ATTRIBUTING REPRESENTATIONAL GEOMETRY TO DATA, ARCHITECTURE, AND TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Any observable property of a large language model (LLM) stems from three sources: the data it reads, the model architecture (the network and tokenizer), and the training process. We argue that training produces little of the representational structure we can measure in an LLM. Instead, most of that structure comes from the data and the architecture. Training adds one thing: the model's ability to tell what a text says from how it is written, and this ability appears early in training. Existing evaluation methods such as scaling laws and ablations compare trained models with one another and do not separate the contributions of these three sources, so claims about what a model has learned, and comparisons between models based on their representations, may rest on properties that training did not produce. To identify the contributions of these three sources, we propose DATS (Data–Architecture–Training Separation), a framework that measures the same documents in three ways and attributes each property to its source: a property already shown by bag-of-words (BoW) representations is attributed to the data, a property first shown by untrained networks to the architecture, and a property shown only by trained models to training. Applied to the geometry of document representations across four model families, DATS finds that (1) the ordering of document types by intrinsic dimension is already present in the data; (2) the similarity between two ways of writing the same text is already present in the word counts, and the tokenizer decides whether an untrained network keeps it; (3) only trained models place two ways of writing the same text together, and in Pythia they do so within the first few percent of training, while the shape of the representation keeps changing to the end.

1 INTRODUCTION

To evaluate, improve, and explain a large language model (LLM), we need to know what it learned. Evaluating a model by its outputs, such as downstream task scores, tells us whether it performs well, but not what drives that performance, what it learned, or when it learned it. Researchers have therefore turned to the model's internals: the vectors a model computes for a piece of text, that is, its representations.

Representations are analyzed in two ways: by examining what a representation vector contains, as probing does (Belinkov, 2022), or by examining the geometry of the representations, which vectors lie near which and how spread out they are. We analyze geometry for two reasons. First, geometry can be compared across models with different vector dimensions, whereas the contents of single vectors cannot. Second, claims about what models learn, such as convergence across models (Huh et al., 2024) and a high-dimensional phase along depth (Cheng et al., 2025), are themselves claims about geometry. In this paper, each vector represents a sentence, and we study the geometry of a document's sentence vectors as a whole.

However, analyzing the representations alone does not tell us which part of their geometry the model learned in training. A trained LLM is produced by fitting a function class with an input encoding to a dataset by a training procedure. Any property of its representations therefore traces back to three sources: the data, which is what the model reads; the architecture, including the network structure and the tokenizer, which determines what the model can represent and how it splits text into tokens; and the training, including the objective, optimizer, training steps, and post-training, which

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.

is everything done to the model during fitting. If we cannot attribute a property of the representations to its source, we cannot tell whether that property reflects the data, the architecture, or the training, and therefore which of the three to change.

Existing methods for attributing a model's properties do not separate the sources. Scaling laws compare trained models of different sizes, and ablations compare trained models with a component removed. These methods analyze and compare only trained models, without an untrained reference or a representation that involves no model at all. Without the separation, the contribution of the other two sources can be attributed to training. Convergence between models is read as a discovery about the real world (Huh et al., 2024); when a model places a text near its translation or another way of writing the same text, the proximity is credited to training; and geometry is used to track training progress without knowing which source produced it.

To separate the three sources, we propose DATS (Data–Architecture–Training Separation), a framework that represents the same documents in three ways: bag-of-words (BoW) representations, which reflect only the data; representations from untrained networks, whose weights, including the token embeddings, are randomly initialized and which therefore reflect the data and the architecture; and representations from trained models, which reflect all three. A property that already appears in the BoW representation is attributed to the data, one that first appears in the untrained network to the architecture, and one that appears only in the trained model to training.

We measure two aspects of geometry: shape, that is, which sentences in a document lie near which and how spread out they are, measured by intrinsic dimension (ID) and representational similarity analysis (RSA); and identity, whether two ways of writing the same sentence end up close together, measured by nearest-neighbor recovery (hit@k). The two measurements are formally independent (Appendix C). We apply DATS to four model families (Qwen2.5, Llama-3.1, Mistral, Mamba) on five kinds of English documents and a set of Chinese documents written in both characters and pinyin, and, to see when the part attributed to training appears, we repeat the measurements on the public intermediate checkpoints of Pythia-410M. We find that most of the document-level geometry we measure is already present before training, in all four families; what training adds is the ability to tell what a text says from how it is written (invariance to the way of writing, tested here on case, script, and romanization), and it adds this early. In Mistral-7B, for example, training leaves the mid-depth shape across the two ways of writing unchanged and raises the recovery of a sentence across them from near chance to near-perfect.

Our contributions are threefold:

1. We introduce DATS, a framework that attributes each property of a representation to the data, the architecture, or the training, without retraining.
2. We attribute shape to the data and the architecture: the ordering of document types by ID is already in the BoW representation, and the similarity between two ways of writing the same text is already in the word counts, kept or lost by the tokenizer in an untrained network.
3. We locate identity in training: only trained models identify the same sentence across two ways of writing it. Pythia does so within the first few percent of training, while the shape continues to change throughout training.

2 Related Work

Representations as evidence of learning. A property of a model's representations is routinely read as evidence about what it learned. Probing classifiers test whether a property is decodable from hidden states (Belinkov, 2022), with control tasks added because a probe's accuracy alone does not establish that the property is represented (Hewitt & Liang, 2019). Circular features for days and months (Engels et al., 2025), a band of intermediate layers that behaves like a global workspace (Gurnee et al., 2026), and a phase of high intrinsic dimension whose representations are the first to transfer to downstream tasks (Cheng et al., 2025) are all read this way. The representations of models trained on different modalities become more alike as the models grow and improve, which the Platonic Representation Hypothesis reads as convergence toward a shared model of reality (Huh et al., 2024). Two recent results qualify the last reading: after calibrating similarity metrics against width and depth confounds, only agreement on local neighborhoods remains (Gröger et al., 2026), and convergence holds for how models encode their input rather than for what follows

(Usama & Chang, 2026). Together these results say that observing a property in a trained representation does not show that training produced it. DATS makes the check explicit: it compares the trained representation with the data itself and with an untrained network of the same architecture, so the question changes from whether models agree to whether what they agree on was ever learned.

Properties present before training. Randomly initialized networks, which we call untrained networks, show properties before any learning: frozen random LSTMs do well on syntactic probing tasks (Zhang & Bowman, 2018), random sentence encoders rival trained ones on classification (Wieting & Kiela, 2019), and random transformers in which only the embedding layers are trained can perform modular arithmetic and associative recall (Zhong & Andreas, 2024). A comparison restricted to trained models will therefore credit training with what the architecture provides. DATS uses the untrained network as an explicit architecture control and, unlike work that studies what untrained networks can do, applies the same measurements to the same documents before and after training.

Attribution by retraining. When a property of a model is attributed to its data, its architecture, or its training, the attribution is made by changing that ingredient, retraining, and comparing the trained models that result: scaling laws attribute loss to parameter count and data volume (Kaplan et al., 2020; Hoffmann et al., 2022), controlled synthetic pretraining attributes reasoning capabilities to architectural choices (Allen-Zhu, 2025), and retraining with different tokenizers attributes part of the multilingual–monolingual gap to the tokenizer (Rust et al., 2021). Every comparison requires a newly trained model. DATS asks at which stage a property first becomes observable, in the data, in the untrained architecture, or only after training, and answers this without training any model.

Geometry from data statistics. Several results trace a geometric regularity in representations to a property of the text: a translation symmetry in co-occurrence statistics produces the circular geometry observed for months and years (Karkada et al., 2026); the periodic structure of number embeddings is already in the corpus token-frequency distribution (Fu et al., 2026); and intrinsic-dimension estimates change regime with the lexical diversity of the dataset (Osman et al., 2026). Each of these traces one regularity to one statistic. DATS traces the geometry of whole documents at once, on the same documents and with the same instruments as the trained model, using the measurements standard in this literature: TWO-NN intrinsic dimension (Facco et al., 2017; Valeriani et al., 2023) and representational similarity (Kornblith et al., 2019), after the standardization that anisotropic representations require (Ethayarajh, 2019; Rudman & Eickhoff, 2024).

3 DATS

DATS represents the same set of documents in three ways and attributes each property of the trained model's representation geometry to the first of the three in which it appears (Figure 1); the attribution is by first appearance, not a causal decomposition. This section defines the objects we measure (§3.1), the three ways of representing a document and the attribution rule (§3.2), and the two measurements of geometry (§3.3).

3.1 Preliminaries

Models, layers, and representations. A language model with L layers reads a text as a sequence of tokens and computes, at every layer $\ell = 0, 1, \ldots, L$, one hidden-state vector per token; we call such a vector a *representation*. Layer 0 is the token-embedding layer, which precedes any contextual processing and is excluded from all layerwise summaries. We call ℓ the *depth* of a layer and ℓ/L its *relative depth*, which lets us compare models with different L.

Sentences and documents. Each vector in this paper represents one sentence. We encode a sentence s on its own and read out one vector per layer, $x^{(\ell)}(s) \in \mathbb{R}^H$, where H is the model's hidden width (§4.2 states which token's hidden state is read out). A document D with n sentences $s_1, \ldots, s_n$ therefore gives, at each layer ℓ, a set of n vectors $X^{(\ell)}(D) = \{x^{(\ell)}(s_1), \ldots, x^{(\ell)}(s_n)\} \subset \mathbb{R}^H$, which we call the document's *point cloud* at layer ℓ. We use sentences because a document has too few paragraphs to form a usable cloud, and because tokens cannot be paired across two ways of writing the same text while sentences can.

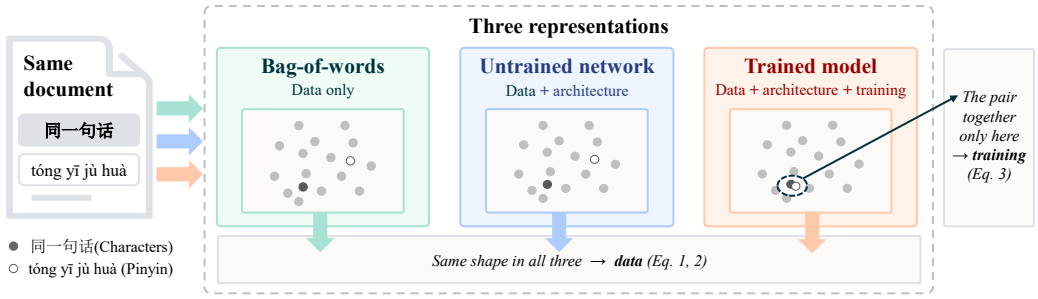

Figure 1: DATS. The same document, here one Chinese sentence in characters (filled) and in pinyin (hollow), is represented in three ways that reflect increasingly many sources, and measured with the same instruments. The three clouds have the same shape, so shape is attributed to the data; only in the trained model does the pinyin form of the sentence land next to its character form, so identity is attributed to training.

Intrinsic dimension. The intrinsic dimension (ID) of a point cloud is the number of directions the points effectively occupy, usually far smaller than H. We estimate it with the TWO-NN estimator (Facco et al., 2017). For each point i, let $r_1(i) \le r_2(i)$ be its Euclidean distances to its nearest and second-nearest neighbors in the cloud, and let $\mu_i = r_2(i)/r_1(i)$. For a cloud on a d-dimensional manifold with locally uniform density, the maximum-likelihood estimate of d is

$$\widehat{\mathrm{ID}}(X) \;=\; \frac{n}{\sum_{i=1}^{n} \log \mu_i}, \tag{1}$$

computed after removing duplicate points and, following Facco et al. (2017), discarding the 10% largest ratios. The estimator uses only the two nearest-neighbor distances of each point, so it does not depend on H, which differs across models. Computing Eq. 1 at every layer gives a curve $\ell \mapsto \mathrm{ID}(X^{(\ell)}(D))$, which we summarize by its *peak*, $\max_{\ell \ge 1} \mathrm{ID}(X^{(\ell)}(D))$: the largest dimensionality a document's cloud attains across depth, independent of where a given architecture places its high-dimensional layers. A fixed relative depth gives the same ordering (Appendix B).

Similarity structure and standardization. The second kind of measurement compares two clouds of the same n sentences. Before any similarity is computed, we standardize each cloud on its own: each of the H coordinates is shifted to mean 0 and variance 1 across the n points, and each vector is then normalized to unit length. We write $\hat{x}_i$ for the standardized vector. This step is necessary because several trained models carry a single coordinate with more than 99% of the variance, which otherwise dominates every cosine similarity (Appendix A). The similarity structure of a cloud X is its $n \times n$ matrix of cosine similarities, $S^X_{ij} = \langle \hat{x}_i, \hat{x}_j \rangle$. Representational similarity analysis (RSA) compares the similarity structures of two clouds X and Y of the same sentences by the Spearman rank correlation ρ_{S} of their upper triangles,

$$\mathrm{RSA}(X, Y) \;=\; \rho_{\mathrm{S}}\big(\{S^X_{ij}\}_{i<j},\ \{S^Y_{ij}\}_{i<j}\big)\,, \tag{2}$$

which equals 1 when the two clouds have the same pattern of which sentences are near which, and 0 when the patterns are unrelated (Kornblith et al., 2019).

3.2 Three ways of representing a document, and the attribution rule

Three ways of representing a document. DATS represents every document in three ways, each reflecting one more of the three sources than the last. In the *bag-of-words (BoW)* representation, each sentence is a vector over the vocabulary of the document: in the word version, the entry for word w in sentence s is its TF–IDF weight, $\mathrm{tf}(w, s) \cdot \log\big(n/\mathrm{df}(w)\big)$, where tf is the count of w in s and $\mathrm{df}(w)$ the number of sentences containing w; in the character version, the entries are counts of character 3- to 5-grams. No model is involved, so this representation reflects the data only. In the *untrained network*, we instantiate the architecture of a trained model from its published configuration and draw every weight, including the token-embedding matrix, from the initialization the configuration specifies; the tokenizer is the model's own and no weight is ever trained. This representation reflects

the data and the architecture, which we take to include the network structure, the tokenizer, and the initialization scheme; the random seed and the initialization scale are nuisance choices, not sources (Appendix B). The *trained model* is the same architecture with its publicly released weights and reflects all three sources. We apply all three to the same documents and measure them with the same instruments.

Attribution rule. For a property of the trained model's geometry, we ask whether the same property is present in the BoW representation, and if not, whether it is present in the untrained network. If the BoW representation has it, the property comes from the data: nothing in the BoW representation was learned or computed by a network. If the BoW representation lacks it but the untrained network has it, the property comes from the architecture: the untrained network adds nothing to the BoW representation except a network and a tokenizer. If only the trained model has it, the property comes from training.

A representation shows an ordering if it produces the same ordering as the trained model, measured by rank correlation, and shows a graded quantity such as RSA or hit@k if its value is at least half of the trained model's value; otherwise it lacks the property.

3.3 Two measurements of geometry

We ask two questions about two clouds of the same sentences and call the answers *shape* and *identity*.

Shape. Do the two clouds have the same spread and the same pattern of which sentences lie near which? We measure spread by the intrinsic dimension of each cloud (Eq. 1) and pattern by the RSA between the two clouds (Eq. 2); RSA is 1 when the patterns agree and 0 when they are unrelated.

Identity. Can we find the representation of a sentence in one cloud from its representation in the other? Let X and Y be the clouds of the same n sentences in two ways of writing (for example, Chinese characters and pinyin), standardized as in §3.1; we call the two ways of writing the two *forms* of the text, and a measurement between them *cross-form*. For sentence i, let $\hat{x}_i$ and $\hat{y}_i$ be its standardized representations in the two forms, and let r_i be the rank of $\hat{y}_i$ among all n vectors of Y when they are sorted by similarity to $\hat{x}_i$,

$$r_i \;=\; 1 + \sum_{j \neq i} \mathbf{1}[\,\langle \hat{x}_i, \hat{y}_j \rangle > \langle \hat{x}_i, \hat{y}_i \rangle\,], \qquad \text{hit@}k(X, Y) \;=\; \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\,r_i \leq k\,]. \tag{3}$$

hit@k is the fraction of sentences whose counterpart is among their k nearest cross-form neighbors. Under a random matching its expected value is k/n; with $n = 200$ and $k = 10$, chance is 0.05.

Shape and identity are independent. Either measurement can be at its maximum while the other is at chance: a cloud rotated onto itself keeps its shape and loses every identity, and a cloud whose points are each pushed in their own direction keeps every identity and loses its shape (Proposition 1, Appendix C). We therefore report both.

Controls. Three checks accompany every correspondence value: a *mismatched-document* control, which pairs X with a different document of the same type; a *length null*, which recomputes hit@k after permuting counterparts within groups of sentences of similar length; and a *length-corrected RSA*, which removes from both similarity matrices the part explained by sentence length, since untrained networks encode length and two forms of a sentence have proportional lengths. Appendix F defines each.

Unless stated otherwise, we report ID as its peak over layers, and RSA and hit@k as the median over the middle layers $0.3 \leq \ell/L \leq 0.7$ and then over documents.

4 EXPERIMENTAL SETUP

4.1 DOCUMENTS

Five English document types. We need documents that differ in how free the choice of words is, from open prose to a fixed formal language, and take five types that span this range and have public, well-studied sources: fiction, essays, scientific articles, statutes, and source code. We use 56 English documents of these five types, ordered by *lexical freedom*: how unconstrained the choice of the next word is in a type of text. The ordering is measured, not assumed: we operationalize lexical freedom as the unigram entropy of a document, $H(D) = -\sum_w p_D(w) \log p_D(w)$, where p_D is the frequency of word w in D, and the median entropy of the five types falls in the order narrative (8.62 bits), essay (8.59), expository (8.26), legal (7.78), code (6.67). Table 7 and Appendix E list the sources and every document with its entropy. Each document is truncated to its first 200 sentences (cleaning in Appendix E). The predicted ordering of the five types was fixed before any model was run (Appendix D).

Chinese documents in three forms. To hold content fixed while changing the way of writing, we use 24 Chinese documents of four types, each a chunk of 200 consecutive sentences from one book (Table 7, Appendix E), each rendered in simplified characters, traditional characters, and pinyin (with and without tone marks) with sentence boundaries and punctuation unchanged. The character and pinyin forms of a sentence share no content token, so correspondence between their clouds cannot come from shared tokens; digits and Latin letters are excluded from the identity measurement.

English documents in two cases. As a second pair of ways of writing the same text, 8 documents each of the narrative, expository, and legal types are rendered in all upper case. Unlike character–pinyin pairs, these share most tokens with the original and serve as the shared-token reference in §5.

4.2 MODELS AND READOUTS

We measure seven trained models and their untrained counterparts (Table 6, Appendix E): Qwen2.5 at 0.5B, 3B, and 7B parameters, which holds the architecture fixed while scaling; Llama-3.1-8B and Mistral-7B, a second attention architecture; and Mamba-130M and Mamba-2.8B, a state-space architecture. The untrained networks are built as described in §3.2; two seeds, two initialization scales, and the Llama-3.1-8B base checkpoint check that none of these affects the results (Appendix B). The training trajectory uses the ten public intermediate checkpoints of Pythia-410M (Biderman et al., 2023), from step 0 to step $143{,}000$, on the English documents and their upper-case versions.

Readouts. At every layer we store three readouts of a sentence: the hidden state of its last token when it is encoded on its own (*last*), the mean of the hidden states over its tokens (*mean*), and the last-token hidden state when it is encoded after its three predecessors in the document (*context*). The ordering of §5.1 uses the *last* readout, the state the model predicts from and the standard readout for intrinsic dimension; the correspondence measurements of §5.2–5.3 use the *mean* readout, the standard sentence embedding, which pools every token so that a cross-form comparison does not rest on one token's state. Appendix B gives the other readouts; every attribution holds but one (Mamba-130M), noted there.

4.3 THE THREE PROPERTIES

We apply DATS to three properties of document geometry. Two are properties of shape. Property ① is the ordering of document types by intrinsic dimension: the coarsest fact about a document's geometry, and one that has been read as a measure of how much a model abstracts from its input (Cheng et al., 2025). Property ② is the similarity structure between a Chinese document written in characters and the same document written in pinyin, which shares no content token with it (a cross-form comparison, §3.3); alignment across two ways of writing the same text has been read as a learned invariance (Muller et al., 2021; Wendler et al., 2024). Property ③ is one of identity: whether a sentence written in pinyin is placed next to the same sentence written in characters. DATS asks where each comes from (§5).

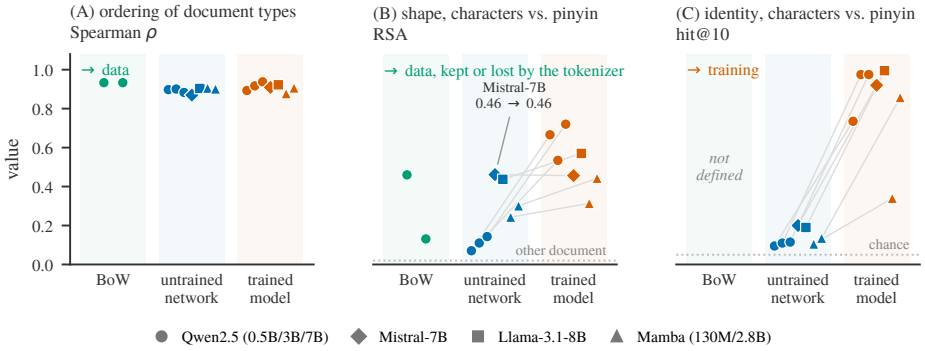

Figure 2: DATS applied to the three properties. Each marker is one representation (BoW green, untrained networks blue, trained models orange; grey lines join the two versions of one architecture). (A) Spearman correlation between the predicted ordering of document types and per-document peak intrinsic dimension: 0.87–0.94 in every representation, so the ordering comes from the data. (B) RSA between a Chinese document in characters and the same document in pinyin: present in TF–IDF, kept by untrained networks whose tokenizer splits characters into bytes, lost by untrained Qwen2.5, and unchanged by training in Mistral-7B. (C) hit@10 for the same pair: near chance before training, 0.34–0.99 after, and not defined for BoW.

4.4 Statistics

Primary tests are rank-based (Spearman correlation for orderings, Wilcoxon signed-rank tests for paired comparisons; confound checks in Appendix B). Document types are ranked from least to most lexically free, so a correct ordering gives a positive correlation.

5 Results

Figure 2 shows the three properties of §4.3 in the three ways of representing the same documents. We report them in the order of the attribution rule: the ordering of document types is already in the BoW representation (§5.1); the cross-form shape is also in the BoW representation, and, under the mean readout, the tokenizer decides whether an untrained network keeps it (§5.2); identity appears only after training (§5.3). Full tables are in Appendix F; Appendix H reports where the attribution stops.

5.1 ① Shape from the data: the ordering of document types

Attribution. The BoW representation, which has no model, orders the five English document types by intrinsic dimension as the trained models do, up to the narrative–essay swap, so DATS attributes the ordering to the data.

Evidence. Table 9 (Appendix F) gives the peaks and the Spearman correlation with the predicted ordering: 0.93 for both BoW representations, 0.87–0.91 for the seven untrained networks (and the two further initializations, Appendix B), and 0.88–0.94 for the seven trained models. The only reordering is narrative–essay, whose medians are within a few units.

Controls. Sentence length does not explain the ordering: it accounts for 59–66% of the variance in log peak across documents, document type for 89–93%, and document type remains significant with length as a covariate (F-test, $p \approx 10^{-16}$). The ordering survives a change of estimator, a halving of the number of sentences, a fixed relative depth in place of the peak, and a change of readout (Appendix B).

5.2 ② Cross-form shape from the data, kept or lost by the tokenizer

Attribution. The BoW representation already has the similarity structure between a document written in characters and the same document written in pinyin: TF–IDF gives $\mathrm{RSA} = 0.46$, as high as any untrained network and equal to trained Mistral-7B (Table 10, Appendix F), so DATS attributes the cross-form shape to the data.

Evidence. What the BoW representation carries across the two forms is the part of the word-count geometry that survives the merging of homophonous characters into one syllable (Lemma 1, Appendix C); the character n-gram version, whose units cut across syllables in pinyin, keeps less of it (0.13). Untrained Llama-3.1-8B and Mistral-7B reach $\mathrm{RSA} = 0.44$–0.46 and untrained Mamba 0.24–0.30; untrained Qwen2.5 stays at 0.07–0.14 from 0.5B to 7B, so the difference is not model size. The Llama, Mistral, and GPT-NeoX (Mamba) vocabularies split Chinese characters into byte pieces; the Qwen vocabulary contains whole characters and words, so the two forms enter the network as unrelated tokens. Training raises the cross-form RSA to 0.53–0.72 in Qwen2.5 and to 0.57 in Llama, and leaves it unchanged in Mistral: 0.46 before and after.

Controls. The structure is not shared across documents: pairing a document's characters with a different document's pinyin gives $\mathrm{RSA} \leq 0.02$ for every representation. It is not sentence length: the length-corrected RSA lowers untrained Llama from 0.44 to 0.39 and the trained models by a similar margin. And it is not dimension: untrained networks represent the pinyin form in fewer dimensions than the character form, and trained models bring the two to the same size at mid-depth (Figure 4B, Appendix F) without changing the RSA in Mistral.

5.3 ③ Identity from training

Attribution. Neither the BoW representation nor any untrained network places a sentence written in pinyin next to the same sentence written in characters, and every trained model does, so DATS attributes identity to training.

Evidence. In Table 10, hit@10 is close to chance for every untrained network (0.10–0.20; chance is 0.05) and 0.34–0.99 for the trained models: Llama-3.1-8B places the pinyin version of a sentence among the ten nearest neighbors of its character version for 99% of sentences. In Mistral, mid-depth RSA is 0.46 before and after training and hit@10 goes from 0.20 to 0.92: under the mean readout, training added identity and nothing else that these measurements see.

Controls. The length null accounts for about half of the untrained excess over chance and none of the trained value (Appendix C.2), and the Llama-3.1-8B base checkpoint gives the same hit@10, so instruction tuning plays no part.

Second pair of forms. For English sentences in mixed and upper case, every trained model reaches $\mathrm{hit@10} = 1.00$ on every document, against 0.26–0.39 for untrained networks, while the BoW RSA is already 0.96–0.99 because a consistent renaming of tokens preserves count structure (Appendix C.3). Here shape is almost fully present before training, and identity is not.

When identity appears. Figure 3 follows Pythia-410M through ten public checkpoints on the mixed- and upper-case pair. hit@10 starts at 0.39 with untrained weights and reaches 1.00 at step $4{,}000$, where it stays for the remaining $139{,}000$ steps. RSA on the same run rises from 0.48 to 0.81 by step $4{,}000$ and to 0.91 by the end; the ordering of document types (§5.1) is present at step 0 ($\rho = 0.40$) and sharpens to 0.62 by step $8{,}000$. Identity is learned within the first three percent of training; shape is present before training and keeps changing for the rest of it.

6 Discussion

Answers to the three consequences. The Introduction named three consequences of attributing properties to training without separating the three sources. On credit: similarity between the representations of two models, or between two ways of writing the same text, is first of all a product of the data and the tokenizer, so a claim that a model learned some geometry should be made only after the BoW and untrained baselines are subtracted. On what to change: under the mean readout the tokenizer sets the cross-form shape before any training, so shape is changed at the tokenizer and

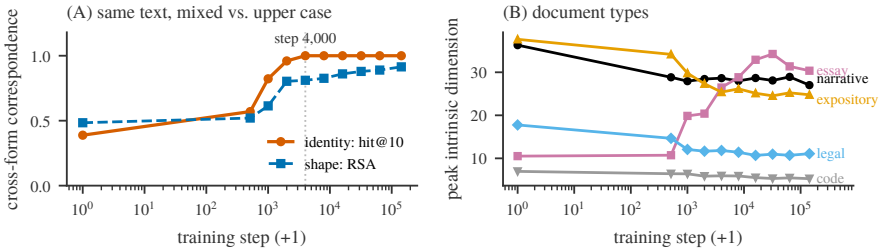

Figure 3: Identity is learned early; shape keeps changing. Pythia-410M across ten public checkpoints (step $+1$ on a log axis). (A) For the same English sentences in mixed and upper case, identity (hit@10, orange) reaches 1.00 by step 4,000 and stays there, while shape (RSA, blue) keeps rising to the end of training. (B) Peak intrinsic dimension of each document type. Legal and code sit below narrative and expository from step 0; essay, whose sentences are the longest, starts low and reaches its trained position by step 8,000 (Appendix H).

identity only by training. On what to measure: shape follows the data and does not follow the loss, so progress in training should be read from identity, not from shape.

What training adds, and why it is cheap. Cheng et al. (2025) found a phase along depth where probes for surface properties fail and probes for content succeed. DATS shows that this ability, identity, is the one property among those we measured that neither the data nor the architecture supplies, and locates it in the middle of the network (Depth, Appendix B) and, in Pythia, within the first few percent of training. Mistral suggests why it is early: training leaves its mid-depth cross-form shape unchanged and adds identity, so identity aligns two clouds that already have the same shape, and aligning is cheaper than constructing.

Prior findings in the same ledger. The peak of intrinsic dimension along depth changes with training (Cheng et al., 2025), and our trajectory shows this too; the ordering of documents by intrinsic dimension does not, and is present in the data. Gröger et al. (2026) found that, after calibration, models agree only on which sentences are each other's nearest neighbors and not on the global structure. In our terms, the agreement that survives is identity, which every trained model learns; the agreement that is removed is shape, which comes from data and tokenizers that differ between models.

Use and limits. Cross-form identity can serve as a training diagnostic: the pairs need no labels and saturate early, so a fall in identity late in training would indicate fitting to the way of writing; we have not tested this. And any claim about cross-form or cross-model alignment should be computed after dimension-wise standardization, because on raw vectors the two measurements in this paper point the opposite way. The results are limited to models of at most 8B parameters; to one training trajectory in one language; to a tokenizer explanation that is an observation, not a mechanism; and to a presence threshold, half of the trained value, which is a convention.

7 Conclusion

We asked which part of the document-representation geometry of a trained language model comes from its data, which from its architecture, and which from its training. DATS answers this by attributing each property to the first of three representations of the same documents that shows it: bag-of-words, untrained network, trained model. Across four model families, the ordering of document types and the shape shared by two ways of writing the same text are present before training; only the ability to place two ways of writing a sentence together is not. Most of the document-level structure we measure is present in the data and the tokenizer. Training gives the model the ability to tell what a text says from how it is written, and this ability appears early in training.

AI USE STATEMENT

We used a large language model (Claude, Anthropic) to draft parts of several sections from the authors' notes, to draft parts of the analysis scripts from the first author's specifications, to draft the constructions and proofs in Appendix C from the first author's formulations, and to polish the prose. The authors reviewed and edited all drafts, ran all scripts, verified the outputs and the proofs, and take full responsibility for all content.

REPRODUCIBILITY STATEMENT

Every number in this paper is produced by analysis scripts from embeddings that a single script per model regenerates. The English documents come from public corpora (Appendix E), the intervention generators are described in Appendix G, the measurements are defined in §3, and the pre-specified predictions are listed in Appendix D. The Chinese documents, and the Japanese documents of the held-out replication in Appendix B, are excerpts of commercially published books and are not redistributed; we release the chunking and transliteration scripts, which rebuild every form from a copy of the source text. Code and the English documents will be released with the paper.

ETHICS STATEMENT

The Chinese documents are excerpts of copyrighted books, used only to compute the measurements reported here and not redistributed in any form.

REFERENCES

Zeyuan Allen-Zhu. Physics of language models: Part 4.1, architecture design and the magic of canon layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2512.17351.

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

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Jade Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2405.15471.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of NAACL-HLT*, 2018.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations (ICLR)*, 2025.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of EMNLP-IJCNLP*, 2019.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7:12140, 2017.

Deqing Fu, Tianyi Zhou, Mikhail Belkin, Vatsal Sharan, and Robin Jia. Convergent evolution: How different language models learn similar number representations. *arXiv preprint arXiv:2604.20817*, 2026.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view. *arXiv preprint arXiv:2602.14486*, 2026.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. Transformer Circuits Thread, `https://transformer-circuits.pub/2026/workspace/`, 2026.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2022.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP-IJCNLP*, 2019.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2405.07987.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Dhruva Karkada, Daniel J. Korchinski, Andres Nava, Matthieu Wyart, and Yasaman Bahri. Symmetry in language statistics shapes the geometry of model representations. *arXiv preprint arXiv:2602.15029*, 2026.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, et al. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*, 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021.

Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 17, 2004.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of EACL*, 2021.

Arwa Osman, Marco Baroni, and Iuri Macocco. Local and global regimes of geometric complexity in language model representations. *arXiv preprint arXiv:2608.14361*, 2026.

William Rudman and Carsten Eickhoff. Stable anisotropic regularization. In *International Conference on Learning Representations (ICLR)*, 2024.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of ACL-IJCNLP*, pp. 3118–3135, 2021.

Muhammad Usama and Dong Eui Chang. Convergence without understanding: When language models agree on representations but disagree on reasoning. *arXiv preprint arXiv:2605.23315*, 2026.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2302.00294.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations (ICLR)*, 2019.

Kelly Zhang and Samuel R. Bowman. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pp. 359–361, 2018.

Ziqian Zhong and Jacob Andreas. Algorithmic capabilities of random transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Anisotropy and standardization

Table 1 reports, for the mid-depth sentence representations of each network, the largest absolute activation, the fraction of the total variance carried by the single highest-variance coordinate, and the RSA between the similarity structure and sentence length. In the three Qwen2.5 models one coordinate carries 98–100% of the variance. Table 2 shows what this does to the correspondence measurements: computed on raw vectors, RSA between characters and pinyin is 0.89–0.93 for every trained transformer and hit@10 is 0.27–0.51; after the dimension-wise standardization of §3.1, RSA falls to 0.46–0.72 and hit@10 rises to the values of Table 10. The raw cosine measured the dominant coordinate, not the sentence.

Table 1: Mid-depth activation statistics (mean readout, English documents). "Top-1 var" is the fraction of variance in the single highest-variance coordinate.

Representation	max |activation|	top-1 var	RSA with length
untrained Qwen2.5-0.5B	5.4	0.08	0.01
untrained Qwen2.5-0.5B (scale 0.006)	0.5	0.14	0.01
untrained Llama-3.1-8B	26.0	0.08	0.37
Qwen2.5-0.5B	323.9	0.99	0.40
Qwen2.5-3B	598.0	0.98	0.37
Qwen2.5-7B	2327.0	1.00	0.37
Mistral-7B	33.4	0.92	0.29
Llama-3.1-8B	54.6	0.96	0.33
Mamba-130M	31.2	0.10	0.23
Mamba-2.8B	6.7	0.07	0.25

Table 2: Character–pinyin correspondence on raw and on standardized vectors (median over documents and middle layers).

	raw		standardized	
Representation	RSA	hit@10	RSA	hit@10
untrained Qwen2.5-0.5B	0.07	0.10	0.07	0.10
untrained Llama-3.1-8B	0.45	0.19	0.44	0.19
Qwen2.5-0.5B	0.90	0.27	0.67	0.73
Qwen2.5-3B	0.89	0.34	0.53	0.97
Qwen2.5-7B	0.91	0.32	0.72	0.97
Mistral-7B	0.93	0.40	0.46	0.92
Llama-3.1-8B	0.93	0.51	0.57	0.99
Mamba-130M	0.30	0.31	0.31	0.34
Mamba-2.8B	0.45	0.81	0.44	0.86

B ROBUSTNESS

Seeds and initialization scales. Untrained networks use the initialization specified by each model's configuration with seed 42; a second Llama-3.1-8B initialization uses seed 1, and a second Qwen2.5-0.5B initialization uses standard deviation 0.006 in place of the default 0.02. The two Llama seeds give character–pinyin RSA of 0.44 and 0.44 and hit@10 of 0.19 and 0.18; the two Qwen2.5-0.5B scales give 0.07 and 0.05 and 0.10 and 0.09. The ordering correlation of §5.1 is 0.90 and 0.87 for the two Llama seeds and 0.90 and 0.88 for the two Qwen2.5-0.5B scales.

Confound checks. Confounds are checked by ordinary least squares on log intrinsic dimension with document-level covariates (sentence length, token count, type–token ratio); sentence length alone explains 59–66% of the variance in log peak and document type 89–93%, and document type remains significant with sentence length as a covariate (F-test, $p \approx 10^{-16}$).

Threshold. The criterion for presence in §3.2 uses half of the trained model's value. Every attribution in §5 is unchanged for any threshold between 0.32 and 0.64 of the trained value. Below 0.31, untrained Mamba-130M would count as showing identity (hit@10 $= 0.105$ against a trained 0.34); above 0.64, the BoW cross-form shape (RSA $= 0.46$) would fall below the threshold for Qwen2.5-7B (trained RSA $= 0.72$). The ordering property is rank-based and does not use the threshold.

Estimator, sample size, depth. The ordering of document types is positive in Spearman correlation under the maximum-likelihood estimator of Levina & Bickel (2004) and under the participation ratio, weaker under the latter ($\rho = 0.82$ and 0.67 for Llama-3.1-8B); the correlation between peaks computed from 100 and from 200 sentences is 0.95–0.96 across models, and between the peak and the value at a fixed relative depth of 0.35 is 0.95–0.97.

Readouts. Table 3 gives the peak intrinsic dimension per document type under the three readouts of §4.2. The ordering is the same under all three; the mean readout keeps the untrained 7B networks within the estimator's range. For the cross-form measurements, the last readout gives untrained RSA of 0.03–0.06 and hit@10 of 0.06–0.10, against trained RSA of 0.15–0.29 and hit@10 of 0.28–0.84 for every model but Mamba-130M, whose last-token state identifies no better than an untrained network (0.10). The attribution of cross-form shape to the data (TF–IDF $= 0.46$ under either readout) is unchanged; identity is attributed to training for every model but Mamba-130M; and the structure that untrained Llama keeps under the mean readout is not present in its last-token state.

Table 3: Peak intrinsic dimension per document type under the three readouts (median over documents).

Representation	readout	narrative	essay	expository	legal	code
Qwen2.5-0.5B	last / mean / context	29.2 / 46.5 / 29.6	35.0 / 50.9 / 31.4	22.2 / 30.2 / 22.6	11.7 / 13.6 / 13.1	5.6 / 6.3 / 4.9
Qwen2.5-7B	last / mean / context	43.0 / 44.8 / 32.0	42.0 / 54.5 / 29.9	25.2 / 34.9 / 26.6	12.9 / 14.7 / 16.7	6.1 / 6.7 / 6.4
Mistral-7B	last / mean / context	46.8 / 38.3 / 38.0	54.3 / 42.7 / 42.8	28.2 / 29.8 / 27.0	12.7 / 14.3 / 16.0	7.1 / 5.5 / 6.3
Llama-3.1-8B	last / mean / context	43.3 / 41.9 / 33.2	47.1 / 50.1 / 33.9	26.1 / 29.3 / 30.8	13.2 / 15.8 / 20.1	6.4 / 5.9 / 6.3
Mamba-2.8B	last / mean	38.9 / 53.9	46.8 / 71.2	26.5 / 36.8	12.2 / 15.0	5.9 / 6.4
untrained Llama-3.1-8B	last / mean	397 / 64.2	383 / 77.8	342 / 51.0	243 / 43.8	57.8 / 12.4
untrained Qwen2.5-0.5B	last / mean	73.6 / 44.8	74.0 / 46.1	61.0 / 33.0	30.1 / 18.2	11.1 / 5.2

Depth. Identity is a property of the middle of the network: in Llama, hit@10 rises from 0.35 at 5% of depth to 0.94 at 15% and stays above 0.97 until the last layers, while RSA is highest in the first third of the network and declines toward the output, so shape and identity peak at different depths. Qwen2.5-7B and Mamba-2.8B reach their plateau later and fall back toward the output.

Other pairs of ways of writing the same text. Table 4 repeats the correspondence measurements for three further pairs. Simplified and traditional Chinese characters share most tokens, and identity is at ceiling for trained and untrained networks alike; the pair is the Chinese shared-token reference. Pinyin without tone marks merges more characters into one syllable than pinyin with tone marks, and the results follow those of Table 10 with slightly lower values throughout. Japanese written in kanji and kana and the same text in romaji is a held-out replication in a second language: the six documents were not used in developing the method, and the pattern is the same, with trained Llama and Mistral at hit@10 $= 0.92$ and 0.90 against 0.13–0.15 for their untrained counterparts.

Table 4: Correspondence for further pairs of ways of writing the same text (RSA / hit@10, median over documents and middle layers).

Representation	simplified–traditional (24 docs)		characters–pinyin, no tones (24)		Japanese–romaji (6)	
untrained Qwen2.5-0.5B / 7B	0.33 / 0.75	0.37 / 0.90	0.07 / 0.10	0.14 / 0.12	0.34 / 0.04	0.55 / 0.06
untrained Llama-3.1-8B	0.69 / 0.98		0.47 / 0.19		0.63 / 0.15	
untrained Mistral-7B	0.79 / 0.99		0.49 / 0.21		0.72 / 0.13	
untrained Mamba-2.8B	0.63 / 1.00		0.32 / 0.15		0.33 / 0.05	
Qwen2.5-0.5B / 7B	0.94 / 1.00	0.95 / 1.00	0.67 / 0.39	0.75 / 0.90	0.71 / 0.28	0.80 / 0.78
Llama-3.1-8B (Instruct / base)	0.97 / 1.00	0.96 / 1.00	0.50 / 0.96	0.48 / 0.94	0.57 / 0.92	0.57 / 0.92
Mistral-7B	0.94 / 1.00		0.41 / 0.88		0.53 / 0.90	
Mamba-2.8B	0.94 / 1.00		0.39 / 0.65		0.46 / 0.69	

C THREE SMALL RESULTS ABOUT THE INSTRUMENTS

The two correspondence measures of §3.3 are used throughout the paper as answers to two different questions. This appendix records that the two questions are formally independent, what the untrained identification rates contain, and why bag-of-words correspondence is bounded by the injectivity of the surface transformation.

C.1 RSA AND HIT@k ARE INDEPENDENT

Proposition 1. *For every $n \geq 3$ there exist clouds $X, Y \in \mathbb{R}^{n \times H}$ of the same n sentences in two forms such that $\mathrm{RSA}(X, Y) = 1$ and $\mathrm{hit@1}(X, Y) = 0$; and for every $\varepsilon > 0$ there exist clouds with $\mathrm{hit@1}(X, Y) = 1$ and $\mathrm{RSA}(X, Y) < \varepsilon$.*

Proof. Same shape, every identity wrong. Let $H = 2$ and let $x_1, \ldots, x_n$ be the vertices of a regular n-gon on the unit circle, $x_i = (\cos \frac{2\pi i}{n}, \sin \frac{2\pi i}{n})$. Define $y_i = x_{i+1}$ (indices mod n), the same cloud rotated by one vertex. Then

$$\begin{aligned} S^Y_{ij} = \langle x_{i+1}, x_{j+1} \rangle = \cos\!\big(\tfrac{2\pi(i-j)}{n}\big) = S^X_{ij} \qquad & \text{(similarity depends only on } i-j) \\ \mathrm{RSA}(X, Y) = \rho_{\mathrm{S}}\big(\{S^X_{ij}\}, \{S^Y_{ij}\}\big) = 1 \qquad & \text{(identical upper triangles)} \\ \langle x_i, y_i \rangle = \cos\!\big(\tfrac{2\pi}{n}\big) < 1 = \langle x_i, y_{i-1} \rangle \qquad & (y_{i-1} = x_i) \end{aligned}$$

so the nearest cross-form neighbor of every x_i is y_{i-1}, not y_i, and $\mathrm{hit@1} = 0$.

Every identity right, arbitrary shape. Let $H \geq 2n$, let $x_1, \ldots, x_n$ be distinct unit vectors spanning a subspace V, and let $z_1, \ldots, z_n$ be unit vectors in a subspace $W \perp V$ with Gram matrix $G_{ij} = \langle z_i, z_j \rangle$. Any symmetric positive definite matrix with unit diagonal is the Gram matrix of unit vectors in $\mathbb{R}^n$, so G may be chosen as $G = I + \varepsilon K$ with $K_{ii} = 0$ and $K_{ij} = -S^X_{ij}$ for $i \neq j$; for $\varepsilon < 1/n$ this matrix is positive definite by Gershgorin's theorem, and the ranks of its off-diagonal entries are the ranks of $\{S^X_{ij}\}_{i<j}$ reversed. For $d > 0$ define $y_i = (x_i + d\, z_i)/\sqrt{1 + d^2}$, a unit vector. Then

$$\begin{aligned} \langle x_i, y_j \rangle = \langle x_i, x_j \rangle / \sqrt{1 + d^2} \qquad & (z_j \perp x_i) \\ \langle x_i, y_i \rangle = 1/\sqrt{1 + d^2} > \langle x_i, y_j \rangle \quad (j \neq i) \qquad & (\langle x_i, x_j \rangle < 1 \text{ for } x_j \neq x_i) \\ S^Y_{ij} = \big(S^X_{ij} + d^2\, G_{ij}\big)/(1 + d^2) \quad (i \neq j) \qquad & (z_i \perp x_j,\, x_i \perp z_j) \end{aligned}$$

so $\mathrm{hit@1} = 1$ for every d, while the off-diagonal entries of S^Y are a convex combination of S^X_{ij} and G_{ij} whose weight on G tends to 1. Because the entries of εK are distinct whenever those of S^X are, there is a d_0 such that for $d > d_0$ the ranks of $\{S^Y_{ij}\}_{i<j}$ equal the ranks of $\{G_{ij}\}_{i<j}$, and then $\mathrm{RSA}(X, Y) = \rho_{\mathrm{S}}(S^X, G) = -1 < \varepsilon$. (If S^X has tied entries, perturb K by an arbitrarily small amount to break the ties; this changes neither $\mathrm{hit@1}$ nor the positive definiteness of G.) □

RSA is blind to a consistent relabeling of the points, and hit@k is blind to a uniform collapse of the similarity structure. Neither measure can be read off the other, which is why Table 10 reports both and why the untrained and trained representations can agree on one while differing on the other.

C.2 What the untrained identification rates contain

Under a random matching of the n pinyin sentences to the n character sentences, each sentence's counterpart lands among its k nearest cross-form neighbors with probability k/n; with $n = 200$ and $k = 10$ this is 0.05, the chance level quoted in the text. The length null of §3.3 is a permutation test against a stronger null that keeps sentence length: the counterparts are permuted within groups of sentences of similar length, and hit@10 is recomputed. For untrained Llama-3.1-8B this null gives 0.11 against the observed 0.19 (median over 24 documents and middle layers); for untrained Qwen2.5 it gives 0.05 against 0.10–0.12, and for untrained Mistral 0.11 against 0.20. The excess of the untrained networks over 0.05 is therefore real, and about half of it is sentence length, which §5.2 attributes to the tokenizer. The same null applied to trained models gives 0.11–0.13 against observed values of 0.34–0.99, so nothing of the trained identity is length. Every untrained value is below half of the trained value on the same model, so under the rule of §3.2 the untrained networks lack identity; the excess over 0.05 is recorded here so that *near chance* in §5.3 is not read as *chance*. We do not use a binomial approximation: the n indicators within one document share a similarity matrix and are not independent, and the permutation null needs no such assumption.

C.3 Bag-of-words correspondence is bounded by the injectivity of the surface map

Lemma 1. *Let ϕ be a surface transformation acting on token types (for example, upper-casing, or the character-to-pinyin map applied character by character). If ϕ is injective on the token types that occur in a document, the TF–IDF count vectors of the transformed sentences are a fixed coordinate permutation of the originals. Consequently the transformed cloud is isometric to the original, its TWO-NN intrinsic dimension is unchanged, and* $\mathrm{RSA} = 1$ *exactly.*

Proof. An injective ϕ on types induces a bijection between the coordinate index sets of the two count matrices, so $C^{\phi} = C\,P$ for a permutation matrix P. IDF weights depend only on document frequencies of types, which P preserves, so the TF–IDF matrices satisfy $T^{\phi} = TP$. Row-wise Euclidean and cosine distances are invariant under a right multiplication by P, hence all pairwise distances, the TWO-NN estimate, and the similarity matrix are preserved. □

When ϕ is not injective, distinct types merge and the transformed similarity structure differs from the original by an amount that grows with the mass of merged types. Two entries of Table 10 and §5.2 are explained by this. Upper-casing merges `The` with `the` and similar sentence-initial pairs, a small mass, and bag-of-words RSA is 0.96–0.99 rather than 1. Character-to-pinyin merges every set of homophonous characters into one syllable (the *pinyin* form) or, with tone marks removed, a larger set (the *toneless* form), a large mass, and bag-of-words RSA drops to 0.46 and 0.33 respectively. The lemma thus gives the BoW representation's cross-form structure a mechanical reading: it is exactly the part of the document's word-count geometry that survives the merging of types under ϕ, and nothing in it involves content.

D Pre-specified predictions

Each prediction in Table 5 was written down before the corresponding data were analyzed.

E Documents and models

English. Table 8 lists the 56 English documents with their number of sentences and unigram entropy. Narrative documents are chapters from novels in the BookSum collection; essays are pre-1929 English essays from Project Gutenberg, one document per author; expository documents are article bodies from the arXiv summarization corpus; legal documents are sections of the U.S. Code from the Pile of Law; code documents are source files from The Stack in Python, JavaScript, Java, C, and Go. For every document we removed front matter, tables of contents, running headers, and watermark lines, split the text into sentences, kept only sentence-terminated paragraphs, and truncated to the first 200 sentences.

Table 5: Pre-specified predictions and their outcomes.

Prediction	Outcome
The five English document types order by intrinsic dimension as narrative > essay > expository > legal > code, in every representation.	Held up to the narrative–essay pair (Table 9).
Each intervention changes the peak intrinsic dimension of trained models in the stated direction (Table 11).	Held for all but the within-sentence shuffle in Qwen2.5-0.5B, which was flat.
Across documents, the change in geometry under the cross-sentence shuffle correlates with the change in next-token loss.	Did not hold (Appendix H).
The cross-form similarity structure of untrained Llama is explained by sentence length.	Did not hold (Appendix H).

Table 6: Models. An untrained network is instantiated from the same configuration as the trained model, with every weight, including the token embeddings, drawn from the configuration's initialization, and is never trained.

Family	Sizes	Trained	Untrained	Base	Trajectory
Qwen2.5	0.5B, 3B, 7B	✓	✓ [a]	–	–
Llama-3.1	8B	✓	✓ [b]	✓	–
Mistral	7B	✓	✓	–	–
Mamba	130M, 2.8B	✓	✓	–	–
Pythia	410M	–	–	–	✓ [c]

[a] 0.5B also at initialization scale 0.006. [b] Two seeds. [c] Steps 0, 512, 1k, 2k, 4k, 8k, 16k, 32k, 64k, 143k.

Chinese. The 24 Chinese documents are chunks of 200 consecutive sentences, split at sentence-final punctuation, from four books (Table 7). The traditional-character form is produced by OpenCC (simplified to traditional); the pinyin form by pypinyin with tone marks, and the toneless form with tone marks removed. Sentence boundaries and punctuation are preserved in every form.

Models and software. Trained models are the Hugging Face checkpoints `Qwen/Qwen2.5-{0.5B,3B,7B}-Instruct`, `meta-llama/Llama-3.1-8B-Instruct` and `meta-llama/Llama-3.1-8B`, `mistralai/Mistral-7B-Instruct-v0.3`, `state-spaces/mamba-{130m,2.8b}-hf`, and `EleutherAI/pythia-410m` at revisions `step0` to `step143000`, all at the default revision current in September 2026. Untrained networks are built with `AutoModelForCausalLM.from_config` from the same configurations with `torch.manual_seed(42)` (seed 1 for the second Llama initialization) and, for the second Qwen2.5-0.5B initialization, `initializer_range` set to 0.006. All models run in `float16` with `transformers` $\geq$ 4.37 and `torch` $\geq$ 2.0; the estimators use `scikit-learn` and `scipy`. English sentences are split at sentence-final punctuation followed by whitespace, with a fixed list of abbreviations protected, and sentences shorter than four words are dropped; Chinese and Japanese sentences are split at 。 ！ ？ . The transliterations use `pypinyin` (tone marks on and off), `OpenCC` (`s2t`), and `pykakasi`. For every Chinese and Japanese document the released manifest gives the book, the chunk index, and the sentence offsets, so that the chunk can be rebuilt from any copy of the text.

Japanese. The six Japanese documents used in Appendix B are chunks of 200 sentences from one novel; the romaji form is produced by pykakasi.

F Full tables for §5

Controls. The *mismatched-document* control replaces Y by the same-index sentences of a different document of the same type, which removes all shared content and keeps sentence position, so correspondence that survives is not about the document. The *length null* recomputes hit@k after permuting the counterparts within groups of sentences of similar length, and gives the part of identification that sentence length alone accounts for. The *length-corrected RSA* removes from both

Table 7: Documents. English types are listed by decreasing lexical freedom (unigram entropy). Each Chinese document exists in simplified characters, traditional characters, and pinyin.

Language	Type	Source	Documents
English	narrative	BookSum novels (Kryściński et al., 2021)	12
English	essay	Project Gutenberg essays (pre-1929)	8
English	expository	arXiv articles (Cohan et al., 2018)	12
English	legal	U.S. Code, Pile of Law (Henderson et al., 2022)	12
English	code	The Stack, five languages (Kocetkov et al., 2022)	12
Chinese	narrative	《白鹿原》	4
Chinese	essay	《常识》 《末代皇帝》 《万历十五年》	12
Chinese	expository	《数学之美》	4
Chinese	legal	《民法典》	4

similarity matrices the part explained by differences in sentence length; it is written RSA | length in the tables.

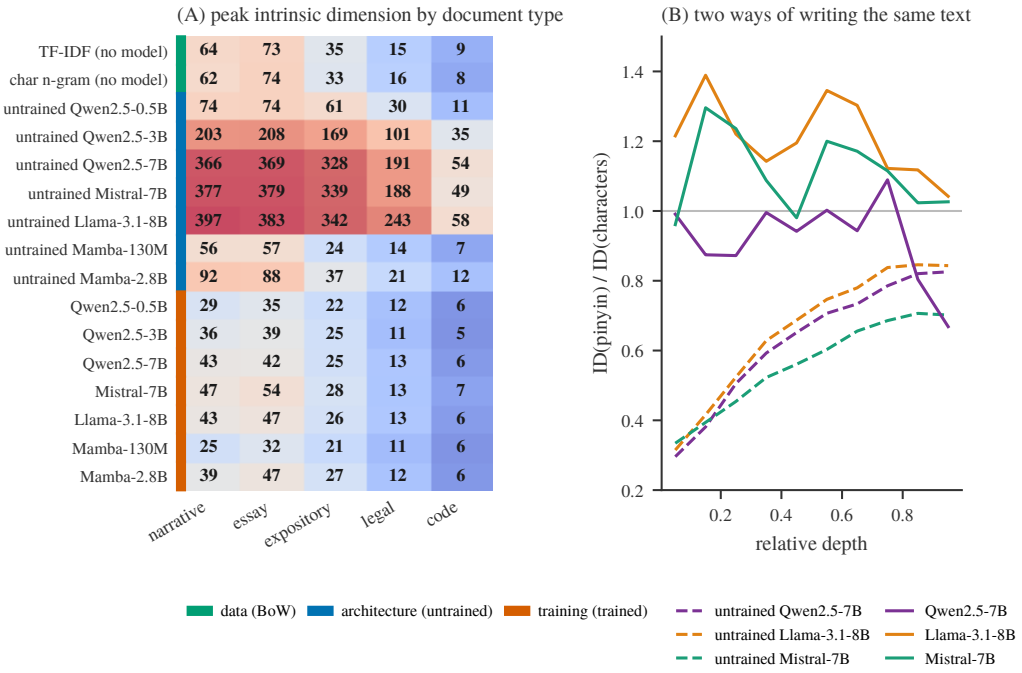

Figure 4: Shape is present before training. (A) Peak intrinsic dimension (last-token readout, median over documents) for the BoW representations (green), seven untrained networks (blue), and seven trained models (orange). Every row orders the five document types the same way up to the narrative–essay pair. (B) Ratio of the intrinsic dimension of a Chinese document written in pinyin to the same document written in characters, along depth. Untrained networks represent the pinyin form in fewer dimensions than the character form; trained models bring the two to the same size at mid-depth.

G Interventions

To locate the boundary of what training adds (Appendix H), we modify English documents in five ways: permuting the words within each sentence, redistributing all words of a document across its sentences, renaming code identifiers, replacing sentences by entropy-matched real words or pseu-

dowords, and stripping punctuation. Each modification has a known effect on the BoW representation, which is the reference against which the models' responses are read. We predicted the direction of each intervention's effect on the trained models before running it (Appendix D).

- *Within-sentence shuffle.* We permute the words inside each sentence. The word counts of every sentence are unchanged, so the BoW representation is unchanged. Applied to 8 narrative documents, two random draws each.
- *Cross-sentence shuffle.* We redistribute all words of a document across its sentences at random. The document's word counts are unchanged but each sentence's are not, so the BoW representation changes by a predictable amount, which we measure and use as the zero point. Applied to 8 documents each of the narrative, expository, and legal types, two draws each.
- *Identifier renaming.* In code documents we replace every variable and function name with a common English word, consistently within a document. The renaming is one to one, so the BoW similarity structure is unchanged (Lemma 1). Applied to 8 code documents.
- *Real words and pseudowords.* We replace each sentence by a sentence of the same length drawn from a real-word vocabulary or from a vocabulary of pronounceable non-words, matched in unigram entropy. The two versions have the same BoW statistics. Applied to 8 and 16 narrative documents respectively.
- *Punctuation stripping.* We remove all punctuation from the narrative, expository, and legal documents (8 each). The BoW representation loses only the punctuation tokens.

H BOUNDARIES OF WHAT TRAINING ADDS

This appendix reports four measurements that mark where the attribution of §5 stops.

What the BoW representation cannot see. Table 11 reports the interventions of Appendix G. Permuting the words within each sentence raises the peak intrinsic dimension of untrained Qwen2.5-0.5B ($74 \to 96$) and of trained Mamba-2.8B ($39 \to 45$, $p = 0.03$) but lowers it for trained Llama ($43 \to 34$, $p = 1.6 \times 10^{-5}$); Qwen2.5-0.5B is flat. The sign of the response to word order flips between the untrained 0.5B and the trained 7B–8B transformers; the trained state-space model does not flip, and two Mamba sizes cannot separate architecture from training budget. Renaming code identifiers to common English words raises the peak in every model and, as intended, changes nothing in the BoW representation. Redistributing all words of a document across its sentences preserves the document's word counts but not each sentence's, so the BoW response ($\times 1.51$ narrative, $\times 2.14$ expository, $\times 4.26$ legal) is the reference against which the models are read: every trained model falls below it in every document type, and Llama is the only model whose response is significantly below 1 ($\times 0.80$ on narrative, $p = 0.008$). Untrained Llama could not be measured under this intervention because its shuffled clouds exceed the estimator's range (ID ≈ 388 at width 4096).

Shape is not coupled to next-token loss. We predicted that the cross-sentence shuffle would raise per-token negative log-likelihood (it does, for all 5×24 document pairs) and that the size of a document's geometric response would track its loss increase. It does not: the Spearman correlation between $\Delta\mathrm{NLL}$ and $\Delta \log \mathrm{ID}$ lies between -0.10 and 0.17 for every model and is never significant. A property that comes mostly from the data has no reason to track the loss, and this one does not.

Sentence length in untrained networks. Every untrained network encodes sentence length: the mid-depth similarity structure of untrained Llama correlates with sentence length at $\rho = 0.37$, comparable to trained models (0.29–0.40). The essay document type, whose sentences are the longest in the corpus (150–180 characters against 90–125 for narrative), has a peak of only 10 at Pythia's initialization against 36 for narrative; training raises it to 30 while lowering every other document type. An untrained network with near-uniform attention averages more tokens in a longer sentence, and its sentence vectors converge; training breaks that averaging. Length is one of the input statistics that an untrained network already represents and that training reorganizes, and the length-corrected RSA of Table 10 keeps it out of the cross-form result.

Anisotropy. Before standardization, the raw cosine RSA between characters and pinyin was 0.89–0.93 for every trained transformer while hit@10 was 0.27–0.51. A single coordinate carries 99%

of the variance in Qwen2.5's mid-depth states (maximum activation 2,327 at 7B). Standardization removes it; both return to the values of Table 10. On raw vectors the two measurements pointed the opposite way. Every correspondence value in this paper is computed after standardization (Appendix A).

Table 8: The English documents. Entropy is unigram entropy in bits.

Type	Document	Sentences	Entropy
narrative	booksum_1412	200	8.02
narrative	booksum_1944	192	8.50
narrative	booksum_1990	200	8.40
narrative	booksum_214	193	8.78
narrative	booksum_3267	200	8.55
narrative	booksum_4528	199	9.33
narrative	booksum_4877	200	8.69
narrative	booksum_4878	200	8.74
narrative	booksum_561	200	8.84
narrative	booksum_63	200	8.50
narrative	booksum_8653	181	8.23
narrative	booksum_914	200	8.92
essay	addison_spectator	200	8.63
essay	bacon_essays	200	8.46
essay	emerson_essays_first	200	8.59
essay	emerson_essays_second	200	8.50
essay	hazlitt_table_talk	200	8.59
essay	lamb_essays_of_elia	200	8.94
essay	mill_utilitarianism	200	8.47
essay	ruskin_sesame_lilies	200	8.75
expository	arxiv_111	200	8.64
expository	arxiv_139	176	8.32
expository	arxiv_14	200	8.43
expository	arxiv_154	200	8.28
expository	arxiv_197	200	7.82
expository	arxiv_200	200	8.24
expository	arxiv_214	200	8.10
expository	arxiv_225	200	8.44
expository	arxiv_34	191	7.94
expository	arxiv_45	200	8.29
expository	arxiv_6	200	8.03
expository	arxiv_80	177	8.08
legal	uscode_0	200	8.05
legal	uscode_1	200	7.82
legal	uscode_2	200	6.27
legal	uscode_3	200	7.97
legal	uscode_4	200	8.09
legal	uscode_5	200	7.57
legal	uscode_6	200	7.80
legal	uscode_7	200	7.76
legal	uscode_8	200	7.62
legal	uscode_9	200	7.69
legal	uscode_10	200	7.79
legal	uscode_11	200	7.68
code	python_2	200	6.54
code	java_21	200	6.90
code	javascript_7	200	6.17
code	c_2	200	6.92
code	go_1	200	5.56
code	python_3	200	5.53
code	python_4	200	5.08
code	python_7	200	7.38
code	python_8	200	6.85
code	python_13	200	7.29
code	python_30	200	6.80
code	python_36	200	5.71

Table 9: Peak intrinsic dimension (TWO-NN, last-token readout, median over documents) per document type, and Spearman correlation ρ between the predicted order and per-document peaks. Every representation orders the types the same way up to the narrative–essay pair. Absolute values are not comparable across representations: they depend on the vector dimension, and the untrained 7B networks sit near the estimator's ceiling.

Source	Representation	narrative	essay	expository	legal	code	ρ
data	TF–IDF	64.4	73.1	35.0	15.4	8.5	0.93
data	character n-gram	62.4	74.0	33.3	16.0	7.7	0.93
architecture	untrained Qwen2.5-0.5B	73.6	74.0	61.0	30.1	11.1	0.90
architecture	untrained Qwen2.5-3B	203	208	169	101	35	0.90
architecture	untrained Qwen2.5-7B	366	369	328	191	54	0.88
architecture	untrained Llama-3.1-8B	397	383	342	243	58	0.90
architecture	untrained Mistral-7B	377	379	339	188	49	0.87
architecture	untrained Mamba-130M	56.4	56.5	23.9	14.2	7.4	0.91
architecture	untrained Mamba-2.8B	91.5	88.5	37.3	20.5	11.6	0.90
training	Qwen2.5-0.5B	29.2	35.0	22.2	11.7	5.6	0.89
training	Qwen2.5-3B	35.5	39.0	24.6	11.1	5.5	0.92
training	Qwen2.5-7B	43.0	42.0	25.2	12.9	6.1	0.94
training	Mistral-7B	46.8	54.3	28.2	12.7	7.1	0.91
training	Llama-3.1-8B	43.3	47.1	26.1	13.2	6.4	0.92
training	Mamba-130M	25.1	32.3	21.3	10.6	5.5	0.88
training	Mamba-2.8B	38.9	46.8	26.5	12.2	5.9	0.91

Table 10: Character–pinyin correspondence, median over 24 Chinese documents and middle layers. RSA: shape (Eq. 2); RSA | length: the same after removing sentence-length structure; hit@10: identity (Eq. 3; chance 0.05); length null: hit@10 after permuting counterparts within length groups. Mismatched-document RSA is ≤ 0.02 in every row. The BoW representation has no cross-form neighbor structure, hence no hit@10. Values at or above the trained median are in bold.

Source	Representation	RSA	RSA | length	hit@10	length null
data	TF–IDF	0.46	–	–	–
data	character n-gram	0.13	–	–	–
architecture	untrained Qwen2.5-0.5B / 3B / 7B	0.07 / 0.11 / 0.14	0.07 / 0.11 / 0.14	0.10 / 0.11 / 0.12	0.05
architecture	untrained Llama-3.1-8B (seed 42 / 1)	**0.44 / 0.44**	0.39 / 0.39	0.19 / 0.18	0.11
architecture	untrained Mistral-7B	**0.46**	**0.42**	0.20	0.11
architecture	untrained Mamba-130M / 2.8B	0.24 / 0.30	0.24 / 0.30	0.10 / 0.14	0.05
training	Qwen2.5-0.5B / 3B / 7B	**0.67 / 0.53 / 0.72**	**0.59 / 0.49 / 0.68**	0.73 / **0.97 / 0.97**	0.13
training	Llama-3.1-8B (Instruct / base)	**0.57 / 0.56**	**0.55 / 0.53**	**0.99 / 0.99**	0.11
training	Mistral-7B	**0.46**	**0.42**	**0.92**	0.11
training	Mamba-130M / 2.8B	0.31 / **0.44**	0.27 / **0.42**	0.34 / 0.86	0.11 / 0.12

Table 11: Peak intrinsic dimension before $\rightarrow$ after each intervention (medians; Mann–Whitney $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 10^{-3}$; Wilcoxon for the paired ratios), and the ratio of shuffled to original peak for the cross-sentence shuffle. The BoW column gives the part of each response that word counts alone produce.

Intervention	BoW	untrained Qwen2.5-0.5B	Qwen2.5-0.5B	Llama-3.1-8B	Mamba-2.8B
within-sentence shuffle (narrative)	$67 \leftrightarrow 60$	$74 \rightarrow 96 \uparrow$	$29 \rightarrow 28$	$43 \rightarrow 34 \downarrow^{***}$	$39 \rightarrow 45 \uparrow^{*}$
identifier renaming (code)	$8.6 \leftrightarrow 7.5$	$11 \rightarrow 19 \uparrow$	$5.6 \rightarrow 9.4 \uparrow$	$6.4 \rightarrow 9.7 \uparrow$	$5.9 \rightarrow 10.5 \uparrow$
real words $\rightarrow$ pseudowords	matched	$121 \leftrightarrow 115$	$50 \rightarrow 44 \downarrow$	$47 \rightarrow 59 \uparrow^{***}$	$58 \rightarrow 71 \uparrow^{***}$
cross-sentence shuffle, narrative ($\times$ original)	1.51	1.27	1.01	$\mathbf{0.80}^{**}$	1.08
cross-sentence shuffle, expository	2.14	1.46	1.32	1.15	1.64
cross-sentence shuffle, legal	4.26	2.85	2.09	2.14	2.58