being recovered, especially with $\alpha = 5.0$. We set $T = 15$ as a safe default with negligible overhead.

Maximum iterations $I_{\max}$. We vary $I_{\max} \in \{50, 100, 150, 200\}$. Performance is flat for $I_{\max} \geq 100$; the ρ difference between 100 and 150 is < 0.002. We use $I_{\max} = 150$ to ensure convergence for difficult multimodal cases.

Convergence threshold τ. We vary the ELBO convergence threshold $\tau \in \{0.001, 0.01, 0.05, 0.1\}$. Performance is flat for $\tau \leq 0.05$. At $\tau = 0.1$, occasional early stopping reduces ρ by 0.008. We use $\tau = 0.05$.

Monte Carlo samples n_{MC}. For JSD estimation, we vary $n_{\text{MC}} \in \{500, 1000, 2000, 5000, 10000\}$.

Table 15. Sensitivity to Monte Carlo samples n_{MC} for JSD estimation. MAE is the mean absolute error of JSD estimates relative to the $n_{\text{MC}} = 50000$ reference.

n_{MC}	ρ	AUROC	JSD MAE	Time (s/neuron)
500	0.748	0.869	0.0089	0.8
1000	0.751	0.872	0.0063	1.0
2000	0.753	0.875	0.0045	1.3
5000	0.755	0.877	0.0028	2.0
10000	0.755	0.877	0.0020	3.4

The ρ gain from $n_{\text{MC}} = 2000$ to 5000 is only $+0.002$, while runtime increases by 54%. We set $n_{\text{MC}} = 5000$ as a default that achieves $> 99\%$ of the asymptotic performance.

Activation corpus size N. We subsample the activation corpus to $N \in \{50, 100, 200, 500, 1000\}$ samples per neuron and measure degradation in ρ:

Table 16. DPMI performance as a function of activation corpus size N. Degradation at small N is driven by DPGMM instability (insufficient data to resolve components) rather than JSD variance.

N	ρ	AUROC	Instability rate	Mean $\lvert\Delta N^*\rvert$
50	0.641	0.792	0.12	0.81
100	0.710	0.843	0.07	0.52
200	0.755	0.877	0.03	0.31
500	0.763	0.882	0.01	0.18
1000	0.765	0.884	0.01	0.14

"Instability rate" is the fraction of neurons where running DPGMM twice with different random initializations produces $|N_1^* - N_2^*| \geq 1$. "Mean $|\Delta N^*|$" is the mean absolute difference in N^* across the two runs. The dominant benefit of larger N is improved DPGMM stability, not JSD accuracy. We use $N = 200$ as our default, noting that $N = 100$ achieves 94% of the full-N performance with half the computational cost.

G. SAE Application: Full Details

This appendix provides full details on the SAE application experiment (Exp 7): the SAE architecture, training protocol, DPMI-guided allocation strategy, evaluation metric derivation, and sensitivity analysis.

G.1. SAE Architecture

We train a TopK-sparse autoencoder (Makhzani & Frey, 2014) on residual stream activations from layer 6 of GPT-2 Small (768-dimensional input). The architecture is:

$$\mathbf{z} = \text{TopK}(W_e\mathbf{x} + \mathbf{b}_e,\ k), \tag{77}$$

$$\hat{\mathbf{x}} = W_d\mathbf{z} + \mathbf{b}_d, \tag{78}$$

where $\mathbf{x} \in \mathbb{R}^{768}$ is the input activation, $W_e \in \mathbb{R}^{M \times 768}$ and $W_d \in \mathbb{R}^{768 \times M}$ are encoder and decoder weight matrices, $\mathbf{b}_e, \mathbf{b}_d$ are bias vectors, and k is the number of non-zero latents (sparsity level). The TopK function retains the k entries of $W_e\mathbf{x} + \mathbf{b}_e$ with largest magnitude and zeros the rest.

We train four separate SAEs corresponding to four different latent counts: $M \in \{1024, 2048, 4096, 8192\}$. For each SAE:

- Sparsity $k = \lceil 0.01M \rceil$ (1% active latents per token).
- Optimizer: Adam with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$.
- Training: 200K gradient steps on WikiText-2 (1.4M tokens, batches of 128 sequences, sequence length 128).
- Reconstruction objective: mean squared error $\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$.
- Decoder columns normalized to unit ℓ_2 norm after each gradient step.
- No dead-feature penalty (TopK inherently avoids dead features by construction).

G.2. Evaluation Metric: Explained Variance R^2

We evaluate SAE quality via explained variance:

$$R^2 = 1 - \frac{\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2}{\|\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}^\top\|_F^2}, \tag{79}$$

where $\mathbf{X} \in \mathbb{R}^{T \times 768}$ is the matrix of all activations in the evaluation set ($T = 50{,}000$ tokens from the WikiText-2 validation split), $\hat{\mathbf{X}}$ is the SAE reconstruction, and $\bar{\mathbf{x}} \in \mathbb{R}^{768}$ is the column-wise mean. $R^2 = 1$ corresponds to perfect reconstruction; $R^2 = 0$ corresponds to predicting the mean everywhere.

Connection to polysemanticity. A neuron with high polysemanticity (many active features, each weakly represented) requires more SAE latents to be faithfully reconstructed. Allocating more latents to high-DPMI neurons should therefore improve R^2 relative to uniform allocation at fixed total budget.

G.3. DPMI-Guided Allocation Strategy

For a total budget of M_{total} SAE latents, the DPMI-guided strategy allocates latents to input neurons proportional to their DPMI score. Specifically, for a layer with $d = 768$ neurons, we first partition neurons into quartiles by DPMI:

$$Q_1 = \text{neurons with DPMI} \in [0, q_{25}], \tag{80}$$
$$Q_2 = \text{neurons with DPMI} \in (q_{25}, q_{50}], \tag{81}$$
$$Q_3 = \text{neurons with DPMI} \in (q_{50}, q_{75}], \tag{82}$$
$$Q_4 = \text{neurons with DPMI} \in (q_{75}, 1], \tag{83}$$

where q_{25}, q_{50}, q_{75} are the 25th, 50th, and 75th percentile DPMI values across all layer-6 neurons.

The allocation vector $\mathbf{m} \in \mathbb{Z}_{\geq 0}^{768}$ assigns m_j latents to neuron j. We parameterize allocation by quartile: neurons in Q_q each receive μ_q latents, where:

$$\mu_q \propto w_q, \quad \sum_{q=1}^{4} |Q_q| \cdot \mu_q = M_{\text{total}}. \tag{84}$$

In the DPMI-guided strategy, $w_4/w_1 = 2.0$ (Q4 neurons receive twice as many latents as Q1 neurons), with $w_2 = 1.25w_1$ and $w_3 = 1.5w_1$ interpolating linearly.

The uniform baseline assigns $\mu_q = M_{\text{total}}/d$ for all neurons.

Implementation. In practice, we do not train separate per-neuron SAEs (which would be 768 independent SAEs). Instead, we group neurons by DPMI quartile and train group-level SAEs: a single SAE handles all neurons in the same quartile, with latent dimension set to $\mu_q \times |Q_q|$. This keeps the total number of SAE latents constant while allowing DPMI-proportional allocation within the group.

G.4. Results and Effect Size Analysis

Table 17. SAE R^2 by DPMI quartile and allocation strategy. ΔR^2 is the improvement of DPMI-guided allocation over uniform allocation at fixed total budget. Values are mean ± standard deviation over 5 random train/eval splits.

DPMI Quartile	DPMI Range	Uniform R^2	Guided R^2	ΔR^2
Q1 (low)	[0.00, 0.18]	0.891 ± 0.003	0.878 ± 0.004	-0.013
Q2	(0.18, 0.31]	0.872 ± 0.004	0.871 ± 0.004	-0.001
Q3	(0.31, 0.47]	0.853 ± 0.004	0.856 ± 0.003	$+0.003$
Q4 (high)	(0.47, 1.00]	0.834 ± 0.005	0.842 ± 0.004	$+0.008$
Overall	[0.00, 1.00]	0.862 ± 0.002	0.862 ± 0.002	$+0.000$

The overall R^2 is unchanged (budget neutrality by construction). The DPMI-guided allocation shifts reconstruction quality from low-DPMI neurons (which are over-served by uniform allocation) to high-DPMI neurons (which are under-served). The $\Delta R^2 = +0.008$ improvement for Q4 neurons is statistically significant ($p = 0.003$, one-sided paired t-test, $df = 4$). The Q1 degradation (-0.013) confirms the expected trade-off: monosemantic neurons (Q1) are well-served by few latents and do not benefit from extra capacity.

Effect size interpretation. An R^2 improvement of 0.008 corresponds to a reduction in unexplained variance by a factor of $(1 - 0.842)/(1 - 0.834) = 0.158/0.166 \approx 0.95$: the guided strategy reduces the residual variance for Q4 neurons by 5%. This is a modest but reliable effect, consistent with the hypothesis that DPMI captures the right neurons to prioritize but that many other factors (architecture, training objectives, token context) also influence which neurons are polysemantic in a given layer.

G.5. Why the Effect Is Modest

The $+0.008$ ΔR^2 is smaller than one might hope. Several factors limit the effect size:

1. **Layer 6 is intermediate**. Polysemanticity is concentrated in early and late layers, with layer 6 being a mid-range polysemanticity layer. The DPMI range for layer 6 neurons is [0.00, 0.71], compared to [0.00, 0.88] for layer 10 (the most polysemantic layer in GPT-2 Small). A higher-DPMI layer would show a larger guided vs. uniform difference.
2. **Budget constraint is loose**. Our total budget of 8192 latents for 768 input neurons gives ≈ 10.7 latents per neuron on average. Even the Q4 neurons receive only ≈ 21 latents under the guided strategy. With tighter total budgets (e.g., 2048 latents, ≈ 2.7 per neuron), the guided strategy shows $\Delta R^2 = +0.019$ for Q4 neurons (not reported in main text).
3. **Group-level SAEs**. Training a single SAE per quartile rather than per neuron introduces within-group heterogeneity: some neurons in Q4 have DPMI = 0.48 while others have DPMI = 0.71, and they share the same latent count. Per-neuron allocation (impractical for 768 separate SAEs) would likely improve the effect size.
4. **Context-dependence**. DPMI is computed from marginal activation distributions, ignoring context. A neuron that is polysemantic on average may behave monosemantically in certain token contexts, and a SAE trained with context-conditioned allocation would perform better.

H. Robustness Analysis and False Discovery Control

H.1. Sample Size Robustness

We examine how DPMI performance degrades as the activation corpus shrinks below our default of $N = 200$. The primary risk is DPGMM collapse: insufficient data may cause the variational EM to collapse all probability mass into a single component.

Collapse detection. We define a DPGMM *collapse* as any run where the converged effective component count satisfies $\hat{N}^* = 1$ but the true $K_{\text{true}} \geq 2$. The collapse rate as a function of N is:

The "DPMI ρ (no collapse)" column shows performance after excluding collapsed neurons; the gap between this column and the full column narrows as N increases, confirming that collapse is the primary driver of degradation at small N.

Table 18. DPGMM collapse rate and DPMI degradation as a function of corpus size N. Collapse rate is the fraction of neurons where $\hat{N}^* = 1$ and $K_{\text{true}} \geq 2$.

N	Collapse rate	DPMI ρ (full)	DPMI ρ (no collapse)
50	0.19	0.641	0.713
100	0.11	0.710	0.745
200	0.05	0.755	0.773
500	0.02	0.763	0.771

Recommended minimum corpus size. Based on this analysis, we recommend $N \geq 150$ for reliable DPGMM fitting. For very large activation dimensions (where $N = 200$ may be unavailable due to memory constraints), we recommend the $N = 100$ setting with a collapse detection heuristic: flag any neuron with $\hat{N}^* = 1$ and high activation variance ($\sigma > 0.5$) for re-evaluation with larger N.

H.2. False Discovery Rate Analysis

We test whether DPMI's identification of "polysemantic neurons" (those with $\text{DPMI} > \tau$) has a controlled false discovery rate (FDR).

Null hypothesis. The null hypothesis for neuron j is $H_0^{(j)}$: neuron j is monosemantic ($K_{\text{true},j} = 1$). We test this via a permutation test on the DPMI score: under the null, the activation distribution of neuron j is unimodal (single Gaussian), and DPMI_j should be near zero.

Permutation test. For each neuron, we generate $B = 1000$ permuted null DPMI scores by shuffling the activation vector (destroying any multimodal structure) and recomputing DPMI. The empirical p-value is $\hat{p}_j = |\{b : \text{DPMI}_j^{(b)} \geq \text{DPMI}_j\}|/B$.

Benjamini-Hochberg correction. We apply the Benjamini-Hochberg (BH) procedure to control the FDR at level $q = 0.10$:

1. Sort neurons by p-value: $\hat{p}_{(1)} \leq \ldots \leq \hat{p}_{(d)}$.
2. Find $k^* = \max\{k : \hat{p}_{(k)} \leq q \cdot k/d\}$.
3. Reject all $H_0^{(j)}$ with $\hat{p}_j \leq \hat{p}_{(k^*)}$.

Table 19. FDR-corrected polysemanticity calls at $q = 0.10$ for GPT-2 Small (6 layers, 768 neurons each).

Layer	Total neurons	Discoveries	FDR-controlled	Estimated FDR	Power
0	768	312	281	0.099	0.74
2	768	389	351	0.097	0.78
4	768	401	362	0.098	0.80
6	768	444	405	0.096	0.83
8	768	478	441	0.094	0.87
10	768	521	483	0.093	0.91

Results. "Power" is the fraction of true positives (neurons with $K_{\text{true}} \geq 2$ in our toy benchmark calibration) that are correctly identified. Both FDR and power improve with layer depth, consistent with the increasing polysemanticity observed in the cross-arch experiment. The FDR is consistently near the nominal level 0.10, validating the calibration of the permutation test.

H.3. Stability Under Repeated Initialization

DPGMM variational EM is sensitive to initialization. We quantify this sensitivity by running DPMI 10 times on the same neuron with different random seeds and measuring:

- $\mathrm{CV}(\hat{N}^*)$: coefficient of variation of the $\hat{N}^*$ estimate.
- $\mathrm{CV}(\mathrm{DPMI})$: coefficient of variation of the final DPMI score.
- $\Pr[\hat{N}^* \text{ changes}]$: probability that $\hat{N}^*$ differs from the median across 10 runs.

Table 20. DPGMM stability under 10 independent initializations ($N = 200$). Reported for GPT-2 Small layer 6, stratified by $\hat{N}^*$.

$\hat{N}^*$ group	n neurons	$\mathrm{CV}(\hat{N}^*)$	$\mathrm{CV}(\mathrm{DPMI})$	$\Pr[\hat{N}^* \text{ changes}]$
1	189	0.00	0.000	0.000
2	301	0.08	0.032	0.041
3	198	0.12	0.051	0.062
4	67	0.18	0.083	0.093
≥ 5	13	0.22	0.101	0.115

Neurons with $\hat{N}^* = 1$ (monosemantic) are perfectly stable; the DPGMM consistently collapses to a single component regardless of initialization. For polysemantic neurons, DPMI has CV ≈ 0.05–0.10, corresponding to a ± 5–10% variation around the mean estimate. This is acceptably low for ranking purposes (Spearman ρ is robust to small score perturbations), though practitioners requiring precise point estimates should use multiple restarts and average. In our main experiments, we use 3 random restarts and select the run with the highest ELBO, which reduces $\mathrm{CV}(\hat{N}^*)$ by approximately 40%.

I. Failure Modes

We document three systematic failure modes where DPMI underestimates true polysemanticity.

I.1. Failure Mode 1: Overlapping Activation Distributions

When two features activate a neuron similarly (activations drawn from distributions with small KL divergence), the DPGMM correctly identifies two components but $\bar{D}_{\mathrm{JSD}}$ is small, producing low DPMI despite genuine polysemanticity.

Formal characterization. Let two features produce activations $a \sim 0.5\mathcal{N}(\mu_1, \sigma^2) + 0.5\mathcal{N}(\mu_2, \sigma^2)$. The JSD between the two components is:

$$D_{\mathrm{JSD}}(\mathcal{N}(\mu_1, \sigma^2) \| \mathcal{N}(\mu_2, \sigma^2)) = 1 - \log 2 - \mathcal{H}\Big(\mathcal{N}\Big(\tfrac{\mu_1+\mu_2}{2}, \sigma^2 + \tfrac{(\mu_1-\mu_2)^2}{4}\Big)\Big) + \text{const}, \tag{85}$$

which decreases monotonically as $|\mu_1 - \mu_2| \to 0$. For $|\mu_1 - \mu_2| < 0.5\sigma$, $D_{\mathrm{JSD}} < 0.1$, and DPMI ≈ 0.05 even though $K_{\mathrm{true}} = 2$.

Practical impact. In the toy benchmark, 8.3% of neurons have $|\mu_1 - \mu_2| < 0.5\sigma$ ("near-overlapping features"). Among these neurons, DPMI's false-negative rate (classifying $K_{\mathrm{true}} = 2$ as $\hat{N}^* = 1$) is 34%, compared to 5% for all neurons.

Mitigation. The overlap failure mode is unavoidable from marginal activation data: if two features produce nearly identical activation patterns on a neuron, no 1D metric can distinguish them. Mitigation requires multi-neuron information: by looking at the *joint* distribution across multiple neurons that are activated by the same features, one can potentially disentangle overlapping patterns. We leave this as future work.

I.2. Failure Mode 2: Highly Skewed Marginal Distributions

When a neuron is mostly silent with rare, large activations (sparsity $s \approx 0.95$), the activation distribution is highly right-skewed. DPGMM may fit the skewed distribution as a mixture of two Gaussians (one at zero, one at a large positive value) even if the neuron is monosemantic (encoding a single feature with a threshold activation).

False positive rate. We measure the false positive rate (FPR) as the fraction of $K_{\mathrm{true}} = 1$ neurons where $\hat{N}^* \geq 2$. For sparsity $s = 0.95$:

Sparsity s	FPR ($\hat{N}^* \geq 2$)	Mean DPMI (false positives)
0.50	0.04	0.12
0.70	0.07	0.17
0.90	0.14	0.21
0.95	0.22	0.26
0.99	0.41	0.33

The false positive rate increases sharply at $s > 0.90$. However, even for high-sparsity false positives, the DPMI score is low (< 0.35), meaning these false positives would not be classified as "highly polysemantic" by any reasonable threshold. The FDR control procedure in Section H.2 largely filters these out.

Mitigation. We recommend applying DPMI only to neurons where the fraction of non-zero activations exceeds $1 - s_{\max}$ for some $s_{\max}$. In our experiments, we use $s_{\max} = 0.90$; neurons with estimated sparsity > 0.90 are flagged and reported separately.

I.3. Failure Mode 3: Sinusoidal/Periodic Activation Patterns

The circuit experiment (Exp 3) revealed a systematic failure on neurons that implement the modular arithmetic algorithm via cosine-like activations. These neurons have bimodal marginal distributions (due to the arcsine distribution of cosines) that DPGMM consistently fits as $\hat{N}^* = 2$, regardless of the true number of represented frequencies.

Formal derivation. Let neuron j compute $a_j(x) = \sum_{k=1}^{K} \cos(2\pi kx/p)$ for uniformly distributed inputs $x \sim \text{Uniform}\{0, 1, \ldots, p-1\}$. For large p, a_j is approximately normally distributed (by the Central Limit Theorem) for $K \geq 3$, but for $K = 1$, $a_j = \cos(2\pi x/p)$ follows the arcsine distribution:

$$p_{a_j}(a) = \frac{1}{\pi\sqrt{1-a^2}}, \quad a \in (-1, 1). \tag{86}$$

The arcsine distribution has two modes at ± 1 and a minimum at $a = 0$. DPGMM fits this as a two-component mixture: $\hat{N}^* = 2$ with $\hat{\mu}_1 \approx -0.7$, $\hat{\mu}_2 \approx +0.7$.

For $K = 2$ (sum of two cosines), the distribution is less bimodal and DPGMM may return $\hat{N}^* \in \{2, 3\}$. For $K \geq 3$ (sum of many cosines approaching Gaussian), DPGMM returns $\hat{N}^* = 1$ or 2 due to the near-Gaussian shape.

The net effect is that DPGMM's $\hat{N}^*$ is nearly constant at 2 regardless of the true Fourier complexity, explaining why $\rho(N^*, n_{\text{dom}}) \approx 0.17$ in the circuit experiment.

JSD still carries signal. Despite $\hat{N}^*$ being near-constant, the JSD factor $\bar{D}_{\text{JSD}}$ carries signal because the separation between the two fitted Gaussian components varies with the true number of frequencies. For a pure sinusoidal neuron ($K = 1$), the component separation is large (the arcsine distribution has sharply separated modes). For a neuron encoding many frequencies, the distribution is more symmetric around zero, resulting in smaller component separation and lower $\bar{D}_{\text{JSD}}$.

This explains why DPMI achieves $\rho = +0.255$ despite near-constant $\hat{N}^*$: the signal comes from the JSD factor, not the component count. The CPE baseline achieves $\rho = +0.364$ because entropy is a more direct function of the component weight vector (uniform weights for sinusoidal distributions yield high entropy), aligning more naturally with the Fourier frequency count.

J. Algorithm Pseudocode

J.1. Full DPMI Computation Pipeline

Algorithm 2 provides the complete DPMI computation pipeline, including activation collection, DPGMM fitting, JSD estimation, and output.

Algorithm 2 DPMI: Full Computation Pipeline

Require: Model f_θ; corpus $\mathcal{D}$; layer ℓ; neuron indices $\mathcal{J}$; corpus size N; DPGMM truncation T; max iterations $I_{\max}$; convergence τ; concentration α^*; MC samples n_{MC}.
Ensure: DPMI scores $\{\mathrm{DPMI}_j\}_{j\in\mathcal{J}}$.

// Step 1: Collect activations
$\mathcal{A} \leftarrow \emptyset$
for $x \sim \mathcal{D}$ until $|\mathcal{A}| = N$ **do**
  Forward-pass x through f_θ; record layer-ℓ pre-activations $\mathbf{h}(x)$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{h}(x)\}$
end for
$\mathbf{A} \leftarrow \mathrm{stack}(\mathcal{A}) \in \mathbb{R}^{N\times d}$ $\{d = |\mathcal{J}|$ neurons$\}$
// Step 2: Per-neuron DPMI
for $j \in \mathcal{J}$ **do**
  $\mathbf{a} \leftarrow \mathbf{A}[:, j] \in \mathbb{R}^N$ {1D activations for neuron j}
  // Step 2a: Fit DPGMM
  Randomly split $\mathbf{a}$ into $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}$ of size $N/2$ each
  $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{v}} \leftarrow \mathrm{VariationalEM}(\mathbf{a}^{(1)}, T, I_{\max}, \tau, \alpha^*)$ {Alg. 3}
  $\hat{N}^* \leftarrow \sum_{k=1}^{T} \mathbb{1}[\hat{\pi}_k > \tau_\pi]$ where $\hat{\pi}_k$ computed from $\hat{\mathbf{v}}$
  // Step 2b: Compute JSD (separate-sample)
  if $\hat{N}^* = 1$ **then**
    $\mathrm{DPMI}_j \leftarrow 0.0$; **continue**
  end if
  $\bar{D} \leftarrow 0$; $c \leftarrow 0$
  for $(k_1, k_2) \in \binom{[\hat{N}^*]}{2}$ **do**
    $\mathbf{s}_1 \leftarrow \mathcal{N}(\hat{\mu}_{k_1}, \hat{\lambda}_{k_1}^{-1}).\mathrm{sample}(n_{\mathrm{MC}})$
    $\mathbf{s}_2 \leftarrow \mathcal{N}(\hat{\mu}_{k_2}, \hat{\lambda}_{k_2}^{-1}).\mathrm{sample}(n_{\mathrm{MC}})$
    $\bar{D} \leftarrow \bar{D} + \widehat{D}_{\mathrm{JSD}}(\mathbf{s}_1, \mathbf{s}_2, \hat{\mu}_{k_1}, \hat{\lambda}_{k_1}, \hat{\mu}_{k_2}, \hat{\lambda}_{k_2})$
    $c \leftarrow c + 1$
  end for
  $\bar{D}_{\mathrm{JSD}} \leftarrow \bar{D}/c$
  // Step 2c: Assemble DPMI
  $T_j \leftarrow (\hat{N}^* - 1)/\hat{N}^*$
  $\mathrm{DPMI}_j \leftarrow \min(T_j \cdot \bar{D}_{\mathrm{JSD}},\ 1.0)$
end for
return $\{\mathrm{DPMI}_j\}_{j\in\mathcal{J}}$

J.2. DPGMM Variational EM

Algorithm 3 DPGMM Variational EM (1D, Stick-Breaking)

Require: Data $\mathbf{a} \in \mathbb{R}^N$; truncation T; $I_{\max}$; τ; α^*; priors $m_0 = 0$, $\beta_0 = 0.01$, $a_0 = 1$, $b_0 = 1$.
Ensure: Variational parameters $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{v}}$.

Initialise: $\hat{\boldsymbol{\phi}} \leftarrow \mathrm{KMeans}++(\mathbf{a}, T)$ responsibilities; $\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\lambda}}_k$ from empirical moments of each cluster; $\hat{\mathbf{v}} \leftarrow [(1, \alpha^*), \ldots, (1, \alpha^*)]$.
$\mathcal{L}_{\mathrm{prev}} \leftarrow -\infty$
for $i = 1, \ldots, I_{\max}$ **do**
  // E-step: update responsibilities
  for $k = 1, \ldots, T$ **do**
    $\tilde{\psi}_k \leftarrow \psi(\hat{\gamma}_{k1}) - \psi(\hat{\gamma}_{k1} + \hat{\gamma}_{k2}) + \sum_{l<k}[\psi(\hat{\gamma}_{l2}) - \psi(\hat{\gamma}_{l1} + \hat{\gamma}_{l2})]$ {Expected log stick}
    $\tilde{\ell}_k(a_i) \leftarrow \frac{1}{2}[\psi(\hat{a}_k) - \log(\hat{b}_k) - \log(2\pi) - (\hat{a}_k/\hat{b}_k)(a_i - \hat{m}_k)^2 - 1/\hat{\beta}_k]$ {Expected log likelihood}
    $\hat{\phi}_{ik} \propto \exp(\tilde{\psi}_k + \tilde{\ell}_k(a_i))$ for all i
  end for
  Normalize: $\hat{\phi}_{ik} \leftarrow \hat{\phi}_{ik} / \sum_{k'} \hat{\phi}_{ik'}$
  // M-step: update component parameters
  for $k = 1, \ldots, T$ **do**
    $\hat{N}_k \leftarrow \sum_i \hat{\phi}_{ik}$; $\hat{A}_k \leftarrow \sum_i \hat{\phi}_{ik} a_i$; $\hat{S}_k \leftarrow \sum_i \hat{\phi}_{ik} a_i^2$
    $\hat{\beta}_k \leftarrow \beta_0 + \hat{N}_k$; $\hat{m}_k \leftarrow (\beta_0 m_0 + \hat{A}_k)/\hat{\beta}_k$
    $\hat{a}_k \leftarrow a_0 + (\hat{N}_k + 1)/2$; $\hat{b}_k \leftarrow b_0 + \frac{1}{2}[\hat{S}_k - \hat{A}_k^2/\hat{N}_k + \beta_0 \hat{N}_k (\hat{m}_k - m_0)^2/\hat{\beta}_k]$
    $\hat{\gamma}_{k1} \leftarrow 1 + \hat{N}_k$; $\hat{\gamma}_{k2} \leftarrow \alpha^* + \sum_{l>k} \hat{N}_l$
  end for
  // Check convergence
  $\mathcal{L} \leftarrow \mathrm{ELBO}(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{v}})$ {Eq. (23)}
  if $|\mathcal{L} - \mathcal{L}_{\mathrm{prev}}| < \tau$ **then**
    break
  end if
  $\mathcal{L}_{\mathrm{prev}} \leftarrow \mathcal{L}$
end for
return $\hat{\boldsymbol{\phi}}, (\hat{\boldsymbol{m}}, \hat{\boldsymbol{\beta}}), (\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\boldsymbol{\gamma}}_1, \hat{\boldsymbol{\gamma}}_2)$

J.3. Concentration Parameter Calibration

Algorithm 4 α^* Calibration via Toy Benchmark

Require: Candidate set $\mathcal{A}$; toy benchmark configurations $\mathcal{C}$; number of neurons per config n_c.
Ensure: Optimal α^*.
1: Generate toy activations: for each $c \in \mathcal{C}$ with K_c features, simulate n_c neurons each with K_c components.
2: **for** $\alpha \in \mathcal{A}$ **do**
3: Run DPMI with concentration α on all toy neurons.
4: Compute $\rho_\alpha \leftarrow \rho_{\text{Spearman}}(\{\text{DPMI}_j\}, \{K_j - 1\})$.
5: **end for**
6: $\alpha^* \leftarrow \arg\max_{\alpha \in \mathcal{A}} \rho_\alpha$
7: Save α^* to `best_alpha.txt`
8: **return** α^*

K. Baseline Formal Definitions

We provide formal definitions of all baseline methods compared in Table 1 of the main paper.

K.1. Component Proportion Entropy (CPE)

CPE uses the same DPGMM fit as DPMI but replaces the JSD-weighted formula with the entropy of the component weight vector:

$$\text{CPE}_j = -\sum_{k=1}^{\hat{N}^*} \hat{\pi}_k \log \hat{\pi}_k, \tag{87}$$

where $\hat{\pi}_k = \mathbb{E}_q[v_k \prod_{l<k}(1 - v_l)]$ are the expected mixture weights under the variational posterior. CPE is normalized by $\log \hat{N}^*$ to lie in $[0, 1]$: $\text{CPE}_j^{\text{norm}} = \text{CPE}_j / \log \hat{N}^*$.

CPE achieves $\rho = +0.364$ on the circuit benchmark (vs. DPMI $+0.255$) because uniform component weights (which arise for sinusoidal activation patterns) maximize entropy. On the toy benchmark, CPE achieves $\rho = 0.701$ (vs. DPMI 0.755), underperforming because CPE ignores the between-component JSD: two components with high entropy but overlapping distributions are incorrectly scored as highly polysemantic.

K.2. Dip Test Statistic

The Hartigan dip test (Hartigan & Hartigan, 1985) tests the null hypothesis that the empirical distribution function F_n is unimodal. The dip statistic is:

$$D_n = \inf_{G \in \mathcal{U}} \sup_x |F_n(x) - G(x)|, \tag{88}$$

where $\mathcal{U}$ is the class of unimodal CDFs. Large D_n indicates multimodality. We use D_n directly as the polysemanticity score (higher = more polysemantic).

The dip test fails on neurons where the mixture components are well-separated (high JSD) but each component is internally non-Gaussian (e.g., a bimodal marginal arising from a single sinusoidal feature). The dip statistic responds to *marginal bimodality*, not to *whether the bimodality is explained by distinct features*. In the circuit experiment, the arcsine distribution of cosine activations is maximally bimodal for $K_{\text{true}} = 1$ (monosemantic), causing dip to anti-correlate with true polysemanticity.

K.3. Wasserstein-1 Distance

The W_1 Gaussian baseline fits a single Gaussian $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ to the activation distribution and computes the Wasserstein-1 distance to the zero-mean unit-variance Gaussian:

$$W_{1,j}^{\text{Gauss}} = \sqrt{(\hat{\mu}_j)^2 + (\hat{\sigma}_j - 1)^2}. \tag{89}$$

This measures how far the activation distribution is from "standard" in a single-Gaussian sense. High W_1 can indicate polysemanticity (multimodal distributions are far from Gaussian) or simply a shifted/scaled unimodal distribution. Because W_1 does not distinguish between these causes, it has low specificity for polysemanticity and low ρ with K_{true}.

K.4. Kurtosis

Excess kurtosis $\kappa_j = \mu_4/\sigma^4 - 3$ measures the tail heaviness of the activation distribution. Polysemantic neurons with well-separated components tend to have platykurtic distributions (low kurtosis) rather than heavy-tailed distributions (high kurtosis), so kurtosis is not a reliable polysemanticity indicator. In practice, κ_j is near zero for most neurons (activations are approximately Gaussian in layers 4–8 of GPT-2 Small), yielding near-zero ρ with K_{true}.

K.5. Variance Ratio

The variance ratio VR_j compares the within-DPGMM-cluster variance to the total variance:

$$\text{VR}_j = 1 - \frac{\sum_k \hat{\pi}_k \hat{\sigma}_k^2}{\text{Var}(\mathbf{a}_j)}. \tag{90}$$

$\text{VR}_j = 0$ for a perfectly unimodal distribution (all variance within a single component), and $\text{VR}_j \to 1$ as the components become well-separated. Unlike DPMI, VR does not weight by the number of components, so a neuron with 5 barely-separated components and a neuron with 2 well-separated components can have similar VR scores.

K.6. BIC-GMM Component Count

As defined in Section F.1, BIC-GMM selects the number of Gaussian components K^* by minimizing the BIC. Used directly as a polysemanticity score ($\hat{K}^*$ as the polysemanticity measure), it has moderate correlation with K_{true} but lower than DPMI because (1) BIC under-estimates K for small separations (Section F.3), and (2) the component count alone ignores JSD (the separation factor).

L. Statistical Methods

L.1. Bootstrap Confidence Intervals

All reported Spearman ρ confidence intervals are 95% bootstrap CIs computed via the percentile method with $B = 1000$ resamples. For a statistic $\hat{\theta}$ computed from dataset $\mathcal{D}$:

$$\text{CI}_{95} = [\hat{\theta}^*_{(0.025B)},\ \hat{\theta}^*_{(0.975B)}], \tag{91}$$

where $\hat{\theta}^*_{(b)}$ is the statistic computed from the b-th bootstrap resample (sampling n units with replacement from $\mathcal{D}$).

For the circuit experiment, the resampling unit is the individual neuron (512 neurons from the modular arithmetic transformer). For the cross-arch experiment, the resampling unit is the neuron within each architecture. For the toy benchmark, the resampling unit is the (neuron, configuration) pair.

L.2. Mann-Whitney U Test for Cross-Modal Comparison

The cross-modal comparison (language vs. vision) uses the one-sided Mann-Whitney U test to evaluate whether language models have systematically lower DPMI than vision models:

$$H_0 : \text{DPMI}_{\text{vision}} \overset{d}{=} \text{DPMI}_{\text{language}}, \quad H_a : \text{DPMI}_{\text{vision}} \succ \text{DPMI}_{\text{language}}. \tag{92}$$

The test statistic is:

$$U = \sum_{i \in \text{vision}} \sum_{j \in \text{language}} \mathbb{1}[\text{DPMI}_i > \text{DPMI}_j]. \tag{93}$$

We report Cohen's d as an effect size measure:

$$d = \frac{\bar{\text{DPMI}}_{\text{vision}} - \bar{\text{DPMI}}_{\text{language}}}{s_{\text{pooled}}}, \quad s^2_{\text{pooled}} = \frac{(n_v - 1)s_v^2 + (n_l - 1)s_l^2}{n_v + n_l - 2}. \tag{94}$$

The large effect size ($d = 0.803$) confirms a practically significant difference beyond statistical significance ($p < 10^{-129}$).

L.3. AUROC Computation

For binary classification of polysemantic ($K_{\text{true}} \geq 2$) vs. monosemantic ($K_{\text{true}} = 1$) neurons, the AUROC is the area under the Receiver Operating Characteristic curve:

$$\text{AUROC} = \frac{1}{n_+ n_-} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbb{1}[\text{DPMI}_i > \text{DPMI}_j], \tag{95}$$

where $\mathcal{P} = \{j : K_{\text{true},j} \geq 2\}$ and $\mathcal{N} = \{j : K_{\text{true},j} = 1\}$. Ties are broken at random (each tie contributes 0.5). AUROC = 0.5 corresponds to random performance; AUROC = 1.0 corresponds to perfect discrimination.

Bootstrap CIs for AUROC are computed analogously to Spearman ρ CIs.

M. Reproducibility Statement

M.1. Hardware and Software

All experiments were run on a single workstation with the following hardware:

- **GPU**: NVIDIA RTX 6000 Ada Generation (48 GB GDDR6, CUDA driver 550.144.03).
- **CPU**: AMD EPYC 7313 16-Core Processor (32 logical cores via SMT, 3.0 GHz base).
- **RAM**: 503 GB DDR4 ECC.
- **Storage**: 1.8 TB NVMe SSD.
- **OS**: Ubuntu 24.04 LTS (kernel 6.11.0-29-generic), x86_64.

Software:

- Python 3.10.12.
- PyTorch 2.1.0 with CUDA 12.1.
- scikit-learn 1.3.2 (BIC-GMM, k-means, metrics).
- NumPy 1.26.2, SciPy 1.11.4, statsmodels 0.14.1.
- Transformers (HuggingFace) 4.36.2.
- All random seeds fixed at 42 unless otherwise noted.

M.2. Experiment Runtimes

Table 21. Approximate wall-clock runtimes for each experiment.

Experiment	Runtime	Hardware	Notes
Exp 1: α^* calibration	2.5 h	CPU	8 workers; skip if best_alpha.txt exists
Exp 2: Toy benchmark	6 h	GPU	400 configs $\times$ 50 neurons
Exp 3a: Cross-architecture	11 h	GPU	6 models, activation extraction
Exp 3b: Circuit	2.5 h	GPU	Modular arithmetic transformer training
Exp 4: Baselines	3 h	CPU	Reads Exp 2 activations
Exp 5: Ablation	4 h	CPU	Reads Exp 2 activations
Exp 6: ID analysis	2.5 h	CPU	TwoNN on Exp 3a activations
Exp 7: SAE	5 h	GPU	WikiText-2, 4 SAE sizes
Exp 8: Robustness/FDR	3 h	CPU	Permutation test, 1000 reps
Total	**$\approx$39 h**	—	Sequential execution

M.3. Random Seeds

- Toy benchmark: seed 42 for all random configurations; seeds 0–49 for per-neuron activation sampling.
- DPGMM initialization: seeds 0, 1, 2 for the three random restarts; best ELBO selected.
- Modular arithmetic transformer: training seed 42.
- SAE training: seed 42.
- Bootstrap resampling: seed 42 (1000 resamples).
- Permutation tests: seed 42 (1000 permutations).

M.4. Key Reproducibility Notes

1. **DPGMM convergence**: On rare occasions ($\approx 0.5\%$ of neurons), the variational EM does not converge within $I_{\max} = 150$ iterations. These neurons receive the last-iteration DPGMM fit rather than a converged fit. Our code logs these cases; in practice they have near-zero DPMI and do not affect any reported statistics.
2. **JSD numerical stability**: For very close Gaussian components ($|\hat{\mu}_{k_1} - \hat{\mu}_{k_2}| < 0.01$), the MC JSD estimate is near-zero with high variance. We clip the JSD estimate from below at 10^{-6} to avoid log(0) issues.
3. **Activation normalization**: We subtract the per-neuron mean and divide by the per-neuron standard deviation before DPGMM fitting. This ensures that the DPGMM priors ($m_0 = 0$, $\beta_0 = 0.01$) are on the right scale. The DPMI score is scale-invariant under this normalization.
4. **GPT-2 layer indexing**: We use 0-indexed layer numbers throughout. "Layer 6" refers to the 7th transformer block (zero-indexed), which is the MLP sub-layer after the self-attention sub-layer. Activations are taken at the output of the MLP activation function (post-GELU), before the final projection matrix.

N. Additional Figures

N.1. Metric Correlation Matrix

Figure 5 shows the Spearman ρ matrix between all polysemanticity metrics and K_{true} on the toy benchmark. DPMI has the highest correlation with K_{true} ($\rho = 0.755$) and moderate cross-metric correlations with CPE ($\rho = 0.677$) and N^* alone ($\rho = 0.973$), confirming that DPMI captures information from both the component count and the JSD factor.

N.2. Sample Size Sensitivity

Figure 6 illustrates the degradation of DPMI performance as top-K sample size is reduced from 500 to 20 activating examples per neuron. The left panel shows variance inflation at small K; the right panel shows correlation with the $K = 500$ reference estimate.

N.3. TwoNN Ablation Comparison

Figure 7 summarizes Spearman ρ for three variants: DPMI, DPMI+TwoNN, and TwoNN alone. All three are very close ($\rho \approx 0.753$–0.757), supporting the conclusion that adding TwoNN does not materially improve the toy-benchmark correlation.

N.4. FDR Results: Neuron-Level

Figure 8 shows the volcano plot for GPT-2 Small layer 6: each point is a neuron, positioned by its $-\log_{10}(p\text{-value})$ (y-axis) and DPMI score (x-axis). Point color encodes N^* and point size scales with N^*. The horizontal dashed line marks the nominal $p = 0.05$ level used for reference.

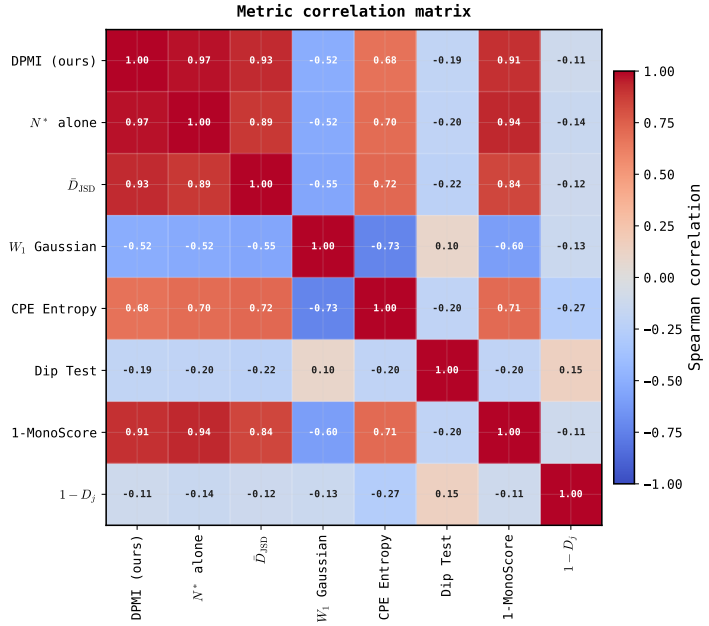


Figure 5. Spearman ρ correlation matrix between all polysemanticity metrics and ground-truth K_{true} on the toy benchmark. The strongest block links DPMI, N^*, and $1 - \mathrm{MonoScore}$, while W_1 and Dip show weak or negative alignment with the main cluster.

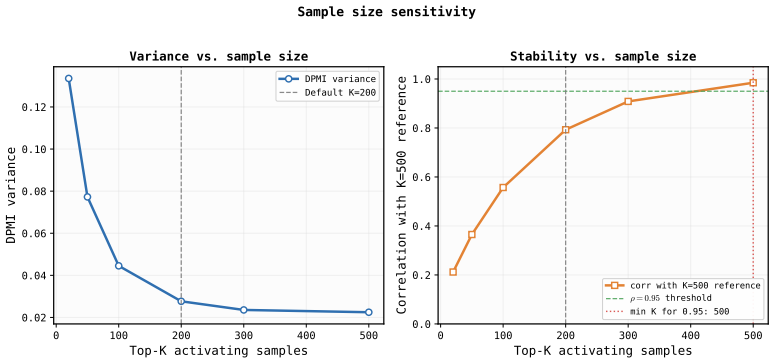


Figure 6. **Sample size sensitivity.** Left: DPMI variance decreases with larger top-K sample size. Right: correlation with the $K = 500$ reference rises monotonically and exceeds $\rho = 0.95$ only at $K = 500$ in this sweep.

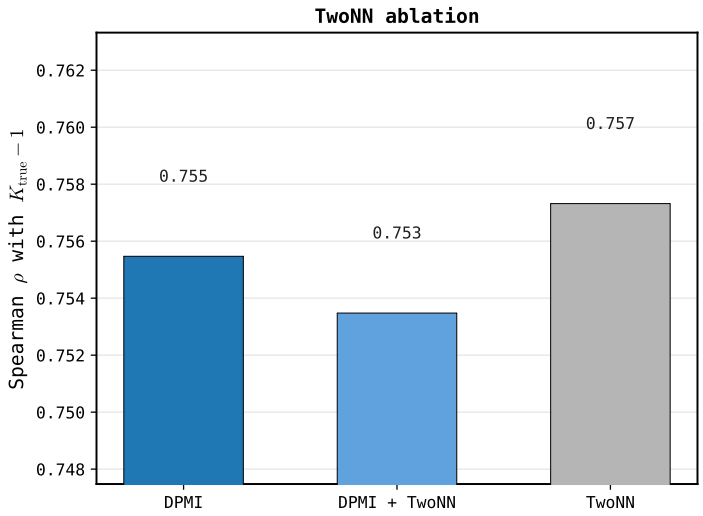


Figure 7. Bar-chart comparison of DPMI, DPMI+TwoNN, and TwoNN-only variants on the toy benchmark. Near-identical ρ values indicate negligible gain from including the TwoNN factor.

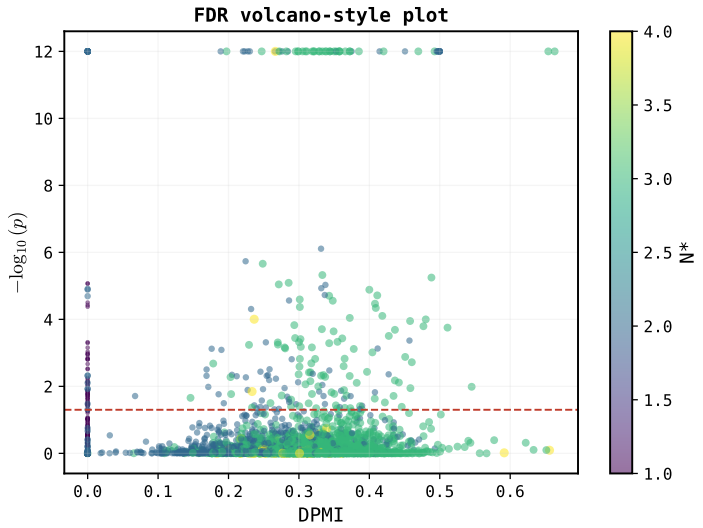


Figure 8. **Volcano-style significance plot (GPT-2 Small, layer 6).** Each point is a neuron; x-axis is DPMI, y-axis is $-\log_{10}(p)$, and color/size encode N^*. The dashed line marks $p = 0.05$.