SPARSE CROSS-LAYER MODULES: A MESOSCOPIC VIEW OF LANGUAGE MODEL ORGANIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpreting a language model requires connecting individual response patterns to the organization of representations across the network. We study sparse cross-layer modules as a mesoscopic unit for this purpose. Joint sparse coding associates each within-document token response with a decoder direction across the layer stack. In Pythia-410M, the resulting population exhibits depth-dependent differentiation: salient decoder coordinates concentrate in adjacent-layer bands, and response breadth, lexical coherence, and description scores vary across the resulting depth groups. Responses remain heterogeneous within groups, while response similarities extend across modules with little salient-coordinate overlap. Across text domains, the same dictionary exhibits shared usage with distinct preferences, and independently fitted dictionaries retain partial response correspondence. Dictionaries fitted at successive training checkpoints show an early increase in best-match response similarity and a more gradual increase in salient-coordinate overlap. Across three Pythia sizes, matched response subsets retain substantial similarity at comparable standardized-space reconstruction quality. Automated descriptions, activating examples, and selected-module interventions connect these population measurements to inspectable patterns and local output effects. Together, the analyses characterize language-model organization through the distribution of individual responses, their relationships within a population, and their recurrence across inputs and model conditions.

1 INTRODUCTION

An explanation of a language model should connect the meanings of its internal responses to the way those responses are organized across the network. A feature that responds to a word pattern, a type of entity, or a recurring context is informative at the level of individual examples. Understanding its place in the model requires further questions: how is its representation distributed across depth, which other responses have similar distributions, and how does its use change across inputs? At the population level, these questions concern how the model distributes, differentiates, and reuses its representations.

Interpretability methods offer several ways to examine this organization. Neuron analyses and sparse autoencoders (SAEs) recover response patterns that can be inspected through activating examples and natural-language descriptions [1–3]. Representation-similarity methods reveal continuity and block structure across layers and models [4–6]. Cross-layer methods connect these resolutions: Sparse Crosscoders learn features spanning layers and models [7], RouteSAE shares sparse learning across layers [8], and Feature Flow traces relations among layer-local sparse features [9]. Causal feature circuits and attribution graphs examine how features interact to affect outputs [10, 11]. These advances make it possible to ask how an entire population of interpretable responses is organized through a language model.

We study this question using *sparse cross-layer modules*. A module pairs one sparse response with a decoder direction across the layer stack. A compact joint dictionary provides a mesoscopic resolution: each response can be inspected in text, located through its decoder coordinates, and compared with other responses. We use this common unit to characterize depth differentiation and recurrence across domains, training stages, and model sizes (Figure 1).

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

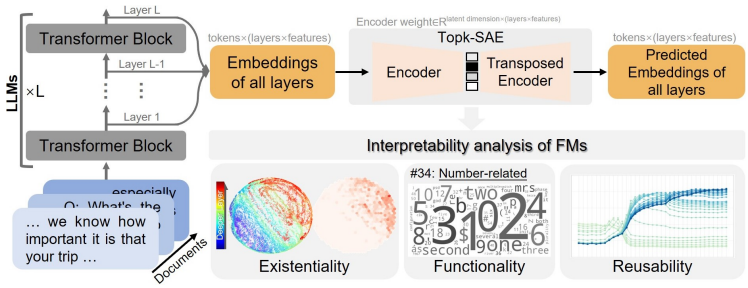

Figure 1: A mesoscopic view of language-model organization. Joint sparse coding connects response patterns to decoder directions across layers. The schematic's three themes concern localization, interpretation, and recurrence across conditions. AN denotes a residual-stream layer–coordinate pair in the original panels; its embedding visualizes coordinate-response geometry.

Our investigation follows three questions. First, what can individual modules explain? Activating contexts and automated descriptions reveal recurring response patterns, while selected interventions measure local output effects. Second, how are these responses organized? We relate decoder depth profiles to response breadth and coherence, and compare response correlations with salient-coordinate overlap. Third, which relationships recur across conditions? Domain usage, independently fitted dictionaries, training checkpoints, and capacity-adjusted scale comparisons examine the same response–direction unit at different resolutions.

The resulting picture is one of differentiation within reuse. Depth orders the allocation of decoder coordinates and accompanies changes in response breadth, yet similar depth profiles can carry distinct responses. Across domains, a shared inventory is recruited with different emphases; across fits and checkpoints, related responses recur with partly changing directions. Signed-weight summaries, layer energy, repeated fits, and alternative decompositions trace these relationships through changes in how modules are measured. Each activating example gains a place in a larger organization: where its response concentrates, which other responses it resembles, and when those relationships recur.

2 Sparse Cross-Layer Modules

Models and data. Pythia provides a common model family for studying training and scale [12]. Our main setting is Pythia-410M, with $L = 24$ layers and hidden dimension $d = 1024$. We fit dictionaries on the first 8,000 Pile-10K documents (approximately 5.5M tokens) and evaluate on the remaining 2,000 (approximately 1.5M tokens) [13]. The study spans ten Pythia sizes from 14M to 12B and 35 independently fitted 410M checkpoints from step 0 to 143,000. Auxiliary model and domain settings are described in Appendix K and Appendix U.

Joint sparse representation. For each token t, we concatenate residual-stream hidden states across layers and standardize every coordinate over token positions within its document:

$$\mathbf{x}(t) = \mathcal{N}_{\text{doc}}\Big([\mathbf{h}^{(1)}(t); \ldots; \mathbf{h}^{(L)}(t)]\Big) \in \mathbb{R}^{Ld}. \tag{1}$$

The main setting contains 24,576 residual-stream layer–coordinate pairs (ANs in the original figures). A dictionary $D \in \mathbb{R}^{M \times Ld}$, with rows normalized to unit length, is shared by the encoder and decoder:

$$\mathbf{c}(t) = \text{TopK}_K(\text{ReLU}(D_{\text{norm}}\mathbf{x}(t))), \qquad \hat{\mathbf{x}}(t) = D_{\text{norm}}^{\top}\mathbf{c}(t). \tag{2}$$

We minimize token-averaged squared reconstruction error, using $M = 1024$ and $K = 32$ by default [3, 7]. At most K positive codes are retained per token. The $M = 1024$ directions summarize coarse modes in the $Ld = 24576$-dimensional stack. Each coefficient weights one full-stack direction, whose signed entries describe its reconstruction contribution. Means and scales use the entire document, so responses describe retrospective, within-document variation. This preprocessing defines the response variable used in the interpretation and usage analyses. Training details appear in Appendix A.

Module responses and decoder profiles. Module m consists of response $c_m(\cdot)$ and decoder direction $\mathbf{w}_m = [\mathbf{w}_m^{(1)}; \ldots; \mathbf{w}_m^{(L)}]$. Its activation frequency is $\mathrm{AF}_m = N^{-1} \sum_t \mathbf{1}[c_m(t) > 0]$; an active module has evaluation-set $\mathrm{AF}_m > 0$. To summarize location, we z-normalize weights within each module and select the upper tail $S_m = \{j : z_{m,j} > 1.65\}$. The depth profile $\mathbf{v}_m$ counts these salient coordinates in each layer; the full direction retains both signs. Correlation-based average-linkage clustering yields dominant depth templates (Appendix E). Profiles and overlaps refer to this selected set in the model's residual coordinate basis.

Correspondence between modules. Response correspondence uses Pearson correlation on common token positions. Population, checkpoint, and scale panels use dense projection scores $s_m(t) = \mathbf{w}_m^\top \mathbf{x}(t)$; activation frequency and marked examples use gated codes $c_m(t)$. The paired-fit comparisons evaluate both. Coordinate overlap is $|S_m \cap S_n|/|S_m \cup S_n|$. For reference A, query B, and similarity matrix S,

$$\mathrm{Align}(A \leftarrow B) = \frac{1}{|B|} \sum_{j \in B} \max_{i \in A} S_{ij}. \tag{3}$$

This summary averages each query's best reference match, allowing shared references. Response and overlap select matches separately; within-set comparisons exclude self-matches. Domain refits use reciprocal neighbors. Capacity comparisons and the repeated-fit and checkpoint controls also use one-to-one assignments, with their inventories specified alongside the results.

Interpreting a module. DeepSeek-V3 [14] generates descriptions and participates in their automated scoring [15]. On separate evaluation examples, Detection measures prediction of active versus inactive windows; Fuzzing measures classification of correctly marked trigger spans (Appendix O). These scores assess descriptions within the automated procedure. Input-embedding similarity measures lexical coherence (Appendix P).

3 Interpretable Responses and Cross-Layer Organization

3.1 Locating individual responses across depth

Joint sparse coding associates an individual response with a direction across the layer stack. Its upper-tail decoder coordinates concentrate in bands of adjacent layers, as illustrated by two modules in Figure 2c,d. The fitting objective contains no layer-locality penalty. The depth profile quantifies this concentration within the learned full-stack direction.

The coordinate-response geometry provides a broader view of this distribution. A CEBRA embedding [16] of the 24,576 residual coordinates, using their responses to 32 text stimuli, displays a depth gradient (Figure 2a,b). Highlighting a module's salient coordinates in that space connects the response to a region of the stack; its layer-count profile gives the direct quantitative description. The same module thus has inspectable token responses and a depth profile of its upper-tail decoder coordinates.

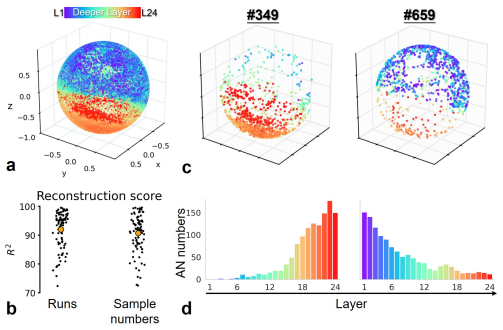

Figure 2: Module localization across depth. (a) Coordinate-response embedding. (b) Embedding consistency under stimulus resampling and count changes; linear-regression R^2 compares AN embeddings. (c) Two modules' salient coordinates. (d) Their depth profiles. See Appendix AG and Appendix AI.

On the original 50,000-token panel, adjacent-layer CKA averages 0.897, and raw-coordinate matching predominantly follows the same residual coordinate. In the 100,000-token fitting panel, adjacent layer-local SAE codes also correspond (test $r = .405$); chaining matches from layer 1 to 24 gives endpoint $r = .0085$ (Appendix N). A joint module associates one response with a full-stack direction, allowing its distribution over depth to be inspected directly.

3.2 DEPTH ORGANIZATION ACCOMPANIES RESPONSE DIFFERENTIATION

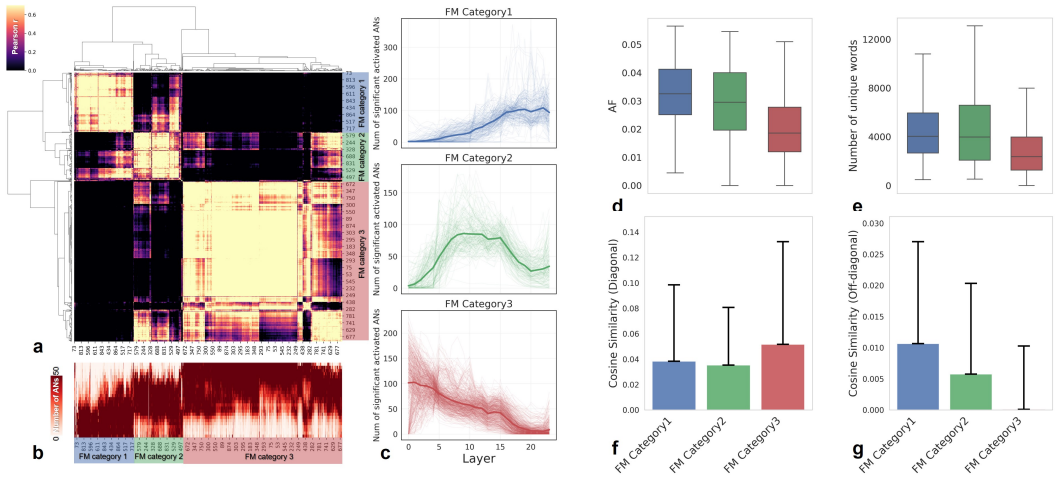

Figure 3: Depth organization and response differentiation. (a) Profile correlations and clustering. (b) Salient-coordinate counts by layer. (c) Deep, middle, and shallow profiles (thin: individual modules; thick: group means). (d,e) Activation frequency and activated-token-type coverage. (f,g) Within-module and within-group between-module input-embedding similarity. Category 1/2/3 denotes deep/middle/shallow; the color–FM lookup is in Appendix F.

The population's default depth-profile clustering is summarized by shallow, middle, and deep templates (Figure 3a–c). These groups locate concentrations of upper-tail decoder coordinates. Their response properties show how the resulting geometry relates to the input patterns that the modules track.

Deep modules activate more frequently and cover more token types, while shallow modules respond to narrower sets (Figure 3d,e). Shallow modules also have greater within-module input-embedding similarity and lower similarity between modules in the same group (panels f,g; Appendix P). They thus distinguish relatively narrow lexical neighborhoods. The deeper group combines broader coverage with less tightly localized lexical responses. Depth gives a graded organization of response properties, with substantial variation among the modules in each region.

The coarse geometry recurs across decomposition choices. A 144-setting sweep varies width, sparsity, saliency threshold, and cluster count. Among the 36 three-cluster settings, 35 have matched-template similarity at least 0.9 and 34 preserve ordered peaks (Appendix H). Alternative groupings by decoder directions or response correlations recover similar average profiles despite different module assignments (Appendix G). The recurring objects are the broad depth templates; individual responses give each template its internal variation.

PCA [17] also recovers ordered depth structure, indicating that part of the geometry is present in activation covariance. Its profile silhouette is 0.6473, compared with 0.7182 for the SAE and approximately 0.024–0.027 for random directions (Appendix J). The sparse representation associates this geometry with token-selective responses. Global-coordinate and per-token/per-layer standardizations retain ordered templates, as do four auxiliary families: BLOOM-560M [18], H2O-Danube3-500M [19], SmolLM2-360M [20], and OPT-350M [21] (Appendix B; Appendix K). These controls establish the recurrence of coarse depth order across the tested fits.

The same depth order appears when the decoder profile measures weight mass. Holding the original 925 displayed modules and their group labels fixed, shallow/middle/deep layer-energy centroids are 9.45/12.97/16.51, compared with 8.13/12.97/17.65 for positive-tail counts. Negative-tail and norm-based profiles also retain this order (Appendix I). Energy places more weight on the broad portions of a direction, bringing the shallow and deep centroids closer together. Thus the depth templates organize responses at a coarse scale, while lexical coherence and the activating contexts distinguish what individual modules track.

3.3 Response neighborhoods and salient-coordinate overlap

A module enters two neighborhoods. Its responses place it among other input patterns, and its decoder profile locates it across depth. In Figure 4, salient-coordinate overlap is strongest within the same or adjacent depth groups, while response correlations extend across groups. Similar responses can occupy different salient-coordinate sets; a shared depth region can contain distinct responses.

This response–direction pairing also recurs across fits. On the 100,000-token fitting panel, validation-selected matches between three tied seeds yield held-out projection-score correlations of .639–.640 and layer-energy cosines of .974–.976 (Appendix M). Untied TopK and ReLU crosscoder fits give projection-score correlations of .592 and .497 with the tied fit, and energy cosines of .975 and .956 (Appendix N). These comparisons use the same inputs with the respective architecture and activity budgets. The depth allocation recurs closely, while the corresponding responses retain variation across decompositions.

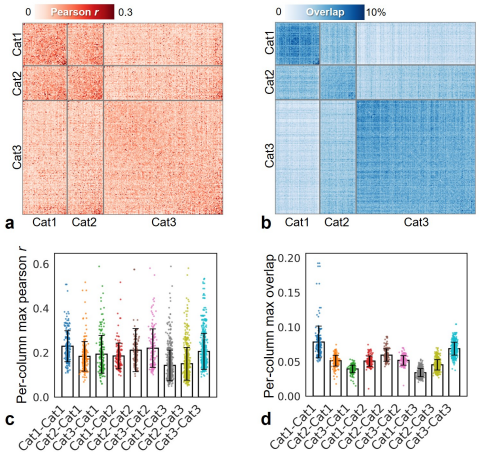

Figure 4: Response relations and coordinate overlap. (a) Token-response Pearson correlations. (b) Salient-coordinate Jaccard overlap. (c,d) Best-match summaries within and across depth groups. The two measurements describe complementary relations among the same modules.

3.4 Response descriptions and output effects

The activating contexts make these population differences concrete (Figure 6). Module descriptions cover surface markers, domain terms, entity fragments, and broader context-associated cues. The accompanying marked examples allow the description to be read against the module's actual responses, while held-out description scoring measures its predictive consistency.

Across 921 scored modules with their recorded depth labels (Appendix F), Detection and Fuzzing correlate at $r = 0.5603$ (Figure 5). Detection medians are 0.600/0.625/0.585 for deep/middle/shallow modules; Fuzzing medians are 0.575/0.620/0.625. The distributions overlap substantially. The deep group's lower Fuzzing scores accompany broader token coverage and lower lexical coherence: descriptions of these responses are less accurate at classifying marked spans in this procedure. The scores therefore characterize variation in description predictability across the fitted population. Marked examples show which aspects of a response a short description captures.

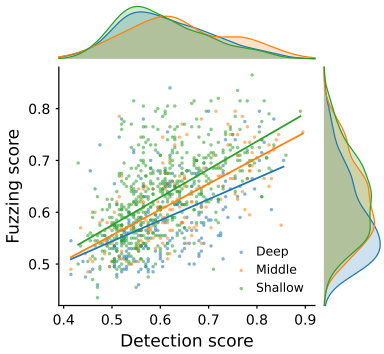

Figure 5: Interpretation scores. Detection versus Fuzzing across 921 modules, with descriptive group fits and marginal densities.

The Fuzzing median differences are 0.045 for middle versus deep and 0.050 for shallow versus deep. Adjusting for log activation frequency leaves mean deep-minus-middle and deep-minus-shallow contrasts of $-.0309$ and $-.0299$ (Appendix R). The association between depth group and description predictability therefore remains after this adjustment within the fitted population. Full score distributions and tests appear in Appendix Q; topic-word coverage appears in Appendix S.

The examples in Figure 6 show variation within each depth region. Deep module 349 marks function words and punctuation, while module 648 marks delimiters and operators in technical contexts. Middle modules 15 and 854 emphasize subword components and structured alphanumeric strings. Shallow modules 659 and 893 illustrate pronoun-related responses and document references. Modules sharing a depth template can thus differ in the patterns that activate them.



Figure 6: Inspectable responses within the module population. Examples show module IDs, Detection/Fuzzing scores, word clouds, generated descriptions, and five marked contexts; darker red indicates stronger activation. Rows correspond to deep, middle, and shallow modules. The supplementary atlas contains the full interpretation collection.

Module-linked output effects. For six selected modules, suppressing salient coordinates produces larger mean target-response reductions than matched random coordinates (9.603 versus 2.655) and larger local distribution changes (KL 0.1327 versus 0.0984). Next-token logit and log-probability changes do not reach statistical significance in this analysis (Appendix AE). A separate study tests 16 fixed targets with document-disjoint discovery and evaluation contexts and residual-norm-matched perturbations. Discovery-defined logit-change signatures are more reproducible than same-depth learned controls for all 16 targets (Holm-adjusted $p < .001$); effects are larger in 14/16 comparisons with five random directions (Appendix AF). These selected targets link measured module responses to local output effects, including signatures that recur across documents.

4 Reuse and Evolution of Module Organization

4.1 Domain Preferences within an Overlapping Population

How does the use of these responses vary across text domains? For each of 19 Pile source categories, we measure module frequency after document-wise normalization, $a_{d,m} = N_d^{-1} \sum_{t \in d} \mathbf{1}[c_m(t) > 0]$. Correlations between these domain vectors describe shared usage and relative preferences within the same fitted dictionary (Figure 7).

Mean off-diagonal invocation correlation is 0.74 for deep modules and 0.48/0.51 for middle/shallow modules. The broader responses of the deep group recur more consistently across domains, while the middle and shallow groups vary more in their preferences. Depth-dependent differentiation and shared use are therefore visible in the same module population.

Independent retraining. Separate dictionaries on Pile-10K, CodeSearchNet [22], GSM8K [23], and Chinese Wikipedia use $M = 1024$, $K = 32$. Within each evaluation domain, the own-domain fit has the highest EV (Table 1). Cross-domain reciprocal-match proportions are 21–38%. Dividing by the corresponding same-distribution refit reference gives relative proportions of 38–79% (Appendix U). Own-domain reconstruction and partial response matching capture complementary aspects of domain adaptation.

Table 1: Independent-domain reconstruction. Raw-space explained variance. Rows identify the fitting data; columns are 30,000-token evaluation panels.

Fit	Pile	Code	Math	Chinese
Pile	**.8340**	.9102	.9384	.7836
Code	.7946	**.9585**	.9425	.7638
Math	.7828	.9067	**.9774**	.7286
Chinese	.7673	.9027	.9386	**.9090**

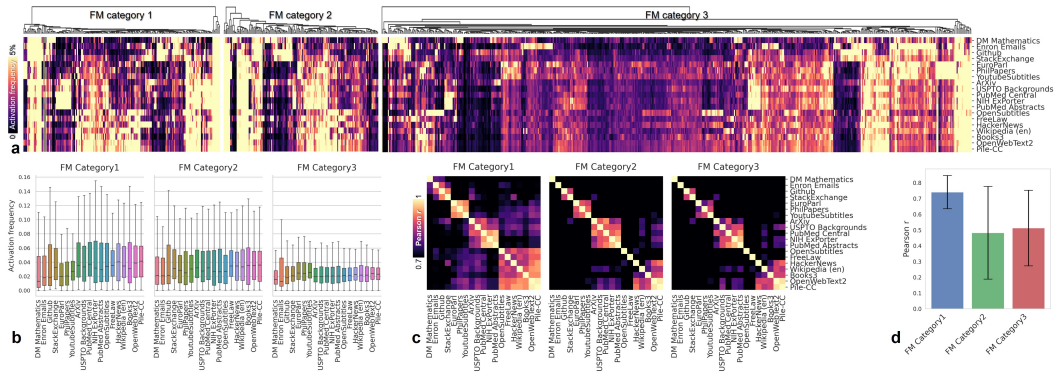

Figure 7: Module reuse and preference across 19 Pile domains. (a) Activation-frequency heatmaps by depth group. (b) Per-domain usage distributions. (c) Correlations between domain invocation vectors. (d) Off-diagonal means and standard deviations. All panels evaluate the same fitted dictionary.

The invocation correlations summarize the relative usage of the same modules across domains. A high correlation means that modules frequently activated in one domain also tend to be frequently activated in another. The deeper group's higher correlation accompanies its broader token coverage; the shallower groups retain stronger differences in which modules are preferred. Domain reuse can therefore be read at two resolutions: the recurring population and the relative emphasis assigned to its members.

4.2 Module populations across training checkpoints

To study organization through training [24], we fit separate dictionaries at 35 Pythia-410M checkpoints. The active-module count falls from roughly 1,000 to 640 over steps 0–256 and returns to approximately 900–1,000 after step 2,000 (Figure 8).
The count measures how much of each fitted inventory is used by evaluation tokens. Its early contraction and later recovery describe changing occupancy under the same sparse budget. The usage panel measures the frequency of positive responses. Together, these quantities show how each independently learned dictionary distributes activity over its available coordinates.

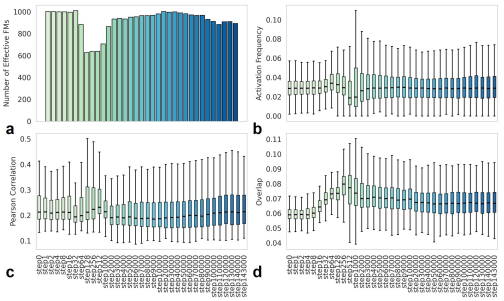

Figure 8: Within-checkpoint organization. (a,b) Active counts and usage. (c,d) Best-match response correlations and salient-coordinate overlaps, excluding self-matches.

For a fixed sparse budget, active count and mean usage satisfy

$$\overline{\mathrm{AF}}_{\mathrm{active}} = \frac{\mathbb{E}_t[\|\mathbf{c}(t)\|_0]}{M_{\mathrm{active}}} \leq \frac{K}{M_{\mathrm{active}}}.$$

Thus a smaller active population can have higher mean usage even with an unchanged per-token budget. The two within-checkpoint similarity panels describe relationships among the active members of that population, excluding each module's self-match.

Two views of cross-checkpoint correspondence. Best-match response similarity rises rapidly early in training and then changes more gradually (Figure 9). Best-match salient-coordinate overlap increases across a longer portion of training. These curves compare independently fitted populations using two separate matching criteria. They describe when similar token-response patterns and similar coordinate sets can be recovered across checkpoints, extending the occupancy analysis to relations among modules.

Each cross-checkpoint curve fixes a reference inventory and varies the query checkpoint. Response and salient-coordinate matches are selected separately, and several queries can choose one reference. The lower panels summarize variation across query checkpoints for each reference. One-to-one assignments recover the increase with a fixed candidate count. Using final step 143,000 as reference, 100 draws of 627 candidates per side give mean response correlations of .093 at step 0 and .633 at step 140,000; matched Jaccard rises from .051 to .335 (Appendix X). With the full inventories, reciprocal response matches cover .167 and .866 of queries at these endpoints.
The fitted populations become more closely connected as training proceeds: response patterns find closer counterparts, and their salient-coordinate sets overlap more. The original curves resolve the two time courses; the assignment controls show their growth under unique matches and equal candidate counts.

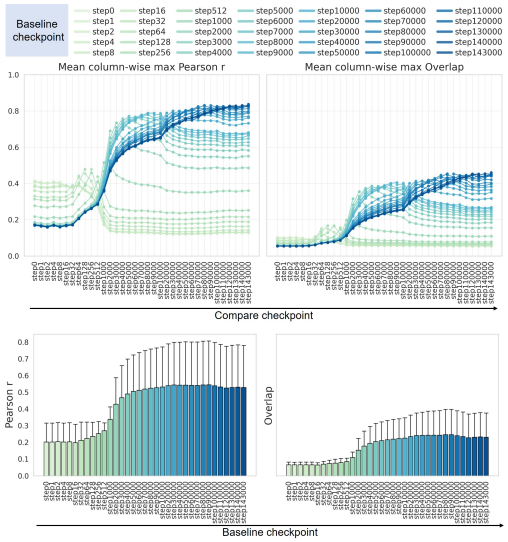

Figure 9: Cross-checkpoint correspondence. Top: separately maximized response similarity and salient-coordinate overlap at a fixed reference checkpoint. Bottom: means and standard deviations across query checkpoints, grouped by reference.

4.3 Organization across model scales and dictionary capacities

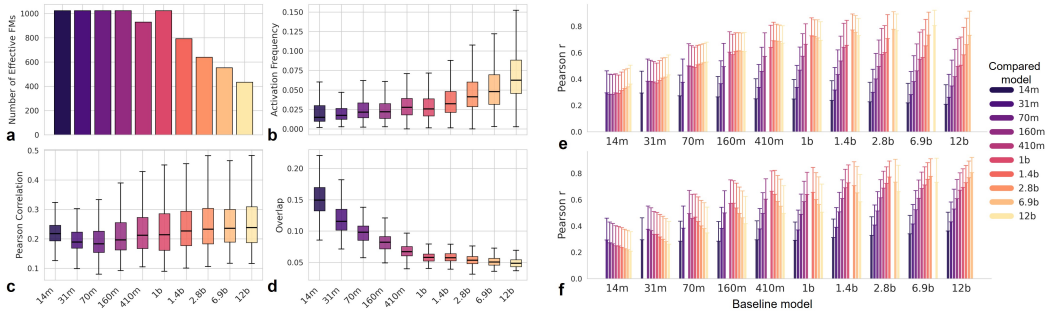

Figure 10: Scale comparisons at a common dictionary budget. Ten Pythia sizes use $M = 1024$, $K = 32$. (a,b) Active counts and usage. (c,d) Within-model best-match responses and salient-coordinate overlaps. (e,f) Cross-scale response alignment in both directions. Coordinate overlaps are measured within each model.

Changing model size changes the activation space that the dictionary must represent. We first hold $M = 1024$ and $K = 32$ across ten Pythia sizes to describe organization at a common decomposition budget. The fitted dictionaries of larger models use fewer active modules; within-model best-match response correlations rise, while salient-coordinate overlap falls (Figure 10a–d). These trends characterize populations recovered at this fixed budget.

At 70M, 410M, and 1.4B, the original fixed-width evaluation gives standardized EV of 0.5615/0.4932/0.3937 (Appendix Y). Evaluation of the existing 6.9B and 12B dictionaries on a separate, fixed 16-document test panel gives .2825/.2446; restoring document means and scales gives raw EV of .8814/.8904 (Appendix Z). On that panel, the two largest dictionaries capture about a quarter of standardized variation. The following capacity comparison varies M within the three smaller scales to examine response correspondence at comparable standardized EV (Table 2).

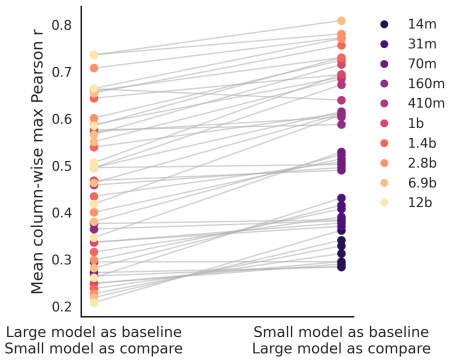

Figure 11: Directional cross-scale matching. Each line compares best-match correspondence with the reference and query model sizes exchanged.

The fixed-budget response matrices depend on matching direction (Figure 10e,f). Most pairs satisfy $\mathrm{Align}(s \leftarrow \ell) > \mathrm{Align}(\ell \leftarrow s)$ for smaller model s and larger model ℓ (Figure 11). A best-match maximum depends on the number and distribution of reference candidates, while the average also depends on the query population. Directional differences therefore summarize both response relationships and the matching inventories.

The capacity-adjusted analysis uses one-to-one assignment, allowing each selected module to enter one pair. For unequal widths, the reported mean describes similarity within the matched subset. Its interpretation concerns the responses in those pairs; the fraction of the larger inventory covered is a separate quantity. This preserves a common response-level comparison as dictionary capacity changes.

Correspondence at comparable reconstruction quality. With $K = 32$, dictionary widths of 256/512/4096 yield standardized-space explained variance of 0.431/0.428/0.443 for 70M/410M/1.4B (Table 2). Active counts then follow the available capacity. One-to-one response matching yields mean similarities in the 0.785–0.824 range, above shuffled-response controls using the same matching procedure ($p < .001$; Appendix AA).

Table 2: Capacity-adjusted comparison. Reconstruction is matched in the standardized training space; raw-space EV is shown separately.

Quantity	70M	410M	1.4B
Width M	256	512	4096
Standardized EV	.431	.428	.443
Raw EV	.988	.845	.856
Active modules	256	511	4067
Nominal K/M	12.50%	6.25%	.78%

Across these three scales, similar response subsets are recovered at comparable standardized-space reconstruction quality, with different dictionary widths and occupancies. Together with the checkpoint results, this connects recurring response structure to the changing model and to the capacity of the representation used to inspect it.

5 Discussion and Conclusion

Sparse cross-layer modules connect individual response patterns to population organization. A response can be inspected in text, located through its decoder profile, and compared with neighboring responses. The resulting picture combines broad depth order with heterogeneous input selectivity: modules within a depth region track different patterns, and correlated responses can have little overlap in their salient coordinates.

The same unit makes recurrence measurable across conditions. Domains assign different usage frequencies to a shared dictionary, while independent domain fits recover partially corresponding responses. Across training checkpoints, best response matches and best coordinate-set overlaps follow different trajectories. Across 70M, 410M, and 1.4B models, matched response subsets remain similar at comparable reconstruction quality. These observations connect the meaning of a particular response to where its decoder coordinates concentrate and where related responses recur.

The repeated fits and decomposition comparisons give this account a characteristic resolution. Broad depth allocations recur closely, while corresponding responses retain differences across fitted dictionaries. Layer energy follows the broad allocation; salient coordinates locate its upper tail in the residual basis. The responses measure variation within complete documents, with automated descriptions and selected interventions connecting that variation to inspectable inputs and local outputs. The response analyses span Pythia, and auxiliary families extend the depth-order comparison. At this mesoscopic scale, an individual interpretation becomes part of a population: its meaning can be read together with its location, its neighbors, and the conditions under which related responses recur.

AI USE STATEMENT

DeepSeek-V3 generated module descriptions and supported the Detection and Fuzzing procedure described in Appendix O. LLMs were used to improve the writing quality of this manuscript, including grammar correction and language polishing. The authors reviewed all AI-assisted output, and take full responsibility for the final content.

REFERENCES

[1] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

[2] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[3] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[4] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

[5] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[6] Thao Nguyen, Maithra Raghu, and Simon Kornblith. On the origins of the block structure phenomenon in neural network representations. *arXiv preprint arXiv:2202.07184*, 2022.

[7] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. https://transformer-circuits.pub/2024/crosscoders/index.html, 2024. Transformer Circuits Thread.

[8] Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models. *arXiv preprint arXiv:2503.08200*, 2025.

[9] Daniil Laptev, Nikita Balagansky, Yaroslav Aksenov, and Daniil Gavrilov. Analyze feature flow to enhance interpretation and steering in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 32593–32616. PMLR, 2025.

[10] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

[11] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. Transformer Circuits, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

[12] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[13] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[14] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[15] Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

[16] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

[17] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720.

[18] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, et al. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[19] Pascal Pfeiffer, Philipp Singer, Yauhen Babakhin, Gabor Fodor, Nischay Dhankhar, and Sri Satish Ambati. H2O-Danube3 technical report. *arXiv preprint arXiv:2407.09276*, 2024.

[20] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

[21] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[22] Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. CodeSearchNet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*, 2019.

[23] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[24] Deniz Bayazit, Aaron Mueller, and Antoine Bosselut. Crosscoding through time: Tracking emergence & consolidation of linguistic representations throughout LLM pretraining. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1353–1377. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.60.

[25] Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

[26] Ruixuan Deng, Xiaoyang Hu, Miles Gilberti, Shane Storks, Aman Taxali, Mike Angstadt, Chandra Sripada, and Joyce Chai. Sparse feature coactivation reveals causal semantic modules in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3419–3451. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.155.

[27] Yanke Yu, Jin Li, Ying Sun, Ping Li, Zhefeng Wang, and Yi Zheng. Discovering decoupled functional modules in large language models. *arXiv preprint arXiv:2603.17823*, 2026.

[28] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[29] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Extended Related Work

From response descriptions to computational interactions. Neuron-level probing and sparse dictionary learning identify response patterns in language models [1–3, 25]. Automated interpretation evaluates natural-language descriptions through their predictive relationship to held-out activations [15]. Causal feature circuits construct graphs of feature interactions and test their behavioral relevance [10]; circuit tracing uses replacement models and attribution graphs to analyze how internal features contribute to outputs [11]. Here the empirical object is a population of sparse responses paired with full-stack decoder directions, examined across depth, input domains, training checkpoints, and model scales.

Aggregate and feature-resolved organization. Convergent-learning studies compare representations across independently trained networks [4]. CKA provides a representation-similarity measure [5], and block-structure analyses relate contiguous groups of similar layers to training and data [6]. These methods describe aggregate organization. Our module-level measurements compare response correlations, upper-tail salient-coordinate overlap, and domain invocation within a fitted population. The coordinate-based overlap is measured in each model's residual basis, whereas a layer-to-layer similarity scalar describes representations globally.

Cross-layer sparse coding and feature flow. Sparse Crosscoders learn features spanning layers and models and examine their depth distributions and changing directions [7]. RouteSAE combines routing with shared sparse learning across layers [8]. Feature Flow maps layer-local SAE features through decoder-direction similarity, tracing persistence and transformation across depth [9]. Our tied, undercomplete TopK dictionary belongs to this methodological family. The contribution here is an empirical account of coarse, cross-layer population organization: how response similarity, salient-coordinate profiles, and reuse vary together. Layer-local SAE comparisons describe a different reconstruction target and analysis resolution.

Module discovery and training dynamics. Coactivation-based methods group sparse features into semantic components and test concept, relation, and compositional interventions [26]. ULCMOD jointly discovers cross-layer neuron modules and associated input topics through an iterative decoupling objective [27]. These methods study modular organization through feature coactivation or neuron–input associations; our measurements concern a jointly learned response and decoder direction. Crosscoders have also tracked linguistic representations through training [24]. Our checkpoint analysis compares separately maximized response similarities and salient-coordinate overlaps, describing correspondence between independently fitted populations across training.

Appendix Guide

These appendices specify the cross-layer dictionary and the measurements applied to its responses and decoder directions. An artificial neuron (AN) in the original figures denotes one residual layer–coordinate pair; a functional module (FM) is one sparse-code coordinate together with its signed, full-stack decoder direction. Training and preprocessing lead into depth grouping, its color–FM registry, and signed-weight and energy summaries. Repeated fits and common-input decompositions then examine the response–direction unit. Interpretation scores and activation-frequency adjustment precede domain reuse, checkpoint matching, and scale comparisons. The final sections give the intervention analyses and the complete CEBRA module-map atlas.

A TopK-SAE Training Details and Hyperparameter Settings

The dictionary is learned from token-level activations of a frozen language model. This section specifies the architecture, data processing, optimization, and default hyperparameters used throughout the main analyses.

A.1 Training data and on-demand activation extraction

TopK-SAE is trained on joint cross-layer activations extracted from a frozen language model, using on-demand extraction of each mini-batch of activations.

Dataset split. We use the Pile-10K corpus. The first 8000 documents are used for training the autoencoder, and the remaining 2000 documents are reserved for evaluation. Documents are indexed deterministically, and the training and test splits are fixed across all runs.

On-demand extraction. During training, each mini-batch consists of a batch of document indices. For each batch:

- The corresponding raw texts are loaded from disk.
- Texts are tokenized using the pretrained tokenizer associated with the target Pythia checkpoint.
- The frozen GPT-NeoX model is executed with `output_hidden_states=True`.
- The slices in `hidden_states[1:]` are concatenated: the first $L-1$ block outputs and the final normalized state. The resulting token-level joint activation has dimension

$$n = L \times d_{\text{model}}.$$

Document-wise normalization. For each document independently, each layer–coordinate dimension is standardized over the document's token positions. This normalization is performed in `float32` and includes explicit handling of NaN and infinite values. The resulting normalized activations are then concatenated across documents in the batch to form the input to the SAE.

The transformation removes document-specific means and scales and therefore emphasizes token-varying structure within each document. Because the statistics include all token positions, including later positions, the resulting code is a retrospective full-document measurement rather than a prefix-causal hidden state.

All language-model parameters remain frozen throughout training; gradients are computed only for the SAE parameters.

A.2 TopK-SAE architecture

The TopK-SAE consists of a single shared dictionary matrix

$$D \in \mathbb{R}^{M \times n},$$

where M is the number of latent features (functional modules) and n is the joint activation dimension.

Encoder. Given an input activation vector $\mathbf{x} \in \mathbb{R}^n$, the encoder computes similarity scores using ℓ_2-normalized dictionary rows:

$$\mathbf{s} = D_{\text{norm}}\mathbf{x}, \quad D_{\text{norm},i} = \frac{D_i}{\|D_i\|_2 + \varepsilon}.$$

Top-K sparsification. For each input vector, ReLU truncation and Top-K gating retain at most K positive scores, as in Equation (2). Retaining the top K scores before ReLU gives the same positive code. A module is active for a token precisely when its resulting code coordinate is positive. Sparsity is enforced on these token responses; it does not threshold the signed decoder direction.

Decoder. Reconstruction is performed using the same normalized dictionary:

$$\hat{\mathbf{x}} = D_{\text{norm}}^{\top}\mathbf{c},$$

where $\mathbf{c}$ is the sparse code produced by the encoder.

Loss function. The model is trained using mean squared error:

$$\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

averaged over all tokens in the batch.

A.3 Optimization and training procedure

Optimizer. We use AdamW to optimize the SAE parameters. The default settings are $lr = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay 0.01. The data-fraction and common-panel fits use Adam with their separately stated settings.

Batching. Mini-batches are defined over documents rather than tokens. Because document lengths vary, the effective number of token-level activation vectors per batch is variable. Unless stated otherwise, the document batch size is set to 2.

Epoch-based training. Training proceeds for a fixed number of epochs over the document index set. In each epoch: (i) All training documents are iterated once (with shuffling); (ii) The SAE is trained using online-extracted activations; (iii) Evaluation loss is computed on the held-out document set.

Distributed training. When multiple GPUs are available, training is performed using Distributed Data Parallel (DDP). Each process holds its own instance of the frozen language model and SAE. Training and evaluation losses are averaged across ranks using all-reduce.

A.4 Hyperparameter settings

Unless otherwise specified, all TopK-SAE experiments reported in the paper use the following default configuration:

Table 3: Default TopK-SAE training hyperparameters.

Hyperparameter	Value
Number of latent features M	1024
Top-K sparsity K	32
Document batch size	2
Number of training epochs	512
Maximum sequence length	2048
SAE parameter dtype	float32
Language model dtype	float16 (frozen)
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	0.01

The choice of explicit Top-K sparsity follows Gao et al. [3]; $K = 32$ is the default setting for this study. The sensitivity sweep tests alternatives rather than assuming that one sparsity budget is appropriate at every scale.

B Independent normalization controls

Document-wise preprocessing centers and rescales each layer–coordinate response over all token positions of its document. This retains within-document token variation in a retrospective code. We independently retrained dictionaries using two other standardizations: one mean and scale per

coordinate estimated from the training set, and per-token/per-layer standardization. The comparison also includes a fit on unstandardized concatenated activations. Relative to the document-wise fit, matched salient-coordinate profile similarities are 0.9338 for global-coordinate and 0.8330 for per-token/per-layer standardization. Both exceed the reported permutation controls and retain ordered early-to-late peaks. Unstandardized concatenation yields 0.1054 and concentrates selected coordinates in later layers (Table 4). The endpoint of these controls is coarse depth geometry. Response interpretation and domain invocation in the main analysis use document-wise normalization.

Table 4: Preprocessing controls. Matched salient-coordinate profile similarity to the document-wise reference and recovered depth organization. Each row represents an independent dictionary fit.

Preprocessing	Similarity	Depth organization
Document-wise	1.0000	Reference ordered peaks
Global coordinate	0.9338	Ordered peaks retained
Per-token/per-layer	0.8330	Ordered peaks retained
Unstandardized	0.1054	Later-layer concentration

C Data-fraction sensitivity

The data-fraction panel uses Adam at learning rate 3×10^{-4} and zero weight decay for 128 epochs. It evaluates subset-specific fits separately from the 512-epoch fixed-budget scale panel in Section Y. Its EV averages coordinate-wise values within each batch and then averages batches; the scale panel pools elementwise SSE and SST. It uses Pythia-410M with $M = 1024$, $K = 32$ and fractions of the 8000-document training split. At 25% of the data, EV is 0.5090 versus 0.5114 at full data, and template similarity is 0.9226. Template similarity is non-monotone across fractions, with the 50% run below the 25% run, indicating that coarse organization can emerge with substantially less data while still varying across training subsets.

Table 5: Data-fraction panel. Similarity is to the full-data templates; the observed sequence is not monotone.

Data fraction	Training tokens	EV	Template similarity	Silhouette
10%	0.55M	0.5048	0.8724	0.7023
25%	1.38M	0.5090	0.9226	0.7248
50%	2.75M	0.5104	0.8479	0.7231
100%	5.50M	0.5114	1.0000	0.7509

D Computational resources

Table 6: Compute-cost summary for the main Pythia-410M setting ($M = 1024, K_{\mathrm{SAE}} = 32$)**.**

Data frac.	**Docs**	**Tokens**	**Wall time**	**Peak mem.**	**Throughput**
10%	803	0.55M	4108 s (1.14 h)	3.42 GB	17.1k tok/s
25%	2006	1.38M	9913 s (2.75 h)	3.42 GB	17.8k tok/s
50%	3945	2.75M	18643 s (5.18 h)	3.42 GB	18.9k tok/s
100%	8000	5.50M	34479 s (9.58 h)	3.42 GB	20.4k tok/s

The cost records belong to the 128-epoch data-fraction fits. The full-data run processes 703,893,760 training tokens; dividing by 34,478.7 seconds gives approximately 20.4k tokens/s. Peak memory is the recorded CUDA allocator maximum.

E Choosing the clustering threshold for depth profiles

For module m, let $w_{m,j}$ be its signed decoder coefficient at residual layer–coordinate pair j and let $z_{m,j}$ be that coefficient standardized across the complete decoder direction within module m. We

define the upper-tail salient-coordinate set $S_m = \{j : z_{m,j} > \tau\}$, with $\tau = 1.65$ by default, and count selected coordinates by layer:

$$v_{m,\ell} = \big|\{j \in S_m : \text{layer}(j) = \ell\}\big|, \qquad \ell = 1, \ldots, L.$$

This salient-coordinate profile locates the upper tail in the model's residual layer–coordinate basis. Selection is relative to the within-module mean and scale. The complete signed direction, including coefficients outside S_m, determines the reconstruction contribution. Module activation frequency and output effects are measured separately. Only modules active on the test set enter the grouping analysis.

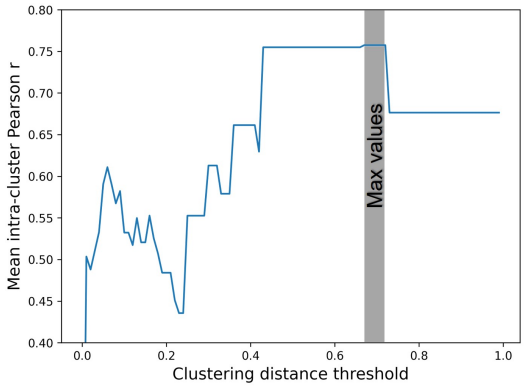

Figure 12: Choosing the hierarchical cut threshold t for depth-profile clustering. The x-axis is the clustering distance threshold t used to cut the dendrogram; the y-axis is the mean, across clusters, of their off-diagonal depth-profile correlations, with singleton scores set to zero. The shaded region marks the tied maximum at $t = 0.67$–0.72; the code uses the first maximizing grid value, $t^\star = 0.67$.

The hierarchical cut used in Figure 3 maximizes within-cluster depth-profile consistency over a grid of thresholds. The implementation uses the first threshold attaining the maximum score.

For each module m, we define a depth-profile vector $\mathbf{v}_m \in \mathbb{R}^L$, where the ℓ-th entry counts its upper-tail salient coordinates in layer ℓ. We stack these vectors as $V = [\mathbf{v}_1, \ldots, \mathbf{v}_{|\mathcal{M}|}] \in \mathbb{R}^{L \times |\mathcal{M}|}$. The grouping includes modules with test-set activation frequency $AF > 0$.

Step 1: Candidate thresholds. Enumerate $t \in \{0, 0.01, \ldots, 0.99\}$.

Step 2: Similarity construction and hierarchical clustering. Compute the module correlation matrix $R_{mm'} = \text{corr}(\mathbf{v}_m, \mathbf{v}_{m'})$, then apply average-linkage hierarchical clustering with correlation distance to the *rows of* R. The dendrogram distance is $d_{mm'} = 1 - \text{corr}(R_{m,:}, R_{m',:})$, comparing each module's similarity pattern to all active modules. Denote the resulting linkage matrix by Z.

Step 3: Cut the dendrogram at threshold t. Use $\text{fcluster}(Z, t, \texttt{criterion = distance})$ to obtain clusters $\{\mathcal{C}_1(t), \ldots, \mathcal{C}_{K(t)}(t)\}$.

Step 4: Define within-cluster consistency and aggregate into $S(t)$. For each cluster $\mathcal{C}_k(t)$, take the within-cluster submatrix $R_k(t)$. For clusters with at least two modules, the consistency score is the off-diagonal mean:

$$s_k(t) = \frac{\sum_{i \neq j,\ i,j \in \mathcal{C}_k(t)} R_k(t)_{ij}}{|\mathcal{C}_k(t)| \cdot (|\mathcal{C}_k(t)| - 1)}.$$

The implementation converts the undefined singleton score to zero. The overall score includes every cluster:

$$S(t) = \frac{1}{K(t)} \sum_{k=1}^{K(t)} s_k(t), \qquad s_k(t) = 0 \quad \text{if } |\mathcal{C}_k(t)| = 1.$$

Step 5: Select the final threshold and dominant groups. The recorded grid has the same maximum score at $t \in \{0.67, 0.68, 0.69, 0.70, 0.71, 0.72\}$. The implementation takes the first of these values, $t^\star = 0.67$, yielding clusters of 205, 166, 4, and 554 active modules. The original panels display the three clusters with more than 20 modules; the four-module residual cluster remains outside those groups. The fixed-k analysis in Section H measures recurrence of their coarse templates.

F DEPTH-CATEGORY COLORS AND FM IDENTIFIERS

The depth panels in Section 3.2 and Figure 3 use blue for deep (Category 1), green for middle (Category 2), and red for shallow (Category 3). Their saved category arrays contain 205 deep, 166 middle, and 554 shallow modules: 925 FMs in the three displayed groups. Four other active modules lie in the small residual cluster at the selected cut. Table 8 lists every displayed FM ID by color, with named headings for monochrome reading.

The interpretation records used in Section 3.4 and Figure 5 have saved labels for 921 uniquely scored modules: 203 deep, 169 middle, and 549 shallow. The figure and scored inventories share 918 IDs; 909 retain the same group label, while nine differ (Table 7). Scored IDs 215, 581, and 921 lie outside the displayed figure groups; figure-group IDs 221, 341, 644, 711, 712, 739, and 750 lack a usable unique score record. Description scores, AF-adjusted contrasts, and scored-group profiles use scored-record labels. The depth panels and this color lookup use figure labels.

Table 7: Category differences between original figure labels and scored-record labels. D/M/S denote deep/middle/shallow.

FM	Figure	Score record	FM	Figure	Score record
40	D	S	544	S	M
234	S	M	561	S	D
505	D	S	564	M	S
534	S	M	596	D	S
539	S	M			

The examples in Figure 6 identify FMs 349/648, 15/854, and 659/893 as deep, middle, and shallow in both records. Red tint on tokens in that figure marks response strength. The CEBRA atlas has a separate color scale: each tile heading identifies its FM, and point hue represents layer depth within that FM (Appendix AI).

G ALTERNATIVE GROUPING WITHOUT EXPLICIT LAYER COUNTS

We also grouped modules by (i) cosine similarity of complete decoder directions, (ii) full test-set response-correlation vectors, and (iii) response-correlation vectors computed on each module's top-document subset. These representations do not explicitly encode layer counts. Relative to the saliency-based grouping, their average depth-template similarities are 0.8783, 0.8448, and 0.8386, respectively. The response-based alternatives have near-zero adjusted Rand index and normalized mutual information against the saliency-group labels. Ordered average templates can therefore coexist with different assignments of individual modules. The saliency profile is the direct measurement of decoder-support location used for the main depth analysis.

H ROBUSTNESS OF DOMINANT DEPTH TEMPLATES

We performed a 144-setting sweep on Pythia-410M varying latent count $M \in \{512, 1024, 2048\}$, sparsity $K_{\mathrm{SAE}} \in \{16, 32, 64\}$, saliency threshold $\tau \in \{1.0, 1.65, 2.0, 2.5\}$, and cluster count $k \in \{2, 3, 4, 5\}$. Among the 36 settings with $k = 3$, median template similarity to the default solution is 0.9491, 97.2% have similarity at least 0.9, and 94.4% preserve strict increasing peak order. Under the default threshold $\tau = 1.65$, 8 of the 9 (M, K_{SAE}) settings preserve strict ordering. For $k = 4$ and $k = 5$, the additional clusters usually appear as tiny residual splits rather than qualitatively new depth regimes. The sweep measures recurrence of coarse templates; module assignments under alternative groupings are examined separately in Section G.

I SIGNED SUPPORT AND LAYER ENERGY

The shallow–middle–deep ordering of the decoder profiles in Section 3.2 persists when support is summarized by either sign of the salient tail, layer norm, or squared layer energy (Table 9). Positive and negative tails have nearly identical group centroids; energy gives more weight to the broad

Table 8: FM depth-category lookup for Figure 3. All 925 displayed FM IDs, listed in ascending order within each original color group. The headings retain the color mapping in monochrome.

Group / color	FM identifiers
Deep / blue	
$n = 205$	3, 7, 10, 11, 30, 38, 40, 54, 59, 60, 61, 63, 83, 97, 99, 108, 125, 132, 134, 144, 154, 158, 159, 164, 165, 168, 171, 172, 176, 186, 188, 195, 203, 204, 206, 207, 214, 226, 237, 238, 239, 243, 245, 250, 256, 258, 261, 268, 278, 280, 290, 291, 292, 294, 304, 305, 312, 313, 315, 318, 323, 348, 349, 367, 368, 373, 376, 386, 391, 399, 405, 406, 414, 418, 420, 425, 427, 436, 438, 439, 440, 443, 450, 451, 453, 454, 465, 469, 471, 489, 493, 494, 495, 502, 505, 508, 514, 516, 519, 520, 521, 526, 528, 532, 535, 540, 546, 553, 567, 571, 575, 577, 586, 587, 595, 596, 599, 603, 604, 614, 615, 620, 621, 624, 634, 636, 639, 643, 646, 647, 648, 652, 653, 656, 660, 664, 667, 669, 674, 677, 679, 688, 689, 690, 695, 696, 697, 698, 703, 720, 722, 729, 736, 737, 740, 751, 757, 758, 761, 763, 765, 767, 769, 780, 789, 791, 795, 796, 797, 813, 822, 827, 828, 830, 833, 835, 847, 852, 858, 864, 868, 871, 872, 882, 896, 902, 903, 910, 918, 927, 933, 935, 941, 943, 961, 963, 964, 970, 972, 977, 979, 987, 990, 1002, 1013
Middle / green	
$n = 166$	1, 15, 17, 18, 20, 21, 23, 34, 43, 45, 50, 52, 55, 62, 65, 77, 80, 88, 91, 103, 111, 124, 129, 156, 160, 169, 179, 190, 191, 196, 199, 200, 211, 233, 235, 240, 248, 252, 254, 267, 272, 277, 281, 284, 285, 287, 288, 317, 322, 327, 330, 340, 354, 359, 372, 385, 394, 398, 403, 411, 421, 433, 434, 442, 444, 447, 470, 473, 474, 476, 479, 492, 504, 547, 548, 550, 563, 564, 568, 570, 576, 590, 593, 600, 606, 608, 610, 611, 613, 632, 638, 641, 655, 658, 662, 663, 665, 675, 680, 682, 684, 693, 701, 704, 714, 715, 716, 718, 721, 723, 735, 746, 773, 776, 779, 783, 784, 785, 786, 790, 794, 798, 800, 803, 806, 814, 820, 825, 826, 829, 838, 843, 846, 851, 854, 861, 873, 876, 878, 886, 888, 891, 899, 901, 906, 907, 909, 913, 916, 936, 945, 950, 952, 954, 956, 969, 982, 985, 992, 996, 997, 999, 1000, 1006, 1007, 1023
Shallow / red	
$n = 554$	0, 2, 4, 5, 6, 8, 9, 12, 13, 14, 16, 19, 22, 24, 25, 26, 27, 28, 32, 33, 35, 36, 37, 39, 41, 42, 44, 46, 48, 49, 51, 53, 56, 57, 58, 66, 68, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 81, 82, 84, 85, 86, 87, 90, 92, 93, 94, 95, 100, 101, 102, 104, 106, 107, 109, 110, 112, 113, 116, 117, 118, 119, 120, 121, 122, 123, 126, 127, 128, 130, 131, 133, 135, 136, 137, 139, 140, 141, 142, 145, 146, 147, 148, 149, 150, 151, 152, 153, 155, 157, 161, 162, 163, 166, 167, 170, 173, 174, 175, 177, 178, 180, 181, 182, 183, 185, 189, 192, 193, 194, 197, 201, 202, 205, 209, 212, 213, 217, 218, 220, 221, 222, 223, 224, 225, 227, 229, 230, 231, 232, 234, 242, 244, 246, 247, 249, 251, 253, 255, 257, 259, 260, 262, 263, 264, 265, 266, 269, 270, 271, 273, 274, 275, 276, 282, 283, 286, 289, 295, 296, 298, 299, 300, 301, 302, 306, 307, 308, 309, 310, 311, 314, 316, 319, 320, 321, 324, 325, 326, 328, 329, 331, 332, 333, 334, 335, 336, 337, 338, 339, 341, 342, 343, 345, 346, 347, 350, 351, 352, 353, 356, 357, 358, 360, 361, 363, 364, 365, 370, 371, 375, 377, 380, 381, 382, 383, 384, 387, 388, 390, 393, 395, 396, 397, 400, 401, 402, 407, 409, 410, 413, 415, 416, 417, 422, 423, 426, 428, 431, 432, 437, 441, 446, 448, 449, 452, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 466, 467, 468, 472, 477, 478, 480, 481, 483, 484, 485, 486, 487, 488, 490, 491, 496, 497, 498, 499, 500, 501, 503, 506, 507, 509, 510, 511, 512, 517, 522, 523, 525, 527, 529, 530, 531, 533, 534, 536, 537, 538, 539, 541, 543, 544, 545, 549, 551, 552, 554, 555, 556, 559, 560, 561, 562, 565, 569, 573, 578, 580, 583, 585, 588, 589, 592, 594, 597, 598, 601, 602, 605, 607, 609, 616, 617, 618, 619, 622, 623, 625, 626, 627, 628, 630, 631, 633, 635, 637, 642, 644, 645, 649, 650, 654, 659, 666, 670, 671, 672, 676, 678, 681, 683, 685, 686, 687, 691, 694, 699, 700, 702, 705, 707, 709, 710, 711, 712, 713, 717, 725, 726, 727, 728, 730, 731, 732, 733, 734, 738, 739, 741, 742, 745, 747, 749, 750, 752, 753, 754, 756, 759, 760, 764, 766, 768, 770, 771, 772, 774, 775, 777, 781, 782, 787, 788, 792, 793, 801, 802, 804, 805, 807, 808, 809, 810, 811, 815, 816, 817, 818, 819, 824, 831, 832, 834, 836, 837, 839, 840, 841, 842, 844, 848, 849, 850, 853, 855, 856, 857, 859, 862, 863, 866, 867, 870, 874, 875, 877, 879, 880, 881, 883, 884, 885, 887, 889, 890, 892, 893, 894, 895, 897, 898, 900, 904, 905, 908, 911, 912, 914, 915, 917, 919, 920, 922, 923, 924, 925, 926, 928, 929, 930, 931, 932, 934, 937, 938, 939, 940, 942, 944, 946, 947, 951, 953, 957, 958, 959, 960, 965, 966, 967, 968, 971, 973, 974, 975, 980, 981, 983, 984, 986, 988, 989, 991, 993, 994, 995, 998, 1001, 1003, 1004, 1005, 1008, 1010, 1011, 1012, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022

portions of each signed direction. The default Pythia-410M registry contains 929 modules with positive test activation frequency. The fixed-group calculation uses the 921 uniquely scored modules and their interpretation-record labels: 549 shallow, 169 middle, and 203 deep (Appendix F).

Profile definitions. For decoder block $\mathbf{w}_m^{(\ell)}$, define layer energy

$$e_{m,\ell} = \frac{\|\mathbf{w}_m^{(\ell)}\|_2^2}{\sum_{q=1}^{24} \|\mathbf{w}_m^{(q)}\|_2^2}, \qquad \mu_m = \sum_{\ell=1}^{24} \ell e_{m,\ell}.$$

The other profiles replace the numerator by the layer ℓ_1 norm, the layer ℓ_2 norm, positive or negative squared-weight mass, or counts of coordinates satisfying $z > 1.65$, $z < -1.65$, or $|z| > 1.65$. Weight z-scores are computed within each full dictionary row using the sample standard deviation. Every profile is normalized to sum to one across layers. Group centroids average the module centroids; group peaks are the maxima of the mean normalized profiles.

Table 9: Alternative profiles with fixed scored-module groups. Centroids use layer indices 1–24. The right column gives shallow/middle/deep group-mean peak layers.

Profile	Shallow	Middle	Deep	Peaks
Positive tail	8.20	12.84	17.60	2/10/23
Negative tail	8.20	12.85	17.59	2/10/23
Absolute tail	8.21	12.85	17.60	2/10/23
Squared layer norm (energy)	9.50	12.88	16.47	1/10/23
Layer ℓ_2 norm	10.87	12.86	14.93	2/10/23
Layer ℓ_1 norm	10.85	12.84	14.92	2/10/23
Positive squared mass	9.50	12.88	16.47	1/10/23
Negative squared mass	9.50	12.89	16.47	1/10/23

The deep–shallow centroid difference is 9.40 layers for positive-tail counts, 6.96 for energy, and 4.06 for layer ℓ_2 norms. Group membership remains fixed throughout these comparisons.

Repeating the calculation with the original 925-module figure inventory gives energy centroids 9.45/12.97/16.51 and positive-tail centroids 8.13/12.97/17.65 for shallow/middle/deep groups. The respective group-mean peaks remain 1/10/23 and 2/10/23. Layer ℓ_2 centroids are 10.84/12.91/14.96; all eight summaries retain the ordering. The registry in Appendix F specifies the two inventories.

Reclustering sensitivity. When all 929 active modules are reclustered with a fixed maximum of three clusters, positive-tail profiles yield sizes 371/4/554; energy gives 363/6/560 and adjusted Rand index (ARI) 0.952 against that partition. Layer ℓ_2 and ℓ_1 norms give 376/6/547 and ARI 0.915. Negative and absolute tails have ARI 0.697 and 0.692, with smallest clusters of seven and six modules. The high agreement is chiefly between two large partitions and a small third cluster. This fixed-count cut measures sensitivity to the grouping rule; the original panels use the dominant-group cut described in Appendix E.

Within-layer basis change. Three orthogonal changes of basis within each 1,024-coordinate layer block leave normalized energy unchanged to at most 4.45×10^{-9}. Median correlations of the positive-tail layer profiles before and after transformation are 0.9848–0.9853, although median same-index Jaccard of the selected masks is 0.0406–0.0409. These values locate the stable layer allocation and the basis dependence of individual coordinate indices. Each predetermined transform multiplies coordinates by random signs and applies a normalized Walsh–Hadamard matrix (seeds 20260924–20260926). Only fixed decoder weights are transformed; language-model activations are unchanged.

Response relationships and decoder summaries. Using a fixed uniform sample of 32,768 evaluation token rows (seed 20260924), we compute dense pre-TopK response Pearson correlations for the 921 scored modules. Across all unordered module pairs, Spearman correlation is 0.128 between response Pearson and positive-coordinate Jaccard, 0.037 between response Pearson and layer-energy cosine, and 0.764 between the two decoder summaries. Mean within-group energy cosine is 0.913/0.931/0.880 for shallow/middle/deep groups; the shallow–deep mean is 0.556. The

decoder summaries capture related depth structure, and their pairwise associations with signed response correlation are weaker. The pairwise entries share modules and evaluation tokens; these values describe one fitted dictionary.

J NULL BASELINES FOR DEPTH ORGANIZATION

We applied the same depth-profile clustering pipeline to random orthogonal and Gaussian directions, PCA components, layer permutations, and row-wise dimension permutations. For $k = 3$, the learned SAE has silhouette 0.7182 and Davies–Bouldin index 0.6358; PCA retains some ordered structure (silhouette 0.6473). Random-direction silhouettes are approximately 0.024–0.027. These measurements locate a structured component in the activations that PCA also partially captures, while the learned sparse directions yield more distinct depth profiles under this pipeline.

K CROSS-FAMILY DEPTH GEOMETRY

We applied the central depth-template analysis with $M = 1024$, $K_{\mathrm{SAE}} = 32$, and saliency threshold 1.65 to four auxiliary model families. Table 10 reports the recovered peak layers. The peak order describes fitted salient-coordinate profiles; this check covers only coarse depth geometry.

Table 10: Auxiliary-family depth peaks. Peak indices are ordered shallow, middle, deep.

Model family	Recovered peak layers	Ordered geometry?
BLOOM	[0, 9, 20]	Yes
H2O-Danube3	[0, 8, 12]	Yes
SmolLM2	[0, 14, 26]	Yes
OPT-350m	[0, 16, 23]	Yes

The models are BLOOM-560M, H2O-Danube3-500M, SmolLM2-360M, and OPT-350M. All four show separated early, middle, and late peaks under this setting. Their distinct peak positions exhibit the same coarse early-to-late order.

L COMPARISON OF EXPLANATORY RESOLUTIONS

The additional comparison uses a common held-out panel of 50,000 tokens. Linear CKA recovers aggregate layer continuity: mean adjacent-layer similarity is 0.897, and mean similarities at layer lags 2/4/8/12/16/20 are 0.821/0.717/0.616/0.488/0.249/0.145. Adjacent-layer mutual-nearest-neighbor matching of raw coordinates retains 1,014–1,024 of 1,024 coordinates per link, with nearly all matches corresponding to the same residual coordinate. The coordinate-level matches largely follow residual-coordinate identity, whereas module responses resolve input-selective patterns within these representations.

Single-layer TopK-SAEs at layers 0/14/23 give held-out raw-space EV of 0.661/0.919/0.594 and reconstruction correlations of 0.875/0.962/0.831. Their outputs remain layer-local. These EV values quantify reconstruction within the respective layer spaces; the joint fit reconstructs the full concatenated stack with its shared dictionary budget. A joint module retains one sparse response with a signed full-stack decoder direction, providing the object used for the depth, domain, and training analyses.

M PAIRED FITS ON A FROZEN ACTIVATION PANEL

At the final Pythia-410M checkpoint, independently initialized tied dictionaries recover similar full-stack depth allocation but partial one-to-one response correspondence. Across three seed pairs, validation-matched test projection scores correlate at .6393–.6398 and gated codes at .5916–.5972; positive-tail coordinate Jaccard is .257–.264 (Table 12). The comparison uses 100,000 training tokens from 132 documents and document-disjoint 50,000-token validation and test panels. Training follows the ordered-prefix policy of the data-fraction experiment; validation and test use the archived

manifest's text normalization and end-of-sequence insertion. Every fit receives identical input IDs, chunk boundaries, and float32 language-model activations. The main dictionaries use approximately 5.5M training tokens, so these results characterize the smaller fixed panel.

Architectures and normalization. Three tied TopK dictionaries use initialization seeds 42, 43, and 44. An untied dictionary uses seed 42, with independently optimized encoder and decoder matrices initialized to the same values. Both matrices have unit-normalized rows; tying is the architecture change in this seed-42 pair. All four use the document-wise normalization in Equation (2), denoted N1. A fifth tied dictionary, also initialized with seed 42, uses N2: coordinate means and population standard deviations estimated from the 100,000 training tokens and then held fixed for validation and test. N2 statistics are accumulated in float64. Both normalizations divide by standard deviation plus 10^{-6}.

All five dictionaries have $M = 1024$ and $K = 32$. Tied dictionaries contain 25,165,824 trainable matrix entries; the untied model contains 50,331,648. Training uses Adam with learning rate 3×10^{-4}, zero weight decay, 128 epochs, and two equally weighted document losses per update. The document-order seed is $20260924 + e$ for zero-indexed epoch e. The final epoch determines each evaluated dictionary. N1 and N2 standardized EV use different normalized targets; their inverse-transformed raw EV uses the same original activation target.

Matching and evaluation. Validation responses determine one-to-one Hungarian assignments, which are then evaluated on the disjoint test panel. Pre-TopK projection scores and post-TopK codes each provide a matching criterion and a test response variable. All 1,024 features remain in the inventory, with zero correlation assigned to a zero-variance response. Global reciprocal-neighbor coverage is computed from the full similarity matrix independently of the Hungarian assignment. One random permutation of feature assignments provides a descriptive reference.

Decoder comparisons use normalized layer-energy profiles and positive-tail coordinate sets. Each row's positive-tail mask uses its sample standard deviation and threshold $z > 1.65$. N1/N2 decoder profiles are expressed in their respective normalized coordinate systems. The deposited summaries include source hashes and document offsets; reproducing the assignments requires the archived response arrays and decoders. Seed comparisons here hold the final language-model checkpoint and training size fixed. The training-trajectory analysis compares dictionaries fitted at different language-model checkpoints.

Held-out reconstruction and code use. The three N1 tied seed fits have similar test reconstruction: raw explained variance (EV) is 0.8439–0.8451 and standardized EV is 0.4677–0.4687 (Table 11). The untied N1 fit reaches 0.8454 raw and 0.4758 standardized EV with twice as many matrix parameters. The tied N2 fit reaches 0.7609 raw and 0.4892 standardized EV. Raw EV uses the same original activation target across fits; standardized EV uses each normalization's target. Every fit has test mean $L_0 = 32$, with eight to 20 modules never active on test.

Table 11: Final-checkpoint paired-fit reconstruction on 50,000 held-out test tokens. EV is pooled over 1,228,800,000 activation elements. L_0 is the mean number of active codes per token; dead modules have zero test activations. N1 uses document-wise normalization and N2 uses training-panel global statistics.

Fit	Raw EV	Standardized EV	L_0	Dead/1024
Tied N1, seed 42	.8439	.4681	32	10
Tied N1, seed 43	.8445	.4677	32	8
Tied N1, seed 44	.8451	.4687	32	10
Untied N1, seed 42	.8454	.4758	32	13
Tied N2, seed 42	.7609	.4892	32	20

Correspondence across fits. The corresponding single-permutation means are 0.0014–0.0031 for scores and -0.0003–0.0025 for codes. On validation, the full correlation matrices contain 742–750 global reciprocal score-nearest-neighbor pairs out of 1,024 queries; 690–703 of the one-to-one assigned pairs are reciprocal. Independently matching sparse codes gives test code correlations of 0.5956–0.5988 and 684–685 reciprocal code-nearest-neighbor pairs. Means include all 1,024 modules, with zero correlation assigned to zero-variance responses.

Table 12: Fixed validation matches evaluated on the disjoint test panel. r_s and r_c are mean matched test Pearson correlations for dense scores and sparse codes. "Recip." counts global mutual nearest neighbors on validation (of 1,024 queries), including ones outside the Hungarian assignment. J is mean positive-tail decoder-coordinate Jaccard; J_{rand} comes from one permutation of assigned partners. C_E is mean cosine similarity of normalized 24-layer decoder-energy profiles. All means include 1,024 assigned pairs.

Pair	Match on	r_s	r_c	Recip.	J	J_{rand}	C_E
42–43, tied N1	scores	.640	.597	747	.264	.027	.976
	codes	.621	.598	684	.260	.027	.976
42–44, tied N1	scores	.640	.596	750	.260	.027	.976
	codes	.621	.599	685	.257	.027	.975
43–44, tied N1	scores	.639	.592	742	.257	.028	.974
	codes	.619	.596	684	.253	.027	.973
42 tied–42 untied, N1	scores	.592	.581	719	.248	.027	.975
	codes	.577	.587	680	.245	.028	.972
42 tied N1–42 tied N2	scores	.569	.546	700	.236	.027	.956
	codes	.546	.554	642	.228	.026	.955

Under score matching, the untied N1 and N2 comparisons have test dense correlations of 0.5919 and 0.5691, below the independent-seed N1 range. Their test sparse-code correlations are 0.5808 and 0.5461. Matching on codes selects distinct assignments and gives the test values in the table. Across the five score-matched comparisons, one random assignment per comparison gives test correlations of 0.0014–0.0056 for dense scores and -0.0010–0.0025 for sparse codes. These single permutations serve as descriptive references rather than a sampling distribution.

Across independent seeds, matched normalized energy-profile cosines are 0.974–0.976; a uniform random permutation of partners has expectation 0.846–0.847, computed as the dot product of the two inventories' mean unit-length profiles. Positive-tail coordinate Jaccard is 0.257–0.264 (random assignment: 0.027–0.028). For tied N1 versus tied N2, energy cosine is 0.956 against a random expectation of 0.829, and coordinate Jaccard is 0.236 under score matching. Each normalization expresses decoder coordinates in its own standardized space. At the final language-model checkpoint and this training size, broad depth distributions recur across fits while individual responses and salient coordinates vary. These paired measurements describe the response and full-stack direction together under changes to initialization, tying, and normalization.

N Decomposition baselines on a common activation panel

The decomposition determines which response coordinates are available even when each fit reconstructs the same 24-layer activation stack. On the fixed panel of Appendix M, rank-1,024 PCA has the highest explained variance but uses 1,024 dense signed coefficients per token. The joint TopK fit uses 32 active coefficients, the selected ReLU crosscoder averages 34.82, and the 24 independent layer dictionaries use 768 in total (Table 13). All methods use the same N1 activations, 100,000 training tokens, and document-disjoint 50,000-token validation and test panels. Full-stack reconstruction uses common pooled denominators. Joint methods have 1,024 components; the local inventory has 24,576. The selected-layer control in Appendix L instead evaluates three individual layer spaces.

Table 13: Held-out full-stack reconstruction and activity on the common N1 panel. Raw and standardized explained variance (EV), mean active coefficients per test token, and parameter budgets come from one reported fit per architecture. PCA uses 1,024 dense signed coefficients, with fitted basis entries as its parameter budget. The local count sums 32 positive codes in each of 24 layers.

Representation	Raw EV	Std. EV	Activity	Parameters/basis
Tied TopK-32	.8439	.4681	32	25,165,824
Untied TopK-32	.8454	.4758	32	50,331,648
ReLU crosscoder	.8255	.4235	34.82	50,357,248
Rank-1,024 PCA	.9080	.7242	1,024 dense	25,165,824
Layer-local TopK (24 SAEs)	.8620	.5291	768	50,380,800

Standard ReLU crosscoder. The crosscoder has an untied affine encoder, ReLU codes, an affine decoder, and a unit-norm gauge for each concatenated decoder row. With $D = 24 \times 1024$ activation coordinates, its per-token objective is

$$\mathcal{L}_{\mathrm{CC}} = \frac{\|x - \hat{x}\|_2^2}{D} + \frac{\lambda}{D} \sum_{m=1}^{1024} a_m \sum_{\ell=1}^{24} \|\mathbf{w}_m^{(\ell)}\|_2,$$

where a_m is the nonnegative ReLU code and $\mathbf{w}_m$ is the normalized decoder row. Token losses are averaged within each document, with two equally weighted document losses per update. It uses the same width, float32 cache, N1 normalization, seed 42, document order, and 128 epochs as the TopK fits. Adam uses learning rate 10^{-4} for encoder and decoder matrices and decoder bias, and 10^{-2} for encoder bias; the TopK fits use 3×10^{-4}.

The penalty was selected using validation mean L_0 alone. Three fixed 128-epoch fits at $\lambda = 0.1, 1, 10$ gave validation L_0 of 817.85, 139.92, and 2.48. A prespecified one-fit interpolation gave $\lambda = 2.3223029464$ and validation $L_0 = 37.0961$, closest to 32 among the four fits but outside the predefined 10% interval $[28.8, 35.2]$. Its test L_0 is 34.8163 and its raw/standardized EV is .8255/.4235. The selection record and final run metadata are archived with the validation results.

Response and decoder correspondence. For each joint baseline, validation responses determine a one-to-one Hungarian assignment to the tied seed-42 modules; the fixed assignment is evaluated on test. Dense pre-gate scores and sparse codes are matched separately. PCA uses the absolute validation correlation of signed coefficients and a validation-selected sign; its correspondence analysis uses signed scores. Decoder geometry uses normalized 24-layer energy-profile cosine and, for sparse models, positive-tail coordinate Jaccard, with $z > 1.65$ using each row's sample standard deviation. Table 14 reports means across all 1,024 assigned pairs, retaining zero-variance responses with correlation zero. Each random column is one fixed random assignment, a descriptive scale reference.

Table 14: Validation-selected joint-component matches on held-out test responses. r is mean Pearson correlation; C_E is mean matched decoder layer-energy cosine; J is mean positive-tail coordinate Jaccard for the sparse models. PCA contributes signed-score and decoder-energy comparisons.

Tied versus	Match on	Test r	Random r	C_E	J
Untied TopK	Scores	.5919	.0084	.9745	.2480
	Codes	.5867	.0002	.9725	.2449
ReLU crosscoder	Scores	.4975	.0115	.9563	.2139
	Codes	.4905	.0022	.9547	.2077
PCA	Signed scores	.0821	.0168	.9151	—

Under the validation assignment, crosscoder responses correlate near .5 with tied-module responses and their decoder-energy profiles have cosines near .95. PCA allocates energy broadly across layers while producing different response coordinates. The comparison uses one selected fit per architecture at this training size; repeated tied fits are in Appendix M.

Randomized PCA. We fit rank-1,024 PCA after subtracting the training-token coordinate mean. A Gaussian sketch with 64 oversampling directions and two power iterations uses seed 20260924. Matrix products stream over training-token batches; the final small Gram matrix, eigendecomposition, and recovery of right singular vectors use float64. The fitted basis is then stored in float32. Its maximum orthogonality error is 3.58×10^{-7}, and its training projection captures .7609 of standardized variance. PCA weights training tokens equally; the SAE updates average two document losses equally. Test standardized and inverse-transformed raw EV are .7242 and .9080, respectively. All 1,024 coefficient directions are used for reconstruction.

Each PCA direction's stored sign makes its largest-magnitude loading positive. To compare response coordinates with joint modules, validation Hungarian assignment maximizes absolute Pearson correlation; the sign of each selected pair is also determined on validation and held fixed on test. Mean test correlation with joint projection responses is .0821, compared with .0168 for one random assignment using validation-selected signs. The corresponding mean layer-energy cosine is .9151. PCA directions have a mean effective layer count of 23.68, compared with 21.64 for the tied dictionary, where effective layer count is $\exp[-\sum_\ell e_{m,\ell} \log e_{m,\ell}]$. The two decompositions therefore have broad layer allocation and different response coordinates on the shared activation panel.

Independent layer-local SAEs. Each of the 24 layers has an untied TopK SAE with 1,024 features, $K = 32$, learned encoder and decoder biases, an unconstrained encoder, and unit-normalized decoder rows. Its encoder receives only that layer's activations. The 50,331,648 matrix parameters equal the joint untied model's matrix count; local biases add 49,152 parameters. The local inventory contains 24,576 features. Training uses the common seed 42, document ordering, Adam learning rate 3×10^{-4}, and 128 epochs. Microbatches accumulate each complete document's loss before the optimizer update. Full-stack test standardized/raw EV are .5291/.8620, with a mean total of 768 active coefficients per token. All local features activate at least once on test. This reconstruction uses a separate sparse code at each layer.

Matching local features. For each adjacent layer pair, a validation Pearson matrix determines a one-to-one assignment of all 1,024 features. Chaining the 23 assignments yields 1,024 trajectories beginning in layer 1. A second analysis independently matches the joint dictionary to each layer's local dictionary. Dense scores and gated codes define separate assignments; the test panel evaluates the fixed pairs. Zero-variance features remain in the inventory with correlation zero. The random reference uses one independently permuted assignment per comparison.

Table 15: Held-out correspondence of independently encoded local features. Means average the 1,024 assigned feature pairs and, for the first two comparisons, the indicated layer links. Assignments use validation only.

	Projection scores		Gated codes	
Comparison	Matched	Random	Matched	Random
Adjacent layers (23 links)	.3676	.0151	.4049	.0014
Joint to local (24 layers)	.3275	.0033	.3716	.0007
Trajectory: layer 1 to 6	.0701	.0107	.1280	.0018
Trajectory: layer 1 to 12	.0283	.0033	.0562	−.0002
Trajectory: layer 1 to 18	.0139	.0009	.0260	.0002
Trajectory: layer 1 to 24	.0041	.0015	.0085	.0008

Adjacent-layer assignments retain held-out response correspondence, and endpoint correspondence decreases along long chains. Each link optimizes a local relation; the composed assignments do not preserve a common held-out response across the entire chain. A joint module assigns one coefficient to its full-stack reconstruction direction. The local matching analysis characterizes these response relations alongside each model's reconstruction.

O Automated Module Interpretation and Scoring

Data and inclusion criteria Given a token sequence $\{x_t\}_{t=1}^{T}$ and a set of modules $\{FM_i\}_{i=1}^{M}$, we collect token-level activations $a_{t,i}$ on the test corpus. For storage efficiency, at each token t we keep only the top-K_{keep} module activations (here $K_{\text{keep}} = 32$) after ReLU truncation and set the rest to zero. For each module i, count its positive token responses

$$n_i^+ = |\{t : a_{t,i} > 0\}| .$$

If $n_i^+ < 200$, we skip interpretation generation and scoring to avoid unstable estimates due to insufficient samples.

Window construction and quantile-stratified sampling For each module i, we construct a fixed-length context window of size $W = 32$ around a center position c: $[c - 15,\ c + (W - 16)]$ (clipped at sequence boundaries). We compute decile thresholds (0.1–0.9 quantiles) over positive activations $\{a_{t,i} : a_{t,i} > 0\}$ and bucket activations into 10 quantiles. We then sample center positions uniformly across buckets to cover different activation strengths.

For interpretation generation, we sample 40 active windows per module. For Detection scoring, we sample 100 active windows as positives and 100 all-zero windows (within the window, $a_{t,i} = 0$) as negatives. Interpretation generation and scoring use held-out contexts, separating description generation from the examples used to evaluate predictive consistency.

Prompt construction and interpretation generation Within each sampled window, we merge consecutive triggered tokens (where $a_{t,i} > 0$) into spans and mark them with delimiters (e.g., « »). We also attach activation-strength information per triggered token. The prompt concatenates 40 examples, each containing (i) the marked window text and (ii) a (token, quantile) list. We constrain the output to a one-sentence explanation and require the final line to start with `[interpretation]:`. We follow the interpretation framework of Paulo et al. [15].

Scoring: Detection and Fuzzing The two reported evaluation tasks follow Paulo et al. [15]: **Detection**: given an interpretation and unmarked windows, predict whether each window is active vs. inactive; score is accuracy. **Fuzzing**: given an interpretation and marked windows, predict whether the marks correctly cover all trigger tokens; score is accuracy. We randomize evaluation items and score them in small batches. Detection evaluates whether the description predicts module activation on unseen windows, whereas Fuzzing evaluates how precisely the description identifies the marked trigger span. The scores describe predictive consistency of generated descriptions within the automated procedure. Score distributions and counts above the descriptive 0.5 reference are summarized in Section Q.

P EMBEDDING-BASED SEMANTIC CONSISTENCY

Let M be the number of modules. For each token t we have: (i) module activations $A \in \mathbb{R}^{T\times M}$, and (ii) input embedding $e_t \in \mathbb{R}^d$. We construct a module similarity matrix $S \in \mathbb{R}^{M\times M}$ where S_{ij} is the mean cosine similarity between embeddings of tokens activated by module i and those activated by module j. Let $\Omega_i = \{t \mid A_{t,i} > 0\}$. If Ω_i or Ω_j is empty, we skip the pair. Otherwise we sample with replacement $K_{\text{sim}} = 1000$ indices from each set and compute

$$S_{ij} = \frac{1}{K_{\text{sim}}} \sum_{k=1}^{K_{\text{sim}}} \frac{\langle e_{t_i^{(k)}}, e_{t_j^{(k)}} \rangle}{\|e_{t_i^{(k)}}\|_2 \, \|e_{t_j^{(k)}}\|_2}.$$

Within a depth group, we report *Intra-module consistency* (mean±std of diagonal entries) and *Inter-module similarity* (mean±std of off-diagonal entries).

Q MODULE INTERPRETATION AND SCORE SUMMARIES

The descriptive unit is one scored module in the fitted Pythia-410M dictionary. The analysis covers 921 such modules: 203 deep, 169 middle, and 549 shallow. Each supplementary interpretation entry gives the module's generated one-sentence description, its Detection and Fuzzing scores, and five marked example contexts. The > 0.5 counts in Table 16 summarize the score distributions using a common descriptive threshold. In the full scored population, 837 modules exceed this threshold for Detection, 873 for Fuzzing, and 804 for both.

Table 16: Interpretation-score coverage. Counts and within-group percentages for Detection and Fuzzing. The > 0.5 threshold is used as a descriptive reference point.

Group	Scored	Detection > 0.5	Fuzzing > 0.5	Both > 0.5
Deep	203	188 (92.6%)	177 (87.2%)	168 (82.8%)
Middle	169	153 (90.5%)	164 (97.0%)	151 (89.3%)
Shallow	549	496 (90.3%)	532 (96.9%)	485 (88.3%)
All	921	837 (90.9%)	873 (94.8%)	804 (87.3%)

Population-level score distributions. Detection and Fuzzing are positively correlated across modules ($r = 0.5603$), but their depth dependence differs. Detection remains broadly similar across groups, whereas the deep group shows lower Fuzzing scores than the middle and shallow groups. Table 17 gives the corresponding means and medians, and Figure 5 visualizes the joint score distribution.

Between-group comparisons. Table 18 reports nominal two-sided Welch tests for the three depth-group pairs on each score, with Benjamini–Hochberg adjustment across all six comparisons. Detection

Table 17: Score distributions by depth group. Means and medians over the scored module population.

	Detection		Fuzzing	
Group	Mean	Median	Mean	Median
Deep	0.6110	0.600	0.5889	0.575
Middle	0.6372	0.625	0.6237	0.620
Shallow	0.6030	0.585	0.6306	0.625
All	0.6111	0.600	0.6202	0.615

differs most clearly for middle versus shallow; Fuzzing differs for deep versus middle and deep versus shallow. The tests summarize module-level contrasts within one fitted dictionary under the Welch sampling assumptions. Shared contexts, Top-K competition, and decoder relationships introduce dependence; the nominal p-values accompany the observed distributions and effect sizes within this fit.

Table 18: Between-group score comparisons. Two-sided Welch tests with Benjamini–Hochberg adjustment across all six comparisons.

Metric	Groups	Welch t	BH-adjusted p
Detection	Deep / Middle	-2.553	0.0167
Detection	Deep / Shallow	1.068	0.3146
Detection	Middle / Shallow	3.869	2.789×10^{-4}
Fuzzing	Deep / Middle	-4.250	8.166×10^{-5}
Fuzzing	Deep / Shallow	-6.327	4.486×10^{-9}
Fuzzing	Middle / Shallow	-1.007	0.3146

The original aggregate scoring summary reports marginal activation-frequency correlations of $r = -0.1443$ for Detection and $r = -0.1894$ for Fuzzing. The record-level analysis in Appendix R joins the 921 unique scored modules by FM ID to the default AF registry and estimates group contrasts in that joined population.

R Activation-frequency adjustment of description scores

The middle-group Detection advantage and lower deep-group Fuzzing score in Section 3.4 remain after adjustment for activation frequency (AF; Table 19). In the 921 uniquely scored modules, median AF is .0225/.0341/.0374 for shallow/middle/deep groups. AF itself has Pearson correlation $-.151$ with Detection and $-.218$ with Fuzzing. Each score record joins uniquely by FM ID to the default Pythia-410M AF registry, and all joined modules have positive AF.

For each score y_m, we fit an additive least-squares model

$$y_m = \beta_0 + \beta_M \mathbf{1}[g_m = \text{Middle}] + \beta_D \mathbf{1}[g_m = \text{Deep}] + \gamma u_m + \epsilon_m,$$
$$u_m = \log_{10} \text{AF}_m - \text{median}_j \log_{10} \text{AF}_j.$$

A nonlinear sensitivity adds u_m^2 and u_m^3, retaining the same group indicators and sample. The median log-AF is -1.5545; the common observed AF range of the three groups is .00699–.11539. The additive specification estimates constant group contrasts across AF.

Middle and shallow Fuzzing are close. The linear models explain .056 and .064 of score variance for Detection and Fuzzing; cubic terms yield .060 and .076. The contrasts summarize associations within one fitted dictionary, corpus, and automated evaluator, with substantial residual variation among modules.

The figure and score inventories share 918 modules, of which nine have different saved group labels (Appendix F). Holding these 918 records fixed and substituting the original figure labels changes each linear adjusted contrast by at most .0029. Detection differences for middle minus shallow and deep minus shallow change from .0462/.0263 to .0469/.0291; the Fuzzing deep-minus-shallow difference changes from $-.0298$ to $-.0270$. This comparison measures sensitivity to the saved labeling difference within the common population.

Table 19: Score differences adjusted for activation frequency. Entries are descriptive contrasts within the fixed 921-module dictionary; the nonlinear model uses cubic log-AF terms.

Score	Contrast	Linear log-AF	Cubic log-AF
Detection	Middle − Shallow	.0463	.0470
Detection	Deep − Shallow	.0261	.0272
Detection	Deep − Middle	−.0202	−.0198
Fuzzing	Middle − Shallow	.0010	.0028
Fuzzing	Deep − Shallow	−.0299	−.0279
Fuzzing	Deep − Middle	−.0309	−.0307

S KEYWORD-BASED ANALYSIS OF INTERPRETATION TEXT

We characterize the vocabulary of generated descriptions using deterministic substring matching over ten topic keyword lists. The analysis uses the same 921 scored modules as Table 16; it reads only the Interpretation field, without using the example contexts or external corpora. Text is lowercased, with no lemmatization or synonym expansion. Each topic is counted at most once per module, and a description may match several topics. The keyword lists are:

- Syntax / discourse relations and function words: syntactic, discourse, grammatical, function word, preposition, conjunction, determiner, clause, subordinator
- Punctuation / symbols / quotation markers: punctuation, symbol, quote, quotation, parenthe, bracket
- Delimiters / operators: delimiter, operator
- Identifiers / sub-token components: identifier, sub-token, variable
- Domain terms / technical vocabulary: technical term, domain-specific, specialized, scientific, programming, technical
- Entities / definitional phrases: entity, entities, named entit, proper noun, knowledg, defining phrase
- Numbers / measurements / time: numeric, number, year, date, percent, measurement
- Negation / removal / absence semantics: negation, negative, remove, removal, absence, without, prohibit
- Code / math / markup contexts: code, programming, mathemat, equation, latex, markup, html, xml
- References / structural labels: reference, citation, cross-reference, figure, table, section, label

For any FM (f) and topic (t), if the Interpretation text contains any keyword from the topic's keyword set (K_t) via substring matching, we define it as a hit:

$$\text{Hit}(f,t) = \mathbb{I}\left(\exists k \in K_t : k \subset \text{Interpretation}(f)\right).$$

The topic coverage for category c is defined as:

$$\text{Coverage}(c,t) = \frac{\sum_{f \in c} \text{Hit}(f,t)}{|c|}.$$

Table 20 reports coverage within each depth group. Deep-module descriptions more often contain syntax/discourse and punctuation terms; terminology and entity-related descriptions are more prevalent in middle and shallow groups. The overlapping topics characterize graded preferences in the vocabulary of generated explanations.

Coverage summarizes the vocabulary of the generated descriptions. Topic lists overlap, and substring matches include occurrences within longer words.

Table 20: Topic coverage in generated descriptions. Counts and percentages use the same 921 scored modules. Topics may overlap; matching is literal and case-insensitive.

Topic	Deep ($N = 203$)	Middle ($N = 169$)	Shallow ($N = 549$)
Syntactic/discourse/function words	162 (79.8%)	95 (56.2%)	263 (47.9%)
Punctuation/symbols/quotes	115 (56.7%)	75 (44.4%)	146 (26.6%)
Domain/technical terms	109 (53.7%)	105 (62.1%)	326 (59.4%)
Code/math/markup contexts	61 (30.0%)	64 (37.9%)	152 (27.7%)
Delimiters/operators	40 (19.7%)	41 (24.3%)	75 (13.7%)
Identifiers/subtokens/variables	28 (13.8%)	26 (15.4%)	67 (12.2%)
Numbers/measurements/time	25 (12.3%)	24 (14.2%)	71 (12.9%)
Entities/definitional phrases	23 (11.3%)	39 (23.1%)	124 (22.6%)
References/structural labels	15 (7.4%)	22 (13.0%)	59 (10.7%)
Negation/removal/absence	4 (2.0%)	2 (1.2%)	21 (3.8%)

T Invocation of one dictionary across Pile domains

The within-Pile analysis keeps the trained dictionary fixed and partitions evaluation tokens into 19 source categories. For domain d with N_d tokens, module m has activation frequency

$$a_{d,m} = \frac{1}{N_d} \sum_{t \in d} \mathbb{I}[c_m(t) > 0].$$

Each domain is thus represented by a vector of frequencies over the same modules; the denominator is the number of evaluation tokens in that domain. Correlating domain vectors within each depth group gives mean off-diagonal correlations of 0.74 for deep, 0.48 for middle, and 0.51 for shallow modules (Figure 7). These comparisons measure invocation and preference within one fitted inventory. Independent retraining on more distinct domains is analyzed in Section U.

U Independent-domain transfer and correspondence

Four dictionaries with the same budget ($M = 1024$, $K = 32$) are trained independently on Pile-10K, CodeSearchNet, GSM8K, and Chinese Wikipedia. Each is evaluated on a 30,000-token panel from each domain, giving 16 fit–evaluation combinations (Table 1). The table reports raw-activation explained variance: off-diagonal values range from 0.7286 to 0.9425, and the own-domain dictionary has the highest value in each evaluation column. This is a reconstruction transfer measurement. The standardized-space explained variance used to select cross-scale dictionary widths in Section AA measures a different representation space.

Response correspondence compares modules from two independently fitted dictionaries by their token-level response vectors. A pair is reciprocal when the module from fit A selects the module from fit B as its highest-response-similarity neighbor and that module selects the original module in the reverse direction. The cross-domain reciprocal-match proportion is 21–38%. The ratio of a cross-domain proportion to its same-distribution repeated-fit reference is 0.38–0.79. The first range describes matches across domains; this ratio expresses their magnitude relative to the refit reference. These comparisons describe partial response correspondence across the reported fits. The diagonal reconstruction advantage describes adaptation to the fitting domain, while reciprocal matching measures recurrence of token-response patterns.

V Additional distribution and task-input panels

The following panels extend the invocation analysis to additional input distributions using a fixed dictionary. With one fixed dictionary, evaluation on BoolQ passages and WikiText2 gives active-set Jaccard overlap of 0.999–1.000 and AF-vector correlations of 0.865–0.894 with in-domain usage over

the 25%–100% training-data settings. Full-data AF correlations are 0.8938 for BoolQ and 0.8709 for WikiText2. The fixed decoder weights retain their depth profiles across input panels (template similarity 1.0). The response measurements describe how those modules are invoked by different inputs.

A small task-input panel compares usage vectors with the global pool. Table 21 reports the correlations, Jensen–Shannon divergences, and depth composition among the 20 most enriched modules. The task panels differ in which parts of the shared population are enriched while maintaining high overall AF-vector correlation.

Table 21: Task-input invocation panel. Last three columns are proportions among the 20 most enriched modules, not model performance.

Input panel	AF correlation	JS divergence	Shallow	Middle	Deep
BoolQ	0.956	0.0105	0.10	0.30	0.60
COPA	0.898	0.0373	0.25	0.25	0.50
WikiText continuation	0.936	0.0147	0.00	0.20	0.80

W CHECKPOINT SAMPLING FOR TRAINING-TRAJECTORY ANALYSIS

Pythia provides dense early checkpoints and wider spacing later in training [12]. We fit a new dictionary at each selected Pythia-410M checkpoint and compare active modules on the evaluation data. "Training step" is the GPT-NeoX optimizer-step index used in checkpoint names (e.g., `.../step143000`). The 35 selected steps cover 0–143000 with dense early and regular later sampling:

- **Stage I (very early, log-spaced):** $\{0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$.
- **Stage II (early-to-mid transition):** $\{1000, 2000, \ldots, 9000, 10000\}$ (every 1000).
- **Stage III (mid-to-late):** $\{20000, 30000, \ldots, 110000, 120000, 130000, 140000, 143000\}$ (every 10000 plus final).

Table 22: Selected training steps for Section 4.2 (35 checkpoints).

Selected training steps				
0	1	2	4	8
16	32	64	128	256
512	1000	2000	3000	4000
5000	6000	7000	8000	9000
10000	20000	30000	40000	50000
60000	70000	80000	90000	100000
110000	120000	130000	140000	143000

Because a separate dictionary is fitted at each of the 35 checkpoints, module IDs do not identify corresponding objects across checkpoints. For a reference checkpoint A and query checkpoint B, let S_{ij} denote either Pearson response correlation on common evaluation tokens or Jaccard overlap of upper-tail salient-coordinate sets. The reported directional summary is

$$\mathrm{Align}(A \leftarrow B) = \frac{1}{|\mathcal{M}_B|} \sum_{j \in \mathcal{M}_B} \max_{i \in \mathcal{M}_A} S_{ij},$$

where $\mathcal{M}_A$ and $\mathcal{M}_B$ contain test-active modules. Each query module contributes one best reference similarity, so the denominator is the number of active query modules. Multiple queries may choose the same reference, and the response and salient-coordinate statistics may choose different references for the same query. Within a checkpoint, best-match summaries omit self-comparisons. Response alignment rises rapidly early in training; salient-coordinate overlap increases over a longer interval (Figure 9). Each curve fixes a reference population and varies the query checkpoint. Its values depend

on the fitted response and coordinate distributions; comparisons across references additionally change the active candidate inventory. The trajectories therefore concern separately maximized relations among independently fitted populations.

The activation-frequency average across active modules shares a budget with their count. If M_{active} modules have positive response on at least one evaluation token, then

$$\overline{\text{AF}}_{\text{active}} = \frac{1}{M_{\text{active}}} \sum_{m \in \mathcal{M}_{\text{active}}} \text{AF}_m = \frac{\mathbb{E}_t \|\mathbf{c}(t)\|_0}{M_{\text{active}}} \leq \frac{K}{M_{\text{active}}}.$$

This identity explains why mean frequency rises when the occupied module count falls under a fixed Top-K budget; the realized number of positive entries can be below K.

X CHECKPOINT MATCHING AND CANDIDATE-COUNT CONTROLS

The convergence toward final-checkpoint responses in Section 4.2 persists under reciprocal matching, one-to-one assignment, and equal candidate counts (Table 23). Early queries often select the same final module, whereas late queries occupy a broader set of partners. Cached similarity matrices pair final step 143,000 with each of 35 query checkpoints. The reference contains 894 modules with positive sparse-code activation frequency; query inventories contain the active modules at each checkpoint. Pearson similarity uses pre-TopK projections, and Jaccard uses positive-tail coordinate sets.

Matching rules and denominators. Directional matching averages column-wise maxima. A reciprocal pair consists of a column and row that select each other as maxima; its query fraction is the number of such pairs divided by the query count. Rectangular Hungarian assignment maximizes summed similarity subject to one use per row and column. Its mean divides by the number of assigned pairs, $\min(n_{\text{ref}}, n_{\text{query}})$, and its query coverage divides that number by n_{query}. When the query inventory is larger, this assigned-subset mean can exceed the all-query directional mean because the denominators refer to different populations.

For the candidate-count calculation, we draw 627 modules per side, the smallest active inventory across checkpoints. Each of 100 repeats uses one uniformly drawn reference subset shared across query steps and both similarities; query subsets are independently drawn at each nonreference step (seed 20260924). Quantiles across draws measure inventory sensitivity. The self-comparison at step 143,000 uses identical indices and yields identity matches; temporal summaries end at step 140,000.

Table 23: Final-reference checkpoint controls. Best-match response and reciprocal fraction use full inventories. The last two columns average one-to-one assignment over 100 draws of 627 candidates per side. Reciprocal values are fractions of queries, while the other columns are mean similarities.

Query step	Query count	Response best	Reciprocal	Matched Pearson	Matched Jaccard
0	1003	.171	.167	.093	.051
2,000	935	.472	.673	.365	.123
50,000	991	.754	.811	.571	.265
140,000	911	.829	.866	.633	.335

At step 0, 1,003 queries select 241 distinct reference rows, with as many as 22 queries selecting one row. At step 140,000, 911 queries select 802 rows, with at most four queries per row. The corresponding excess-assignment fractions, $1 - n_{\text{distinct}}/n_{\text{query}}$, decline from 76.0% to 12.0%. Response and Jaccard assignments are solved separately. Each point also reflects variation among independently fitted dictionaries.

Depth-profile correspondence in a secondary inventory. We additionally compare layer-energy profiles at steps 64, 256, 2,000, 5,000, 20,000, 50,000, 120,000, and 143,000. This analysis uses frozen module IDs from an existing signature cache with at least 200 supporting activations, giving query counts of 623/633/933/952/1002/990/878/892, respectively. This eligibility rule defines a separate population from the 35-checkpoint matrices above. The final reference has 892 modules. At step 64, best-match and Hungarian energy cosines are .997 and .971; reciprocal matches cover 5.3% of queries. Positive-coordinate Jaccard is .071 and .065 under the two assignments. At step

120,000, energy cosine is .997/.996, reciprocal coverage is 60.1%, and Jaccard is .455/.450. Broad depth distributions recur early by energy cosine; reciprocal matches and coordinate overlap increase across the reported steps.

Y RECONSTRUCTION QUALITY UNDER A FIXED DICTIONARY BUDGET

We evaluate reconstruction quality for representative Pythia scales from the ten-size fixed-$M = 1024$, $K = 32$ comparison. These measurements characterize the changing relative capacity of that budget as model size grows.

Table 24: Reconstruction metrics under the default fixed-capacity setting. Reconstruction quality degrades with model size when latent capacity is held fixed.

Model	Explained variance	Normalized MSE	Pearson
Pythia-70M ($M = 1024, K_{\mathrm{SAE}} = 32$)	0.5615	0.4385	0.7503
Pythia-410M ($M = 1024, K_{\mathrm{SAE}} = 32$)	0.4932	0.5068	0.7026
Pythia-1.4B ($M = 1024, K_{\mathrm{SAE}} = 32$)	0.3937	0.6063	0.6275

The decline in explained variance, together with the corresponding rise in normalized MSE, shows that a fixed $M = 1024$ budget represents progressively less of the activation variance at larger scales. The standardized-space width comparison in Section AA therefore provides the reconstruction-controlled basis for the cross-scale correspondence analysis in the main text.

Z HELD-OUT RECONSTRUCTION OF THE LARGER DICTIONARIES

Section 4.3 compares module populations across Pythia scales. On 16 held-out documents, the existing 6.9B and 12B fixed-width dictionaries reconstruct .2825 and .2446 of standardized variation, respectively; inverse-transformed raw EV is .8814 and .8904 (Table 25). The different ordering in raw and standardized space reflects their different reconstruction targets. Both dictionaries have width $M = 1024$ and sparsity $K = 32$.

The 16 documents are sampled without replacement from the original 2,000-document test partition with NumPy's default random generator (seed 20260924). Their raw Pile-10K row indices are 8083, 8087, 8130, 8138, 8310, 8532, 8544, 8736, 9006, 9035, 9065, 9251, 9290, 9321, 9432, and 9861. Text is tokenized with the corresponding local Pythia tokenizer and truncated at 2,048 tokens, yielding the same 11,360 tokens for each model.

Extraction and reconstruction. Language-model inference and dictionary operations use float32. The extracted slices reproduce `hidden_states[1:]`: the first $L - 1$ block outputs and the final normalized output. Each document is standardized per coordinate as $z = (h - \mu)/(\sigma + 10^{-6})$, using the population standard deviation. The existing normalized dictionary produces the TopK–ReLU code and standardized reconstruction $\hat{z}$. The reconstructed raw activation is $\hat{h} = \hat{z}(\sigma + 10^{-6}) + \mu$, using the same document's recorded statistics. Reconstruction is processed in 64-token chunks; sums are accumulated in float64 on CPU.

For either space, we pool all scalar elements from the 16 documents. With N elements, the reported quantities are

$$\mathrm{SSE} = \sum_{i=1}^{N} (x_i - \hat{x}_i)^2, \qquad \mathrm{EV} = 1 - \frac{\mathrm{SSE}}{\sum_i x_i^2 - (\sum_i x_i)^2/N}, \qquad \mathrm{NMSE} = \frac{\mathrm{SSE}}{\sum_i x_i^2}.$$

The sums and the denominators for EV and NMSE are recomputed in each space. Raw EV includes document means restored by the inverse transform, so standardized and raw EV quantify different reconstruction targets.

Per-document sums, token counts, model and dictionary identifiers, numerical settings, and source hashes accompany the results. The bootstrap varies sampled documents while holding both fitted dictionaries fixed.

Table 25: Large-model reconstruction on 16 fixed held-out documents. Intervals are the 2.5th and 97.5th percentiles of 2,000 document-bootstrap replicates (seed 20260924). Each replicate recomputes pooled SSE/SST from sampled documents.

Model	Standardized EV	Standardized NMSE	Raw EV	Raw NMSE
6.9B	.2825 [.2581, .3009]	.7175	.8814 [.8687, .8966]	.1186
12B	.2446 [.2217, .2634]	.7554	.8904 [.8801, .9031]	.1096

AA STANDARDIZED-SPACE CAPACITY COMPARISON

The separate capacity comparison keeps $K = 32$ while varying M for three Pythia sizes: 70M, 410M, and 1.4B. It aligns held-out explained variance in the standardized training space: EV is 0.431/0.428/0.443 at $M = 256/512/4096$, respectively (Table 2). Raw-space EV for these fits is 0.988/0.845/0.856 and describes reconstruction in the original activation space. Width selection uses the standardized-space endpoint. The active counts are 256/511/4067. Thus occupied inventory changes with dictionary budget at approximately matched standardized-space reconstruction quality.

For cross-scale response comparison, a one-to-one Hungarian assignment matches modules by common-token response similarity. The three reported matched means span 0.785–0.824; each exceeds shuffled-response controls evaluated with the same matching procedure ($p < .001$). Unlike the directional best-match operator, this assignment uses each selected module at most once. With unequal dictionary widths, it summarizes the matched response subset rather than equal-size inventories. The nominal K/M budgets for the three fits are 12.50%, 6.25%, and 0.78%; the number of positive codes per token can be lower than K after ReLU. These budgets and the varying active counts couple average usage to dictionary capacity.

AB WITHIN-SCALE RESPONSE-CORRELATION MATRICES

We provide the raw within-model response-correlation structure used in Figure 10c. For each model scale, we compute Pearson correlations between token-level response vectors of active modules, set the diagonal to NaN, and show a 5×5 block-max pooled version for readability.

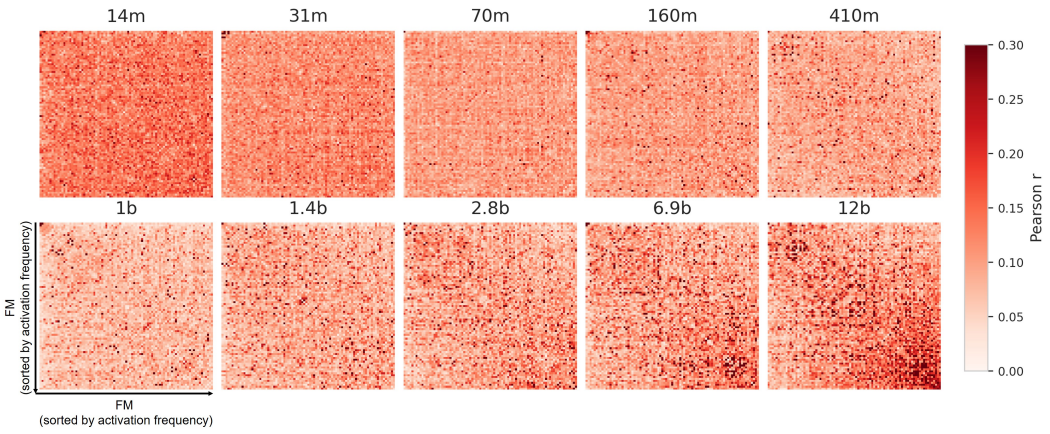

Figure 13: Within-scale response correlation (Pearson r). Each subplot is one model scale; entries are correlations between active modules' token-level response vectors (diagonal omitted). We apply 5×5 block-max pooling (ignoring NaNs) and use a shared color range $[0, 0.3]$.

AC WITHIN-SCALE SALIENT-COORDINATE OVERLAP MATRICES

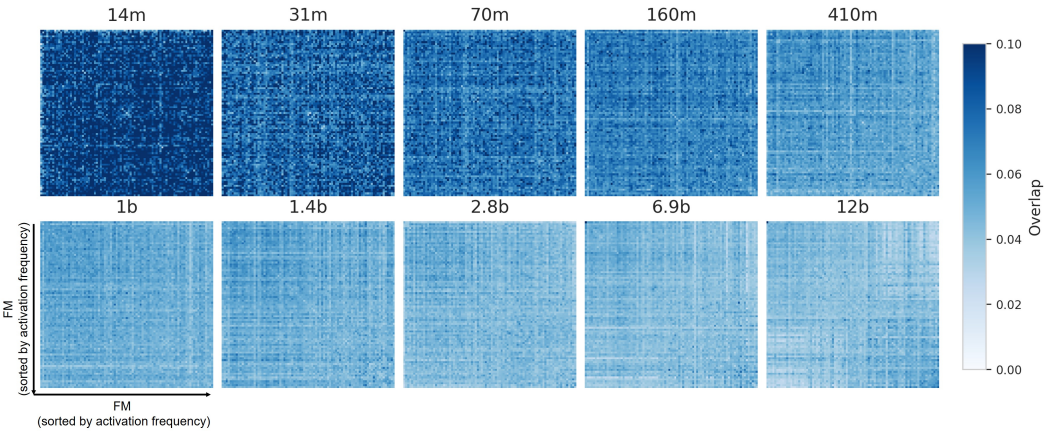

Figure 14: Within-scale salient-coordinate overlap (Jaccard). Each subplot is one model scale; entries are Jaccard overlaps between upper-tail salient-coordinate sets (diagonal omitted). We apply 5×5 block-max pooling (ignoring NaNs) and use a shared color range $[0, 0.1]$.

AD CROSS-SCALE RESPONSE-CORRELATION MATRICES

For reference model A and query model B, we compute cross-scale similarity

$$R_{i,j}^{A \leftarrow B} = \mathrm{corr}\left(s_i^A,\, s_j^B\right),$$

which is the matrix summarized by $\mathrm{Align}(A \leftarrow B)$ in the main text.

Each response vector contains the module's dense projection scores at common evaluation-token positions. With n_A reference modules and n_B query modules, the matrix has shape $n_A \times n_B$. A column describes one query's relationships to the entire reference population; its maximum identifies the closest response under this criterion. Averaging those maxima gives the directional summary. Exchanging the two inventories changes which axis supplies the maxima and which population supplies the average. The two summaries can therefore differ even when they derive from the same pairwise relationships.

Response correlation compares scalar sequences indexed by tokens, so it remains defined when models have different layer counts and hidden dimensions. A matched pair locates related response patterns in the two fitted populations. Each member retains its own full-stack decoder direction and depth profile. The within-model overlap matrices in Appendix AC complement this comparison by describing coordinate sharing in each model's residual basis. Figure 15 displays the response matrices with 5×5 block maxima, making strong local correspondences visible across the large pairwise grids.

AE SALIENT-SUPPORT INTERVENTION ANALYSIS

This analysis asks whether a module's upper-tail salient coordinates identify locations relevant to its measured response. For each of six selected modules, the evaluation uses 50 high-activation contexts and compares suppression of those coordinates with a matched random-coordinate control. The target module's mean activation drop is 9.603 under selected coordinates versus 2.655 under random coordinates; the corresponding medians are 9.820 and 3.105. The mean local next-token output-distribution change, measured by KL divergence, is 0.1327 versus 0.0984. Next-token log-probability and logit changes are numerically larger under selected coordinates but are not statistically significant in this six-module analysis. These results concern the selected modules and tested contexts.

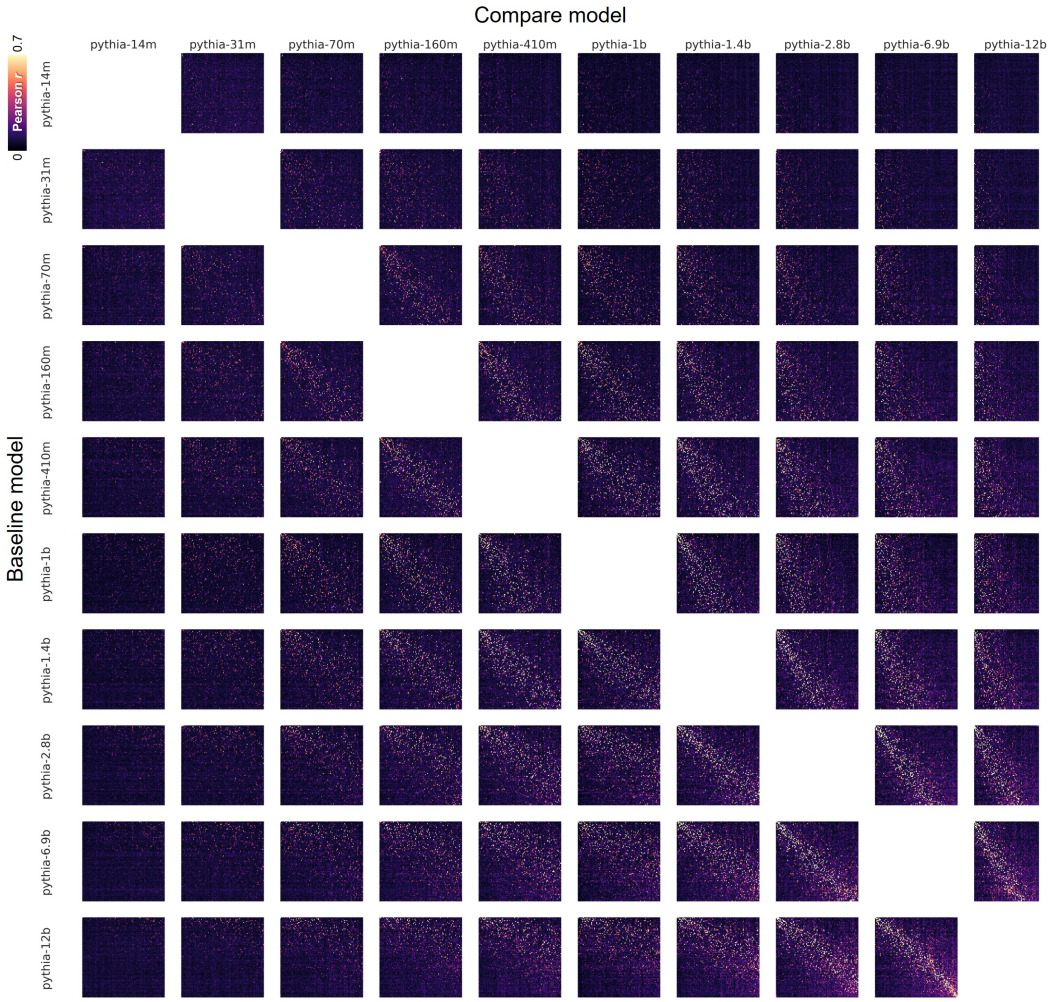

Figure 15: Cross-scale response correlation matrices (Pearson r). Rows are reference models and columns are query models; each off-diagonal subplot is $R^{A \leftarrow B}$. We apply 5×5 block-max pooling for visualization and use a shared color range. Diagonal subplots (same model) are omitted.

AF DOCUMENT-DISJOINT SELECTED-MODULE INTERVENTIONS

This separate analysis tests whether a selected module's output-change pattern recurs across documents. The study fixes 16 target modules and draws non-overlapping discovery and evaluation arms from Pile documents held out from SAE training. Each arm supplies 24 contexts from distinct documents per module. Intervention magnitude is matched by residual ℓ_2 norm at the actual hook tensor. For each target, a next-token-logit change signature is defined from discovery contexts alone and then assessed on evaluation contexts.

Fixed learned modules from the same depth provide an identity-specificity comparison. All 16 target signatures are more reproducible than these controls, with Holm-adjusted $p < .001$. Five additional random directions are matched in residual norm; the target effect is larger in 14 of 16 module-level comparisons against them. The same-depth comparison measures how distinctive the selected module's output signature is across documents, while the random directions calibrate effect magnitude. The evidence is for the selected modules and their local next-token output effects.

AG ROBUSTNESS OF CEBRA EMBEDDINGS

The CEBRA view embeds 24,576 Pythia-410M layer–coordinate responses to 32 text stimuli in three dimensions. Resampling and stimulus-count comparisons assess consistency of the resulting coordinates.

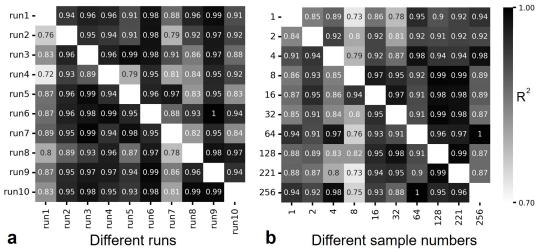

Figure 16: CEBRA embedding consistency (Pythia-410M). a. 10 runs with fixed stimulus count $N = 32$ but different random stimulus resamples; heatmap shows pairwise consistency (R^2). **b.** One run per stimulus count $N \in \{1, 2, 4, 8, 16, 32, 64, 128, 221, 256\}$; heatmap shows pairwise consistency (R^2).

For embeddings $E^{(a)}, E^{(b)} \in \mathbb{R}^{M_{AN} \times 3}$ with common residual layer–coordinate rows, fit the affine prediction $\widehat{E}^{(b)} = E^{(a)}\widehat{B} + \mathbf{1}\widehat{\mathbf{b}}^{\top}$ by ordinary least squares. Following CEBRA's released simulation code and consistency routine, the score is the in-sample coefficient of determination, averaged equally over the three target coordinates:

$$R^2(a \to b) = \frac{1}{3}\sum_{k=1}^{3}\left[1 - \frac{\sum_i (E_{ik}^{(b)} - \widehat{E}_{ik}^{(b)})^2}{\sum_i (E_{ik}^{(b)} - \overline{E}_{\cdot k}^{(b)})^2}\right],$$

for nonconstant coordinates. The regression includes an intercept. This ordered-pair score measures coordinate-level linear predictability between embeddings. Depth localization is measured separately from decoder profiles in Sections E, H and J.

AH VISUALIZATION WITH OTHER DIMENSIONALITY REDUCTION METHODS

t-SNE [28] and UMAP [29] provide alternative projections of the same residual layer–coordinate response data. The main module maps use CEBRA (Figure 2 and Section AG).

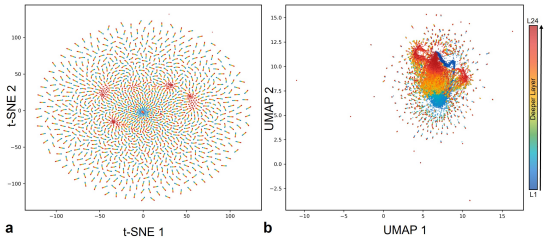

Figure 17: t-SNE and UMAP visualizations of ANs (Pythia-410M). Each point is a residual layer–coordinate pair (AN in the original figures); color indicates layer depth. **Left:** t-SNE (2D). **Right:** UMAP (2D).

AI COMPLETE CEBRA MODULE-MAP ATLAS

The following 15 panels highlight each module's upper-tail salient-coordinate set in the common CEBRA embedding of Figure 2. They display selected decoder coordinates across the fitted inventory; the quantitative profiles and controls appear in Sections E, H and J.

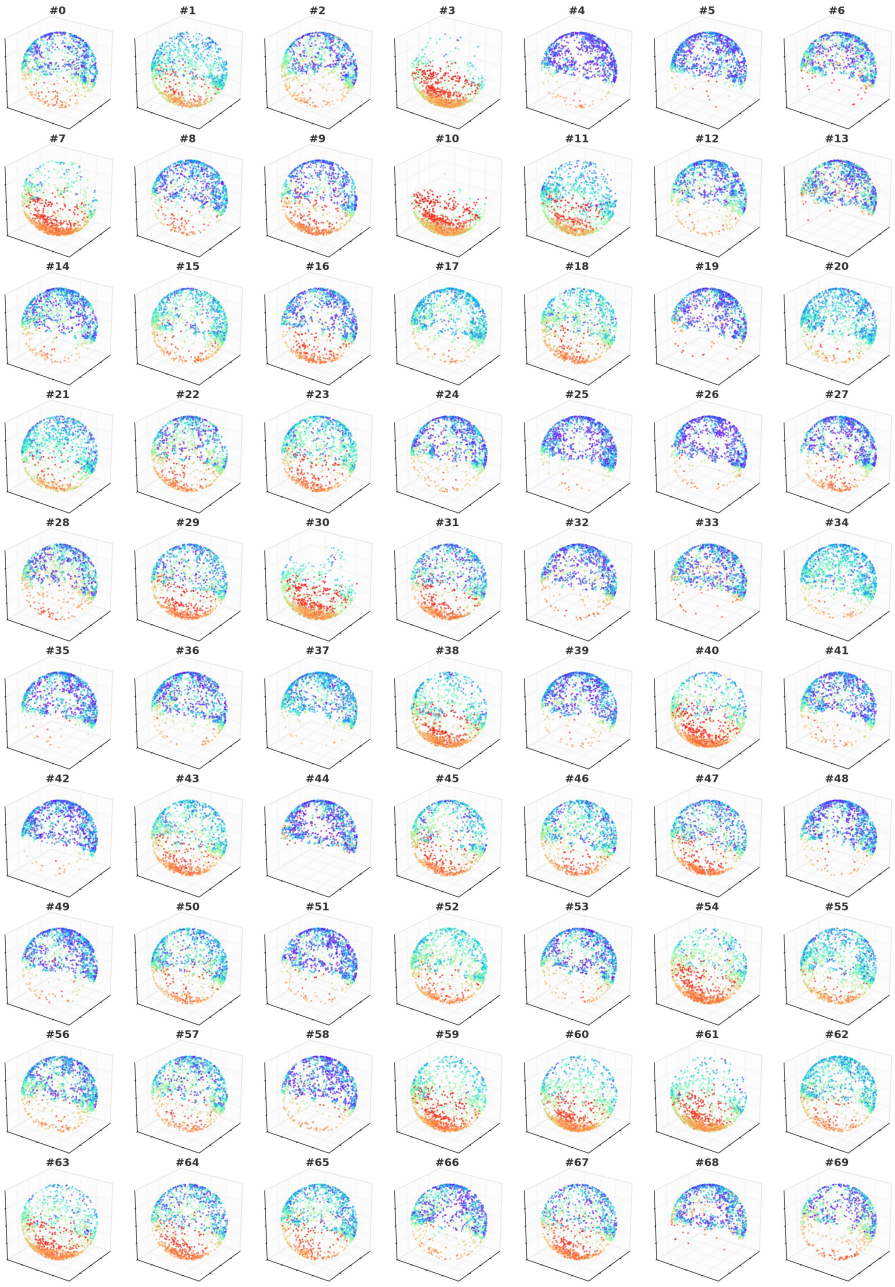

Figure 18: Module support maps (1/15). Highlighted points are upper-tail salient coordinates for the indicated module IDs in the common CEBRA embedding.

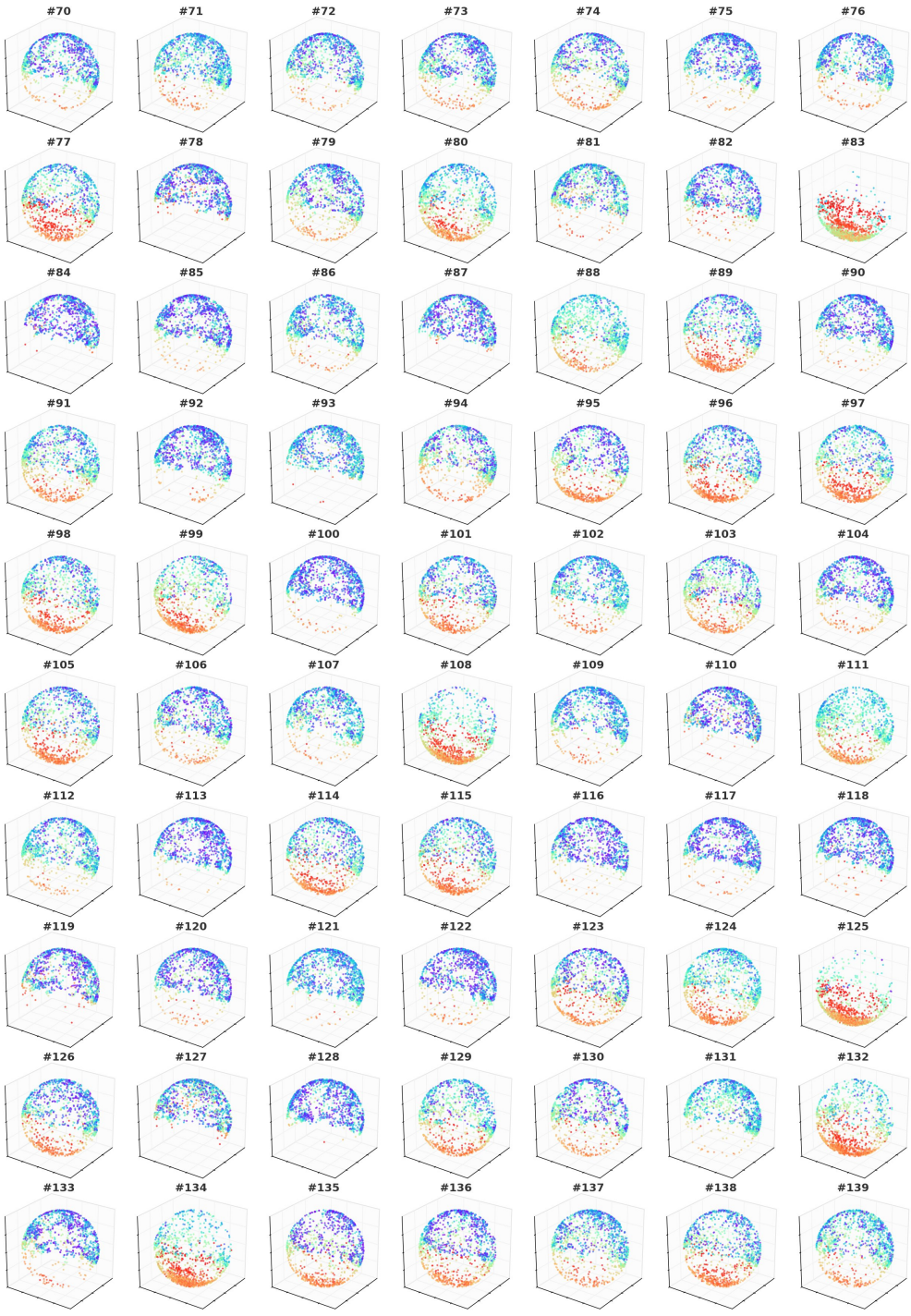

Figure 19: Module support maps (2/15).

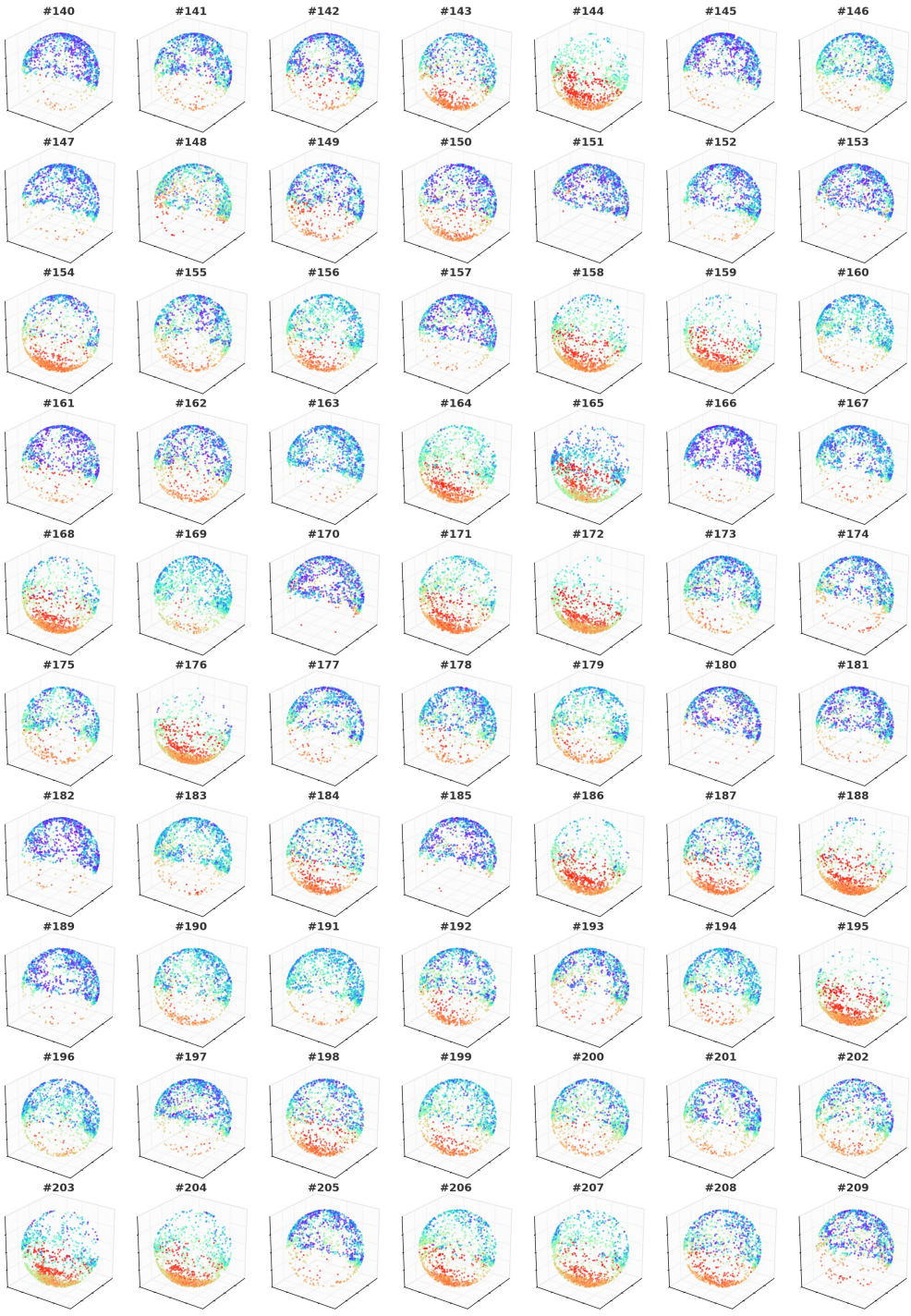

Figure 20: Module support maps (3/15).

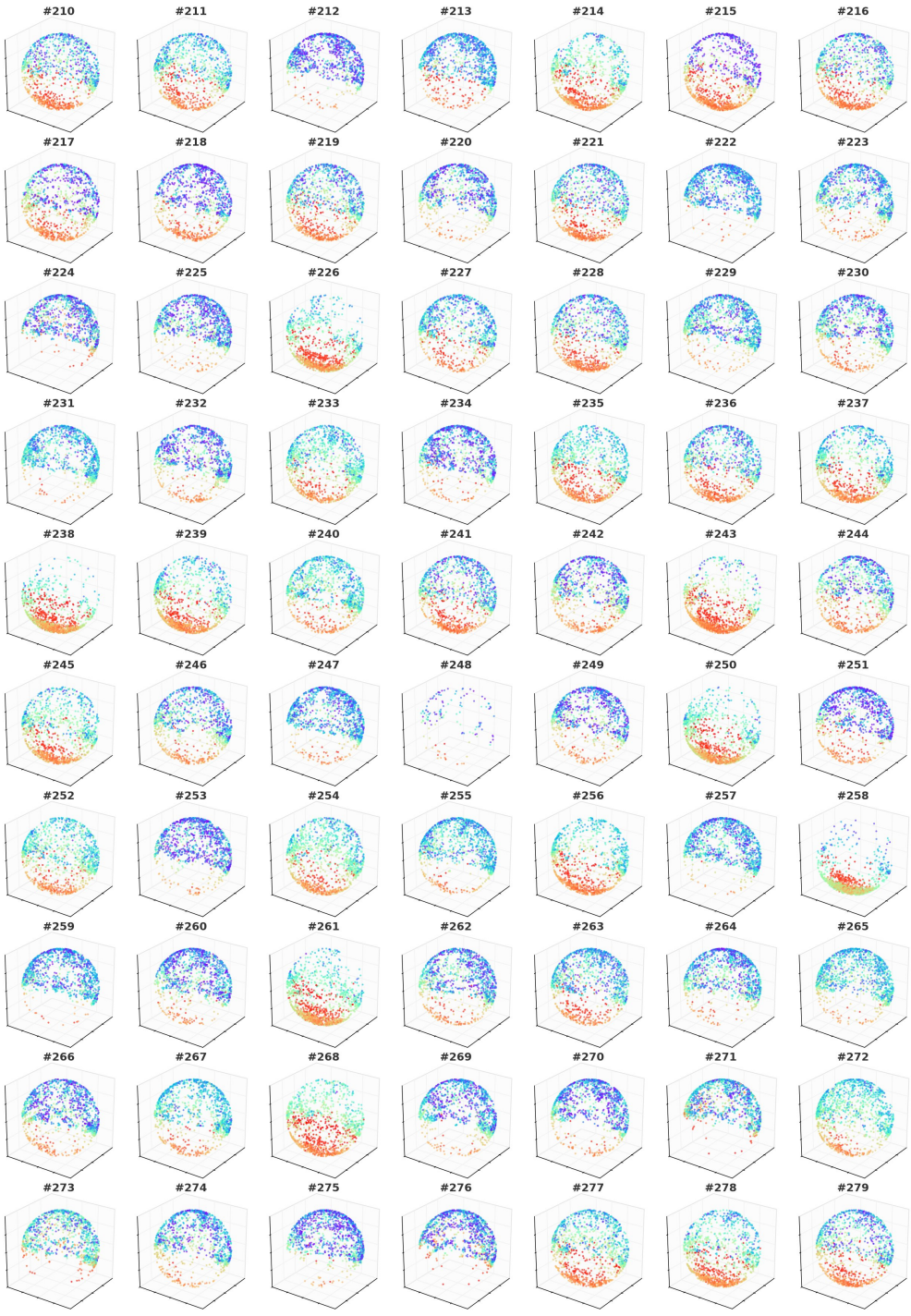

Figure 21: Module support maps (4/15).

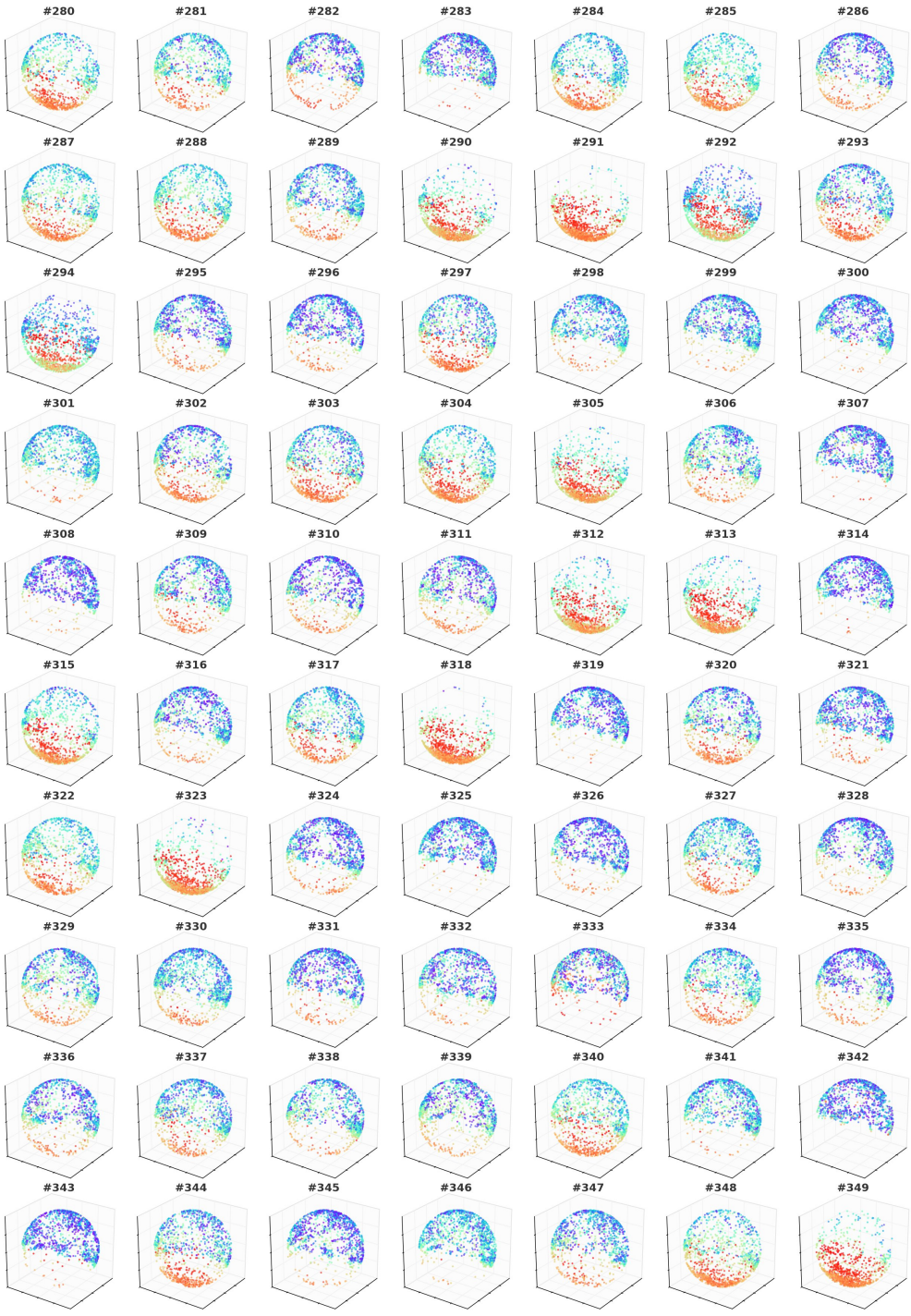

Figure 22: Module support maps (5/15).

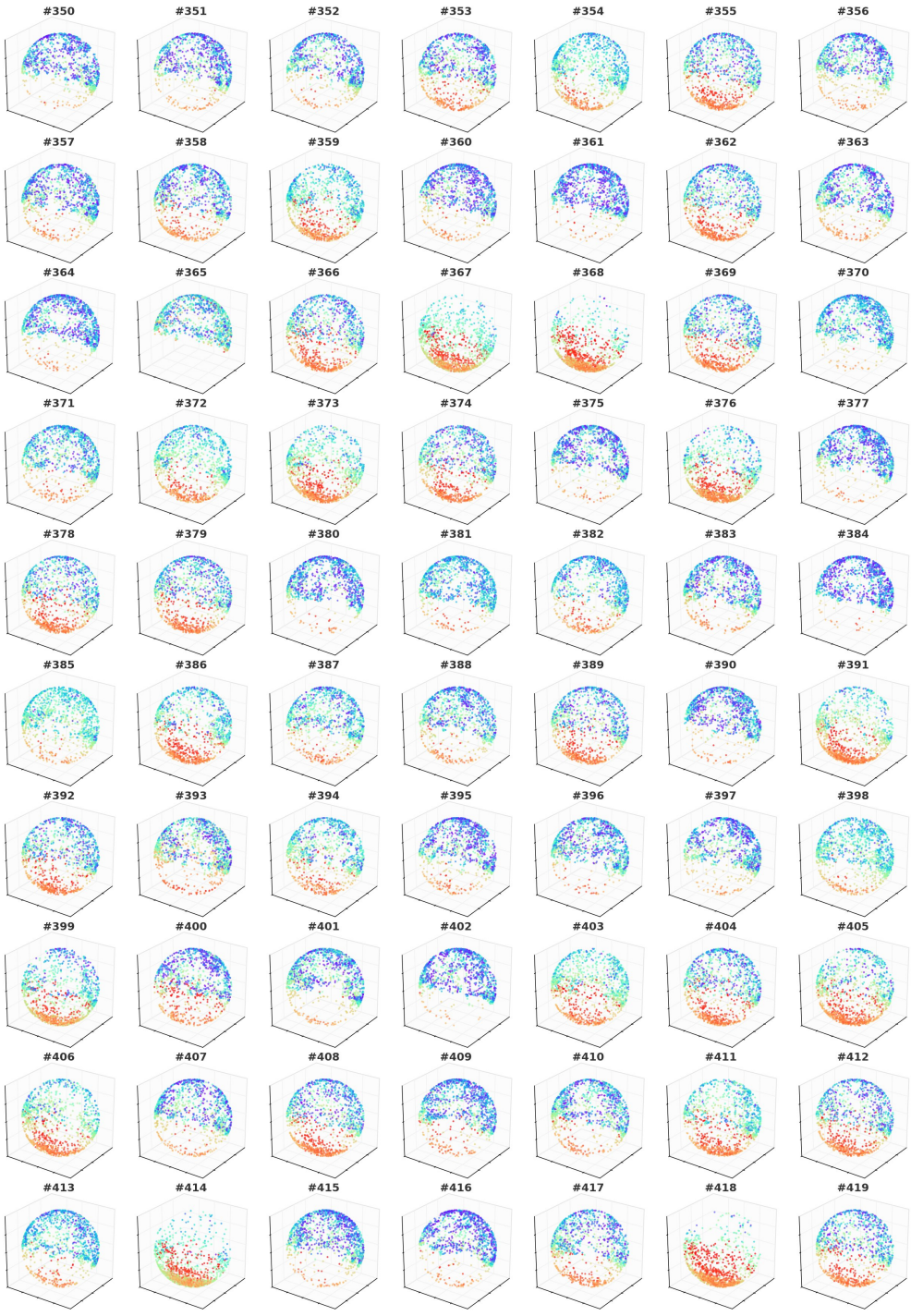

Figure 23: Module support maps (6/15).

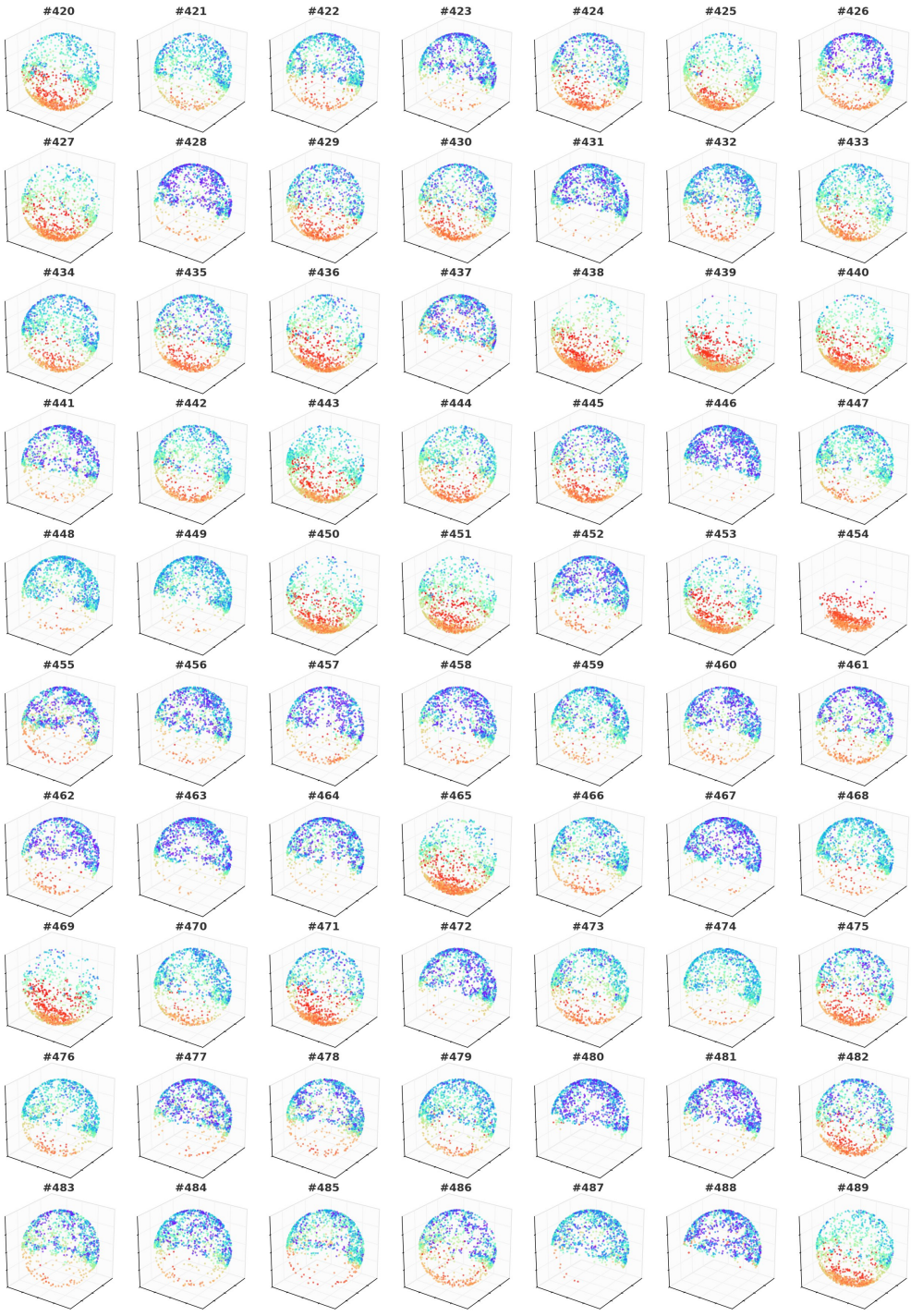

Figure 24: Module support maps (7/15).

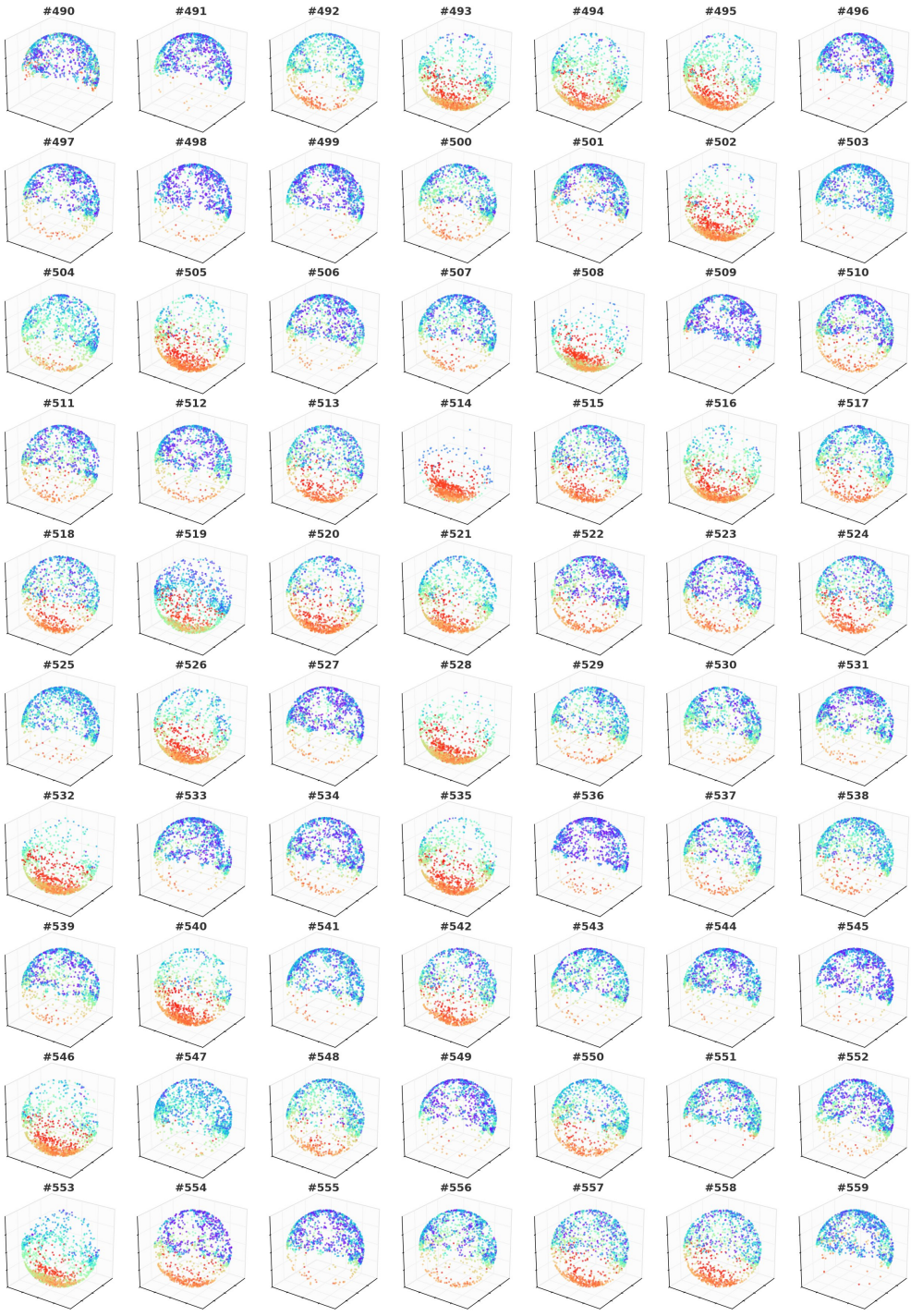

Figure 25: Module support maps (8/15).

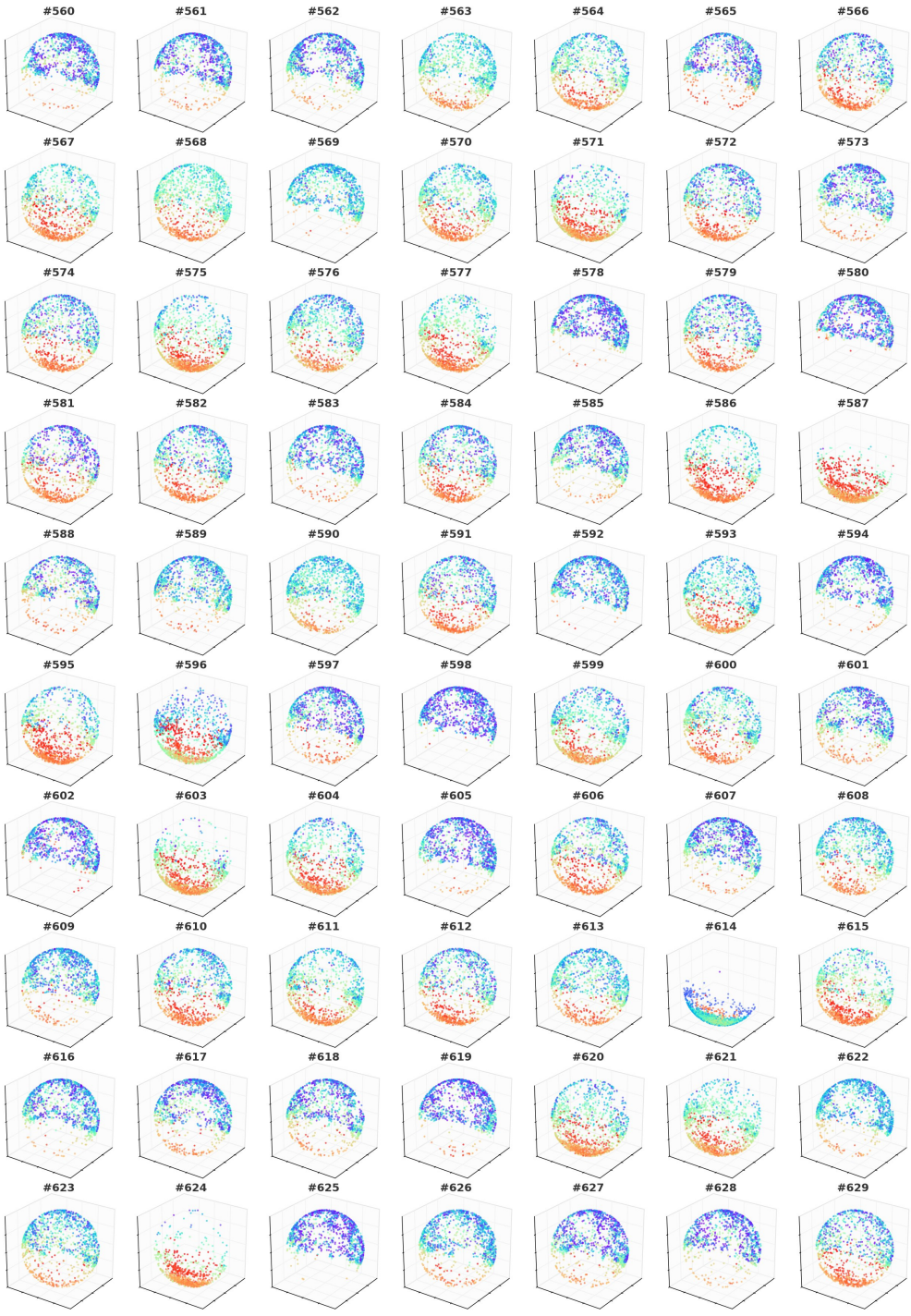

Figure 26: Module support maps (9/15).

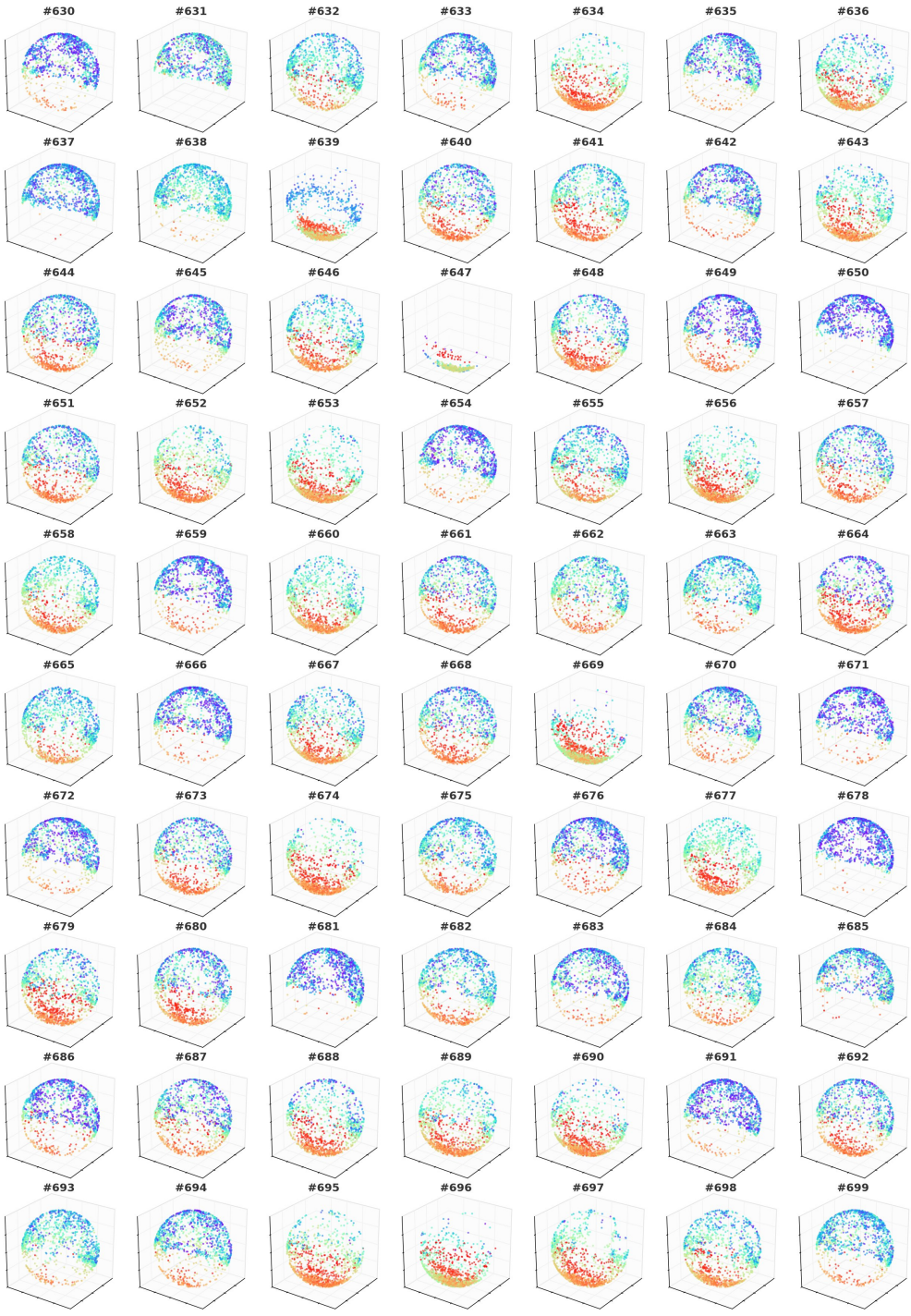

Figure 27: Module support maps (10/15).

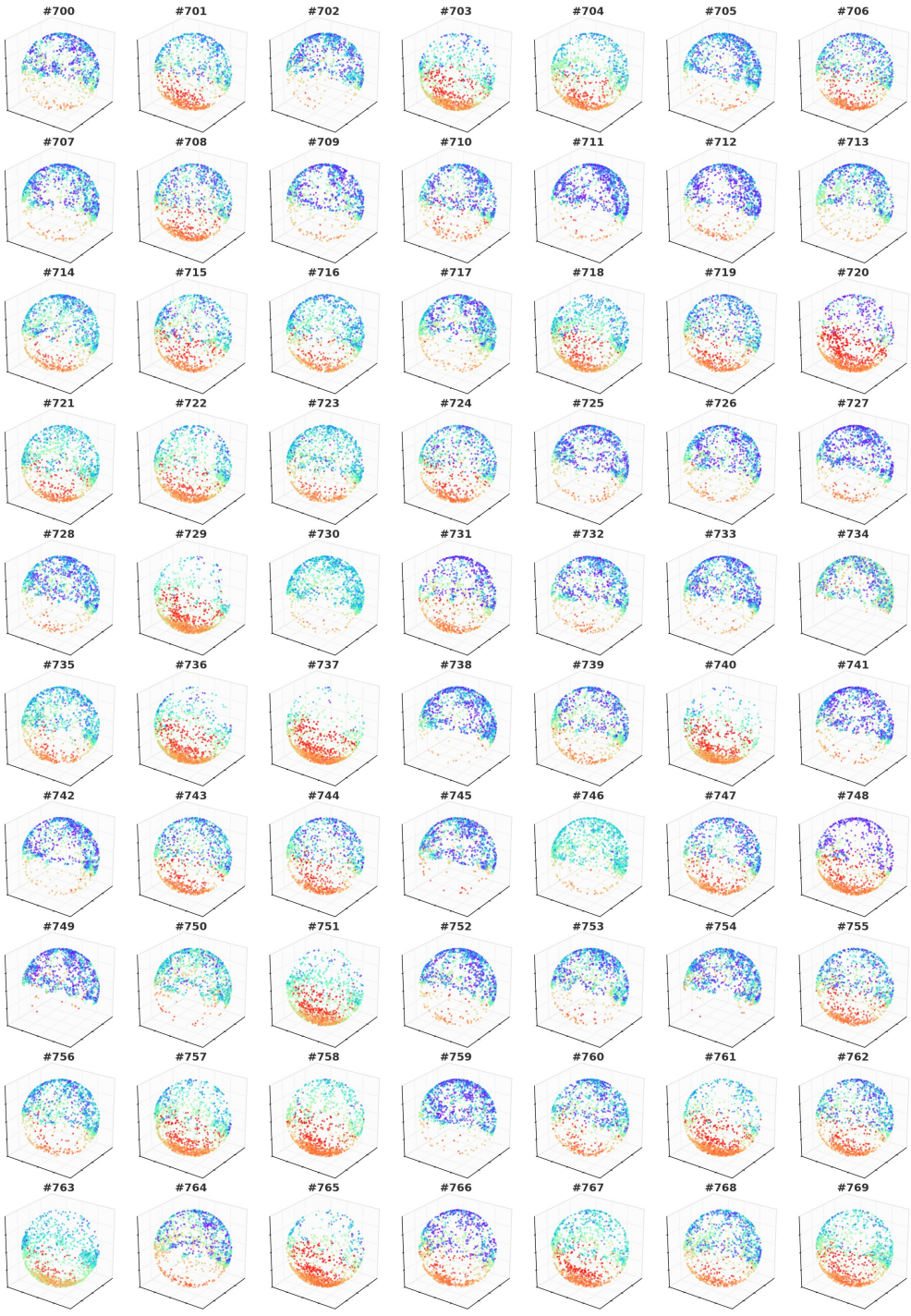

Figure 28: Module support maps (11/15).

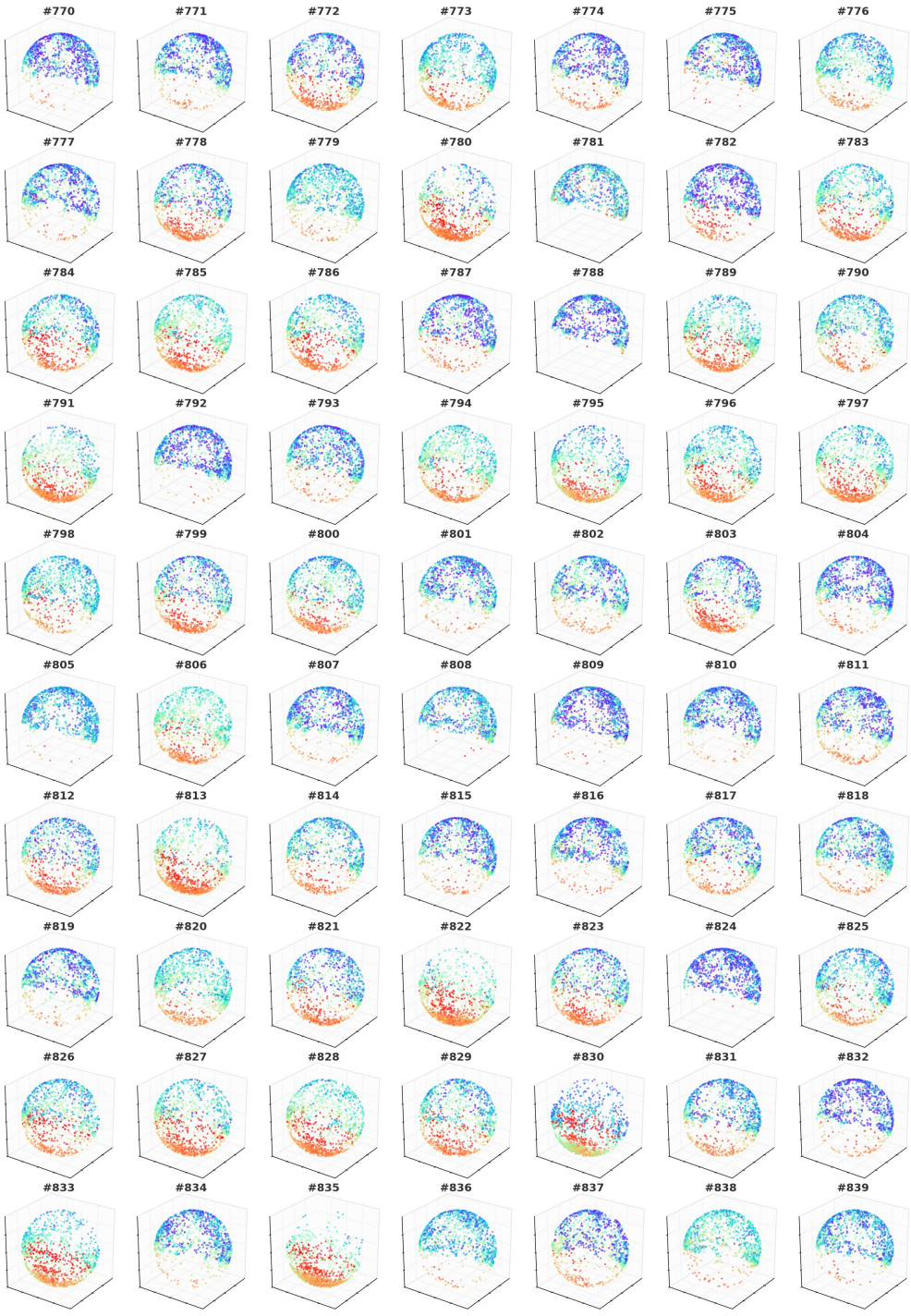

Figure 29: Module support maps (12/15).

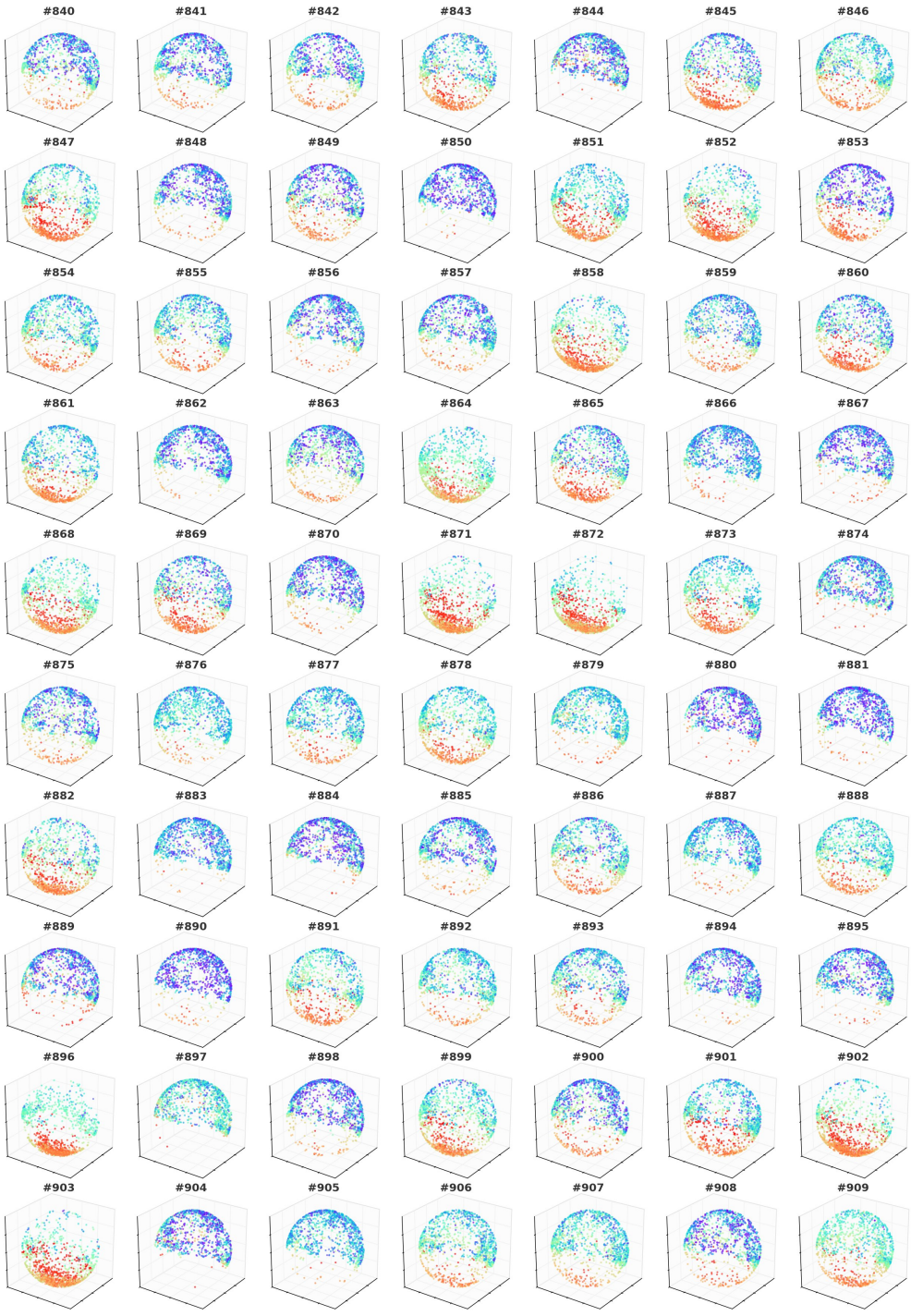

Figure 30: Module support maps (13/15).

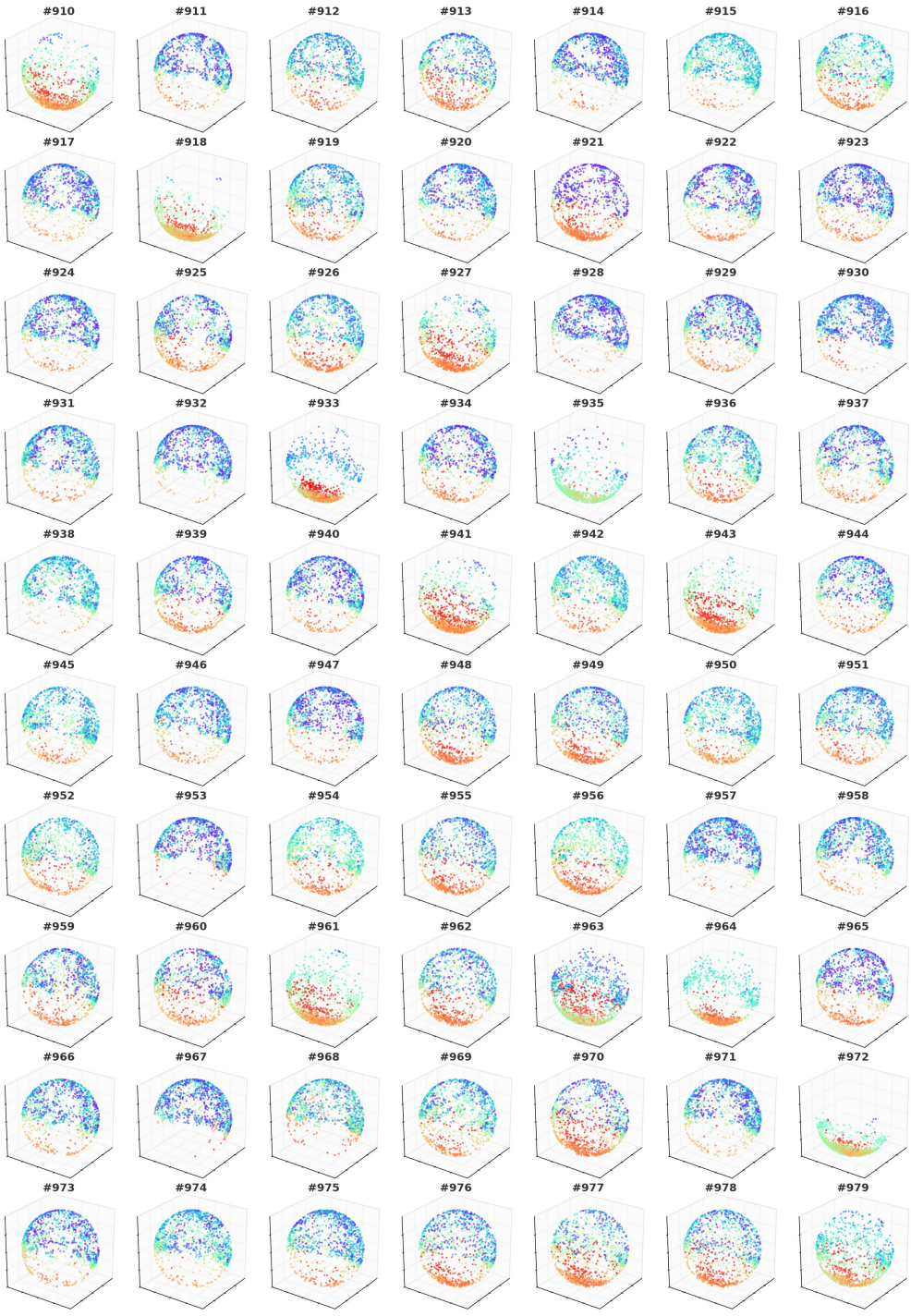

Figure 31: Module support maps (14/15).

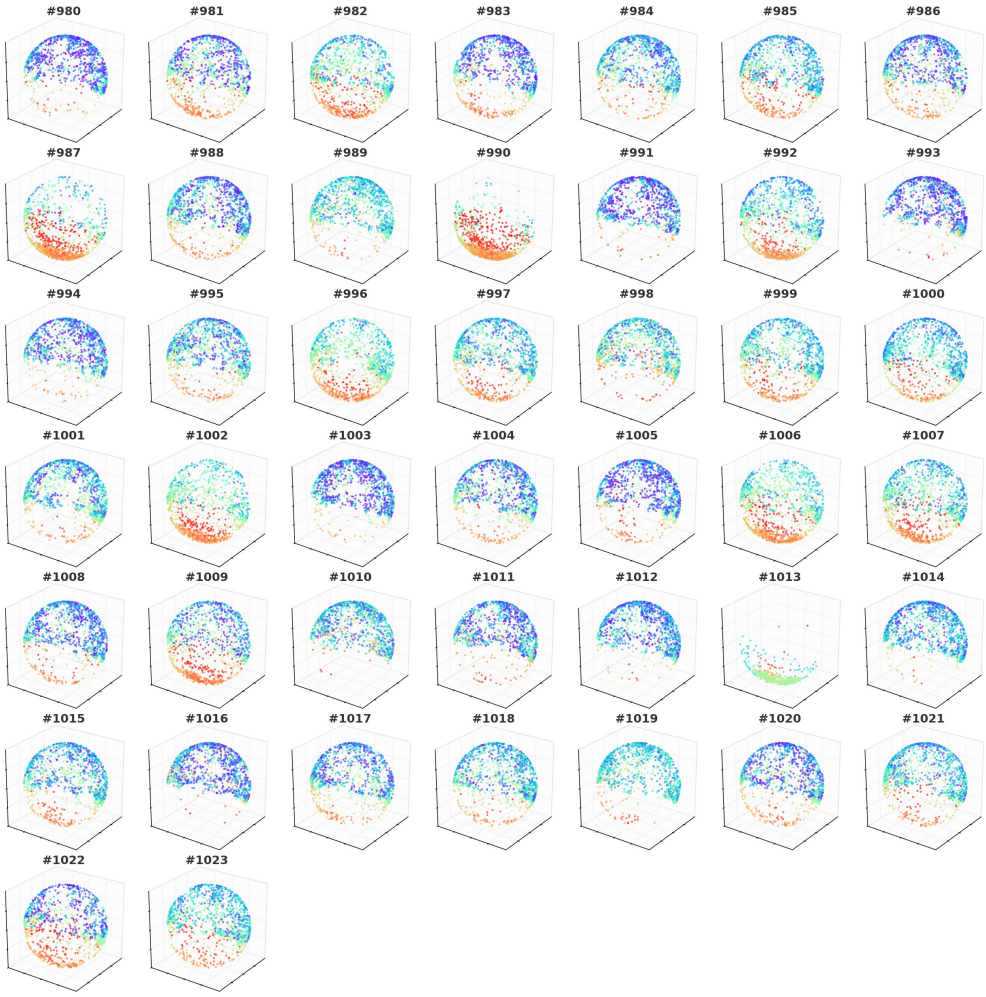

Figure 32: Module support maps (15/15).