Construct Emergence Tracing: From Meaning to Mechanism in Transformer Interpretability via Generative Psychometrics

Hudson Golino[1]

[1] University of Virginia

Author Note

Hudson Golino, Department of Psychology, University of Virginia. This work was supported by a grant from The Jefferson Trust. The Construct Emergence Tracing pipeline uses the HuggingFace Transformers library for activation extraction and the R packages EGAnet and aricode for network estimation and metric computation.

The authors made the following contributions. Hudson Golino: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft.

Correspondence concerning this article should be addressed to Hudson Golino, Department of Psychology, University of Virginia, Charlottesville, VA, USA. E-mail: hfg9s@virginia.edu

Abstract

How do language models organize a multidimensional psychological construct across their layers? Construct Emergence Tracing (CET) addresses this question by combining Dynamic Exploratory Graph Analysis with normalized mutual information (NMI), the Total Entropy Fit Index, and a compression-based network complexity measure. We evaluated fourteen models spanning nominal sizes of 117 million to 32 billion parameters, using five items from each of five attributes of grandiose narcissism in every condition. The simulations yielded 10,640,520 network estimates across model states, embedding dimensions, and repeated item samples. Construct recovery generally increased from early to intermediate states, but its onset, magnitude, and terminal persistence varied substantially. Mean NMI increased from 0.504 to 0.653 across Qwen2.5-Instruct sizes from 0.5B to 32B. Five models exhibited zero or near-zero final-state mean NMI: GPT-2, Phi-4-mini, Qwen2.5-32B, GPT-OSS-20B, and Muse-Glimmer-30B. The remaining nine retained final-state mean NMI between 0.504 and 0.777. Peak coherence, entropy fit, and composite maturity were not always attained at the same depth. These results identify distinct emergence and terminal-retention profiles while showing why parameter count and final-layer extraction alone provide incomplete descriptions of construct representation. CET offers a construct-centered complement to mechanistic interpretability; controlled interventions and matched checkpoint comparisons are needed to establish the causes of the observed trajectories.

Keywords: interpretability, transformer representations, network psychometrics, DynEGA, TEFI, Kolmogorov complexity, representational maturity, GPT-2, Phi-2, Phi-4-mini, semantic emergence

Construct Emergence Tracing: From Meaning to Mechanism in Transformer Interpretability via Generative Psychometrics

Introduction

Transformer-based language models, from GPT-2's 117 million parameters to models with hundreds of billions of parameters, have transformed natural language processing (Brown et al., 2020; Radford et al., 2019). Their capacity to generate sophisticated text, reason, and interpret context has increased, with advances in mathematical reasoning as well (Brown et al., 2020; Shao et al., 2024). However, how these models work remains an open question. If an LLM can generate text that expresses complex and abstract ideas and interpret similarly complex prompts, a question worth asking is: *how do these models represent and organize abstract semantic information inside their layers*? As transformers are used in consequential fields such as medicine, law, education, finance, and governance, interpretability becomes central to transparency, accountability, alignment, and safety (Bommasani et al., 2021).

The interpretability literature has developed an impressive set of methods in a short period of time. Feature attribution techniques such as *Shapley Additive Explanations* (SHAP) (Lundberg & Lee, 2017) and *Integrated Gradients* (Sundararajan, Taly, & Yan, 2017) assign importance scores to input tokens. Attention visualization and neuron-activation analyses, on the other hand, reveal patterns of attention and activation across contexts (Olah et al., 2020; Vig, 2019). Circuit tracing and attribution graphs (Lindsey, Gurnee, Ameisen, et al., 2025; Olah et al., 2020) uncover specific computational pathways and functional specialization in large language models. This work recalls early connectionist efforts to understand learned internal representations (Rumelhart, Hinton, & Williams, 1986). Causal (Feder et al., 2022) and counterfactual approaches (Wachter, Mittelstadt, & Russell, 2017) probe robustness and cause-effect structure, while representation similarity analyses, including *Centered Kernel Alignment* (CKA) (Kornblith,

Norouzi, Lee, & Hinton, 2019), *Singular Vector Canonical Correlation Analysis* (SVCCA) (Raghu, Gilmer, Yosinski, & Sohl-Dickstein, 2017), and *Representational Similarity Analysis* (RSA) (Kriegeskorte, Mur, & Bandettini, 2008), compare geometric properties of activation spaces. Probing classifiers (Alain & Bengio, 2017; Belinkov, 2022) assess whether specific linguistic features are linearly decodable from hidden states. These methods offer specific, and sometimes mechanistic, insights into individual computations of large language models, but they usually employ a *bottom-up* approach. In other words, they start from neurons, attention heads, or individual prompts and work toward a broader understanding of the models' internal states.

They address questions such as *how does the model compute X?* or *what information can be decoded from its hidden states?*, but do not directly establish *where and when a coherent abstract concept emerges across the layers of a large language model* (Belinkov, 2022; Lindsey et al., 2025).

Psychology has studied latent constructs for over a century (Cronbach & Meehl, 1955; Spearman, 1904). Constructs such as narcissism, empathy, or anxiety can encompass multiple dimensions rather than a single characteristic. Their observed indicators are expected to form a coherent structure (Cronbach & Meehl, 1955; Epskamp, Maris, Waldorp, & Borsboom, 2018). To quantify how such a structure emerges, a method must begin with the construct and assess how well the model's internal representations recover it at each processing stage.

This gap in interpretability research points to a possible new direction: psychological measurement science. Generative AI has begun to reshape research practices in psychometrics. Large language models are now used to generate item pools and create embeddings for dimensionality assessment before data collection (Garrido, Russell-Lasalandra, Rodríguez-Montoya, & Golino, 2026; Lara L. Russell-Lasalandra, Christensen, & Golino, 2024). Redundant and unstable items can also be identified and

removed automatically. These procedures have produced item pools whose structural organization agrees with that found in nationally representative samples (Garrido et al., 2026; Lara L. Russell-Lasalandra et al., 2024). These developments have given rise to what has been termed *Generative Psychometrics* (Lara L. Russell-Lasalandra et al., 2024), a research paradigm that integrates large language models with quantitative psychometric evaluation. Early work in this field focused on what LLMs can do *for* psychometrics: generate items, optimize prompts, and accelerate scale development (Garrido et al., 2026; Lara L. Russell-Lasalandra et al., 2024; Lara L. Russell-Lasalandra & Golino, 2026). However, Generative Psychometrics can also be used to map and understand when and where an abstract psychological construct emerges inside the layers of a large language model.

A key methodological advance was the discovery that LLM embedding spaces are searchable and represent abstract psychological constructs in a non-linear and non-uniform way (H. Golino, 2026). This finding made Generative Psychometrics a possible paradigm for interpretability research. H. Golino (2026) used Dynamic Exploratory Graph Analysis (DynEGA) (H. F. Golino, Christensen, Moulder, Kim, & Boker, 2022) to search embedding coordinates systematically, with the dimension index treated as a pseudo-temporal ordering. Dimensional structure was concentrated in specific regions of the embedding landscape rather than distributed uniformly across all coordinates. This *landscape search* approach optimizes jointly over structural accuracy, via Normalized Mutual Information (*NMI*) (Strehl & Ghosh, 2002), and entropy-based organization, via the Total Entropy Fit Index (*TEFI*) (H. F. Golino et al., 2021). It consistently outperformed standard cross-sectional analysis of full embedding vectors in the reported study (H. Golino, 2026). The same insight may extend beyond API-level embeddings to the internal representations of transformers: if the output embedding is a non-uniform and non-linear landscape, the hidden states at every layer of an LLM might also be.

The current paper introduces **Construct Emergence Tracing** (CET), a set of

interpretability tools for analysis across model depth. The embedding landscape search approach (H. Golino, 2026) applied DynEGA to final-layer embeddings from a single model. CET applies it across *every layer* of multiple architectures and combines structural accuracy (NMI), entropy-based organization (TEFI), and a compression-based proxy for Kolmogorov complexity (H. Golino, 2025; Kolmogorov, 1965). The resulting *Representational Maturity Index* tracks how construct structure emerges, matures, and sometimes collapses as representations propagate through the LLM layers.

CET starts from a *predefined psychological construct*: a set of textual items that represent behavioral, emotional, or cognitive attributes, with a known dimensional structure. It then traces how this structure is represented across every layer of a transformer. The core insight is that each layer's hidden states are searchable landscapes. DynEGA is applied to derivatives obtained from time-delay embeddings of the activation vectors, and community structures are estimated from the resulting networks (H. Golino, 2026; H. F. Golino, Christensen, Moulder, et al., 2022). CET quantifies three complementary properties of this structure at each layer: semantic coherence, via NMI against ground-truth labels; organizational structure, via TEFI; and representational efficiency, via a compression-based proxy for Kolmogorov complexity (H. Golino, 2025). These three indices are combined into a *Representational Maturity Index* that summarizes how well-formed the target construct's representation is at each layer and embedding configuration. **CET thus represents the interpretability direction of Generative Psychometrics**: psychology's measurement science is applied *to* generative AI. Model hidden states serve as the data, and layers serve as the developmental axis along which latent structure emerges.

CET makes three contributions. First, it formalizes construct representation as a measurable trajectory across model depth. Second, it connects this trajectory to network-psychometric indices. Third, it evaluates the framework across fourteen language models with a common five-item-per-attribute design. The comparison tests whether

construct recovery follows a similar profile across models and whether high intermediate-layer coherence persists into the final state.

Related Work

Interpretability Methods for Transformers

The interpretability literature for transformers can be organized along several axes. *Feature attribution* methods, i.e., SHAP, Integrated Gradients, and attention rollout, provide token- or feature-level importance scores but do not capture multidimensional structure. *Attention and neuron analyses* examine how models allocate focus and which neurons activate for specific inputs, leading to insights into functional specialization but typically at a single layer or for specific prompts. *Mechanistic interpretability*, including circuit tracing and attribution graphs (Lindsey et al., 2025; Olah et al., 2020), systematically identifies computational pathways and causal circuits within models, revealing how models implement multi-step reasoning, entity tracking, and factual recall. However, circuit tracing is inherently micro-level. In other words, it traces individual computations rather than mapping the macro-level trajectory of how an abstract concept organizes across all layers.

Representation Geometry and Probing

A complementary line of work studies the geometry of transformer representations. Centered Kernel Alignment (CKA) and Singular Vector Canonical Correlation Analysis (SVCCA) compare representational similarity across layers, models, or training stages. Probing classifiers assess whether specific linguistic properties (part-of-speech, dependency relations, semantic roles) are linearly decodable from hidden states at each layer. These methods shows that syntactic information peaks in middle layers while semantic features concentrate in upper layers (Jawahar, Sagot, & Seddah, 2019; Rogers, Kovaleva, &

Rumshisky, 2020; Tenney, Das, & Pavlick, 2019). Representational Similarity Analysis compares neural activation geometries to behavioral or computational models.

These approaches assess *whether* a property is present but not *how well-organized* or *how efficiently encoded* a multi-dimensional construct is. A probing classifier might detect that narcissism-related features, for example, are decodable at a given layer without revealing whether the five-factor structure of the construct has emerged as a coherent, parsimonious organization. CET addresses this gap by asking whether information is present *and* how well structured the construct representation is.

Network Psychometrics

CET draws on tools developed in network psychometrics (Epskamp, Maris, et al., 2018; Epskamp, Waldorp, Mõttus, & Borsboom, 2018; H. F. Golino, Christensen, & Garrido, 2022; H. Golino, Nesselroade, & Christensen, 2025), a subfield of quantitative psychology. Exploratory Graph Analysis (EGA) (Christensen & Golino, 2021; H. F. Golino & Epskamp, 2017; H. F. Golino et al., 2020) estimates dimensionality by modeling multivariate data as a network using either regularized partial correlations (Epskamp, Maris, et al., 2018; Friedman, Hastie, & Tibshirani, 2008; H. F. Golino et al., 2020) or the triangulated maximally filtered graph (Massara, Di Matteo, & Aste, 2016) and applying community detection algorithms. EGA has demonstrated high accuracy in recovering the number and membership of latent dimensions across extensive Monte-Carlo simulation studies (Christensen, Garrido, Guerra-Peña, & Golino, 2024; H. F. Golino & Epskamp, 2017; H. F. Golino et al., 2020). Dynamic EGA (DynEGA) (H. Golino, 2026; H. F. Golino, Christensen, Moulder, et al., 2022; Tomašević, Golino, & Christensen, 2024) extends this framework to time-series and ordered data by combining state-space reconstruction via time-delay embedding, derivative estimation via Generalized Local Linear Approximation (GLLA) (Boker, Deboek, Edler, & Keel, 2010; Deboeck, Montpetit, Bergeman, & Boker, 2009), and network-based community detection. The Total Entropy Fit Index (TEFI) (H.

F. Golino, Jiménez, Garrido, & Christensen, 2024; H. F. Golino et al., 2021) provides an entropy-based measure of how well a proposed partition organizes the correlation structure, grounded in Von Neumann entropy from quantum information theory. The Kolmogorov Complexity proxy for networks, estimated via compression of serialized edge lists, quantifies the algorithmic complexity of a network's structure (H. Golino, 2025; H. Golino et al., 2025).

CET adapts these tools to a new domain, by treating transformer hidden states as the data, embedding dimensions as the pseudo-temporal ordering, and layers as the developmental axis along which construct structure emerges.

Theoretical Framework

Representational Development in Transformers

Transformer processing can be conceptualized as a *hierarchical abstraction pipeline*. Each layer applies attention and feed-forward transformations that progressively re-encode input representations. The literature broadly supports a developmental view. Early layers encode local lexical and syntactic features, middle layers perform contextual integration and begin forming abstract representations, and late layers compress and reformat representations for the output task (Jawahar et al., 2019; Rogers et al., 2020; Tenney et al., 2019). CET formalizes this view by defining what it means for a construct to be *represented* at a given layer and tracking how this representation evolves.

Definition 1 (Construct Representation)**.** *Let $\mathcal{C}$ be a target construct defined by a partition of a set of textual items into r labeled categories (e.g., five dimensions of narcissism). The construct $\mathcal{C}$ is* ***represented*** *at layer ℓ if the hidden states of items belonging to $\mathcal{C}$ form a coherent cluster structure in the layer's representational space, with clustering quality exceeding chance levels under an appropriate null model.*

Definition 2 (Representational Trajectory)**.** *The* ***representational trajectory*** $\mathcal{T}(\mathcal{C})$ *of construct* $\mathcal{C}$ *across a model with* L *layers is the sequence*

$$\mathcal{T}(\mathcal{C}) = \Big\{\mathcal{R}_1(\mathcal{C}),\ \mathcal{R}_2(\mathcal{C}),\ \ldots,\ \mathcal{R}_L(\mathcal{C})\Big\},$$

where $\mathcal{R}_\ell(\mathcal{C})$ *denotes the construct's representational state at layer* ℓ*, quantified by a tuple of indices measuring semantic coherence, organizational structure, and representational efficiency.*

Indices of Representational Maturity

CET triangulates three complementary properties at each layer.

Semantic Coherence (NMI). Normalized Mutual Information between the communities detected by DynEGA and the ground-truth dimensional labels. NMI ranges from 0 (no agreement) to 1 (perfect recovery). Higher NMI indicates that the model's internal clustering at a given layer aligns with the intended construct structure.

Organizational Structure (TEFI). The Total Entropy Fit Index evaluates how well a dimensional partition organizes the correlation structure, using Von Neumann entropy. Given a correlation matrix $\mathbf{R}$ and its density-scaled version $\rho = \mathbf{R}/p$ (where p is the number of variables), the Von Neumann entropy is

$$\mathcal{S}(\rho) = -\operatorname{tr}(\rho \log \rho) = -\sum_i \lambda_i \log \lambda_i,$$

where $\{\lambda_i\}$ are the eigenvalues. For a partition into N_F dimensions with submatrices $\{\rho_k\}$:

$$\text{TEFI} = \left[\frac{1}{N_F}\sum_{k=1}^{N_F}\mathcal{S}(\rho_k) - \mathcal{S}(\rho)\right] + \left[\left(\mathcal{S}(\rho) - \sum_{k=1}^{N_F}\mathcal{S}(\rho_k)\right)\sqrt{N_F}\right].$$

Lower (more negative) TEFI values indicate better-organized structure. TEFI is sensitive to both the number of dimensions and the assignment of items to dimensions, making it a strict test of structural quality.

Representational Efficiency (KC). A compression-based proxy for Kolmogorov Complexity estimates the minimal informational encoding of the estimated network. The implemented estimator thresholds the DynEGA network to edge presence, constructs an edge list with constant retained-edge encoding, serializes it, and compresses it using gzip. The compressed length serves as the KC estimate (H. Golino, 2025; H. Golino et al., 2025). To obtain a robust estimate, compressed lengths are averaged across 1,000 random shuffles of the edge ordering. Lower KC indicates a more compressible, and therefore more structurally regular and efficient, network representation.

Representational Maturity Index (RM). Coherence, entropy fit, and network compressibility are combined on a common descriptive scale:

$$\mathrm{RM}_{m\ell nt} = 0.5\,\mathrm{NMI}_{m\ell nt} + 0.3\,\frac{T_{\max} - \mathrm{TEFI}_{m\ell nt}}{T_{\max} - T_{\min}} + 0.2\,\frac{K_{\max} - \mathrm{KC}_{m\ell nt}}{K_{\max} - K_{\min}},$$

where m, ℓ, n, and t index model, recorded state, embedding dimension, and simulation iteration. NMI retains its theoretical bounds of zero and one; $T_{\min}, T_{\max}$ and $K_{\min}, K_{\max}$ are the empirical TEFI and KC bounds pooled across all fourteen models. The weights give greatest emphasis to agreement with the stipulated attributes while incorporating entropy fit and topology compressibility. RM lies in $[0, 1]$ for the observed data. Its normalization depends on the comparison set and does not establish measurement invariance across architectures. Raw NMI, TEFI, and KC are therefore reported alongside the composite.

Semantic Manifold Reconstruction and Cluster Stability

The theoretical rationale for CET rests on the observation that transformer hidden states for semantically related items concentrate near a lower-dimensional manifold embedded in the high-dimensional activation space (Bengio, Courville, & Vincent, 2013; Cai, Huang, Bian, & Church, 2021; H. Golino, 2026). CET leverages time-delay embeddings, part of classical reconstruction theorems (Takens, 1981; Whitney, 1936), to recover the structure of this manifold and then applies community detection to identify

semantic clusters.

Theorem 1 (Semantic Manifold Reconstruction)**.** *Let $\mathcal{M} \subset \mathbb{R}^k$ be a compact, C^2 manifold of intrinsic dimension $m < k$ parameterizing the semantic states visited by hidden activations. Let $f : \mathcal{M} \to \mathcal{M}$ be a C^2 diffeomorphism and $\phi : \mathcal{M} \to \mathbb{R}$ a generic C^2 observation function. By Takens' theorem, for generic ϕ and almost all delays $\tau > 0$, the delay map*

$$\Phi(x) = \Big(\phi(x),\ \phi(f^{\tau}(x)),\ \ldots,\ \phi(f^{2m\tau}(x))\Big) \in \mathbb{R}^{2m+1}$$

is a C^1 diffeomorphism onto its image. Suppose the semantic communities $\{C_1, \ldots, C_r\}$ form a partition of $\mathcal{M}$ with positive separation margin
$\Delta = \min_{i \neq j} \inf\{d(x, y) : x \in C_i, y \in C_j\} > 0$*. Then Φ preserves cluster separations:*

$$\min_{i \neq j} \inf\{\|\Phi(x) - \Phi(y)\| : x \in C_i, y \in C_j\} \geq a\,\Delta > 0,$$

where $a > 0$ is the lower bi-Lipschitz constant of Φ on the compact $\mathcal{M}$.

The proof follows from two observations. First, a C^1 embedding of a compact manifold is bi-Lipschitz: compactness and injectivity ensure the Jacobian's smallest singular value is bounded away from zero, yielding constants $0 < a \leq b < \infty$ such that $a\,d(x, y) \leq \|\Phi(x) - \Phi(y)\| \leq b\,d(x, y)$. Second, bi-Lipschitz maps preserve positive separation margins, meaning that any clustering procedure stable under Lipschitz distortions (such as density-based clustering or k-NN graph methods for appropriate k) will recover the same partition up to label permutation. The full proof is provided in Appendix A.

Remark 1. Community structure per se is not a topological invariant. However, when communities correspond to Δ-separated subsets of a compact manifold, bi-Lipschitz embeddings preserve the separation margins that community detection algorithms rely on. CET measures the strength of community recovery empirically via NMI, providing a direct diagnostic of whether the reconstructed space supports faithful dimensional identification.

Methods

CET Pipeline



Figure 1. The CET-DynEGA pipeline. (a) Each item's activation vector from a given layer is treated as a univariate time series indexed by hidden-unit position. (b) A time-delay embedding matrix is constructed by sliding a window of n consecutive values. (c) The L matrix encodes polynomial basis weights for estimating derivatives of different orders. (d) GLLA projects the embedding into derivative space, yielding estimated values, velocities, and accelerations. (e) First-derivative columns are extracted from each item's Y matrix and column-bound to form the input to network estimation. (f) EGA estimates a network and detects communities; varying n across 3–200 searches the activation landscape for the

The CET pipeline consists of nine steps applied to each model:

1. **Item sampling.** From a curated pool of approximately 50 textual items per attribute of the target construct, randomly sample five items per attribute without replacement, yielding 25 items per iteration.

2. **Hidden-state extraction.** Pass each item through the model; at each layer ℓ, compute the mean hidden state across all tokens, yielding one d-dimensional vector per item per layer (where d is the model's hidden size).

3. **Standardization.** Standardize each hidden unit across the stacked item-by-layer observations, before separating layers for network estimation.

4. **Time-delay embedding.** For each layer and item, construct a time-delay embedding matrix with $n \in \{3, 4, \ldots, 200\}$ columns, treating the hidden-state dimension index as a pseudo-temporal ordering.

5. **Derivative estimation (GLLA).** Compute first-order derivatives from the embedding matrix using Generalized Local Linear Approximation.

6. **Network estimation (EBICglasso).** Estimate a Gaussian Graphical Model from the covariance of derivatives, selecting the regularization parameter λ over a grid of 100 values via the Extended Bayesian Information Criterion with adaptive $\gamma \in \{0.50, 0.25, 0\}$.

7. **Community detection.** Apply the Louvain algorithm to the estimated network to identify communities.

8. **Metric computation.** For each (layer, embedding dimension) pair, compute NMI, TEFI, and KC.

9. **Aggregation and maturity.** Repeat steps 1–8 across iterations (500 available for GPT-2; 100 for Phi-2; 100 for each remaining model). Aggregate NMI, TEFI, and

KC using arithmetic means across iterations. Compute RM using the pooled bounds for all fourteen models.

Dynamic Exploratory Graph Analysis (DynEGA)

DynEGA integrates dynamical-systems reconstruction with network psychometrics. Given a set of item-level activation vectors at a particular layer, the procedure treats the hidden-state dimension index as a pseudo-temporal ordering. For each item's d-dimensional activation vector $\mathbf{v} = (v_1, v_2, \ldots, v_d)$, the time-delay embedding with lag τ and embedding size n produces the matrix:

$$\mathbf{X}^{(n)} = \begin{bmatrix} v_1 & v_{1+\tau} & \cdots & v_{1+(n-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ v_M & v_{M+\tau} & \cdots & v_{M+(n-1)\tau} \end{bmatrix}, \quad M = d - (n-1)\tau.$$

First-order derivatives are estimated via GLLA:

$$\mathbf{Y} = \mathbf{X}\,\mathbf{L}\,(\mathbf{L}'\mathbf{L})^{-1},$$

where $\mathbf{L}$ contains polynomial basis weights. The covariance of derivative columns across items is then used to estimate a sparse Gaussian Graphical Model via GLASSO with EBIC model selection. Community detection on the resulting network identifies clusters of variables that co-fluctuate, interpreted as dynamical dimensions. In the CET context, these communities correspond to groups of items whose activation patterns share common dynamical structure, i.e., items whose representations change together across the embedding landscape.

The key innovation of applying DynEGA in this context is that it does not assume the entire activation vector contributes uniformly to dimensional structure (H. Golino, 2026). Instead, by varying the embedding dimension n from 3 to 200, CET systematically searches the activation landscape, identifying regions where construct structure is optimally represented.

Models and Construct

Models. Fourteen LLM models span nominal parameter counts from 117M to 32B and 12 to 64 decoder blocks (Table 1). The comparison includes GPT-2, TinyLlama, Qwen2, Phi-2, Phi-4-mini, six Qwen2.5-Instruct sizes, GPT-OSS, Qwen3.8, and Muse-Glimmer. Large language models, rather than architectural families, are the experimental conditions: the six Qwen2.5 sizes provide a within-family comparison, while the other models broaden variation in depth, hidden size, and attention design.

Table 1

Large Language Models

Model	Parameters	Blocks	Hidden size
GPT-2	117M	12	768
TinyLlama 1.1B	1.1B	22	2,048
Qwen2 1.5B	1.5B	28	1,536
Phi-2	2.7B	32	2,560
Phi-4-mini	3.8B	32	3,072
Qwen2.5 0.5B	0.5B	24	896
Qwen2.5 1.5B	1.5B	28	1,536
Qwen2.5 3B	3B	36	2,048
Qwen2.5 7B	7B	28	3,584
Qwen2.5 14B	14B	48	5,120
Qwen2.5 32B	32B	64	5,120
GPT-OSS 20B	20B	24	2,880
Qwen3.8 27B	27B	64	5,120
Muse-Glimmer 30B	30B	52	6,656

Parameter counts are nominal model sizes, not counts of parameters active for a

particular token. All Qwen2.5 conditions use Instruct model versions, TinyLlama uses the Chat-v1.0 checkpoint, and Phi-4-mini uses its instruction-tuned checkpoint. Qwen3.8 denotes the stored Qwen3.8-27B BF16 GGUF identifier, whose configuration specifies a Qwen3.5 text architecture. Its 64 blocks comprise 48 linear-attention and 16 full-attention blocks. Muse-Glimmer has 52 text blocks, comprising 39 sliding-attention and 13 full-attention blocks. The comparison evaluates text representations throughout.

Construct and item sampling. Grandiose narcissism is operationalized through five attributes based on Pincus and Lukowitsky (2010): Distorting Disconfirming External Information, Exhibitionism, Intense Envy, Lack of Empathy, and Repression of Negative Self-Information. Each iteration samples five items without replacement from each attribute pool, producing 25 items and five stipulated attribute labels. The number of items per attribute is identical for every model. Sampling is repeated to assess variation in construct recovery across item subsets.

Activation Extraction and Computation

Items are supplied as plain text, without a conversational template, and token-level hidden states are averaged to obtain one vector per item and recorded state. Each hidden unit is standardized across the stacked item-by-state observations before the data are separated by state for network estimation. GPT-2 contributes its embedding output and 12 decoder outputs, yielding 13 recorded states. The other models contribute their decoder-state sequences with the initial embedding output excluded. Relative depth therefore indexes the recorded sequence; its first state does not represent an identical computational operation in every architecture.

Activations are evaluated in float32 through HuggingFace Transformers. GPT-2, TinyLlama, Qwen2, and Phi-4-mini use CPU execution. Phi-2, Qwen2.5, GPT-OSS, Qwen3.8, and Muse-Glimmer use an NVIDIA A100 with 80 GB device memory and CPU

placement when required by model size. GPT-OSS's MXFP4 weights are dequantized through BF16 before conversion to float32; this conversion does not recover weights prior to quantization. Qwen3.8 is imported from a BF16/F32 GGUF, whereas Muse-Glimmer uses native safetensors converted to float32. GPT-OSS was also evaluated in a 100-iteration reference run using OpenAI's PyTorch implementation, which dequantizes the released MXFP4 expert weights to BF16 and performs BF16 computation with native float32 normalization parameters. This reference run used the same item count, embedding grid, and final-state extraction convention. The final hidden state follows each model implementation's output-normalization convention. Accordingly, a final returned state may differ from a block output sampled before final normalization.

The GPU runs use Python 3.11.16 and PyTorch 2.5.1, with Transformers 4.57.6 for Phi-2, Qwen2.5, and GPT-OSS and 5.17.0 for Qwen3.8 and Muse-Glimmer. The GPT-OSS BF16 reference run uses a separate environment with Python 3.12.14 and PyTorch 2.7.1+cu118 (CUDA 11.8). Model-specific configurations and available runtime records accompany the analysis code. Network estimation is distributed over embedding configurations using R's `foreach` and `doParallel`; the A100 workstation uses 80 R workers, with one numerical-library thread per worker. These workers parallelize the network analysis rather than loading independent copies of the language model.

Simulation Scale

Each model is evaluated over 198 embedding dimensions ($n = 3, \ldots, 200$) at every recorded state and across repeated item samples. The 13–64 recorded states, 198 embedding configurations, and 100 iterations per model yield 9,801,000 network estimates (Table 2). All estimates use five items per attribute, and 100 iterations per model/layer/embedding configurations.

Result validation checks complete state-by-embedding grids, unique

Table 2

Simulation scale with five items per attribute.

Model	States	Iterations
GPT-2	13	100
TinyLlama 1.1B	22	100
Qwen2 1.5B	28	100
Phi-2	32	100
Phi-4-mini	32	100
Qwen2.5 0.5B	24	100
Qwen2.5 1.5B	28	100
Qwen2.5 3B	36	100
Qwen2.5 7B	28	100
Qwen2.5 14B	48	100
Qwen2.5 32B	64	100
GPT-OSS 20B	24	100
Qwen3.8 27B	64	100
Muse-Glimmer 30B	52	100

iteration/state/embedding keys, finite primary metrics, and NMI values within $[0, 1]$.

Aggregation and Cross-Model Comparison

All models use the same aggregation. Cell-level results are arithmetic means across iterations for each state and embedding dimension. A model's overall NMI first averages the full state-by-embedding grid within each iteration and then averages these iteration means. Ninety-five percent Student-t Monte Carlo intervals use variation across iteration means; the network estimates within a grid are not treated as independent replicates.

These intervals quantify variability across sampled item sets for the models and analysis procedure used.

For a recorded state ℓ in a sequence of length L, relative depth is $100(\ell - 1)/(L - 1)$. NMI depth curves average over embedding dimensions. Maturity trajectories instead report the maximum of the mean RM surface over embedding dimensions at each state, describing the best observed composite score at that depth. Landscape figures retain every state and embedding configuration. The common RM normalization uses pooled TEFI bounds of $[-35.26579, 33.01256]$ and KC bounds of $[201.711, 676.193]$. Because RM is affine in the metrics, its cell mean can be computed directly from the cell-level mean NMI, TEFI, and KC.

Two descriptive summaries support comparisons of trajectory shape. Recovery onset is the first recorded state at which NMI averaged across embedding dimensions reaches 0.50; this operational threshold is not a test against chance. Terminal behavior is classified as near-zero when final-state mean NMI is below 0.05 and retained otherwise. Exact terminal means and maxima are reported so that the distinction does not depend solely on the threshold. Peaks over the state-by-embedding grid are exploratory selected maxima, not held-out estimates.

Results

Construct Recovery Across Models

Construct recovery varies across models despite using the same number of items per attribute and the same embedding-dimension grid (Table 3). Overall mean NMI ranges from 0.382 in TinyLlama to 0.702 in Phi-2. Qwen3.8 and Muse-Glimmer each attain a mean of approximately 0.662, although their entropy-fit and terminal profiles differ substantially. Phi-2 has the highest full-grid mean, and its Monte Carlo interval lies above those of the other models. This separation concerns recovery of the sampled construct

under CET rather than general model capability.

Within Qwen2.5-Instruct, overall mean NMI increases with nominal size: 0.504, 0.563, 0.601, 0.611, 0.629, and 0.653 for 0.5B through 32B. This ordering describes recovery over the entire sampled grid, including early and terminal states. It differs from the ordering of best-cell NMI, which is not monotone across the six sizes. Parameter count, training, and architectural features are not experimentally separated, so the trend is evidence of an association within this checkpoint family rather than a causal scaling law.

Table 3

Construct recovery across model conditions.

Model	Mean NMI	95% MC interval	Peak NMI
GPT-2	0.396	[0.393, 0.400]	0.758
TinyLlama 1.1B	0.382	[0.375, 0.389]	0.883
Qwen2 1.5B	0.575	[0.567, 0.583]	0.879
Phi-2	0.702	[0.692, 0.713]	0.899
Phi-4-mini	0.451	[0.442, 0.461]	0.875
Qwen2.5 0.5B	0.504	[0.495, 0.513]	0.848
Qwen2.5 1.5B	0.563	[0.553, 0.573]	0.824
Qwen2.5 3B	0.601	[0.592, 0.610]	0.895
Qwen2.5 7B	0.611	[0.601, 0.620]	0.864
Qwen2.5 14B	0.629	[0.619, 0.639]	0.900
Qwen2.5 32B	0.653	[0.643, 0.663]	0.899
GPT-OSS 20B	0.463	[0.456, 0.470]	0.891
Qwen3.8 27B	0.662	[0.652, 0.671]	0.922
Muse-Glimmer 30B	0.662	[0.651, 0.673]	0.916

Mean NMI averages all recorded states and embedding dimensions within each

iteration. MC intervals reflect Monte Carlo variability across iterations. Peak NMI is the maximum of the iteration-averaged state-by-embedding surface and is reported separately because optimizing over the grid answers a different question from averaging over it.

Figure 2 displays the same model means and Monte Carlo intervals. The common plotting scale highlights both the separation of lower and higher full-grid means and Phi-2's higher mean, while the intervals for Qwen2.5-32B, Qwen3.8, and Muse-Glimmer overlap.

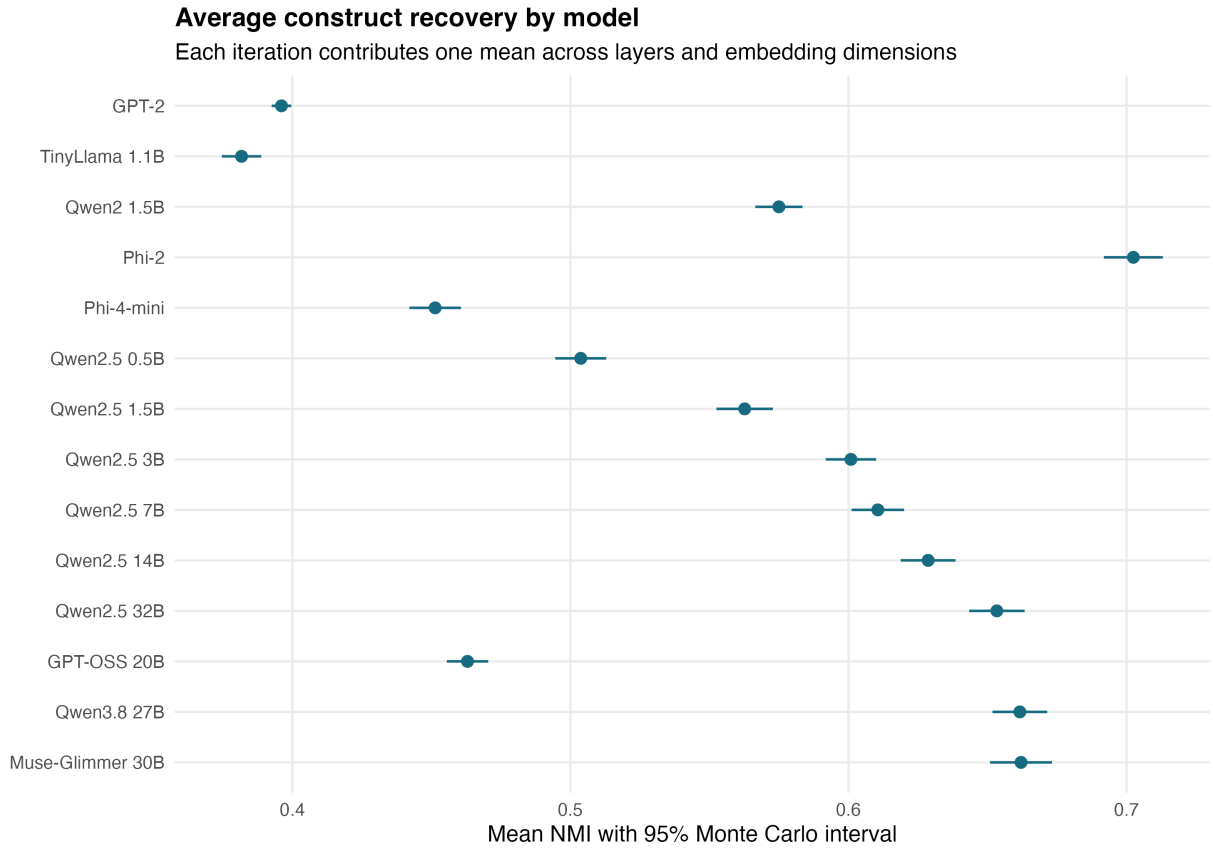


Figure 2. Average construct recovery by model. Each iteration contributes one mean across recorded states and embedding dimensions; points average these iteration means and intervals show 95% Monte Carlo uncertainty. Every model uses five items per attribute and the same aggregation.

Emergence Trajectories and Landscapes

Figures 4 and 5 summarize emergence across all fourteen models using the same RM normalization. Most trajectories show a rise from early states followed by a broad region of higher maturity, but neither the onset nor the location of the optimum is fixed across architectures. Mean NMI first reaches 0.50 at relative depths ranging from 12.9% in Phi-2 to 52.4% in TinyLlama. The six Qwen2.5 models reach this threshold between 19.0% and 25.9% depth despite their different block counts.

The location of maximum composite maturity also varies. GPT-2 peaks at state 8 of 13, Phi-2 at block 20 of 32, Qwen2.5-32B at block 52 of 64, and Muse-Glimmer at block 38 of 52. These differences indicate that a common middle-layer interval would omit some models' best composite states. A high RM value need not coincide with maximum NMI because the composite also depends on entropy fit and network compressibility.

Figure 3 overlays all fourteen trajectories on common axes, allowing direct comparison of recovery timing and terminal changes. The separate panels in Figure 4 make individual trajectories easier to inspect where curves overlap.

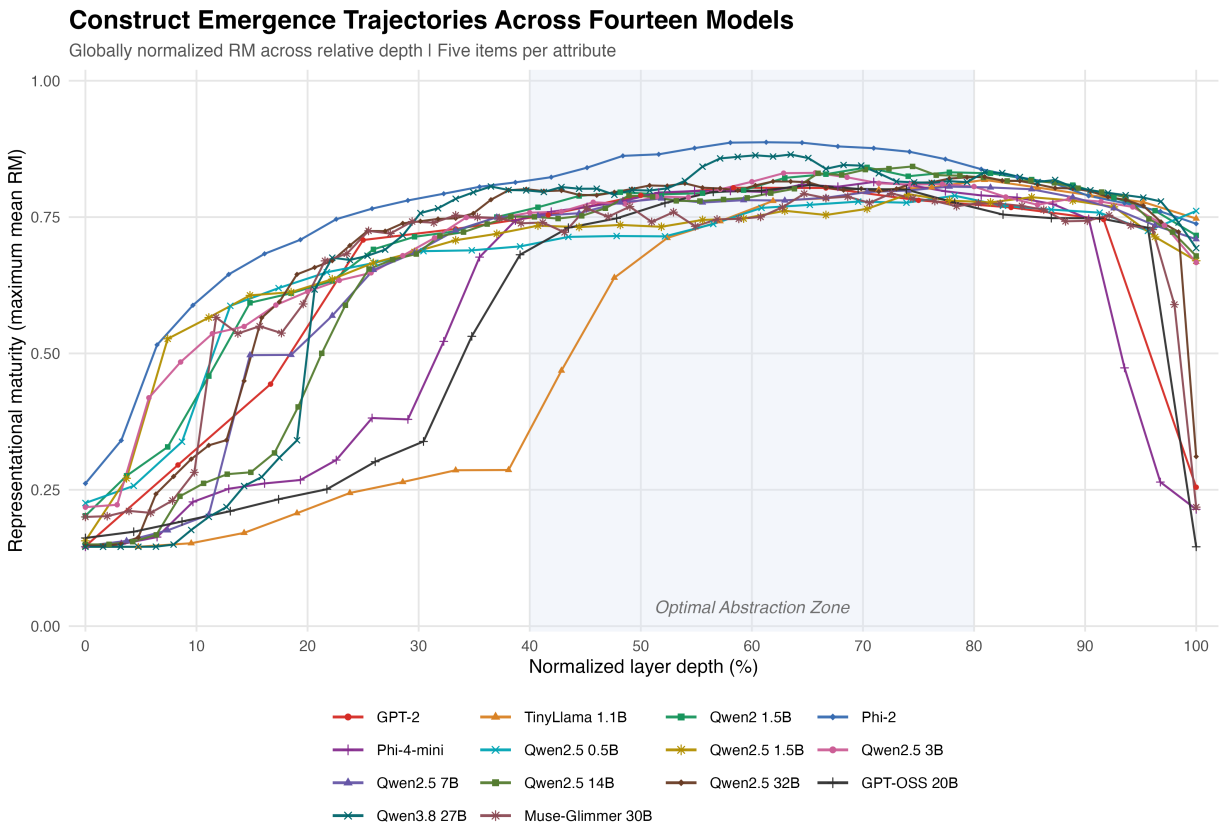


Figure 3. Construct emergence trajectories across fourteen models on common axes. Each point represents the maximum, over embedding dimensions, of mean RM at one recorded state. Colors and markers distinguish models. All curves use five items per attribute and the same TEFI and KC normalization pooled across the fourteen models. The shaded 40–80% depth interval marks the reference Optimal Abstraction Zone; individual model optima may lie outside this interval. Lines connect successive recorded states; relative depth includes the embedding output for GPT-2.

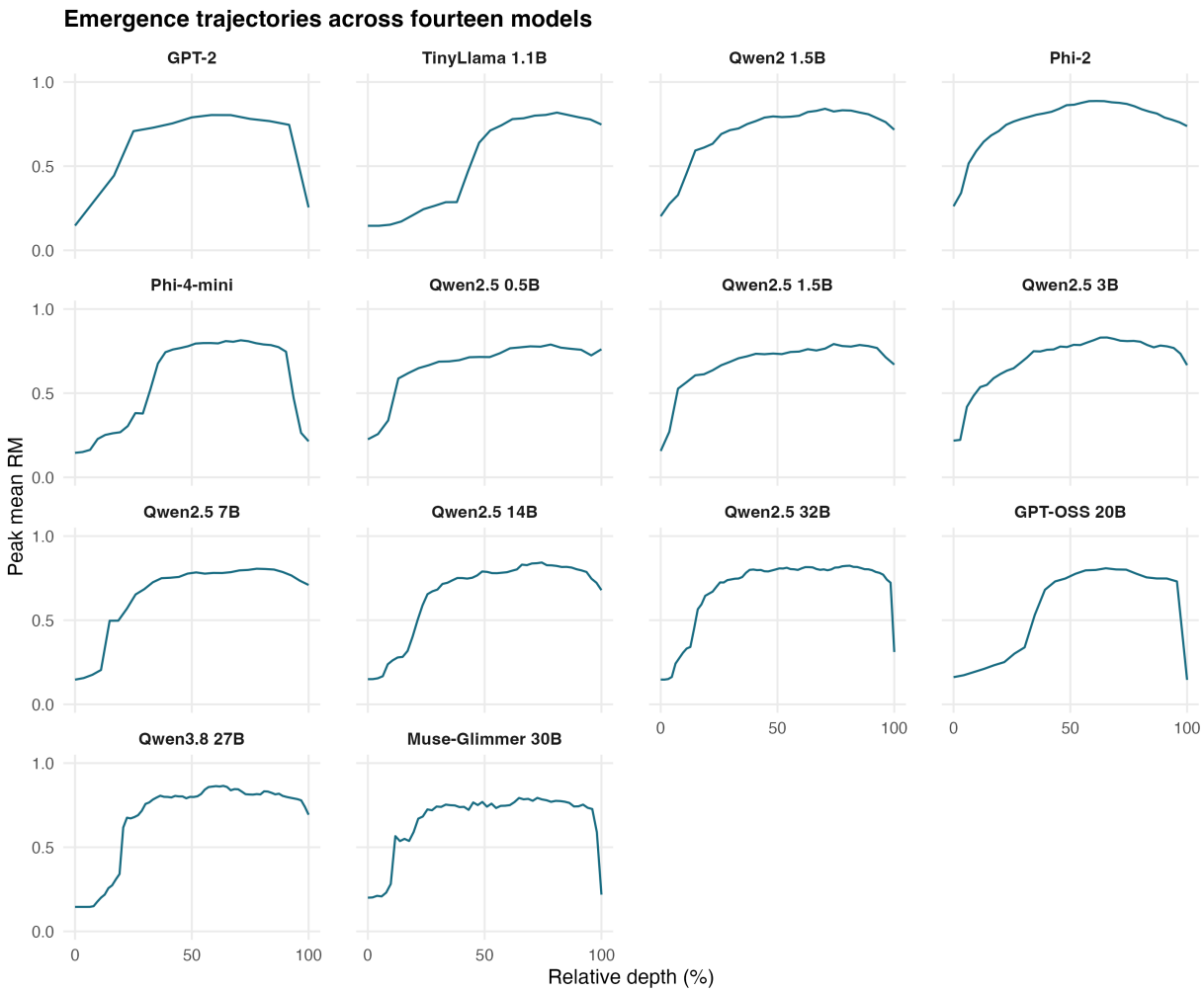


Figure 4. Emergence trajectories across fourteen models. Each curve shows the maximum, over embedding dimensions, of mean RM at each recorded state. TEFI and KC are normalized using bounds pooled across all fourteen models; NMI retains its 0–1 scale. Panels share depth and maturity axes. Relative depth indexes each recorded sequence, including the embedding output for GPT-2.

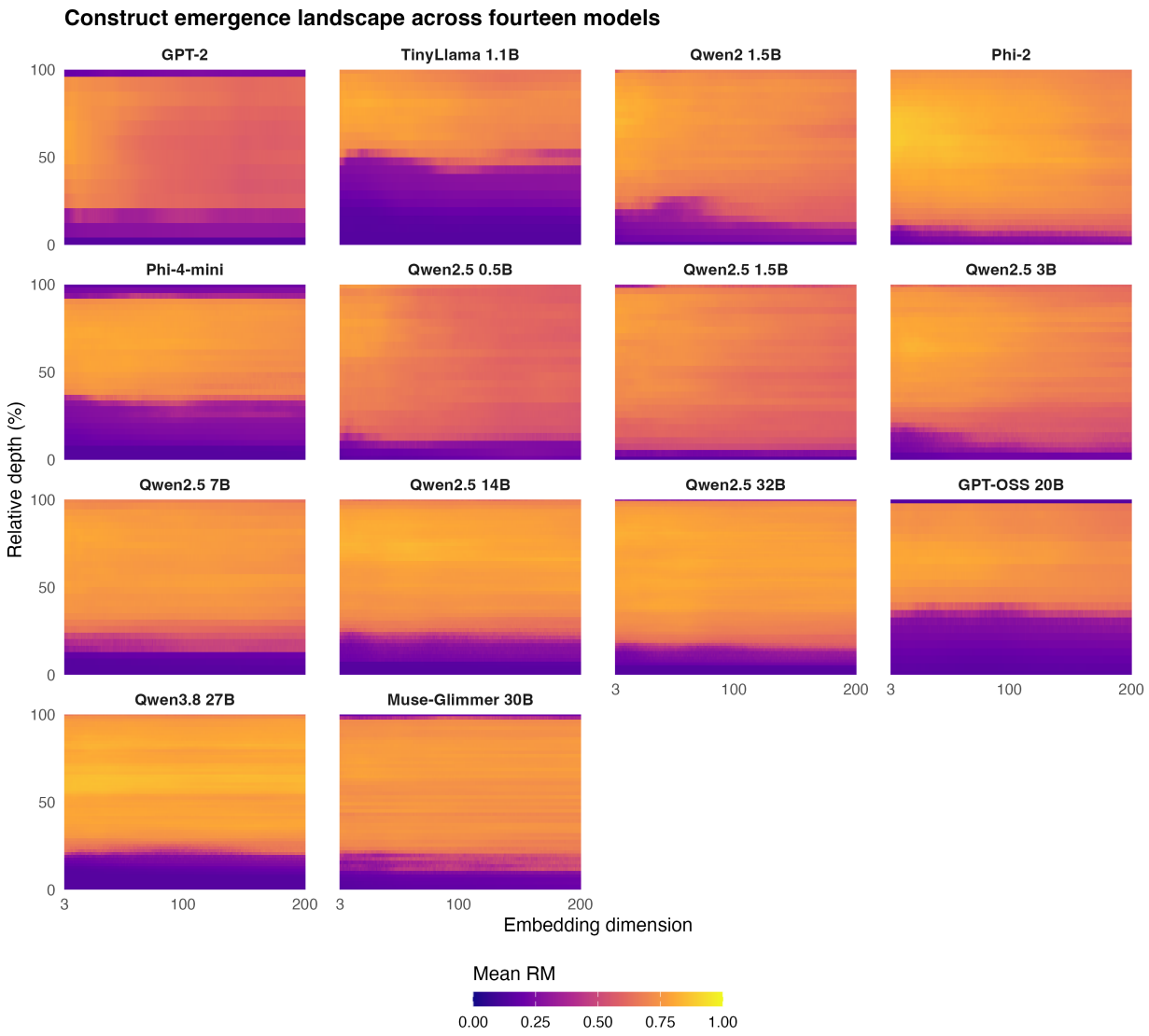


Figure 5. Construct emergence landscape across fourteen models. Each cell shows mean RM at a recorded state and embedding dimension using the same pooled normalization as Figure 4. States are placed on a relative-depth axis without binning, preserving narrow terminal bands. Warmer colors indicate higher composite scores within this comparison set.

The NMI surfaces (Figure 6) distinguish agreement with the attribute labels from the broader composite. The largest selected mean-cell NMI is 0.922 for Qwen3.8 at block 37 and embedding dimension 13, followed by 0.916 for Muse-Glimmer at block 26 and embedding dimension 9. Nevertheless, high intermediate-layer agreement does not imply

terminal retention: Muse-Glimmer reaches zero final-state NMI, whereas Qwen3.8 retains a final-state mean of 0.744.

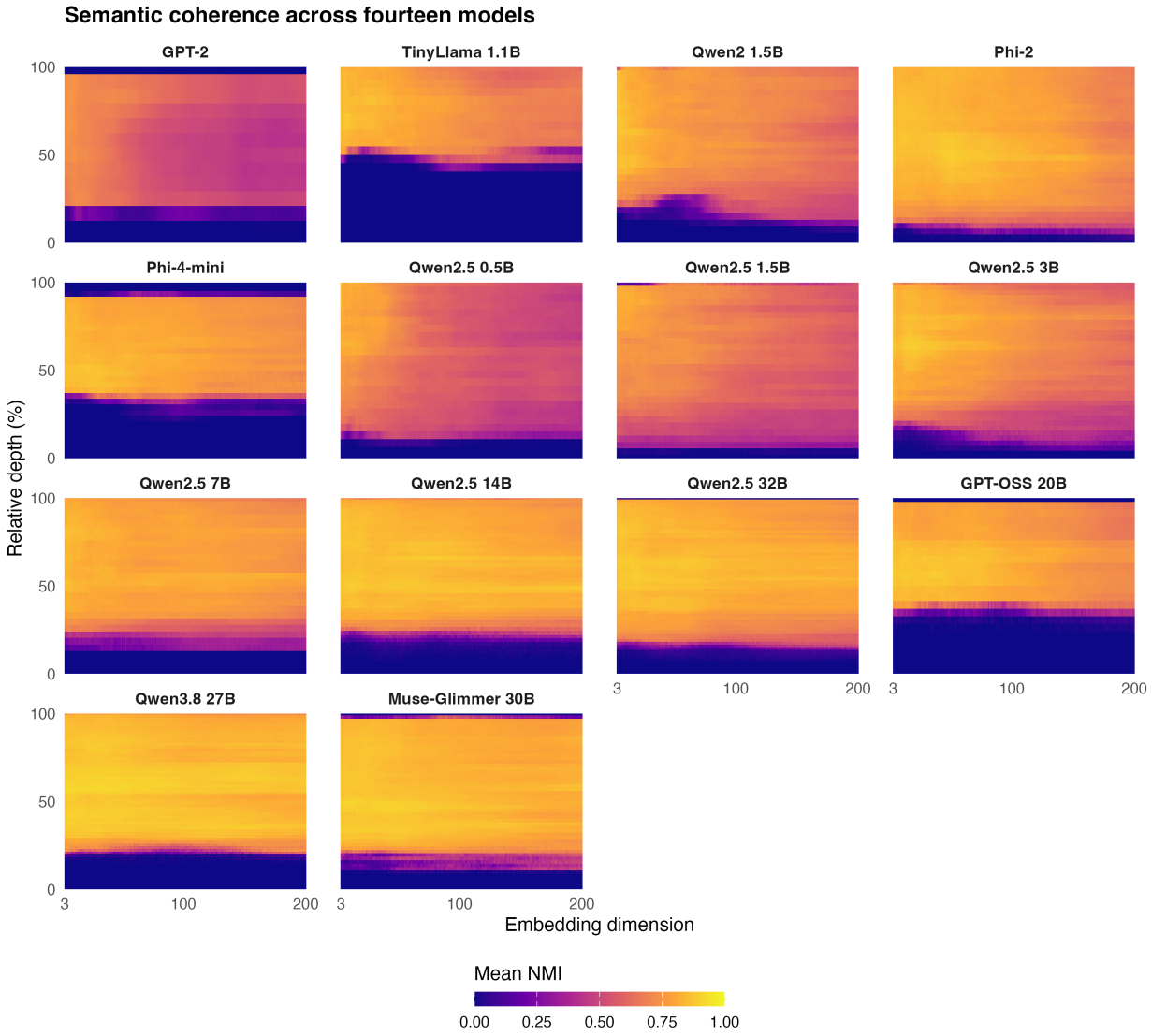


Figure 6. Semantic coherence across fourteen models. Cells show arithmetic-mean NMI across iterations for each recorded state and embedding dimension. All panels use the same color scale and relative-depth axis. The selected maximum of a surface and its mean across the grid describe different aspects of recovery.

Model-Specific Emergence Signatures

GPT-2. GPT-2 exhibits a comparatively short emergence sequence. Mean NMI reaches 0.50 at state 4, while the highest mean-cell NMI is 0.758 at state 9. Mean TEFI and peak composite maturity attain their optima at state 8. Recovery remains substantial through state 12, whose mean NMI is 0.607, but falls to zero at state 13. The distinction between its cell-level NMI peak and its depth-averaged NMI maximum at state 11 illustrates why a single best-layer summary does not fully describe the surface.

TinyLlama-1.1B. TinyLlama has the latest onset under the 0.50 mean-NMI criterion, reaching it at block 12 of 22. Its selected mean-cell NMI nevertheless reaches 0.883 at block 16. NMI averaged across embedding dimensions and composite maturity both peak at block 18, and mean NMI remains 0.699 in the final block. Its relatively low overall mean therefore reflects a long low-recovery portion of the sequence rather than uniformly poor recovery at all depths.

Qwen2-1.5B. Qwen2 reaches the onset criterion at block 8 and sustains recovery across much of the remaining sequence. Its selected mean-cell NMI peaks at 0.879 at block 20, which also gives the best mean TEFI and peak composite maturity. Mean NMI across embedding dimensions reaches its maximum later, at block 26, before declining to 0.699 at block 28. The separation between selected-cell and depth-averaged optima indicates that the width of the favorable embedding region changes with depth.

Phi-2. Phi-2 reaches the onset criterion at block 5, earlier in relative depth than any other model, and has the highest overall mean NMI (0.702). Its selected mean-cell NMI is 0.899 at block 16; mean TEFI and composite maturity reach their optima at block 20. Final-state mean NMI remains 0.777. This combination of early recovery and sustained agreement contributes to its strong full-grid average.

Phi-4-mini. Phi-4-mini reaches mean NMI of 0.50 at block 12 and attains selected mean-cell NMI of 0.875 at block 17. Composite maturity peaks at block 23, while mean

TEFI is lowest at block 25, indicating continued changes in network organization after coherence is already high. The final two blocks have zero NMI. Its trajectory therefore combines high intermediate-layer agreement with terminal loss of agreement under CET.

Qwen2.5-Instruct: 0.5B to 32B. The six Qwen2.5 models combine a consistent ordering of overall mean NMI with different local optima and terminal outcomes. The 0.5B model's selected mean-cell peak (0.848) occurs in its final block, even though its mean NMI across embedding dimensions is highest at block 15. The 1.5B model reaches a selected peak of 0.824 at block 26 and retains a final-state mean of 0.504. The 3B and 7B models attain selected peaks of 0.895 and 0.864 at blocks 24 and 15, respectively; their final-state means are 0.638 and 0.722.

The 14B and 32B models reach nearly identical selected peaks (0.900 and 0.899) at blocks 32 and 37, but diverge sharply at the output. The 14B model retains final-state mean NMI of 0.709. The 32B model falls from 0.784 in block 63 to 0.025 in block 64, despite its higher overall mean. Thus the within-family association between size and average recovery does not extend to terminal retention.

GPT-OSS-20B. GPT-OSS reaches the onset criterion at block 10 of 24. Selected mean-cell NMI peaks at 0.891 at block 14; mean NMI and composite maturity peak at block 16, followed by the best mean TEFI at block 18. Agreement persists through block 23 (mean NMI 0.742) before falling to zero at block 24. The OpenAI BF16 reference run closely reproduced this profile: overall mean NMI was 0.463, and final-state NMI remained zero in all 19,800 terminal configurations. This agreement does not support computation precision as the explanation for the terminal decline, although implementation and precision changed together. Both runs characterize the released MXFP4-derived weights and do not establish how weights without quantization would behave.

Qwen3.8-27B. The Qwen3.8 checkpoint combines early relative onset (block 15 of 64) with the highest selected mean-cell NMI (0.922 at block 37). Mean NMI across

embedding dimensions and peak RM are greatest at block 41; mean TEFI is lowest at block 42. Final-state mean NMI remains 0.744. Its high recovery is therefore sustained across a substantial part of the sequence and remains measurable at the output, despite a terminal decline. The alternation of linear- and full-attention blocks is a potential source of variation, but this comparison does not isolate their contributions.

Muse-Glimmer-30B. Muse-Glimmer reaches the onset criterion at block 12 of 52 and achieves selected mean-cell NMI of 0.916 at block 26. NMI averaged across embedding dimensions peaks at block 20, whereas both mean TEFI and peak RM reach their optima at block 38. The discrepancy between these depths shows that later network organization can improve without increasing average agreement with the attribute labels. Mean NMI declines to 0.347 in block 51 and zero in block 52, producing a high overall mean alongside complete final-state loss of agreement.

Two Terminal Regimes

The final-state results separate into five near-zero and nine retained-recovery profiles under the descriptive mean-NMI threshold of 0.05 (Table 4). GPT-2, Phi-4-mini, GPT-OSS, and Muse-Glimmer have exactly zero final-state NMI throughout the embedding grid. Qwen2.5-32B has a final-state mean of 0.025, although its maximum across embedding dimensions is 0.120; its decline is therefore substantial but not uniformly complete.

The nine retained profiles have final-state mean NMI between 0.504 and 0.777. Retention does not imply an unchanged final state: Qwen2.5-1.5B declines from 0.680 to 0.504, and Qwen3.8 declines from 0.803 to 0.744. Terminal behavior is therefore continuous within the broad regimes. Exact means and maxima provide a more informative comparison than the categorical labels alone.

Final mean averages NMI across embedding dimensions and iterations, while final

Table 4

Final-state NMI and terminal regime.

Model	Final mean	Final maximum	Regime
GPT-2	0.000	0.000	Near-zero
TinyLlama 1.1B	0.699	0.829	Retained
Qwen2 1.5B	0.699	0.820	Retained
Phi-2	0.777	0.842	Retained
Phi-4-mini	0.000	0.000	Near-zero
Qwen2.5 0.5B	0.626	0.848	Retained
Qwen2.5 1.5B	0.504	0.734	Retained
Qwen2.5 3B	0.638	0.771	Retained
Qwen2.5 7B	0.722	0.807	Retained
Qwen2.5 14B	0.709	0.778	Retained
Qwen2.5 32B	0.025	0.120	Near-zero
GPT-OSS 20B	0.000	0.000	Near-zero
Qwen3.8 27B	0.744	0.813	Retained
Muse-Glimmer 30B	0.000	0.000	Near-zero

maximum is the maximum of the iteration-averaged NMI values over embedding dimensions. Near-zero denotes final mean NMI below 0.05. Neither a zero NMI nor a terminal decline demonstrates the absence of semantic information: these scores measure agreement between a recovered network partition and the stipulated attributes under the CET pipeline.

Terminal retention does not follow a simple ordering by parameter count. The 32B Qwen2.5 checkpoint shows a much larger decline than the smaller members of its family, while the 27B Qwen3.8 checkpoint retains substantial agreement and the 30B Muse

checkpoint does not. Differences in final normalization, architecture, training, and checkpoint representation remain plausible explanations. Therefore, while the present design identifies where recovery changes, it cannot distinguish these mechanisms.

Discussion

CET reveals that construct recovery is distributed unevenly across model depth and that different summaries of the same representation can lead to different model comparisons. Across fourteen models evaluated with the same number of items, overall mean NMI, selected peak coherence, composite maturity, and terminal retention capture distinct properties. The six Qwen2.5 sizes show increasing overall mean NMI, but this ordering does not hold for selected peaks or final-state agreement. High intermediate-layer coherence is compatible with either near-zero or retained terminal recovery.

Generative Psychometrics applies psychometric theory to the generation and evaluation of measures using language models (Garrido et al., 2026; Lara L. Russell-Lasalandra et al., 2024; Lara L. Russell-Lasalandra & Golino, 2026). CET uses the same measurement perspective to study model representations. Rather than treating predictive output as the only object of evaluation, it asks whether relationships among items from a predefined construct are recoverable within internal states. This links the embedding-landscape approach (H. Golino, 2026) to a depth-resolved analysis of construct organization.

The contribution of Construct Emergence Tracing is a change in the level of description. Feature attribution and circuit tracing identify components and pathways, while CET characterizes agreement between a construct's stipulated dimensions and the network organization of its item representations. These approaches address complementary questions. CET can identify states suitable for more focused probing or intervention, but network recovery alone does not identify the mechanism responsible for a model's behavior.

Item Generation, Embeddings, and Construct Recovery

Recent developments in Generative Psychometrics have made it possible to move from a construct definition to a large pool of candidate items in a relatively short period of time. Structured prompting guides item generation, while embedding-based analyses help identify semantic redundancy, unstable item membership, and departures from the intended dimensional structure before collecting human responses (Garrido et al., 2026; Lara L. Russell-Lasalandra et al., 2024; Lara L. Russell-Lasalandra & Golino, 2026). These applications change what can be evaluated during the early stages of scale development. However, the semantic organization of item statements and the organization of individual differences in responses are different objects of measurement. An embedding can indicate that two statements express related content without establishing that people respond to them in the same way. Embedding-based evaluation therefore provides evidence about an item pool that must subsequently be connected to response-based evidence of validity.

The choice of method used to analyze these representations is consequential. Garrido et al. (2026) compared PCA-based workflows with EGA across three item-generating LLMs and two embedding models. PCA combined with either Kaiser's rule or parallel analysis overestimated dimensionality, while EGA more accurately recovered the stipulated structure. Network-based item filtering only partly reduced the PCA workflows' overextraction, and an application to the Multidimensional Schizotypy Scale showed a similar discrepancy against a structure estimated from human responses. These findings concern the evaluated extraction and retention workflows; they do not establish that every application of PCA to embeddings is inappropriate.

The limitation is particularly relevant when a component solution is interpreted psychologically. PCA identifies linear combinations that account for total variance, but variance in a representation need not correspond to the distinctions defining a construct. Shared wording, redundancy, and narrow semantic differences can contribute to the

component solution alongside the intended attributes. Conversely, a distinction relevant to a construct may occupy a direction with relatively little variance and be lost when only leading components are retained. In other words, reducing the dimensionality of a representation and recovering the dimensionality of a psychological construct are different tasks. CET addresses the latter by evaluating agreement with an explicit item partition across model depth. The present study does not directly compare CET with PCA, and such a comparison would require matched representations and evaluation criteria. Nevertheless, its results reinforce the need to distinguish properties of the embedding space from substantive claims about the construct represented within it.

A Developmental View of Construct Emergence

The developmental perspective provides a psychological interpretation of why the organization of a representation matters beyond the presence of individual features. Following the synthesis of Piagetian, Vygotskian, and neo-Piagetian accounts in H. Golino (2025), abstraction can be understood as the coordination of existing representations into structures that express more general relationships. In Piaget's account, reflective abstraction involves reorganizing prior coordinations at a new level. The connection to CET is at this relational level: statements with different surface forms may come to be organized according to shared attributes as their representations pass through successive transformations. The resulting trajectory makes this organization measurable without assuming that a particular transformer block corresponds to a stage of childhood development.

Neo-Piagetian theories help refine this connection. Pascual-Leone emphasized the mental-attentional resources needed to coordinate schemes, while Case, Fischer, and Demetriou developed accounts of the integration and organization of increasingly complex representations, as reviewed by H. Golino (2025). From this perspective, the question is not simply how much information a system contains, but how that information can be

coordinated. The increasing average recovery across Qwen2.5 sizes is compatible with this question, although parameter count is not a measure of human mental-attentional capacity. Moreover, the different locations of the NMI and TEFI optima show that agreement with construct labels and entropy-based organization need not develop together. A developmental interpretation must accommodate these differences rather than impose a single sequence of increasing maturity on every model.

Vygotsky adds a complementary point: the organization of meaning has a social and cultural history. Language becomes a psychological instrument through socially mediated activity, and abstraction depends on the systems of meaning available in that activity (as discussed in H. Golino, 2025). LLMs learn from language produced within such systems, so the construct structure recovered by CET may partly reflect culturally shared ways of describing personality. This connection does not equate training on text with a child's participation in dialogue. It does, however, motivate comparisons across languages, cultural contexts, and alternative formulations of the same attributes. A structure recovered from one item pool should not automatically be treated as a culturally invariant organization of the construct.

The developmental view is therefore a framework for asking how representations become differentiated, coordinated, and reorganized. In the present analysis, these changes occur across processing depth in an already trained model, not across learning or maturation over time. Terminal declines also show that the trajectory need not culminate in the state with the greatest construct recovery. Likewise, a compressible network is not, by itself, evidence of a more abstract concept; its organization must be considered alongside coherence and, ultimately, generalization to new expressions and contexts. CET provides descriptive measurements with which to investigate these psychological questions, while tests across training stages and tasks would be needed to evaluate a stronger account of representational development.

Coherence, Organization, and Maturity

The relative timing of NMI and TEFI optima does not support a single sequence in which structure always precedes meaning, or vice versa. Using NMI and TEFI averaged over embedding dimensions, GPT-2's best entropy fit occurs at state 8 and its highest NMI at state 11. The ordering is reversed in Phi-4-mini (NMI at block 17, TEFI at block 25), Qwen2.5-32B (35 versus 51), and Muse-Glimmer (20 versus 38). TinyLlama reaches both optima at block 18. These patterns establish variation in the relationship between the two indices without assigning a cognitive process to each state.

The distinct metric orderings also explain why RM should accompany rather than replace its components. Qwen3.8 and Muse-Glimmer have nearly identical overall mean NMI, but their mean TEFI values are -11.01 and -5.85, respectively, and their terminal profiles differ. Compression-based KC contributes information about the topology of the estimated network, but its constant edge-presence encoding does not measure the precision or storage cost of the model's neural representation. A composite optimum is therefore a weighted compromise among descriptive properties, not a unique developmental endpoint.

Terminal Recovery and Representation Selection

In GPT-OSS, mean NMI falls from 0.742 in the penultimate block to zero in the final state, while in Qwen2.5-32B it falls from 0.784 to 0.025. In contrast, nine models retain substantial agreement at the output. Choosing the final state would therefore produce sharply different conclusions about construct recovery than searching intermediate states.

These findings do not show that information has been destroyed. A change in output normalization or representation geometry can affect correlation structure, network estimation, and community detection even when information remains recoverable by another method. Controlled comparisons of pre-normalization block outputs, returned hidden states, and alternative probes are needed to distinguish an extraction effect from a

more general change in construct accessibility.

Implications for Model Comparison and Interpretability

The within-family Qwen2.5 ordering suggests that average construct recovery may be a useful complement to evaluations that focus on the best available layer. Its interpretation remains conditional on the sampled construct and grid: a model with a long low-recovery segment can have a modest full-grid average despite a high selected peak, as illustrated by TinyLlama. Conversely, a model can score highly on average and still lose agreement at the final state, as illustrated by Qwen2.5-32B and Muse-Glimmer.

For representation selection, CET provides a surface of analysis, or a developmental perspective on construct representation, instead of a single ranking. This allows investigators to distinguish an isolated configuration from a broader region of recovery and to evaluate whether the preferred state changes with the criterion.

CET also differs from linear probing and representational similarity measures such as CKA (Kornblith et al., 2019). A probe evaluates decodability under a specified prediction model, whereas CET evaluates agreement between an estimated community structure and construct labels. CKA compares representations with one another instead of testing a particular multidimensional construct. Agreement among these approaches would strengthen interpretation, while disagreement could reveal sensitivity to geometry, estimation, or the level at which a construct is expressed.

Implications for Safety and Alignment

A construct-centered measure could help evaluate whether an intervention changes the depth profile of a safety-relevant representation. The present results establish a descriptive method for such comparisons. Applications in the domains of deception, harmful intent, or alignment require independently justified item pools, matched negative

controls, and links between representational changes and observable behavior.

Limitations and Future Work

The model conditions differ in parameter count, training, tokenizer, attention design, and checkpoint representation. Multiple Qwen2.5 sizes are members of the same family, so the fourteen conditions are not fourteen independent architectural samples. GPT-OSS is evaluated after dequantization from MXFP4, and Qwen3.8 is identified by local GGUF metadata specifying a Qwen3.5 text architecture. Conclusions apply to the evaluated models, therefore broader claims about release families or quantization effects require especifically matched comparisons, which is beyond of the scope of this paper.

The analysis uses one construct and a single item-pool source. Lexical regularities may contribute to recoverable attribute structure, and agreement with stipulated labels is not proof of semantic understanding. Replication with independent item wording, matched nonsensical constructs, label-permutation controls, and other psychological domains is needed to distinguish semantic organization from item-specific regularities.

The hidden-unit index is treated as a pseudo-temporal ordering for delay embedding, although hidden coordinates do not form an observed physical time series. The theoretical reconstruction argument specifies sufficient conditions for separation preservation, but it does not establish that every model representation satisfies those conditions. Sensitivity to coordinate ordering, embedding ranges, community-detection methods, and activation normalization should be examined directly.

Finally, RM depends on the chosen weights and pooled empirical bounds, while peaks depend on the searched grid. The onset and terminal thresholds are descriptive conventions rather than hypothesis tests. Reporting the component metrics and continuous profiles makes these choices visible, but causal explanations of emergence and terminal decline will require activation interventions, matched checkpoint comparisons, and

convergent measures of information recovery.

Conclusion

Construct Emergence Tracing provides a construct-centered methodology to investigate the internal language-model representations by tracing network recovery across states and embedding dimensions. In the fourteen models evaluated with five items per attribute, the results show substantial differences in recovery onset, mean coherence, composite optima, and terminal retention. Increasing overall mean NMI across Qwen2.5 sizes coexists with a strong final-state decline in the largest model, demonstrating that average recovery and output-layer preservation are distinct properties.

The principal implication is methodological: neither a single selected peak nor the final state adequately summarizes construct representation across all models. CET makes these differences measurable and identifies locations for subsequent probing and causal analysis, providing a psychology-based, measurement-oriented framework to understand how psychological constructs are represented inside large language models, with a specific developmental arc being quntified within the model layers. In this role, network psychometrics complements mechanistic interpretability by connecting internal representational structure to an explicitly defined psychological construct.

References

Alain, G., & Bengio, Y. (2017). Understanding intermediate layers using linear classifier probes. *5th International Conference on Learning Representations, Workshop Track*.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, *48*(1), 207–219.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Boker, S. M., Deboek, P. R., Edler, C., & Keel, P. (2010). Generalized local linear approximation of derivatives from time series. In S. M. Chow, E. Ferrer, & F. Hsieh (Eds.), *The notre dame series on quantitative methodology. Statistical methods for modeling human dynamics: An interdisciplinary dialogue* (pp. 161–178). Routledge/Taylor & Francis Group.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., Arx, S. von, … Liang, P. (2021). *On the opportunities and risks of foundation models*. https://doi.org/10.48550/arXiv.2108.07258

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). *Language models are few-shot learners*. https://doi.org/10.48550/arXiv.2005.14165

Cai, X., Huang, J., Bian, Y., & Church, K. (2021). Isotropy in the contextual embedding space: Clusters and manifolds. *Proceedings of the 9th International Conference on Learning Representations*.

Christensen, A. P., Garrido, L. E., Guerra-Peña, K., & Golino, H. (2024). Comparing community detection algorithms in psychometric networks: A monte carlo simulation. *Behavior Research Methods*, *56*(3), 1485–1505.

Christensen, A. P., & Golino, H. F. (2021). Estimating the stability of psychological dimensions via bootstrap exploratory graph analysis: A monte carlo simulation and

tutorial. *Psych*, *3*(3), 479–500.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin*, *52*(4), 281–302. https://doi.org/10.1037/h0040957

Deboeck, P. R., Montpetit, M. A., Bergeman, C., & Boker, S. M. (2009). Using derivative estimates to describe intraindividual variability at multiple time scales. *Psychological Methods*, *14*(4), 367–386. https://doi.org/http://dx.doi.org/10.1037/a0016622

Epskamp, S., Maris, G., Waldorp, L. J., & Borsboom, D. (2018). Network psychometrics. In P. Irwing, T. Booth, & D. J. Hughes (Eds.), *The wiley handbook of psychometric testing: A multidisciplinary reference on survey, scale and test development* (pp. 953–986). John Wiley & Sons. https://doi.org/10.1002/9781118489772.ch30

Epskamp, S., Waldorp, L. J., Mõttus, R., & Borsboom, D. (2018). The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, *53*(4), 453–480. https://doi.org/10.1080/00273171.2018.1454823

Feder, A., Keith, K. A., Manzoor, E., Pryzant, R., Sridhar, D., Wood-Doughty, Z., et al. (2022). Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *arXiv Preprint arXiv:2206.08353*.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441. https://doi.org/10.1093/biostatistics/kxm045

Garrido, L. E., Russell-Lasalandra, L. L., Rodríguez-Montoya, C., & Golino, H. F. (2026). Estimating dimensional structure in generative psychometrics: Comparing PCA and network methods using large language model item embeddings. *PsyArXiv*. https://doi.org/10.31234/osf.io/2s7pw_v2

Golino, H. (2025). *What i learned with john: On the depth of language and how to measure it with large language models and algorithm (kolmogorov) complexity*. https://doi.org/10.31234/osf.io/b92n5_v4

Golino, H. (2026). Optimizing the landscape of LLM embeddings with dynamic

exploratory graph analysis for generative psychometrics: A Monte Carlo study. *arXiv Preprint arXiv:2601.17010*.

Golino, H. F., Christensen, A. P., & Garrido, L. E. (2022). Exploratory graph analysis in context. *Revista Psicologia: Teoria e Prática*, *24*(3), ePTPPA14197–ePTPPA14197.

Golino, H. F., Christensen, A. P., Moulder, R., Kim, S., & Boker, S. M. (2022). Modeling latent topics in social media using dynamic exploratory graph analysis: The case of the right-wing and left-wing trolls in the 2016 US elections. *Psychometrika*, *87*(1), 156–187.

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS One*, *12*(6), e0174035. https://doi.org/10.1371/journal.pone.0174035

Golino, H. F., Jiménez, M., Garrido, L. E., & Christensen, A. P. (2024). Generalized total entropy fit index: A new fit index to compare bifactor and correlated factor structures in SEM and network psychometrics. *PsyArXiv*. Retrieved from https://doi.org/10.31234/osf.io/5g3hb

Golino, H. F., Moulder, R., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., ... Boker, S. M. (2021). Entropy fit indices: New fit measures for assessing the structure and dimensionality of multiple latent variables. *Multivariate Behavioral Research*, *56*(6), 874–902.

Golino, H. F., Shi, D., Garrido, L. E., Christensen, A. P., Nieto, M. D., Sadana, R., ... Martinez-Molina, A. (2020). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*(3), 292–230. https://doi.org/10.1037/met0000255

Golino, H., Nesselroade, J., & Christensen, A. P. (2025). Toward a psychology of individuals: The ergodicity information index and a bottom-up approach for finding generalizations. *Multivariate Behavioral Research*, *60*(3), 528–555. https://doi.org/https://doi.org/10.1080/00273171.2025.2454901

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does BERT learn about the structure of language? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3651–3657. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1356

Kolmogorov, A. N. (1965). Three approaches to the definition of the concept "quantity of information." *Problemy Peredachi Informatsii*, *1*(1), 3–11. Retrieved from https://www.mathnet.ru/eng/ppi68

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. *Proceedings of the 36th International Conference on Machine Learning*, 3519–3529. PMLR.

Kriegeskorte, N., Mur, M., & Bandettini, P. (2008). Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, *2*, 4. https://doi.org/10.3389/neuro.06.004.2008

Lindsey, J., Gurnee, W., Ameisen, E., et al. (2025). *On the biology of a large language model*. https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*, 4765–4774.

Massara, G. P., Di Matteo, T., & Aste, T. (2016). Network filtering for big data: Triangulated maximally filtered graph. *Journal of Complex Networks*, *5*(2), 161–178. https://doi.org/10.1093/comnet/cnw015

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom in: An introduction to circuits. *Distill*, *5*(3), e00024.

Pincus, A. L., & Lukowitsky, M. R. (2010). Pathological narcissism and narcissistic personality disorder. *Annual Review of Clinical Psychology*, *6*(1), 421–446.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI. Retrieved from OpenAI website: https://cdn.openai.com/better-language-models/language-models.pdf

Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, *30*.

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. https://doi.org/10.1162/tacl_a_00349

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. https://doi.org/10.1038/323533a0

Russell-Lasalandra, Lara L., Christensen, A. P., & Golino, H. (2024). *Generative psychometrics via AI-GENIE: Automatic item generation and validation via network-integrated evaluation*. https://doi.org/10.31234/osf.io/fgbj4

Russell-Lasalandra, Lara L., & Golino, H. (2026). Prompt engineering for scale development in generative psychometrics. *arXiv Preprint arXiv:2603.15909*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., … Guo, D. (2024). *DeepSeekMath: Pushing the limits of mathematical reasoning in open language models*. https://doi.org/10.48550/arXiv.2402.03300

Spearman, C. (1904). "General intelligence," objectively determined and measured. *The American Journal of Psychology*, *15*(2), 201–292. https://doi.org/10.2307/1412107

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*, 583–617. Retrieved from https://www.jmlr.org/papers/v3/strehl02a.html

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *Proceedings of the 34th International Conference on Machine Learning*, 3319–3328.

Takens, F. (1981). Detecting strange attractors in turbulence. In *Lecture notes in mathematics (vol. 898)* (pp. 366–381). Springer. https://doi.org/10.1007/BFb0091924

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, 4593–4601. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1452

Tomašević, A., Golino, H. F., & Christensen, A. P. (2024). Decoding emotion dynamics in videos using dynamic exploratory graph analysis and zero-shot image classification: A simulation and tutorial using the transforEmotion r package. *PsyArXiv*. https://doi.org/10.31234/osf.io/hf3g7

Vig, J. (2019). A multiscale visualization of attention in the transformer model. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 37–42. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-3007

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, *31*, 841–887.

Whitney, H. (1936). Differentiable manifolds. *The Annals of Mathematics*, *37*(3), 645–680. https://doi.org/10.2307/1968482

Appendix A

Proof of the Semantic Manifold Reconstruction Theorem

Lemma 1 (Bi-Lipschitz on Compact Sets)**.** *If $\Phi : \mathcal{M} \to \mathbb{R}^{2m+1}$ is a C^1 embedding and $\mathcal{M}$ is compact, then there exist constants $0 < a \leq b < \infty$ such that for all $x, y \in \mathcal{M}$,*

$$a\, d(x, y) \leq \|\Phi(x) - \Phi(y)\| \leq b\, d(x, y),$$

where d is any Riemannian distance on $\mathcal{M}$ equivalent to the ambient metric.

Proof sketch. Compactness and C^1 regularity ensure the Jacobian of Φ has bounded operator norm (giving b) and its smallest singular value is bounded away from zero (giving $a > 0$, following from injectivity and continuity on a compact domain). □

Lemma 2 (Separation Preservation)**.** *Let $\{C_1, \ldots, C_r\}$ be a partition of $\mathcal{M}$ into closed subsets with positive separation margin $\Delta = \min_{i \neq j} \inf\{d(x, y) : x \in C_i, y \in C_j\} > 0$. Under the bi-Lipschitz Φ above,*

$$\min_{i \neq j} \inf\{\|\Phi(x) - \Phi(y)\| : x \in C_i, y \in C_j\} \geq a\, \Delta > 0.$$

Consequence. Any clustering procedure on $\Phi(\mathcal{M})$ that is stable under Lipschitz distortions and consistent for Δ-separated clusters will recover the same partition (up to label permutation), provided finite-sample noise is sufficiently small.

Appendix B

Implementation Details

Activation and network analysis. Token-level hidden states are averaged within each item. Hidden-unit features are standardized across stacked item-by-state observations, after which each state is analyzed separately. Unit-lag embeddings use dimensions 3–200; first derivatives are estimated through GLLA. EBICglasso estimates the network and Louvain detects communities. NMI evaluates agreement with the five attribute labels.

Complexity encoding. Network entries are thresholded to edge presence. Each retained edge is encoded by its coordinate pair and a constant value of $2^1 = 2$, serialized, randomly reordered, and compressed with gzip. The mean compressed length over 1,000 reorderings is KC. This encoding describes topology compressibility and does not preserve numerical edge weights.

Aggregation. Arithmetic means are used for all models. Empirical TEFI and KC bounds are pooled across every five-item observation before computing RM. Since the transformation is affine, the mean RM surface is obtained from the corresponding component means. Per-state RM trajectories maximize this mean surface over embedding dimensions; mean-NMI trajectories instead average over embedding dimensions.

Reproducibility materials. The analysis materials provide checkpoint identifiers, available software records, input checksums, validation results, cell and layer summaries, and scripts for rebuilding the figures and tables. No language-model inference is required to regenerate the reported analyses from the saved simulation results.

Appendix C

CET Pipeline Pseudocode

Algorithm 1 Construct Emergence Tracing (CET)

Require: Model $\mathcal{M}$ with L layers; item pool $\mathcal{P}$ with r attributes; $k = 5$ items per attribute; N_{iter} iterations; embedding range $[n_{\min}, n_{\max}]$

Ensure: Layer-wise metrics $\{\text{NMI}, \text{TEFI}, \text{KC}\}_{\ell,n}$ and RM

for $t = 1$ to N_{iter} **do**
  Sample k items per attribute from $\mathcal{P} \rightarrow$ item set $\mathcal{I}_t$, labels $\mathbf{y}_t$
  for each item $x \in \mathcal{I}_t$ **do**
    Extract hidden states $\mathbf{h}_\ell(x) \in \mathbb{R}^d$ for each layer ℓ
  end for
  Stack all item-by-layer vectors; standardize each hidden unit across the stack
  for $\ell = 1$ to L **do**
    Select the standardized rows for layer ℓ
    for $n = n_{\min}$ to $n_{\max}$ **do**
      Build time-delay embeddings with n dimensions
      Estimate derivatives via GLLA
      Estimate network via EBICglasso
      Detect communities $\hat{\mathbf{c}}$ via Louvain
      Compute NMI$(\hat{\mathbf{c}}, \mathbf{y}_t)$, TEFI, KC
    end for
  end for
end for
Aggregate NMI, TEFI, and KC using arithmetic means across iterations
Compute RM using TEFI and KC bounds pooled across all models

Appendix D

Component-Metric Landscapes

The following figures report the component metrics on their observed scales using the same five-item conditions and arithmetic aggregation as the main results.

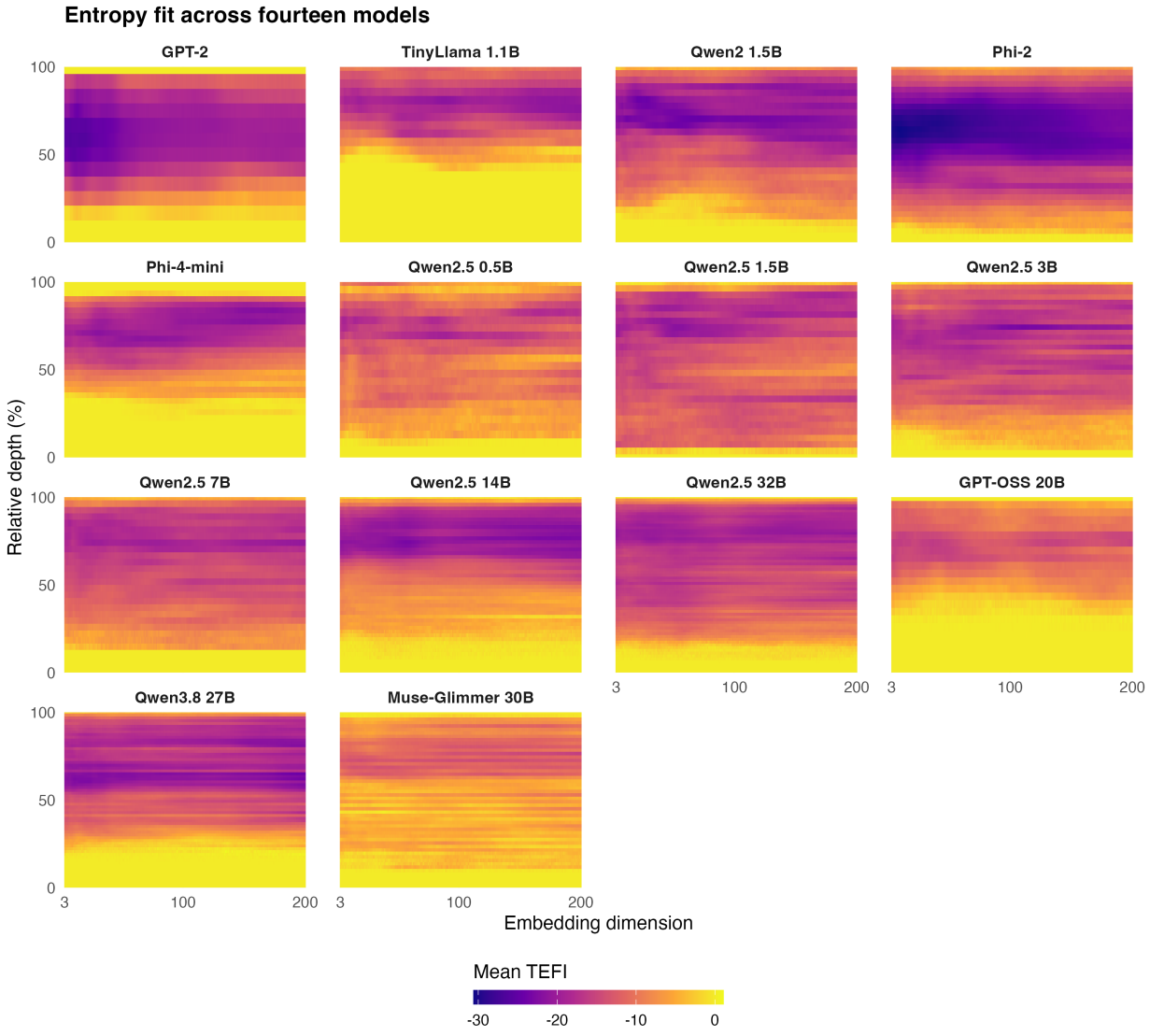


Figure D1. Mean TEFI across fourteen models. All panels use one color scale and the same relative-depth convention. Lower values indicate lower entropy fit under the implemented network analysis.

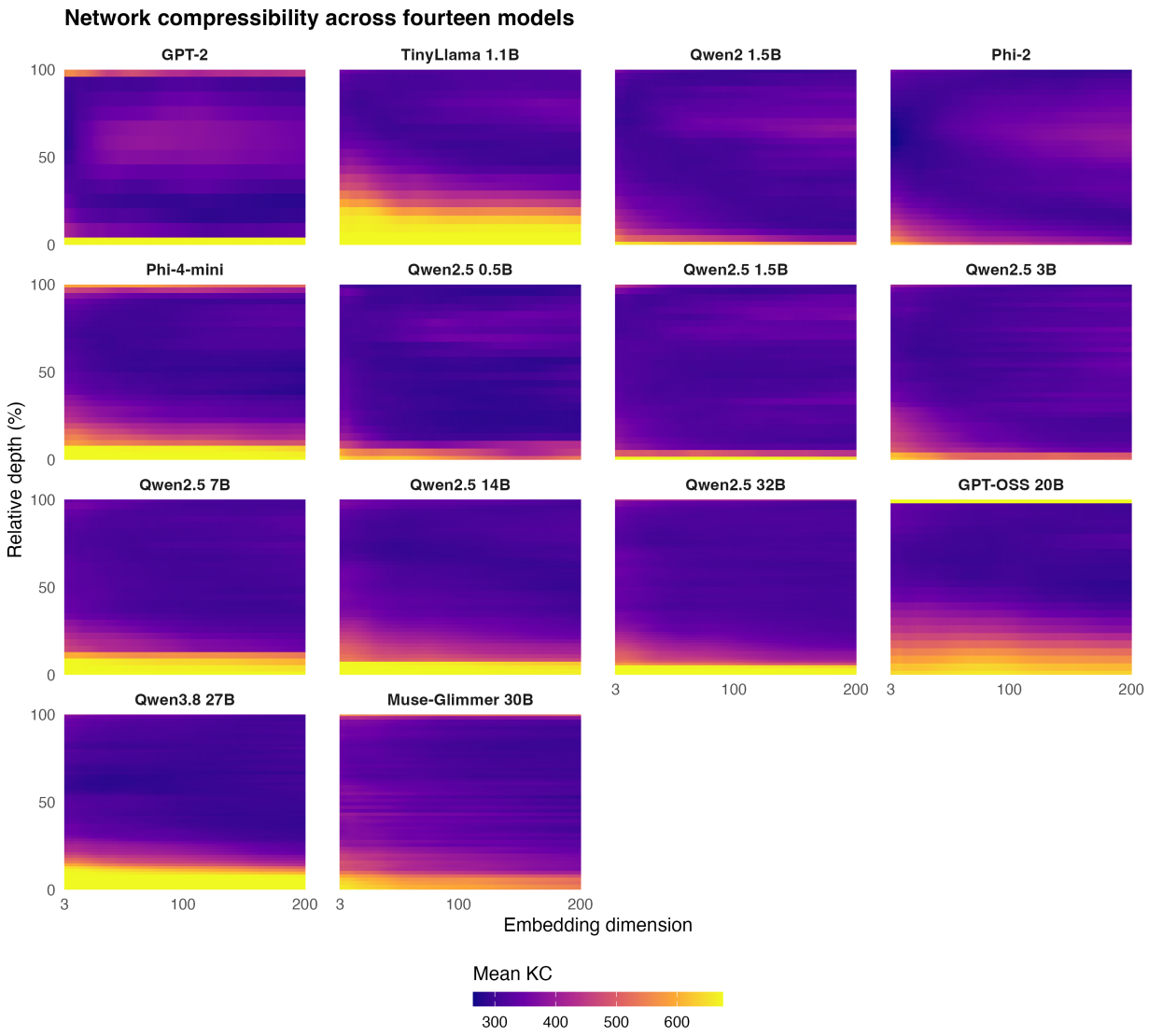


Figure D2. Mean KC across fourteen models. Colors represent gzip-compressed length for the topology encoding. Lower values indicate more compressible network encodings, not a measured reduction in neural storage requirements.

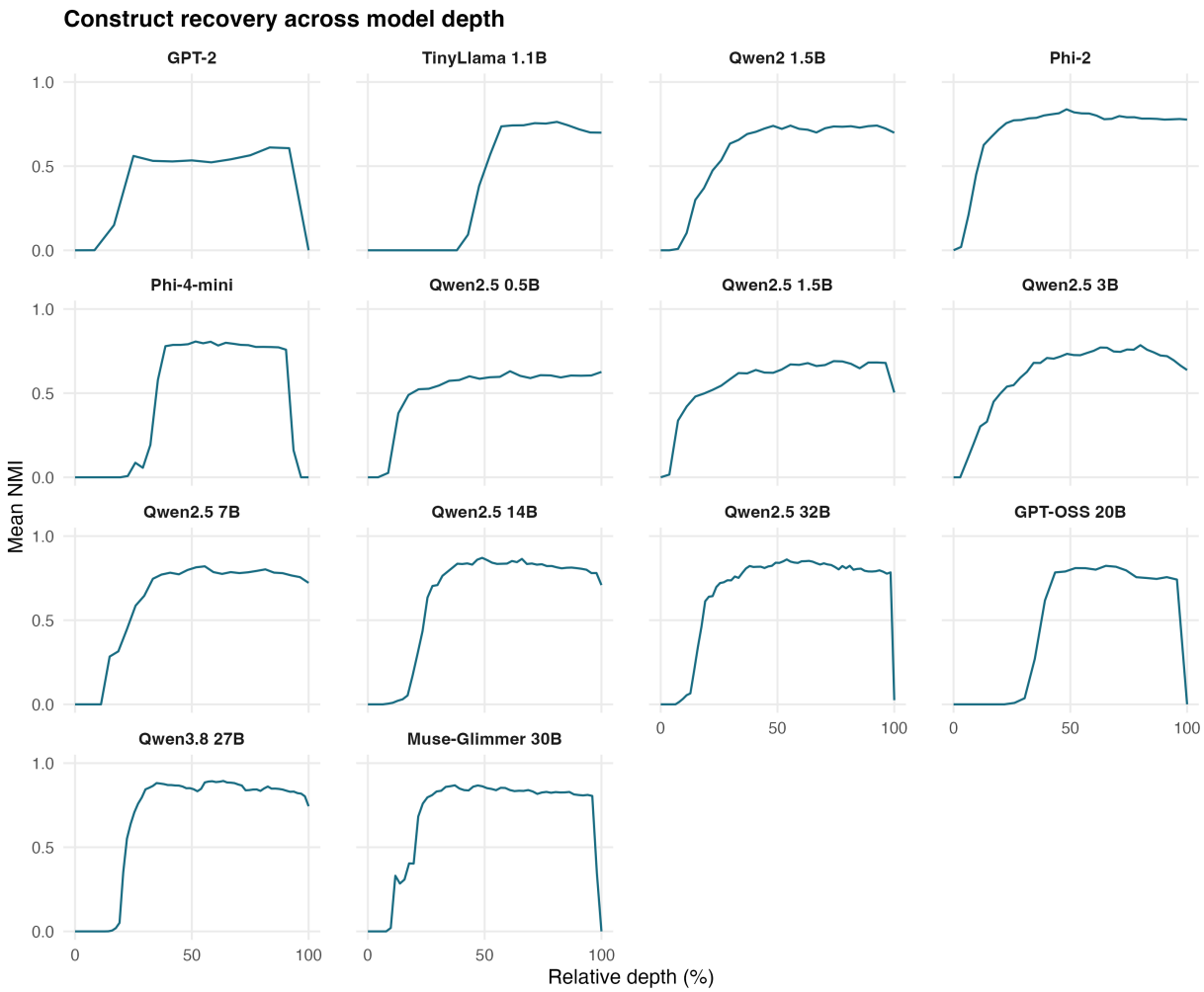


Figure D3. Mean NMI across relative depth for all fourteen models. Curves average over embedding dimensions and iterations, complementing the per-state maxima used for RM trajectories.