A THEORY OF PLATONIC REPRESENTATIONS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representations of translated sentences are similar in the inner layers of multilingual language models – an observation connected to the platonic representation hypothesis, yet unexplained theoretically. We provide an explanation based on the assumption that data have a hidden hierarchical structure whose abstract levels are shared across languages while surface levels are modality- or language-specific. Concretely, we generate synthetic languages from probabilistic context-free grammars sharing upper-level but not lower-level production rules. In this setting the Bayes-optimal next-token predictor is belief propagation (BP); encoding its messages in successive layers yields analytical predictions that agree well with transformers trained on the same data. The framework explains why cross-lingual similarity peaks in middle layers, coexists with language-specific structure, and strengthens with language proximity, model quality and data exposure. It distinguishes similarity (shared neighborhood geometry) from alignment (shared coordinates), showing that the latter occurs when code-switched data, i.e. mixed-language sentences, are abundant enough. It further predicts that subtracting from each layer the component linearly predictable from the preceding one increases cross-lingual similarity, which we confirm in pretrained LLMs.

1 INTRODUCTION

The platonic representation hypothesis posits that models trained on vastly different data converge toward a shared representation of reality (Huh et al., 2024). Consistent with this view, semantically similar inputs elicit similar representations in the inner layers of trained networks (Acevedo et al., 2026). Remarkably, sentences and their translations are similarly represented inside multilingual language models, and the same holds across modalities: if two pictures are encoded similarly by a vision transformer, so are their captions by a language model (Huh et al., 2024). Here, similarity of representations means that they tend to share a local geometry or neighborhood structure. Despite the considerable attention these results have attracted, a theoretical framework explaining these phenomena is missing. A satisfactory theory should account for the following observations:

1. *Similarity coexists with language-specific structure.* Multilingual language models do not collapse their representations into a fully neutral interlingua: they retain language-specific information alongside shared conceptual structure (Singh et al., 2019; Chang et al., 2022; Dumas et al., 2025).

2. *Similarity is organized systematically across layers.* In multilingual LLMs, representations of translated sentences are most similar in the middle layers, surrounded by more language- or modality-specific processing near the input and output (Wendler et al., 2024; Zhao et al., 2024; Liu & Niehues, 2025; Dumas et al., 2025; Inaba et al., 2025; Huh et al., 2024).

3. *Similarity increases with language proximity.* Cross-lingual similarity is stronger between structurally and typologically close languages (Jones et al., 2021; Peng & Søgaard, 2024).

4. *Similarity tracks model quality and data exposure.* It strengthens as models acquire better representations through training and increased capacity (Peng & Søgaard, 2024; Inaba et al., 2025), and it is weaker for low-resource languages, which exhibit poorer cross-lingual transfer (Wu & Dredze, 2020; Miao et al., 2024).

5. *Beyond similarity, models can achieve alignment.* Models fine-tuned on a downstream task in one language often transfer zero-shot to others (Pires et al., 2019). This reveals a stronger property

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

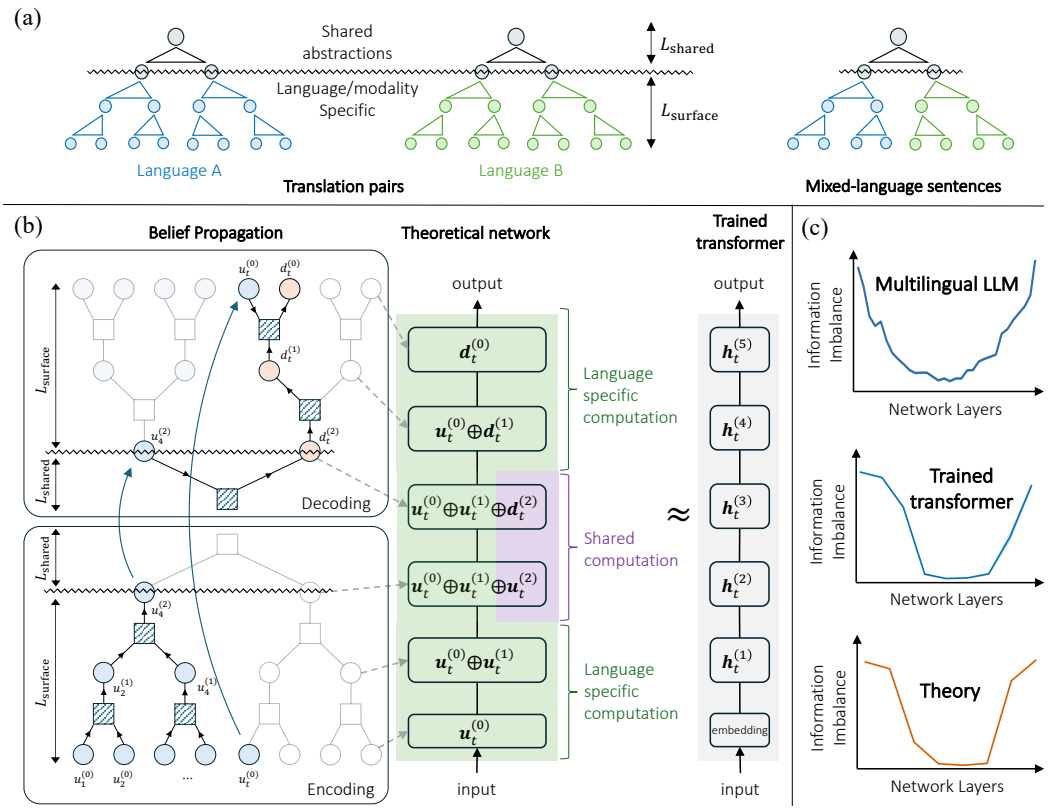


Figure 1: **(a)** Multilingual Random Hierarchy Model. Upper-level production rules are shared between languages, while lower-level rules are language-specific. Zigzag line denotes this separation. Mixed sentences are generated by replacing language-specific subtrees. **(b)** Optimal next-token prediction via inference with belief propagation (BP). *Encoding:* From t observed tokens, beliefs on higher-and-higher latent variables are sequentially computed and propagated upward. The highlighted nodes denote the encoded beliefs, and the lined boxes denote the computations. *Decoding:* Posterior probabilities of latent variables are computed iteratively using the encoded information, toward lower-and-lower levels, ultimately leading to the probability of the $(t+1)$-th token. From this exact algorithm, a theoretical network is constructed, whose layerwise representations encode the steps of the BP algorithm. The computations below L_{surface} are language-specific, while those above L_{surface} have shared components. This leads to a maximal cross-lingual similarity near the center of theoretical networks, confirmed in trained transformers. See Sec.4 for details. **(c)** Cross-lingual similarity measured with Information Imbalance (where lower Information Imbalance corresponds to higher similarities, see Appendix D). OLMo-3-7B tested on English and Polish (Team Olmo et al., 2026), transformers trained on multilingual RHM data, and theoretical networks constructed with BP – all exhibit similar II profiles, with strongest similarity in the middle layers.

than geometric similarity, which we call *alignment*: the two languages encode shared information along the same directions, so that a readout learned in one operates in the other without refitting – a property that rotation-invariant similarity measures cannot detect. Mixed-language contexts, including synthetic code-switching during pretraining, reinforce this alignment (Wang et al., 2025).

To explain these phenomena, we introduce a framework based on the assumption that language and images have a hidden hierarchical structure – a central notion in both linguistics (Chomsky, 1956) and computer vision (Mumford & Desolneux, 2010; Grenander & Miller, 2007) – and that the highest, most abstract levels of this hierarchy are shared across languages and modalities, while the lowest levels encode surface aspects (vocabulary, low-level syntax) that differ between them, as illustrated in Fig. 1(a). This framework accounts for all the facts listed above. It also makes a new prediction: subtracting from each hidden layer the component linearly predictable from the previous layer isolates the information newly computed at that depth, and this residual is more suitable for

predicting the neighbor structure of other languages than the raw representations. We confirm this prediction both in our controlled setting and in pretrained LLMs processing natural languages.

In practice, we design synthetic languages within the class of context-free grammars (Chomsky, 1956). Specifically, we consider a multilingual extension of the Random Hierarchy Model (Cagnetta & Wyart, 2024) in which languages share the upper production rules but differ in the lower ones. The model yields two kinds of predictions: computational, by training GPT-2-style transformers on these data for next-token prediction; and analytical, from belief propagation – the algorithm that solves next-token prediction optimally on such data, sketched in Fig. 1(b). Belief propagation is iterative, and assuming that its successive messages are linearly encoded in successive layers turns it into a network whose representations can be analyzed exactly. The two approaches approximately agree with each other and with measurements on pretrained LLMs, as illustrated in Fig. 1(c).

Technically, several metrics have been introduced to quantify the similarity of representations in LLMs. Recent work (Gröger et al., 2026) shows that global ones, including Centered Kernel Alignment (CKA), based on cross-correlations between representations (Kornblith et al., 2019), in fact provide limited support for the Platonic hypothesis. By contrast, local methods, which measure how well distances in one representation predict the neighborhood structure of another, do support it and thus are favored here. These include mutual k-Nearest Neighbors (mKNN) (Huh et al., 2024) – which, however, suffers from an undesirable dependence on sample size (Koepke et al., 2026) – and the Information Imbalance (II) (Glielmo et al., 2022). The II measures how highly the nearest neighbors in one representation are ranked in the distance ordering of another: it vanishes for perfectly similar representations, and reaches a baseline of unity when representations are unrelated. Detailed definitions of these metrics are provided in Appendix D. II and mKNN lead to consistent results.

2 Related work

Transformers and synthetic grammars. Transformers trained on CFG-generated data have been argued to parse: Zhao et al. (2023) show by construction that BERT-like models can approximate the inside-outside algorithm—which comprises an upward (encoding) and a downward (decoding) phase—and identify parsing information in trained models. Related conclusions hold for next-token prediction (Allen-Zhu & Li, 2025). Garnier-Brun et al. (2025) additionally study how hierarchical inference emerges during training, showing that correlations at successively larger scales are learned sequentially. For frozen tree topologies, Zhou et al. (2026) showed that the encoding sweep suffices (the decoding phase collapsing into a single final layer) and that trained transformers can find this solution. These works characterize the learned algorithm but not how it is learned, and none addresses the multilingual setting.

The Random Hierarchy Model (RHM). Random ensembles of grammars were introduced by De Giuli (2019); frozen-tree hierarchical generative models were proposed to study the learnability of deep functions (Mossel, 2016; Malach & Shalev-Shwartz, 2018). The RHM (Cagnetta et al., 2024) combines both – a frozen tree with random production rules – making learning itself tractable: it predicts sample complexities, that production rules are learned bottom-up (Cagnetta & Wyart, 2024), and neural scaling laws tested on natural language (Cagnetta et al., 2026); it extends to varying topologies (Parley et al., 2026). For diffusion models learning RHM data, belief propagation is the optimal denoiser, predicting a phase transition at finite noise observed in images and text (Sclocchi et al., 2025b;a). Yet the multilingual setting had not been considered, facts 1–5 are unexplained.

3 Multilingual-Multimodal Random Hierarchy Model (MRHM)

Generative grammars and the Random Hierarchy Model. A CFG generates an observed sequence by recursively expanding latent symbols according to production rules, making a generative tree whose leaves constitute the observed tokens (Chomsky, 1956). A probabilistic CFG additionally specifies a distribution over the possible production rules and therefore over sentences and their latent derivations. The RHM is a controlled, finite-depth instance of this framework (Cagnetta et al., 2024). It generates an s-ary tree of depth L. Each node carries a symbol from an alphabet $\mathcal{A}$ of size v; level L contains the root, while level 0 contains the observed tokens. We denote the level-0 leaf at position i by $x_i = X_i^{(0)}$, for $i = 1, \dots, s^L$, and denote its level-ℓ ancestor by $X_i^{(\ell)}$. Thus,

$X_i^{(\ell)} = X_j^{(\ell)}$ whenever x_i and x_j belong to the same level-ℓ sub-tree. A sketch of the RHM tree with notations is available in Appendix A.

At each level, a production rule maps a parent symbol $a \in \mathcal{A}$ to an ordered tuple of children $(b_1, \ldots, b_s) \in \mathcal{A}^s$. Each parent has m possible rules, giving mv rule tuples per level. The tuples are sampled without replacement from $\mathcal{A}^s$, ensuring that each child tuple maps back to a unique parent. A sentence is generated by sampling a root symbol and recursively selecting one of the m rules available to each node. The RHM is therefore specified by (L, s, m, v); in Fig. 1(a) $L = 3$ and $s = 2$.

Multilingual RHM. In natural language, translations share the abstract content while expressing it through different language-specific structures. This is also true for paired images and captions. We model these two situations by considering two RHM grammars that generate two languages A and B. The languages share the upper L_{shared} levels of the generative hierarchy (same production rules and symbols) but use distinct production rules and distinct symbols in the lower L_{surface} levels, with $L = L_{\text{shared}} + L_{\text{surface}}$. Thus at the split level L_{surface} (and only there), each symbol has $2m$ outgoing rules, with m rules for each language, giving $2mv$ rules in total.

A translation pair is generated by sampling a common upper tree down to the split level L_{surface} and then expanding the lower subtrees using the rules of languages A and B, respectively. The resulting sentences therefore share all the latent variables at levels $\ell \geq L_{\text{surface}}$ but differ below this split. We interpret the shared variables as common semantic content and the lower variables as its language-specific realization. Consequently, L_{surface} provides a controlled proxy for language distance: increasing L_{surface} reduces the amount of shared generative structure.

Code-switching as a cross-lingual training signal. Multilingual corpora contain mixed-language contexts that can provide models with evidence connecting otherwise distinct linguistic realizations. Consistent with the evidence that synthetic code-switching can improve cross-lingual alignment (Wang et al., 2025), we introduce a controlled form of code-switching during pretraining. For each level-L_{surface} sub-tree in a language-A sentence, we replace its language-A realization with the corresponding language-B realization with a probability ϵ, independently for all sub-trees; and analogously from B to A.

4 Theory: Next-token Prediction via Belief Propagation

Because the RHM has a tree-structured dependency graph, the Bayes-optimal next-token distribution $p(x_{t+1} \mid x_{1:t})$ can be computed by belief propagation (BP) (Sclocchi et al., 2025b). This remains true in the multilingual setting. As illustrated in Figure 1(b), the computation occurs in two stages: encoding (blue nodes) and decoding (orange nodes). During encoding, evidence from the observed tokens propagates upward through the hierarchy, producing beliefs over the latent variables. During decoding, beliefs propagate downward along the path to x_{t+1}, yielding its posterior predictive distribution.

To compare this computation with trained transformers, we construct theoretical networks whose layerwise representations linearly encode the BP messages. Belief over a latent symbol is represented by a vector in $\mathbb{R}^v$, with one coordinate for each symbol in the alphabet $\mathcal{A}$. Each network layer implements one sequential step of the BP computation.

Encoding. At token position t, the message $\mathbf{u}_t^{(\ell)} \in \mathbb{R}^v$ represents the belief over the latent variable $X_t^{(\ell)}$, inferred from the observed tokens within its subtree. The unobserved tokens are marginalized. The messages are computed recursively, as described in Appendix B.

For the language-specific levels $\ell < L_{\text{surface}}$, beliefs for the two languages occupy separate v-dimensional orthogonal blocks, giving a $2v$-dimensional representation. Separate blocks also occur for the shared levels $\ell \geq L_{\text{surface}}$; however, we will empirically show and quantify in § 7 that some amount of coordinate alignment – a finite scalar product – can occur depending on the code-switching rate ϵ and the training set size P. As shown in Appendix H, however, such alignment has little effect on geometric, neighborhood similarity; thus in the main text theoretical networks assume orthogonal representations for simplicity. The layerwise representations of the theoretical network accumulate the messages computed up to that point:

$$\mathbf{h}_t^{(k)} = \bigoplus_{\ell=1}^{k} \mathbf{u}_t^{(\ell)}, \qquad 1 \le k \le L-1, \tag{1}$$

where $\bigoplus$ denotes concatenation. The messages are accumulated at different levels because they are subsequently used during decoding.[1]

Decoding. Once the observed prefix has been encoded, information propagates downward along the ancestral path to the token being predicted. At level ℓ, the downward message $\mathbf{d}_t^{(\ell)}$ represents the posterior distribution of $X_{t+1}^{(\ell)}$, conditioned on the entire observed prefix $x_{1:t}$. Its recursive computation is given in Appendix B. As in the encoding stage, the beliefs below L_{surface} are language-specific, whereas those at and above L_{surface} are shared.

After the computation of a downward message, the corresponding upward message is no longer needed, and can be discarded. However, in an LLM there is no cost in keeping useless messages, as long as they do not interfere with the downstream computations. To allow for this possibility, we gradually decay used messages with a multiplicative factor $\gamma \in [0, 1]$ at each step after usage. The resulting layerwise representations are:

$$\mathbf{h}_t^{(k)} = \mathbf{d}_t^{(2L-1-k)} \oplus \underbrace{\left(\bigoplus_{\ell=1}^{2L-2-k} \mathbf{u}_t^{(\ell)} \right)}_{\text{not yet used}} \oplus \underbrace{\left(\bigoplus_{\substack{\ell=2L-1-k \\ \ell \neq 0}}^{L-1} \gamma^{\ell-(2L-2-k)} \mathbf{u}_t^{(\ell)} \right)}_{\text{used}}, \quad L \le k \le 2L-1. \tag{2}$$

Instantaneous removal of used messages corresponds to $\gamma = 0$. As shown in Fig. 10 of Appendix B, although the value of γ does not alter qualitative findings, $\gamma = 1/2$ leads to representations that match those found in language models trained on the RHM significantly better. Overall, the final message $\mathbf{d}_t^{(0)}$ is precisely the predictive distribution $p(x_{t+1} \mid x_{1:t})$. The complete theoretical network contains $2L-1$ layers, one for each step of the encoder–decoder computation.

Incomplete inference. Transformers approach optimal performance on RHM data only when trained on sufficiently many samples. With a smaller training set, they recover the production rules only up to an effective depth $\ell_* < L$, which increases with the number of training examples P (Cagnetta & Wyart, 2024). This produces suboptimal next-token predictions. We model this regime using an incomplete BP algorithm that only recovers production rules up to level ℓ_*, and higher level latents are not reconstructed. Note that a transformer has to perform fewer computational steps in this case, and passive layers will appear. For simplicity, we assume that they are the middle layers. Our theoretical results depend on this choice quantitatively, but not qualitatively. Details of this construction are discussed in Appendix B.

Alternative algorithm. Fixed-topology grammars also admit an $(L+1)$-step shortcut in which the downward computation collapses into a single final update (Zhou et al., 2026), while the encoder is exactly the same. We refer to this algorithm and to the associated theoretical network as "encoder-only". Depending on the training set size (P) and the code-switching rate (ϵ), trained transformers exhibit representations consistent with either the encoding-decoding or encoder-only algorithm. We expect encoder-decoder to be the robust solution that characterizes natural languages[2]. We therefore focus on $\epsilon = 0.1$, for which trained transformers consistently exhibit this structure, and study the transition between the two algorithms in Appendix E.

5 Geometric similarity

5.1 Transformer training setup and similarity measure

We train GPT-style decoder-only transformers for next-token-prediction on multilingual RHM data. Unless stated otherwise, the data are sampled from a multilingual RHM with $L = 4, s = 2, m =$

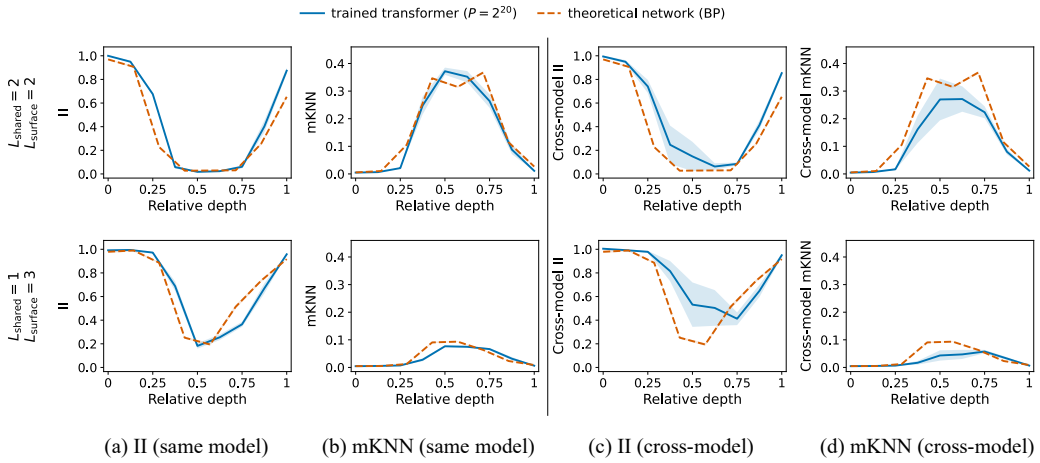
[1] A more parsimonious version of BP would only encode the messages $u_t^{(l)}$ actually used by the decoder. We show in Appendix B that such distinctions do not qualitatively change our findings.

[2] Grammars of natural languages have varying tree topology. In Appendix G, we show empirically that transformers trained on such grammars indeed display an encoder-decoder architecture (Parley et al., 2026).

$4, v = 16$, and two languages with split levels $(L_{\text{shared}}, L_{\text{surface}}) \in \{(3,1),(2,2),(1,3)\}$. The transformers have 8 layers, embedding dimension 512, and 4 attention heads.[3] We trained on datasets containing P sentences using cross-entropy loss and the AdamW optimizer for multiple epochs until convergence. For each run, we select the checkpoint with the lowest validation loss and perform all analyses on a held-out dataset not used during training. We perform similarity measurements on token-averaged representations of the networks.

5.2 Geometric similarity in multilingual models

We compare the geometric similarity in transformers trained on multiple languages with that of the theoretical networks constructed from BP (§4). Both networks are evaluated on the same collection of translation pairs. We study this for both: a single network trained on two languages and independent networks separately trained on the two languages respectively.



(a) II (same model) (b) mKNN (same model) (c) II (cross-model) (d) mKNN (cross-model)

Figure 2: **(a,b)** Representations of a transformer trained on multilingual RHM are geometrically similar in the middle layers when processing pairs of translated sentences. The geometric similarity is stronger for structurally similar languages, controlled by L_{surface} in our setup. **(c,d)** This similarity also exists between representations of independent models, trained separately on languages A and B. Measurements with II and mKNN show similar trends. In all cases, the empirical trends are well-captured by theoretical networks constructed with BP.

Geometric similarity is not perfect and peaks in the middle layers (1 & 2). Across different multilingual RHM settings (L_{surface}), the representations of translation pairs exhibit a non-monotonic similarity profile along transformer layers (Figure 2). Similarity is low near the input, increases substantially in the middle layers (but does not become perfect), and decreases again near the output; producing a U-shaped profile in II, and an inverted-U shaped profile in mKNN – as shown in Fig. 2 (a) and (b), respectively.

The corresponding BP networks reproduce the profiles qualitatively. This layerwise organization follows directly from the structure of inference, as sketched in Fig. 1 (b). During the initial encoding steps, the BP messages depend on the language-specific lower levels of the grammar and similarity of translations is absent. As implied by Eqs. 1 and 2, the middle layers encode the high-level latents that are shared by translated pairs, leading to the observed similarity. The latter is not perfect, since the information on low-level latents is still needed to ultimately be used in the decoder to predict the next token. Finally, during the final decoding steps, the computation returns to language-specific rules to predict the next token, causing the similarity to decrease in the final layers.

Geometric similarity strengthens with language similarity (3). The magnitude of middle-layer similarity depends systematically on the amount of generative structure shared between languages. Increasing the language-specific depth L_{surface}, while correspondingly decreasing the shared depth

[3] For $L = 4$, BP has $2L - 1 = 7$ sequential steps. Appendix F shows that our conclusions are unchanged for any choice of $n_{\text{layers}} \geq 2L - 1$; additional layers simply stretch the layerwise profile across network depth.

L_{shared}, raises the minimum II and lowers maximum mKNN (see Fig. 2). This is expected from the theory, since in the middle layers a larger part of the representation is shared across translated pairs.

Separately trained models also exhibit geometric similarity. Training two independently initialized transformers, one on language-A sentences and the other on language-B, exhibits similar layerwise similarity trends as a single trained model (Fig.2(c,d)).

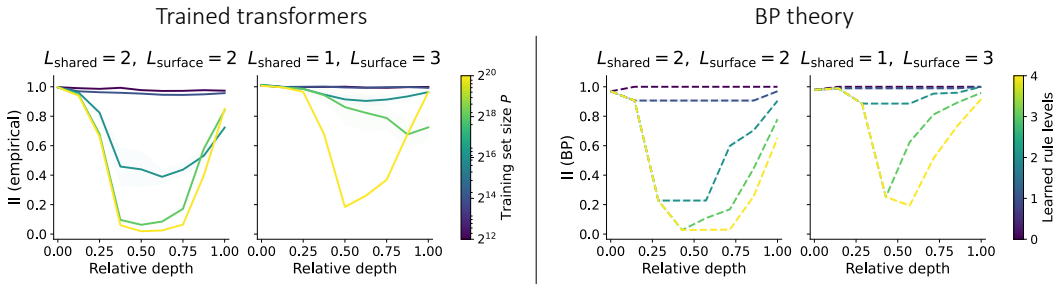
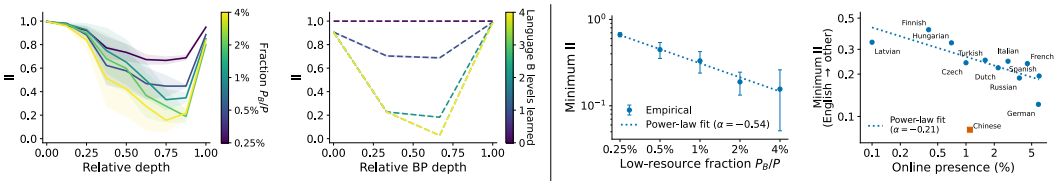


Figure 3: **(Left)** Cross-lingual similarity gets stronger as the training-set size P increases. **(Right)** Our theory captures this behaviour with the imperfect-BP algorithm, which only knows the RHM rules up to level ℓ_*.

Geometric similarity correlates with performance (4). At small training-set sizes, transformers exhibit poor next-token-prediction performance and little cross-lingual similarity. As P increases, geometric similarity progressively emerges in the middle layers, as indicated by lower II (Fig. 3).

This behaviour is explained by the imperfect-BP construction of §4. With finite data, the network learns the production rules only up to level ℓ_*. When ℓ_* does not extend into the shared portion of the hierarchy, the model cannot construct geometrically similar high-level representations. Middle-layer similarity emerges once the learned computation reaches the first shared level, and becomes stronger as additional shared levels are recovered. The discrete progression with ℓ_* in the theoretical networks reproduces the continuous improvement with P observed in the trained transformers.



(a) Trained transformer II ($\epsilon = 0$)
(b) BP theory II (Encoding-only)
(c) Trained transformer
(d) Multilingual LLM (Acevedo et al. 2026)

Figure 4: Cross-lingual geometric similarity with low-resource languages. **(a)** Similarity as measured by II gets weaker as the frequency of language-B decreases ($\epsilon = 0$). **(b)** BP theory captures this behaviour by considering fully learned language-A and partially learned language-B, up to level ℓ. **(c)** The minimum II (across layers) decreases monotonically with the fraction of low-resource language (P_B/P). **(d)** Plot adapted from (Acevedo et al., 2026): Multilingual LLM shows qualitatively similar trend – minimum II between English and other languages decreases monotonically with the online presence of the other language. Both theory and observations display apparent power law behavior in the range probed with exponents $\alpha \approx 0.5$ and $\alpha \approx 0.2$ respectively.

Geometric similarity weakens for low-resource languages. To capture the effect of language-specific exposure in pretraining, we fix the total dataset size (P) and vary the fraction of samples from each language, P_A/P and P_B/P, with $P_A + P_B = P$. To precisely control the sample fractions, we set $\epsilon = 0$ for this experiment. Fig. 4(a) shows that geometric similarity gets stronger as the low-resource fraction P_B/P is increased. BP theory reproduces this trend, with stepwise improvements as more language-B data allows additional grammar levels to be resolved (Fig. 4(b)). Note that here we use the $(L+1)$-step encoder-only BP computation, to match the $\epsilon = 0$ case. Fig. 4(c) shows the maximal geometric similarity (minimum II) across layers, as a function of P_B/P. A clear monotonic decrease of the minimum II shows that the similarity between two languages depends significantly on how much data from each language the model sees during training. These results

are consistent with experiments performed on natural language in Acevedo et al. (2026), where the abundance of languages in training corpora was approximated with their online presence. We show their results in Fig. 4(d) for comparison.

6 Layerwise Novelty

Definition of novelty. Can simple manipulations of LLM representations give more direct access to shared semantic content? Our theory implies that they can. As indicated in Eq.1, at each step of the encoding phase the representation contains a newly computed higher-level message, together with earlier lower-level language-specific messages, retained for subsequent decoding. Because the retained messages are predictable from the preceding layer, subtracting this predictable component isolates the more abstract information introduced by the current layer. We call this residual *layerwise novelty* and predict that it exposes shared computation more clearly than the raw hidden states. Details of the procedure to compute layerwise novelty are in Appendix I.

Novelty improves cross-lingual similarity in the RHM. We first test our prediction in the multilingual RHM. Fig. 5 confirms that layerwise novelty is more similar across languages at intermediate layers than raw representations. Novelty lowers the II in the first part of the network, including the minimum II at the middle layer. In later layers, novelty shows higher II because it removes the high-level information carried by the skip connections. Similarity measured with mKNN shows comparable results (Fig.18). These results indicate that shared structure is concentrated in the information computed in the intermediate layers, but is partially obscured in raw hidden states by language-specific information carried forward from earlier layers.

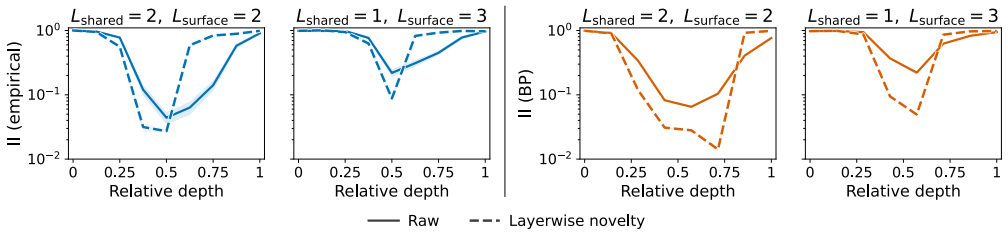


Figure 5: In both trained transformers and theoretical BP networks, layerwise novelty reveals more strongly similar intermediate computations than raw hidden states.

Layerwise novelty improves cross-lingual similarity in natural languages. We test this prediction in pre-trained language models, taking representations from Mistral-7B-v0.1 (Jiang et al., 2023) and OLMo-3-7B on parallel English–Romanian and English–Polish sentence pairs from Europarl (Koehn, 2005). In Fig. 6, we show that layerwise novelty achieves lower II than raw representations across most layers. Furthermore, we confirm in Appendix I that these results (i) hold for other models/language-pairs, (ii) hold for the mKNN metric, (iii) outperform the simple removal of residual connections, and (iv) outperform the supra-word residual (Toneva et al., 2022) – an alternative method obtained by subtracting from contextual embeddings the component linearly predictable from individual word embeddings.

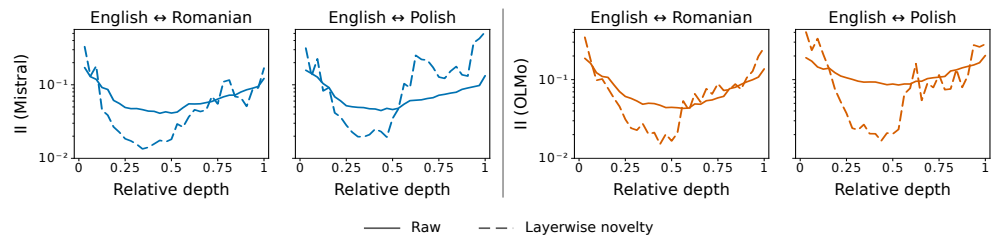


Figure 6: Comparison of II on raw representations and layer-wise novelty in Mistral-7B-v0.1 and OLMo-3-7B for parallel English–Romanian and English–Polish sentence pairs from Europarl.

7 COORDINATE ALIGNMENT

Geometric similarity does not imply coordinate alignment. Geometric similarity is a statement about shape: it compares distances and neighborhoods, and is therefore meaningful even when the two representations live in different spaces – it can be measured across separate models. To observe coordinate alignment, shared information must be encoded along the same directions of a common space, so that a readout learned on one language operates on the other without refitting. It is coordinate alignment, not geometric similarity, that underlies cross-lingual transfer. In this section we establish that the two properties are decoupled in our model: the shared hierarchy alone produces geometric similarity, whereas coordinate alignment requires an explicit cross-lingual signal – it is controlled jointly by the training-set size P and the code-switching rate ϵ.

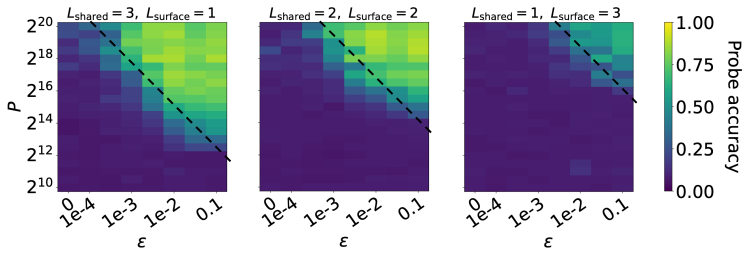


Figure 7: Alignment measured by latent cross-probing. Linear probes for latents at level-L_{surface} are trained on language-A representations and tested on language-B representations. Accuracy is controlled jointly by the code-switching rate ϵ and the training-set size P: the aligned region (green) appears along the predicted boundary $P_{\text{threshold}} \propto \epsilon^{-1}$ (dashed line).

We measure alignment using latent cross-probing: a linear probe is trained on language-A representations to predict a latent variable of the shared hierarchy, then evaluated, without refitting, on language-B representations. High cross-probing accuracy indicates that the two languages encode the shared latent in compatible coordinates. As shown in Figure 7, cross-probing accuracy depends jointly on ϵ and P. At fixed P, increasing ϵ strengthens alignment, thus explaining (5). At fixed $\epsilon > 0$, increasing P allows the model to exploit the increasingly well-sampled cross-lingual signal. In contrast, at $\epsilon = 0$, increasing the amount of data alone does not produce substantial cross-probing accuracy: the model becomes geometrically similar across languages, yet never aligned.

This sample complexity has a simple interpretation. In monolingual RHM, building aligned representations of latents requires seeing pairs of features in the training set a sufficient number of times (Cagnetta & Wyart, 2024). Here, features from different languages are observed only through code-switched subtrees, and are therefore suppressed by a factor proportional to ϵ: a dataset of size P carries an effective cross-lingual signal of order ϵP. Reaching a fixed level of alignment requires ϵP to remain constant, predicting $P_{\text{threshold}} \propto \epsilon^{-1}$, in agreement with the boundary of the aligned region in Fig. 7 (dashed line).

8 CONCLUSION

We have shown that a single assumption –that data possess a hidden hierarchy, with semantic content residing at its top and language-specific structure at its bottom– accounts for the phenomenology of multilingual representations: similarity peaking in the middle layers, coexisting with language-specific structure, strengthening with language proximity, model quality and data exposure, and distinct from the stronger property of alignment, which requires cross-lingual signal in the training data. Since the theory is not specific to language, two modalities can be viewed as two surface realizations of a shared upper hierarchy, suggesting that the same analysis applies to the cross-modal similarities motivating the platonic representation hypothesis. It also carries a practical implication: alignment is purchased with mixed-language signal, identifying code-switched data as the efficient lever for low-resource languages.

A central consequence of the theory is constructive: it prescribes how to build more abstract representations, by removing from each layer what is predictable from the previous one – eliminating surface details to isolate newly computed, higher-level information. This layerwise novelty may prove useful in AI to study how abstractions are processed, but also in neuroscience. Indeed, LLM representations are currently the best predictors of brain activity during language processing (Caucheteux & King, 2022), and novelty may offer a principled, depth-resolved improvement.

AI USE STATEMENT

In this work, we used generative AI tools (ChatGPT, Claude) in two capacities. First, as writing assistants, for grammar checking and local rephrasing of author-written text; no section of the paper was generated by an AI system, and all scientific claims, derivations and interpretations are the authors' own. Second, as coding assistants for analysis, plotting scripts and consolidating supplementary material; all AI-assisted code was reviewed, tested and verified for correctness by at least one author, and the reported numerical results were validated independently of the tools used to write the code. We did not use generative AI for research ideation, experimental design, calculations, or making figures. We have reviewed all AI-assisted content and take full responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is a theoretical and empirical study of representations in language models, using synthetic data, and publicly available models and corpora (Europarl). It involves no human subjects or personal data, and we foresee no direct ethical concerns.

REPRODUCIBILITY STATEMENT

The multilingual RHM and the BP construction are fully specified in Sections 3 and 4 and Appendix B; transformer training details are given in Appendix C; similarity measures are defined in Appendix D; and the natural-language pipeline (models, Europarl data, novelty extraction) is described in Appendix I. Code to generate multilingual RHM data, train transformers, construct the theoretical networks and reproduce all main results is provided as supplementary material.

REFERENCES

Santiago Acevedo, Andrea Mascaretti, Riccardo Rende, Matéo Mahaut, Marco Baroni, and Alessandro Laio. A quantitative analysis of semantic information in deep representations of text and images. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=sBnaFSIuGR.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, learning hierarchical language structures. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=mPQKyzkA1K.

Francesco Cagnetta and Matthieu Wyart. Towards a theory of how the structure of language is acquired by deep neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 83119–83163. Curran Associates, Inc., 2024. doi: 10.52202/079017-2645. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9740da1c07c7b451af14e11523f95271-Paper-Conference.pdf.

Francesco Cagnetta, Leonardo Petrini, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Phys. Rev. X*, 14:031001, Jul 2024. doi: 10.1103/PhysRevX.14.031001. URL https://link.aps.org/doi/10.1103/PhysRevX.14.031001.

Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.07488.

Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5:134, 2022. doi: 10.1038/s42003-022-03036-1.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 119–136. Association for Computational Linguistics, December 2022.

N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956. doi: 10.1109/TIT.1956.1056813.

Eric De Giuli. Random language model. *Physical Review Letters*, 122(12):128301, 2019. doi: 10.1103/PhysRevLett.122.128301.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31822–31841. Association for Computational Linguistics, July 2025.

Jerome Garnier-Brun, Marc Mezard, Emanuele Moscato, and Luca Saglietti. How transformers learn structured data: Insights from hierarchical filtering. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18831–18847. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/garnier-brun25a.html.

Aldo Glielmo, Claudio Zeni, Bingqing Cheng, Gábor Csányi, and Alessandro Laio. Ranking the information content of distance measures. *PNAS Nexus*, 1(2):pgac039, 05 2022. ISSN 2752-6542. doi: 10.1093/pnasnexus/pgac039. URL https://doi.org/10.1093/pnasnexus/pgac039.

Ulf Grenander and Michael I. Miller. *Pattern Theory: From Representation to Inference*. Oxford University Press, 2007.

Fabian Gröger, Shuo Wen, and Maria Brbic. Revisiting the platonic representation hypothesis: An aristotelian view. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=uz0gAAYydl.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Tatsuro Inaba, Go Kamoda, Kentaro Inui, Masaru Isonuma, Yusuke Miyao, Yohei Oseki, Yu Takagi, and Benjamin Heinzerling. How a bilingual LM becomes bilingual: Tracing internal representations with sparse autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13458–13470, Suzhou, China, November 2025. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Alexander Jones, William Yang Wang, and Kyle Mahowald. A massively multilingual analysis of cross-linguality in shared embedding space. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5833–5847, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.471. URL https://aclanthology.org/2021.emnlp-main.471/.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pp. 79–86, Phuket, Thailand, 2005. URL https://aclanthology.org/2005.mtsummit-papers.11/.

A. Sophia Koepke, Daniil Zverev, Shiry Ginosar, and Alexei A. Efros. Back into plato's cave: Examining cross-modal representational convergence at scale, 2026. URL https://arxiv.org/abs/2604.18572.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Danni Liu and Jan Niehues. Middle-layer representation alignment for cross-lingual transfer in fine-tuned LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15979–15996. Association for Computational Linguistics, July 2025.

Eran Malach and Shai Shalev-Shwartz. A provably correct algorithm for deep learning that actually works. *arXiv preprint arXiv:1803.09522*, 2018.

Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, Mateusz Klimaszewski, Duarte M. Alves, José Pombal, Nicolas Boizard, Manuel Faysse, Pierre Colombo, François Yvon, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm-9b: Technical report, 2025. URL https://arxiv.org/abs/2506.04079.

Zhongtao Miao, Qiyu Wu, Kaiyan Zhao, Zilong Wu, and Yoshimasa Tsuruoka. Enhancing cross-lingual sentence embedding for low-resource languages with word alignment. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3225–3236, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.204. URL https://aclanthology.org/2024.findings-naacl.204/.

Elchanan Mossel. Deep learning and hierarchal generative models. *arXiv preprint arXiv:1612.09057*, 2016.

David Mumford and Agnès Desolneux. *Pattern Theory: The Stochastic Analysis of Real-World Signals*. A K Peters/CRC Press, 2010.

Jack T. Parley, Francesco Cagnetta, and Matthieu Wyart. Deep networks learn to parse uniform-depth context-free languages from local statistics. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mJgkPAFdiK.

Qiwei Peng and Anders Søgaard. Concept space alignment in multilingual LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5511–5526, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.315. URL https://aclanthology.org/2024.emnlp-main.315/.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL https://aclanthology.org/P19-1493/.

Antonio Sclocchi, Alessandro Favero, Noam Itzhak Levi, and Matthieu Wyart. Probing the latent hierarchical structure of data via diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.13770.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122 (1):e2408799121, 2025b. doi: 10.1073/pnas.2408799121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2408799121.

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. BERT is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pp. 47–55. Association for Computational Linguistics, November 2019.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961. Version 2, revised April 2026.

Mariya Toneva, Tom M. Mitchell, and Leila Wehbe. Combining computational controls with natural text reveals aspects of meaning composition. *Nature Computational Science*, 2(11):745–757, 2022. doi: 10.1038/s43588-022-00354-6.

Zhijun Wang, Jiahuan Li, Hao Zhou, Rongxiang Weng, Jingang Wang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. Investigating and scaling up code-switching for multilingual language model pre-training. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11032–11046, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.575. URL https://aclanthology.org/2025.findings-acl.575/.

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 (Volume 1: Long Papers)*, pp. 15366–15394. Association for Computational Linguistics, August 2024.

Shijie Wu and Mark Dredze. Are all languages created equal in multilingual BERT? In Spandana Gella, Johannes Welbl, Marek Rei, Fabio Petroni, Patrick Lewis, Emma Strubell, Minjoon Seo, and Hannaneh Hajishirzi (eds.), *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 120–130, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.16. URL https://aclanthology.org/2020.repl4nlp-1.16/.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16513–16542, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.1029. URL https://aclanthology.org/2023.emnlp-main.1029/.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. How do large language models handle multilingualism? In *Advances in Neural Information Processing Systems*, volume 37, pp. 15296–15319, 2024. doi: 10.52202/079017-0489. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1bd359b32ab8b2a6bbafa1ed2856cf40-Paper-Conference.pdf.

Zhiling Zhou, Tianhao Wang, and Zhuoran Yang. How transformers represent hierarchies: A local-to-global mechanism. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ScLLtpPoaE.

A RHM SCHEMATIC

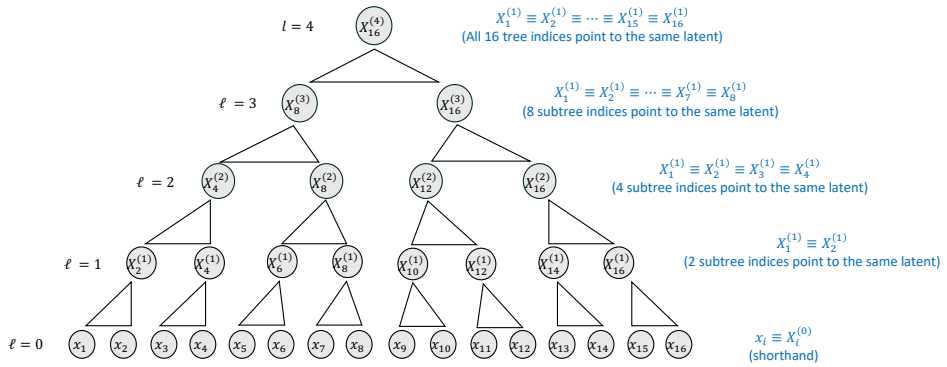


Figure 8: Schematic of RHM with $L = 4$. Visible tokens, i.e. leaves, are denoted as $x_i \equiv X_i^{(0)}$. Level-ℓ parent node of i-th token is denoted as $X_i^{(\ell)}$ – this introduces a redundancy in notation since this latent can be denoted with any of the 2^ℓ subscripts, corresponding to the tokens in its subtree.

B BELIEF PROPAGATION AND THEORETICAL NETWORKS

We describe the belief-propagation (BP) algorithm underlying § 4 and its representation as a theoretical network. Figure 9 illustrates the computation: upward messages summarize evidence within observed subtrees, while downward messages propagate information along the boundary leading to x_{t+1}.

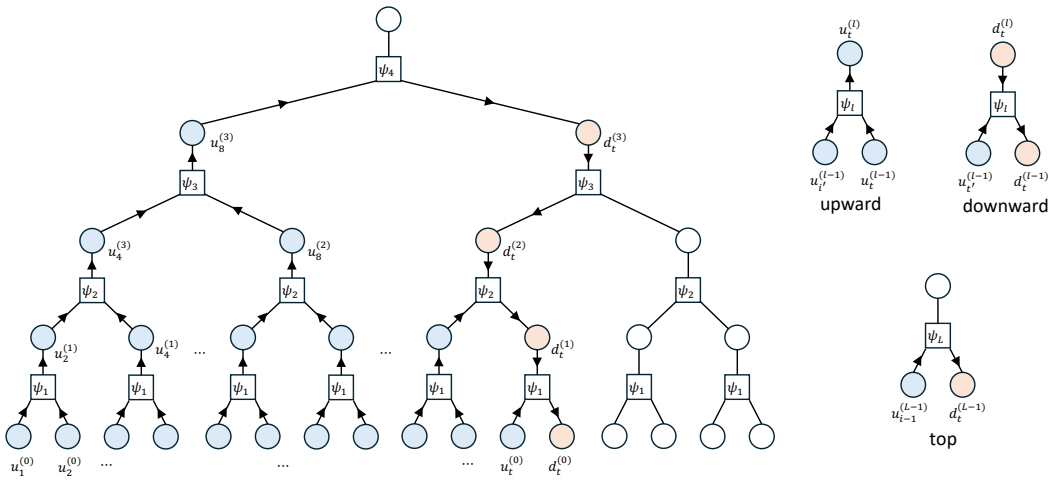


Figure 9: **Belief propagation for causal next-token prediction.** Blue nodes denote upward messages summarizing evidence within observed subtrees; orange nodes denote downward messages along the path to x_{t+1}. Unfilled nodes are unobserved, and squares denote grammar factors ψ_ℓ. The right panels show the local upward, downward, and top-level updates.

Notation and grammar factors. Each message is a vector in $\mathbb{R}^v$. Its level and position, (ℓ, i), uniquely identify the corresponding latent node $X_i^{(\ell)}$. When specifying the BP recursions, we write $[\mathbf{u}_i^{(\ell)}]_a$ for the component associated with the assignment $X_i^{(\ell)} = a$. Thus, a indexes a component of the message. We use i for a generic position; setting $i = t$ recovers the causal-boundary messages used in the main text.

For $1 \le \ell \le L-1$, let $\psi_\ell(b, \mathbf{a})$ indicate whether $b \to \mathbf{a}$ is a valid production rule, where $b \in \mathcal{A}$ and $\mathbf{a} = (a_1, \ldots, a_s) \in \mathcal{A}^s$:

$$\psi_\ell(b, \mathbf{a}) = \begin{cases} 1, & b \to \mathbf{a} \text{ is a valid rule at level } \ell, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Uniform rule probabilities can be omitted because all messages are normalized.

The top factor is root-agnostic:

$$\psi_L(\mathbf{a}) = \begin{cases} 1, & \mathbf{a} \text{ is a valid top-level tuple,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Indeed, sampling one of v root symbols followed by one of its m rules is equivalent to using a single dummy root with vm possible expansions. Thus, no prior or message over the root is required.

The updates below L_{surface} use language-specific production rules. At the split, a shared parent has m admissible expansions in each language, while the production rules above the split are shared. All messages below depend implicitly on the observed prefix $x_{1:t}$.

Upward messages. Consider a level-ℓ node $X_i^{(\ell)}$ with ordered children $X_{i_1}^{(\ell-1)}, \ldots, X_{i_s}^{(\ell-1)}$. Its upward message summarizes the evidence within its subtree:

$$\left[\mathbf{u}_i^{(\ell)}\right]_b = \sum_{\mathbf{a} \in \mathcal{A}^s} \psi_\ell(b, \mathbf{a}) \prod_{j=1}^{s} \left[\mathbf{u}_{i_j}^{(\ell-1)}\right]_{a_j}, \qquad 1 \le \ell \le L-1. \tag{5}$$

The recurrence is initialized at the leaves by

$$\left[\mathbf{u}_i^{(0)}\right]_a = \begin{cases} \delta_{a,x_i}, & i \le t, \\ 1/v, & i > t. \end{cases} \tag{6}$$

Observed leaves therefore provide point-mass evidence, while unobserved leaves are marginalized.

At transformer position t, $\mathbf{u}_t^{(\ell)}$ denotes the message for the level-ℓ subtree intersected by the causal boundary. If this subtree is only partially observed, Equation 6 marginalizes its unobserved leaves. Strictly, $\mathbf{u}_t^{(\ell)}$ is the likelihood of the observed descendants as a function of the latent symbol; we refer to its normalized form as a belief.

Downward messages. The downward computation follows the path from the top of the hierarchy to x_{t+1}. At each factor, let a denote the symbol of the child on this path and let $\mathbf{a}_{\text{sib}} = (a_1, \ldots, a_{s-1})$ denote its siblings. The position of the target child within the ordered tuple is fixed by t and is left implicit. We write $\mathbf{u}_{\text{sib},j}^{(\ell)}$ for the upward message from its j-th sibling.

The first downward update is performed directly by the root-agnostic factor:

$$\left[\mathbf{d}_t^{(L-1)}\right]_a = \sum_{\mathbf{a}_{\text{sib}} \in \mathcal{A}^{s-1}} \psi_L(a, \mathbf{a}_{\text{sib}}) \prod_{j=1}^{s-1} \left[\mathbf{u}_{\text{sib},j}^{(L-1)}\right]_{a_j}. \tag{7}$$

Here $\psi_L(a, \mathbf{a})$ denotes the top-level factor evaluated on the ordered tuple formed by the target child and its siblings. No root prior or root message is required.

At lower levels, the update additionally incorporates the downward message from the parent:

$$\left[\mathbf{d}_t^{(\ell-1)}\right]_a = \sum_{b \in \mathcal{A}} \sum_{\mathbf{a}_{\text{sib}} \in \mathcal{A}^{s-1}} \left[\mathbf{d}_t^{(\ell)}\right]_b \psi_\ell(b; \mathbf{a}) \prod_{j=1}^{s-1} \left[\mathbf{u}_{\text{sib},j}^{(\ell-1)}\right]_{a_j}, \qquad 1 \le \ell \le L-1. \tag{8}$$

The downward message is formally a cavity belief containing evidence from outside the target subtree. The leaf-level message is the next token posterior

$$\left[\mathbf{d}_t^{(0)}\right]_a = p(x_{t+1} = a \mid x_{1:t}). \tag{9}$$

Theoretical-network representations. The BP messages are assembled into layerwise representations as defined in Equations 1 and 2. We realize these direct sums in a fixed ambient space by assigning each message level its own block and setting inactive blocks to zero.

To avoid imposing coordinate alignment, we embed the two languages into orthogonal subspaces. For a message vector $\mathbf{m} \in \mathbb{R}^v$, define

$$\iota_A(\mathbf{m}) = \begin{bmatrix} \mathbf{m} \\ \mathbf{0} \end{bmatrix}, \qquad \iota_B(\mathbf{m}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \in \mathbb{R}^{2v}. \tag{10}$$

We apply this convention even when $\mathbf{m}$ describes a shared latent variable. It preserves the local geometry of the messages within each language without placing the two languages in the same coordinates.

Alternative message representations and algorithms. As discussed in the main text and Eq. 2, while constructing the later layers of the theoretical network we decay the upward messages after their usage with a factor $\gamma \in [0, 1]$ at each step. Here $\gamma = 0$ recovers immediate deletion, whereas $\gamma = 1$ retains all upward messages. While this choice does not affect the qualitative results of the paper, the best quantitative match to trained transformers is achieved at intermediate values of γ. We make one such choice in the main text, namely $\gamma = 1/2$. This is a direct consequence of the residual structure of transformers, where skip connections prevent immediate removal of messages. We compare the layerwise similarity behaviours for different γ in Fig. 11

Finally, a parsimonious implementation encodes a level-ℓ upward message only after its subtree is complete, i.e. when t is divisible by s^ℓ. The construction used in the main text also represents partial-subtree messages by marginalizing unobserved leaves. This choice does not qualitatively affect the results as we show in Fig. 12.

Depth-limited inference. With finite training data, transformers recover the production rules progressively from the surface upward (Cagnetta & Wyart, 2024). We model a network that has learned the rules only through level $\ell_* < L$ by replacing every unresolved factor with a uniform factor over all possible productions:

$$\psi_\ell^{\text{unknown}}(b, \mathbf{a}) = 1, \qquad b \in \mathcal{A}, \quad \mathbf{a} \in \mathcal{A}^s, \quad \ell > \ell_*. \tag{11}$$

Thus, every child tuple is considered equally possible, including tuples that are invalid under the true grammar.

Consequently,

$$\left[\mathbf{u}_t^{(\ell)}\right]_b = \frac{1}{v}, \qquad \ell > \ell_*, \qquad \text{and} \qquad \left[\mathbf{d}_t^{(\ell_*)}\right]_b = \frac{1}{v}. \tag{12}$$

The uniform message at level ℓ_* provides the boundary condition for the downward recursion, which then proceeds through the learned factors at levels $\ell_*, \dots, 1$.

Unlike a truncated computation, this construction retains the complete $(2L-1)$-step BP computation. Updates involving unresolved levels carry only uniform, sample-independent messages and therefore introduce no additional information into the representations.

When $\ell_* < L_{\text{surface}}$, only language-specific rules have been learned. Cross-lingual similarity emerges when ℓ_* reaches L_{surface} and strengthens as further shared levels are recovered. Distinct depths ℓ_*^A and ℓ_*^B model unequal exposure to the two languages.

B.1 ENCODER-ONLY SHORTCUT, I.E. ONE-STEP DECODING.

Fixed-topology grammars admit an $(L+1)$-step shortcut in which the upward messages are combined directly to predict the next token, avoiding a sequential downward pass (Zhou et al., 2026). We call this an "encoder-only" computation (referring to the computational algorithm, not the transformer architecture).

Depending on the training-set size and code-switching rate ϵ, trained transformers approach either this shortcut or the full encoder–decoder computation. Code-switching requires the predictor to determine which language-specific rules govern individual subtrees, favoring an explicit downward computation; we study this transition in Appendix E. Moreover, the shortcut relies on knowing the

tree topology in advance. In Appendix G, we show that varying-topology grammars require the full encoder–decoder computation. Since natural language is generally modeled using variable-topology parse trees, this motivates our focus on $\epsilon = 0.1$, for which trained transformers consistently exhibit the encoder–decoder structure.

Here, we recall for completeness how the transformer can implement an encoder-only computation, as discussed in (Zhou et al., 2026). Indeed, encoder-only algorithm composes all downward updates into a single function evaluated by the final transformer block. After the encoding phase, the network has computed the messages from the sibling sub-trees that contain observed tokens. Denote these messages by

$$u_{0,t}, u_{1,t}, \ldots, u_{L-1,t}.$$

For a fixed tree topology, the subtree associated with each message and the position at which it is stored are deterministic functions of t.

The usual BP decoder processes these messages sequentially. Schematically,

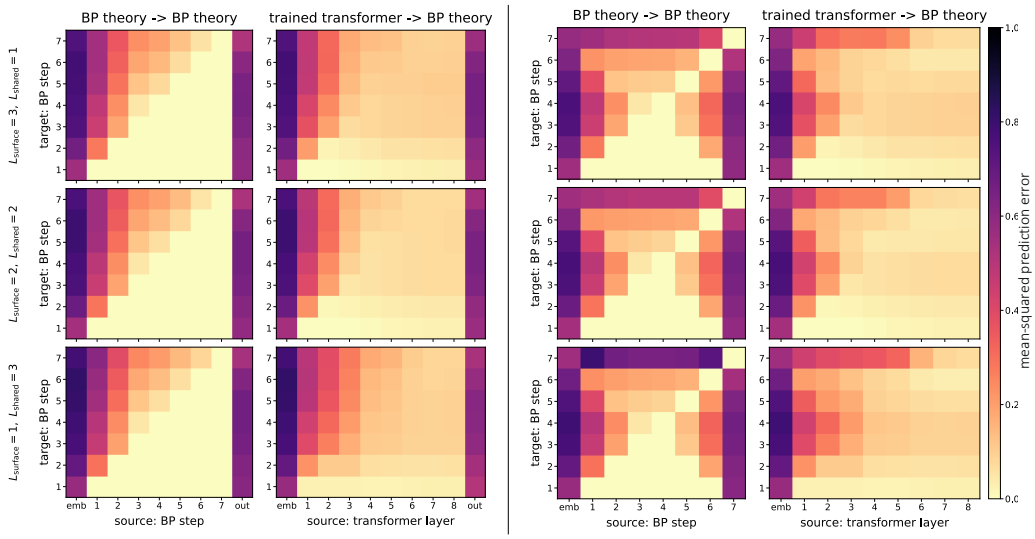
$$d_{L-1,t} = f_{L-1}(u_{L-1,t}), \qquad d_{\ell-1,t} = f_{\ell-1}(d_{\ell,t}, u_{\ell-1,t}),$$

where $d_{0,t} = p(x_{t+1} \mid x_{1:t})$. Substituting each update into the next eliminates the intermediate downward messages and gives

$$d_{0,t} = F(u_{0,t}, u_{1,t}, \ldots, u_{L-1,t})$$

for a fixed function F. Thus, sequential decoding and one-step decoding differ only in how the same function is evaluated. Zhou et al. (2026) provides an explicit construction showing that a single transformer block can perform this computation.

B.2 Transformers approximate BP computation



(a) $\gamma = 1/2$ (b) $\gamma = 0$

Figure 10: Mean-squared error when predicting layerwise representations from one network using another, for $\gamma = 1/2$ **(a)** and $\gamma = 0$. **(b)** BP encodings. Transformer $\rightarrow$ theory predictions confirm the expected layerwise encoding of BP messages in trained transformers. These heatmaps also match with theory $\rightarrow$ theory predictions, further solidifying the correspondence. The match is better for $\gamma = 1/2$.

Here we provide direct evidence that transformers trained on multilingual RHM data approximately implement the BP algorithm described above. We do so by linearly predicting the representations of the theoretical network from transformer representations. Figure 10 shows a heatmap of mean-squared error predicting each layer ℓ of the theoretical network from each transformer layer l. Indeed, the transformer layers accurately predict theoretical representations corresponding to BP

messages it is expected to encode. We compare this heatmap with a prediction from one theoretical model to another (this is trivial on the diagonal, but nontrivial elsewhere). The two prediction heatmaps, transformer $\rightarrow$ theory and theory $\rightarrow$ theory show similar layerwise behaviour, further strengthening the correspondence. Note that the correspondence is especially clear for the encoding of BP where used messages decay at a rate $\gamma = 1/2$ compared to $\gamma = 0$ where messages are immediately eliminated, as shown in panel (b) of Figure 10.

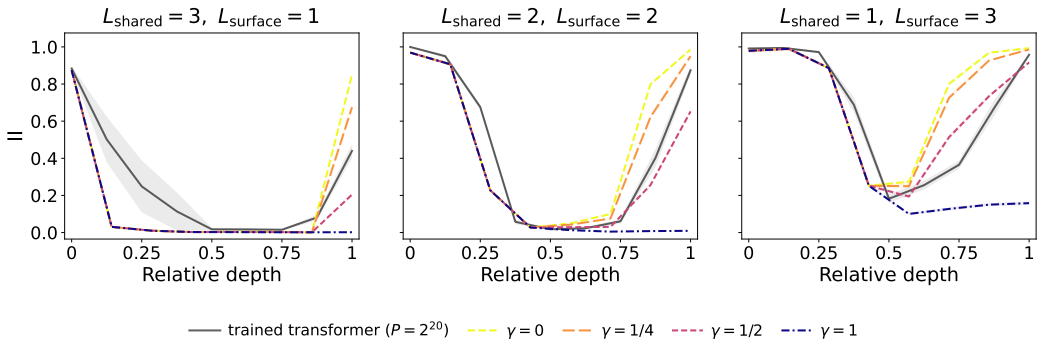


Figure 11: Comparing layerwise II profiles for different $\gamma \in [0, 1]$. Empirical II curves from trained transformers shown for reference.

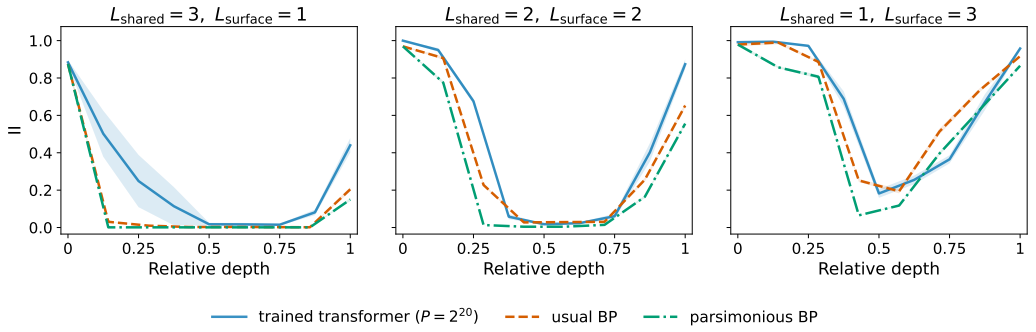


Figure 12: II for parsimonious implementation of BP where the only latent messages stored correspond to completely observed subtrees – compared with the ordinary BP and trained transformers. The qualitative behaviour is unaffected.

C Training Details

We trained 8-layer decoder-only transformers with embedding dimension 512, four attention heads, and ReLU activations using next-token cross-entropy loss. Optimization used AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$) with peak learning rate 10^{-3}, weight decay 10^{-2}, and batch size 1024. We used a warmup–stable–decay schedule: linear warmup over the first 10% of training, a constant learning rate for the middle 80%, and cosine decay over the final 10% to 10^{-2} times the peak rate. Models were trained for up to 128 epochs, and analyses used the checkpoint with the lowest next-token validation loss. Reported results are averaged over seeds $42, 420, 4200$, and 42000. For testing, we use a held-out set of 2^{15} sentences.

In Fig. 13 we show the test loss of trained transformers as a function of the training set size P. The stepwise decrease in the loss follows theoretical entropies calculated with depth-limited BP.

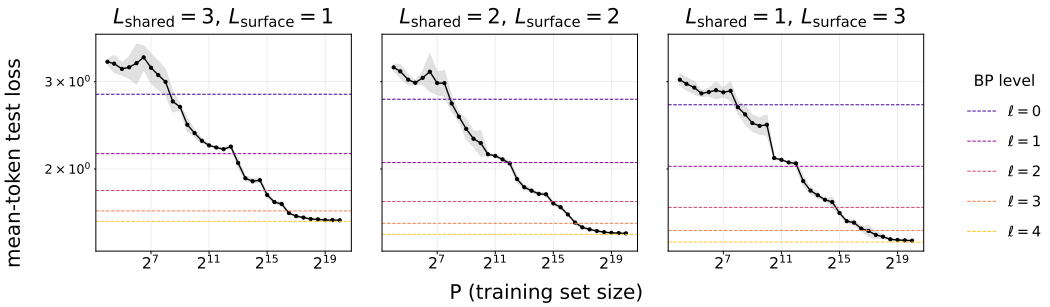


Figure 13: Test loss for transformers trained on multilingual RHM data with $\epsilon = 0.1$ and different training set sizes P. The loss scales with P, showing sequential plateaus at the theoretical entropies computed with depth-limited BP.

D SIMILARITY MEASURES

We measure representational similarity using information imbalance (II) and mutual k-nearest-neighbor overlap (mKNN). These quantities capture neighbourhood similarity – II looks at the nearest neighbour, mKNN compares a larger local neighborhood.

The "network" may be either a trained transformer or the theoretical network constructed by encoding BP messages. We generate N pairs of translated sentences $\{(\mathbf{x}_n^A, \mathbf{x}_n^B)\}_{n=1}^N$ in languages A and B. At layer l, we extract the corresponding token-averaged representations $\mathbf{h}_n^{A,(l)}, \mathbf{h}_n^{B,(l)} \in \mathbb{R}^{d_l}$, and stack them into matrices $\mathbf{H}^{A,(l)}, \mathbf{H}^{A,(l)} \in \mathbb{R}^{N \times d_l}$,

For II and mKNN measurements in the RHM setup, we set $N = 2048$.

Information imbalance. Information imbalance asks whether the closest neighbor of each sentence in one representation remains nearby in the other. For each sentence n, let

$$j_A(n) = \arg\min_{j \neq n} d_A\left(\mathbf{h}_n^{A,(l)}, \mathbf{h}_j^{A,(l)}\right)$$

be its nearest neighbor in space A, and let $r_B(n, j)$ denote the rank of sentence j among the neighbors of n in space B, with rank one denoting the nearest neighbor. The directional information imbalance is

$$\Delta(A \to B) = \frac{2}{N^2} \sum_{n=1}^{N} r_B(n, j_A(n)) \,.$$

Thus, II first identifies a local relationship in A and then measures how well that relationship is preserved in B. If the two representations have similar local geometry, $j_A(n)$ will also have a low rank in B, giving a small value of $\Delta(A \to B)$. If their neighborhood structures are independent, the expected rank is approximately $N/2$, and the normalization gives $\Delta(A \to B) \approx 1$.

II is directional: $\Delta(A \to B)$ need not equal $\Delta(B \to A)$. For example, one representation may separate sentences that are indistinguishable in the other. In the main figures, we report the symmetrized value

$$\overline{\Delta}(A, B) = \frac{1}{2} \left[\Delta(A \to B) + \Delta(B \to A)\right] .$$

Mutual k-nearest-neighbor overlap. Whereas II follows only the closest neighbor, mKNN compares the full set of the k nearest neighbors of each sentence. Let $\mathcal{N}_A^k(n)$ and $\mathcal{N}_B^k(n)$ denote these sets in the two representation spaces. We define

$$\text{mKNN}_k(A, B) = \frac{1}{N} \sum_{n=1}^{N} \frac{\left|\mathcal{N}_A^k(n) \cap \mathcal{N}_B^k(n)\right|}{k}.$$

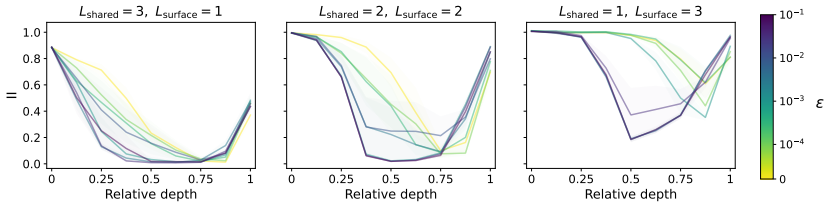
The score lies between 0 and 1. A value of 1 means that every sentence has exactly the same k nearest neighbors in the two spaces, while smaller values indicate less agreement between their local geometries. Unlike II, mKNN is symmetric by construction.

We use $k = 10$ and identify neighbors using cosine similarity after ℓ_2-normalizing each representation. Exact distance ties occur more frequently in the discrete BP representations than in the trained networks. We therefore compute the expected overlap under independent uniform tie breaking, avoiding artificial agreement caused by deterministic, index-based tie breaking.

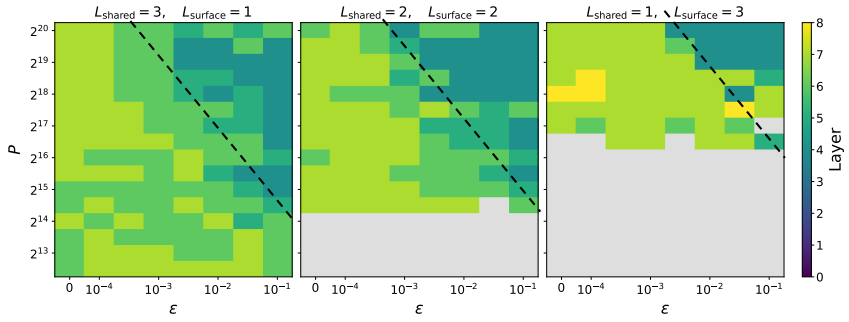
The scale probed by mKNN depends on the choice of k. Moreover, for independent neighborhood structures its expected value is approximately $\frac{k}{N-1}$. Thus, unlike the normalized II score, raw mKNN values depend explicitly on both k and the evaluation-set size N. Comparisons across training-set sizes P are valid in our experiments because we hold N and k fixed, but mKNN is not an intrinsic, parameter-free measure of neighborhood agreement in the same sense as II.

E ALGORITHMIC TRANSITION WITH ϵ AND P

In trained transformers the transition between encoding-decoding and encoder-only behaviour is governed by two factors: the code-switching rate ϵ and training set size P. One way to study this transition is to train transformers for many different P and ϵ values, and identify the layer that shows the lowest II, and therefore the highest similarity. The Encoder-decoder design predicts highest similarity in the middle layers, whereas encoder-only design predicts that at the penultimate layer. Indeed, in Fig. 14 we show that this observable can identify the algorithmic transition with ϵ and P.



(a) II profiles for various ϵ, for fixed $P = 2^{20}$. encoder-only to encoding-decoding transition can be seen with increasing ϵ.



(b) Tracking the transformer layer with the lowest II shows the algorithmic transition wrt ϵ and P. Each grid-box is a separately trained transformer. Cases where minimum II was greater than 0.8 are grayed out. Comparing this with Fig. 7 in the main text, we find that the algorithmic transition approximately co-occurs with coordinate alignment.

Figure 14: Transition from encoder-only to encoding-decoding computations in transformers trained on multilingual RHM data with various ϵ and P. Interestingly, the algorithmic transition co-occurs with coordinate alignment (Fig. 7).

F VARYING THE NUMBER OF TRANSFORMER LAYERS

The correspondence between trained transformers and theoretical networks constructed with BP does not require identical number of layers between the two networks. This is already clear since all experiments in the main text use an 8-layer transformer, while the theoretical network has 7 layers,

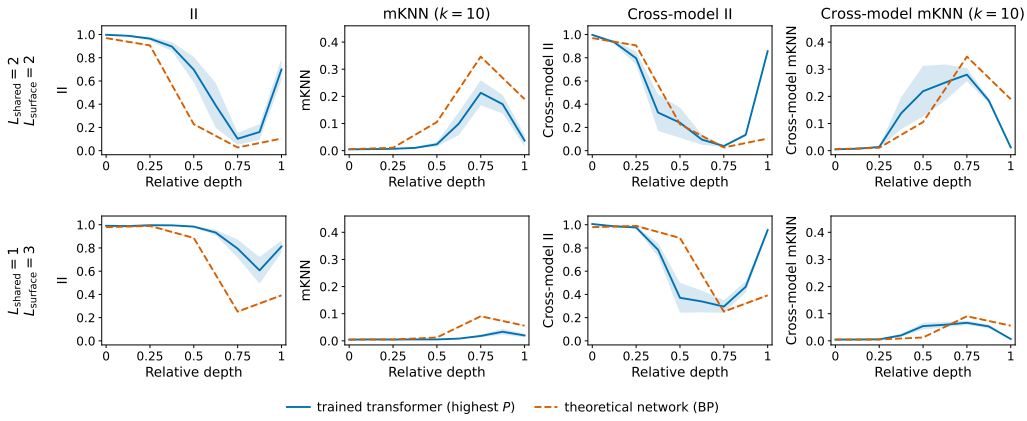


Figure 15: $\epsilon = 0$ version of Fig. 2 in the main text. In this case, the layerwise similarity trends in trained transformers are better captured by that of a theoretical network constructed with an $(L{+}1)$-step encoder-only BP algorithm.

corresponding to the $2L - 1 = 7$ steps of the encoding-decoding BP computation. Transformers tend to "stretch" the same computation over all available layers, resulting in dilation of the expected layerwise trends. We explicitly test this by training transformers of varying depths on the same multilingual RHM data. As shown in Fig. 16, the layerwise similarity trends remain qualitatively unchanged with varying transformer depth. Note that a minimum transformer depth is still required for the computation, below which the performance would suffer from suboptimal predictions.

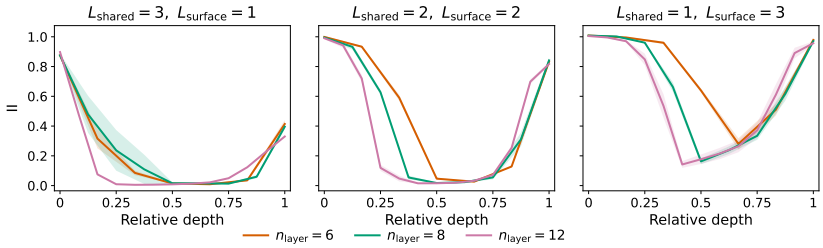


Figure 16: Layerwise similarity profiles with varying number of transformer layers. The trend remains qualitatively unchanged, evidencing that transformers stretch the BP computation steps among its layers when more depth is available.

G VARYING TOPOLOGY

We study the varying tree topology RHM (Parley et al., 2026) and extend it to MRHM.

Varying-tree MRHM. We keep the notation of Section 3, but each node now has two or three children. At each level, every symbol $b \in \mathcal{A}$ has m_2 binary rules $b \to (a_1, a_2)$ and m_3 ternary rules $b \to (a_1, a_2, a_3)$, whose child tuples are sampled without replacement from $\mathcal{A}^2$ and $\mathcal{A}^3$, so each tuple has a unique parent. Each node draws an arity $s \in \{2, 3\}$ with probability p_s, then one of its m_s s-ary rules uniformly; $p_2 \in \{0, 1\}$ recovers the fixed-tree RHM. A level-ℓ node thus spans between 2^ℓ and 3^ℓ tokens, and sentence lengths range over $[2^L, 3^L]$, so sentence length varies and the segmentation of the token string into subtrees is not known in advance: the tree topology must be inferred along with the symbols. Ambiguity is controlled by the fractions of admissible tuples, $f_2 = m_2/v$ and $f_3 = m_3/v^2$. The multilingual version shares the rules above L_{surface}. A translation pair shares its upper tree, including its topology, while its lower subtrees differ.

Encoder–decoder structure with varying topology. Figure 17 shows that cross-lingual II reaches its minimum at intermediate transformer layers and increases again toward the output. This profile is consistent with an encoder–decoder computation, in which shared abstractions emerge during encoding, followed by decoding into language-specific predictions.

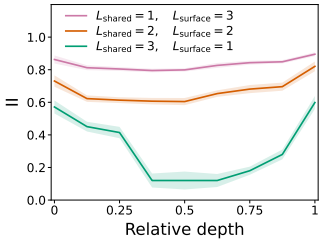


Figure 17: Cross-lingual similarity with varying tree topology. II is shown against relative transformer depth for three splits of a generative hierarchy with $L = 4$. Across all three splits, II reaches an intermediate-layer minimum and rises toward the output, consistent with the encoder–decoder picture.

H Coordinate alignment

We train linear probes from transformer representations of RHM sentences to the corresponding latent variables that generated the sentences. Each transformer layer (l), and each latent level (ℓ) and position (i) has a separate probe, which linearly classifies the latent variable. The test accuracy of this probing tells us whether latents at a given RHM level are encoded in a given transformer layer. For probing a latent, we specifically focus on the representations at the token index i that corresponds to the last token of the subtree spanned by that latent – so that all the information about that latent is causally available to the transformer representation.

To measure coordinate alignment, we use *cross-probing*: independent probes are trained on (pure) language-A and language-B sentences. We test language-A probes on language-B sentences and vice-versa. If we denote the trained linear probes, for latent level ℓ and position i, and transformer level l, by the weight $\mathbf{W}_i^{A(\ell,l)}$ and bias $b_i^{A(\ell,l)}$, then the cross-probing predictions can be written as

$$\hat{y}_i^{A\to B(\ell,l)} = \arg\max_z \left[\mathbf{W}_i^{A(\ell,l)}\mathbf{h}_i^{B(l)} + \mathbf{b}_i^{A(\ell,l)}\right]_z$$

Since the latent variables above the split level L_{surface} are shared, the probes in both languages decode the same information. For cross-probing to be successful, this shared information should be represented in shared coordinates.

Note that below $\ell = L_{\text{surface}}$ coordinates never align, because the latent variables themselves are language-specific. Therefore, we report the cross-probing accuracy at $\ell = L_{\text{surface}}$, which is the first shared layer. We find that once this level is aligned, the levels above also align.

The cross-probing results are shown in Fig. 7, where we find that the coordinate alignment occurs with increasing code-switching rate ϵ and training set size P.

I Layerwise Novelty

Definition. Let $h_{n,t}^{(k)} \in \mathbb{R}^{1\times d_k}$ denote the representation of token t in sentence n at layer k. We define layerwise novelty as the residual after subtracting an affine prediction based on the preceding layer:

$$r_{n,t}^{(k)} = h_{n,t}^{(k)} - f_k\left(h_{n,t}^{(k-1)}\right), \qquad f_k(h) = hW_k + b_k. \tag{13}$$

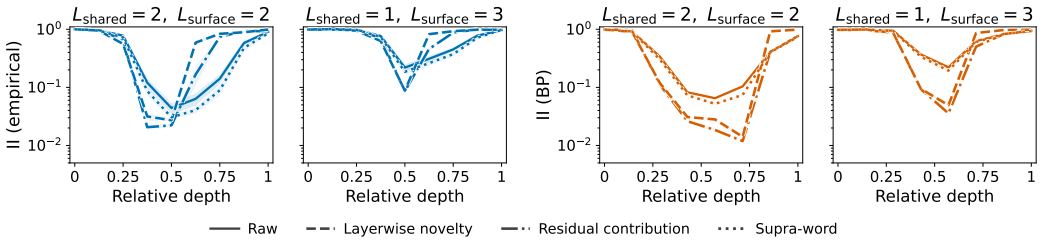
We average these residuals over valid tokens to obtain the sentence representation used for cross-lingual comparison.

Tokenwise affine fit. For each layer transition, we fit a single affine map shared across all token positions and sentences in the fitting set using ridge regression:

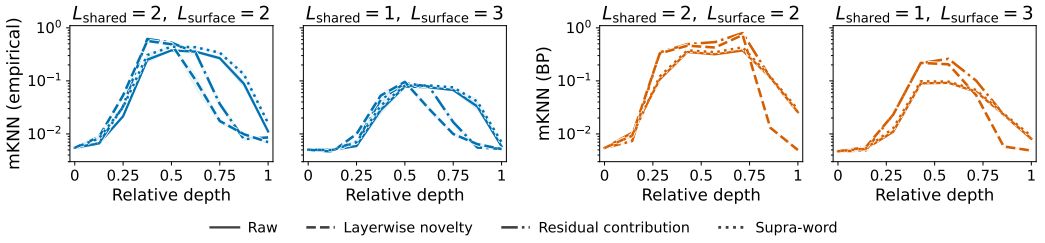
$$(W_k, b_k) = \arg\min_{W,b} \left\{ \frac{1}{S} \sum_{n=1}^{S} \frac{1}{T_n} \sum_{t=1}^{T_n} \left\| h_{n,t}^{(k)} - h_{n,t}^{(k-1)} W - b \right\|_2^2 + \lambda \|W\|_F^2 \right\}. \tag{14}$$

Here S is the number of fitting sentences, T_n is the number of valid tokens in sentence n, and λ is the regularization parameter, selected to minimize the affine map's mean squared prediction error on a validation set. The factor $1/T_n$ gives each sentence equal total weight.

Figure 18 extends the RHM analysis in Fig. 5 by reporting both II and mKNN for trained transformers and theoretical BP networks, with the residual contribution $h^{(k)} - h^{(k-1)}$ and the supra-word residual as additional baselines. Both metrics support the same qualitative picture: novelty enhances cross-lingual similarity during encoding but reduces it during decoding relative to raw representations, consistent with the removal of retained shared information.



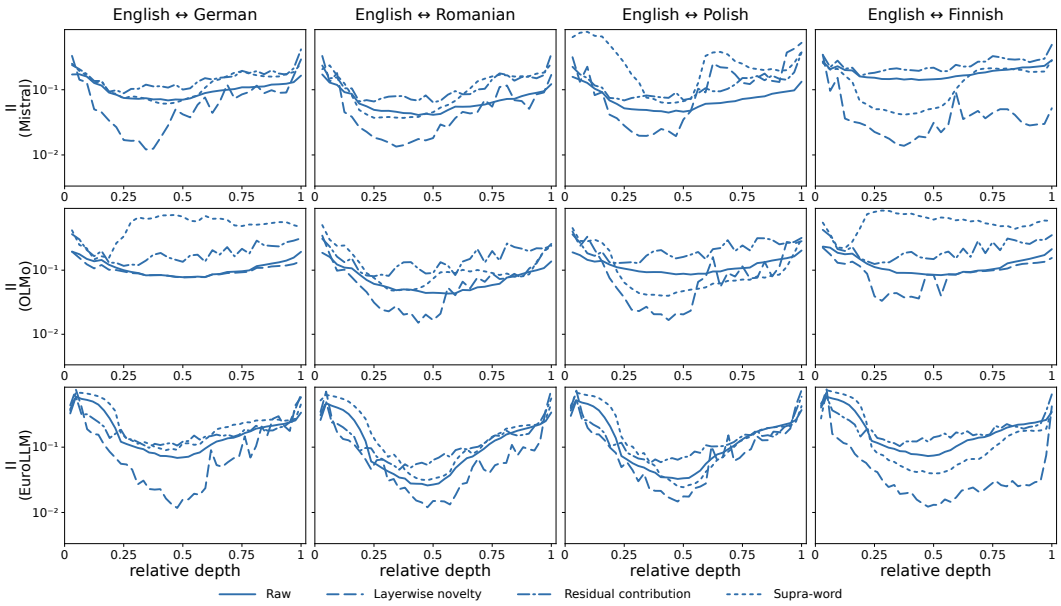
(a) II



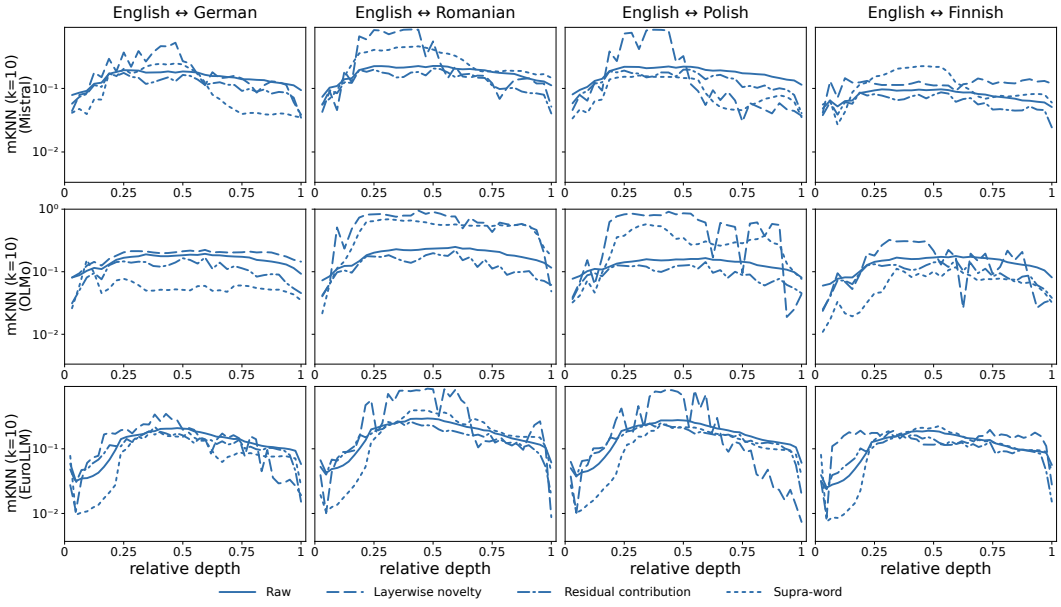
(b) mKNN

Figure 18: Extended version of Fig. 5. Cross-lingual geometric similarity in trained transformers and theoretical BP networks is measured using II **(a)** and mKNN **(b)** for raw representations, layerwise novelty, the residual contribution, and the supra-word residual. Lower II and higher mKNN indicate greater similarity.

Figure 19 extends the natural-language analysis to Mistral-7B-v0.1, OLMo-3-7B, and EuroLLM-9B (Martins et al., 2025) across four language pairs. Layerwise novelty generally yields lower II and higher mKNN than raw representations at intermediate layers, whereas improvements from the residual contribution and supra-word residual are less consistent across models and language pairs.



(a) II



(b) mKNN

Figure 19: Extended version of Fig. 6. Cross-lingual geometric similarity in Mistral-7B-v0.1, OLMo-3-7B, and EuroLLM-9B is evaluated on parallel English–German, English–Romanian, English–Polish, and English–Finnish sentences from Europarl. II **(a)** and mKNN **(b)** are measured for raw representations, layerwise novelty, the residual contribution, and the supra-word residual. Lower II and higher mKNN indicate greater similarity.