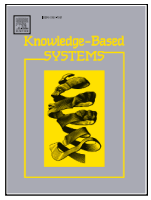
Knowledge-Based Systems 342 (2026) 115850

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



On the impact of pretraining data ordering in transformer encoder- and decoder-only language models

Luca Dini [a,*], Lucia Domenichelli [a,b], Dominique Brunato [a], Felice Dell'Orletta [a]

[a] Institute of Computational Linguistics "Antonio Zampolli" (CNR-ILC), ItaliaNLP Lab, Via Giuseppe Moruzzi, 1, Pisa, 56124, Italy
[b] University of Pisa, Largo Bruno Pontecorvo 3, Pisa, 56127, Italy

ARTICLE INFO

Keywords:
Curriculum learning
Data ordering
Language model pretraining
Linguistic representations
Representation geometry

ABSTRACT

Pretraining large language models typically relies on randomly ordered corpora, implicitly assuming that data order has limited impact on learning. However, curriculum learning suggests that the sequence of training examples can influence optimization and representation dynamics. In this work, we systematically examine pretraining data ordering as an independent design variable for transformer-based language models, analyzing how curriculum-inspired strategies affect learning trajectories, representations, and transfer performance.

We pretrain encoder-only and decoder-only models under controlled conditions, varying only the ordering of training data according to readability-based complexity proxies and their inverted variants, alongside multiple random baselines. Beyond final accuracy, we adopt a multi-dimensional evaluation framework combining intrinsic metrics, linguistic probing across training stages, downstream tasks, and geometric analyses of embedding spaces.

Results indicate architecture-dependent tendencies in response to data ordering. Encoder models generally exhibit stronger sensitivity to curriculum strategies, with noticeable differences in optimization behavior, probing dynamics, and representation geometry. Decoder models appear comparatively more stable under forward curricula, with more pronounced effects emerging under inverted orderings. Probing analyses suggest that early improvements reflect differences in data exposure rather than accelerated linguistic acquisition, while later-stage effects selectively mirror properties emphasized by specific curricula. Geometric analyses show that data ordering reshapes global variance structure, often increasing anisotropy, without substantially altering nonlinear intrinsic dimensionality.

Overall, data ordering functions as a selective inductive bias during pretraining, influencing learning dynamics and representational emphasis rather than consistently improving performance. These findings clarify how curriculum design interacts with transformer architectures and delineate its practical impact on pretraining outcomes.

1. Introduction

Pretraining neural language models (NLMs) has become a cornerstone of today's natural language processing (NLP) research, enabling models to acquire general-purpose linguistic knowledge that can be transferred to a wide range of downstream tasks. Transformer-based architectures, in particular, have demonstrated remarkable success when trained on large-scale text corpora. However, despite extensive interest in architectural and objective-level design choices, most pretraining pipelines rely on randomly ordered data, implicitly assuming that the sequence in which training examples are presented has little effect on the resulting model. In contrast, a growing body of work, particularly inspired by the foundational paper on curriculum learning (CL) [1], challenges this view by showing that non-random training data ordering can shape optimization trajectories, convergence speed, and training stability, as well as influence the quality and organization of learned representations [2–4]. These findings suggest that the order in which training data are presented to a language model is not a neutral design choice, but rather an additional degree of freedom in pretraining with the potential to significantly impact both model behavior and transfer performance.

In this study we expand this line of research by conducting a systematic and controlled investigation into the impact of data ordering strategies on the pretraining of transformer-based language models. We

* Corresponding author.
E-mail addresses: luca.dini@ilc.cnr.it (L. Dini), lucia.domenichelli@ilc.cnr.it (L. Domenichelli), dominique.brunato@ilc.cnr.it (D. Brunato), felice.dellorletta@ilc.cnr.it (F. Dell'Orletta).

https://doi.org/10.1016/j.knosys.2026.115850
Received 24 February 2026; Received in revised form 20 March 2026; Accepted 22 March 2026
Available online 24 March 2026

evaluate a range of ordering methods, from curriculum-inspired approaches that organize training data according to different complexity-related metrics, to multiple random-ordering baselines designed to account for variability inherent in stochastic training.

Recent initiatives have also emphasized structured and cognitively motivated training regimes, most notably efforts such as *BabyLM* [5,6] which bring together insights from linguistics, cognitive science, and machine learning to investigate how language can be acquired from limited and developmentally plausible input. A central concern in these settings is data efficiency, namely how much linguistic input is required for a model to acquire robust language competence under constrained conditions. In contrast, the present study focuses on data ordering, deliberately holding data volume, model capacity, and optimization settings fixed and varying only the sequence in which training examples are presented. Treating data efficiency and data ordering as independent factors allows us to isolate the effect of curriculum design on learning dynamics and representations in large-scale pretraining, positioning our study as methodologically distinct from, yet complementary to, cognitively motivated approaches.

To comprehensively characterize the effects of data ordering, we adopt a multi-dimensional evaluation framework spanning intrinsic, extrinsic and representational analyses, going beyond task-level metrics to examine how different curricula shape learning dynamics and learned representations.

Finally, the importance of data ordering becomes particularly salient when considered in light of the architectural diversity of neural language models. Encoder-only and decoder-only transformers differ fundamentally in their training objectives and inductive biases: bidirectional masked language modeling encourages the integration of global contextual information, whereas autoregressive modeling enforces a unidirectional, sequential prediction process. These differences suggest that curriculum-based data ordering strategies may interact with model architectures in non-uniform ways, potentially shaping learning dynamics and representations differently across training regimes. Motivated by this observation, we explicitly investigate data ordering effects across two widely adopted and architecturally distinct model families, specifically BERT [7] and GPT-2 [8] as representative of encoder and decoder-only models, under identical and tightly controlled experimental conditions.

The key contributions of our work are outlined below:

- We systematically study training data ordering as an independent and controllable design variable in transformer language model pretraining, and provide a controlled experimental setting that isolates ordering effects from data quantity, model capacity, and optimization choices.
- We provide a controlled comparison of curriculum-inspired data ordering strategies, varying both the complexity criterion (e.g., length- and readability-based proxies) and the direction of ordering, and evaluate their impact across intrinsic, extrinsic, and representational metrics.
- We introduce a multi-dimensional evaluation framework that combines intrinsic metrics, layer-wise probing, downstream task performance, and geometric analyses to characterize data ordering effects beyond final loss or accuracy.
- We conduct a comparative analysis across architectures, contrasting encoder- and decoder-only models to reveal architecture-specific sensitivities and trade-offs in response to identical data ordering strategies.

The rest of the paper is organized as follows: Section 2 reviews related work on curriculum learning and data ordering strategies in NLP, with a focus on how training difficulty has been operationalized. Section 3 describes the overall experimental approach and motivates the evaluation perspectives adopted in this study. Section 4 details the experimental framework, including datasets, model architectures, pretraining setup, and evaluation metrics. Section 5 presents the empirical results, covering intrinsic training dynamics, linguistic probing analyses, downstream task performance, and representation geometry. Section 6 discusses the main findings in light of the paper's contributions, highlighting architecture-specific effects and implications for curriculum design.

2. Related work

The order in which training examples are presented to a neural network has long been recognized as a factor influencing both optimization dynamics and final model behavior. Even when trained on the same data, neural networks can converge to different solutions depending on the sequence in which examples are processed, leading to variations in convergence speed, generalization performance, and the internal features that emerge during training [4,9]. This sensitivity indicates that data ordering acts as an implicit form of regularization, shaping not only the optimization trajectory but also the structure of learned representations.

In the context of language modeling, the dominant class of approaches that explicitly exploit data ordering falls under the umbrella of curriculum learning (CL), which organizes training examples according to a predefined notion of difficulty. Alongside curriculum-based methods, however, several other lines of work explore alternative data ordering or scheduling strategies that are not grounded in classic curriculum assumptions. The following section reviews these lines of research and clarify how our work relates to them. While we adopt curriculum learning as one controlled ordering strategy, our goal is not to optimize a model through a specific curriculum, but to use different ordering schemes to study data ordering itself as an independent factor in pretraining.

In this work, we distinguish between two broad families of data ordering strategies. The first comprises static curricula, in which example difficulty is defined using predefined, model-agnostic metrics and the ordering is fixed prior to training. The second includes adaptive or self-paced approaches, where difficulty is derived from model-internal signals such as loss or confidence, and example selection evolves during training in response to the model's current state. While both families may exhibit easy-to-hard progressions, they differ fundamentally in whether ordering is externally specified and fixed, or dynamically determined by the model itself. Our study focuses on the former setting, isolating the effects of predefined complexity-based orderings.

2.1. Curriculum learning frameworks

Curriculum learning was originally motivated by studies of language acquisition and sequence learning [10], and later formalized as a general training paradigm for neural networks by Bengio et al. [1]. It builds on the general idea that the sequence of training examples can shape optimization trajectories and the representations a model learns, often by presenting "easier" instances earlier and gradually increasing difficulty. In NLP, curriculum learning approaches differ primarily in how this notion of difficulty is operationalized.

A substantial body of work defines curriculum orderings using **human-inspired, model-agnostic complexity metrics**, typically motivated by linguistic intuition or cognitive considerations. Early neural NLP studies explored curricula for tasks such as machine translation, commonly operationalizing difficulty through sentence length, word frequency, or related surface-level heuristics. These metrics are interpretable, inexpensive to compute, and independent of the model's internal state, making them particularly attractive for large-scale training.

More recent work has extended these ideas to language model pretraining and to curricula defined over units larger than individual sentences. A prominent example is the line of research that studies curriculum learning through context length and block construction. Nagatsuka et al. [11] propose a curriculum that progressively increases the maximum block size during pretraining, exposing the model to shorter

contexts early on and gradually requiring it to model longer-range dependencies. This strategy can be interpreted as a curriculum over input span and dependency range, rather than purely over surface-level textual complexity, and highlights that curricula may be implemented not only by reordering documents but also by controlling how raw text is segmented into training instances.

Building on this idea, Ranaldi et al. [12] propose a composite complexity measure that combines block size with explicit linguistic proxies, introducing LRC (Length, Rarity, and Comprehensibility) to capture multiple facets of difficulty. By integrating structural properties with lexical and readability-based signals, their approach reflects a shift toward richer curricula that better approximate the multidimensional nature of linguistic complexity, particularly in pretraining settings where simple heuristics such as length alone may be insufficient.

Human-inspired curricula have also been explored in data-efficient and cognitively motivated pretraining settings. Agrawal et al. [13] define curriculum and anti-curriculum schedules using human-inspired complexity proxies (specifically readability and lexical richness measured at both the sentence and document level) and apply them to BERT pretraining. Their results indicate that corpus ordering improves both performance on downstream tasks, encompassing application-oriented and linguistically focused evaluations, and training efficiency compared to random baselines.

Similarly, within the BabyLM framework, Diehl Martinez et al. [14] introduce CLIMB, which evaluates curriculum learning along multiple axes, including vocabulary availability, data scheduling, and objective design. This multifaceted setup underscores that curriculum learning can target different components of the training pipeline, from what tokens are introduced to how learning objectives evolve over time.

In contrast to human-inspired approaches, another line of work defines training difficulty using **machine-derived, model-dependent signals**, such as loss, perplexity, or confidence scores [15–18]. These methods lead to adaptive or self-paced curricula, including self-paced learning [19], competence-based scheduling, and automated curriculum learning frameworks, in which examples are ordered or selected based on the model's current competence rather than on fixed linguistic heuristics. While such curricula can naturally adapt to the evolving state of the model and often improve optimization efficiency or stability, they also tightly couple data ordering with optimization dynamics, making it more difficult to disentangle the effects of ordering from those of adaptive sampling or reweighting.

Other perspectives on data ordering

While much of the literature on data ordering in NLP is framed under the umbrella of curriculum learning, typically defined by an explicit notion of example difficulty, several lines of work consider alternative ordering or scheduling strategies that do not rely on monotonic easy-to-hard progressions. These approaches nonetheless share the core intuition that the temporal organization of training data can meaningfully affect learning dynamics.

One such direction investigates phase-based or annealed data scheduling, in which the distribution of training data changes over time rather than being globally ordered [20]. This is common in multi-domain or multi-style settings, where models are initially exposed to cleaner or more homogeneous data and gradually transitioned toward more diverse or noisy distributions. Here, ordering operates along axes such as domain or register, and is primarily motivated by training stability and robustness rather than linguistic complexity [21,22].

Another direction focuses on utility-based or value-driven data prioritization, where examples are ordered or sampled according to their estimated contribution to learning [23]. Recent studies in language model pretraining rank data using notions such as irreducible loss or data value, and explore both curriculum and anti-curriculum variants based on these signals. In this setting, ordering reflects expected usefulness rather than intrinsic difficulty.

Related findings also emerge in machine translation. Ding et al. [24] and Ding et al. [25] show that modifying the effective training distribution, for instance by counteracting frequency distortions introduced by knowledge distillation, leads to measurable changes in learning dynamics and lexical behavior.

Taken together, these strands show that data ordering in NLP extends beyond classical curriculum learning based on fixed complexity measures. However, existing work often studies these strategies in isolation, focuses on a single architectural family, or evaluates effects using limited metrics, motivating a more unified and controlled analysis of data ordering strategies and their interaction with model architecture.

3. Approach

Our goal is to provide an in-depth, multidimensional analysis of how different pre-training data orderings affect language models' learning and generalization capabilities. To this end, we pre-train several language models while systematically varying the ordering of the pre-training data according to the strategies defined in Section 3.1.

Rather than focusing solely on final performance, we adopt a dynamic perspective and evaluate models at multiple stages throughout pretraining. This enables us to characterize not only end-state outcomes but also the learning trajectories induced by different curricula, including differences in convergence behavior, representational development, and stability over time.

Finally, we apply this approach to both encoder-based and decoder-based language models, allowing us to assess how curriculum-induced training dynamics interact with different architectures and pre-training objectives.

All experiments are conducted on Italian, a language that has received comparatively less attention than English in large-scale curriculum learning studies.

3.1. Data ordering strategies

We structure the pretraining corpus around three complexity-based curricula computed at the sentence level:

Sentence Length Ordering: Sentence length is widely recognized as a shallow proxy for syntactic complexity. The curriculum inspired to this metric orders sentences by their length in tokens from shortest to longest. **Gulpease Ordering**: The Gulpease index [26] is a traditional readability measure for Italian that estimates textual difficulty based on two shallow proxy of sentence complexity, i.e. sentence length and word length (corresponding to the average number of characters per token). The final index ranges from 0 to 100, with higher values indicating easier readability. We construct a curriculum by ordering sentences from most to least readable, corresponding to decreasing Gulpease scores.

Read-IT Ordering: Read-IT [27] is a state-of-the-art readability assessment system for Italian that models sentence complexity using a wide range of lexical, morpho-syntactic, and syntactic features. For each sentence, Read-IT outputs probabilistic complexity scores at different linguistic levels (base, lexical, syntactic), as well as a global complexity score deriving from the interplay of all features from each linguistic level, which ranges from 0 (easy) to 1 (difficult). In our experiments, we use this global score to order sentences from easiest to most difficult.

Although the considered metrics can also be applied to larger text units, we adopt sentence-level ordering to preserve fine-grained control over training difficulty. Aggregating complexity scores across multiple sentences can obscure substantial variation in sentence-level difficulty, introducing noise into the curriculum and weakening the correspondence between ordering and actual linguistic complexity. Conversely, sentence-level ordering mitigates this issue by aligning the curriculum more closely with the properties of individual training examples.

The resulting curricula have thus been constructed over the same pre-training data and differ only in the ordering imposed on the sentences.

For each complexity-based curriculum, we also define an **inverted variant** (i.e. "anti-curriculum"), by reversing the corresponding sentence ordering, enabling direct comparisons between easy-to-hard and hard-to-easy training schedules. For all curricula, when multiple sentences receive the same complexity score, their relative order is randomized.

Finally, we train models on multiple **random orderings** of the same pre-training data, which serve as baselines to account for variability due to stochastic data presentation and to contextualize the effects of structured curricula.

3.2. Model evaluation

To characterize the effects of data ordering on language model learning, we evaluate models trained under different ordering strategies across multiple complementary dimensions, spanning intrinsic model behavior, representation-level properties, and downstream task performance. Intrinsic metrics capture optimization dynamics and generalization over time, representation-based analyses and probing tasks examine how linguistic information is organized across layers during training, and downstream evaluations assess whether curriculum-induced differences persist at the task level. Across all evaluation perspectives, we adopt a dynamic protocol: models are evaluated at multiple checkpoints throughout pretraining, enabling us to analyze learning trajectories rather than focusing solely on final performance. Checkpoints are sampled more densely during the early stages of training, where curriculum effects are expected to be most pronounced, and more sparsely as training progresses.

Each evaluation dimension is detailed in the following subsections.

3.2.1. Likelihood-based intrinsic metrics

To assess the intrinsic effects of curriculum learning on model behavior, we adopt likelihood-based intrinsic metrics that directly quantify a model's fit to the training distribution. The pretraining loss provides a fine-grained view of optimization dynamics, capturing convergence speed, stability, and sensitivity to data ordering throughout training. Complementarily, uncertainty-based measures derived from the model's predictive distribution, such as perplexity, reflect how confidently the model assigns probability mass to observed token sequences, offering insight into the quality of the learned probabilistic representation. Together, these metrics characterize how different data ordering strategies shape both the trajectory and outcome of learning, independently of downstream task performance.

3.2.2. Linguistic probing

Probing frameworks are widely used in the *interpretability* literature as diagnostic tools to assess whether specific types of linguistic or semantic information are encoded in a model's internal representations [28–30]. In a typical probing setup, the pretrained model is kept frozen, and a lightweight classifier is trained on top of layer-wise sentence representations to predict targeted linguistic properties. The underlying assumption is that if a simple probe can reliably recover a given property, then the corresponding information is accessible in the representations, even if it was not explicitly optimized for during pre-training. Beyond detecting the presence of linguistic information, probing also enables analysis of where such knowledge is most salient within the model architecture, providing insight into how representations are distributed and evolve across layers.

In our study, we adopt the *linguistic profiling* framework introduced by Miaschi et al. [31]. This approach provides a unified and interpretable probing methodology based on a broad set of linguistically motivated features, enabling fine-grained analysis of the linguistic competences captured by pretrained language models.

3.2.3. Geometry of the embedding space

Beyond analysing the linguistic competence encoded by models during training, we study how different data ordering strategies influence the geometry of the learned embedding space. Prior work has shown that Transformer representations are highly structured and often exhibit strong anisotropy, with embeddings concentrating along a limited set of directions rather than being uniformly distributed [32,33].

In recent studies, geometric analyses have been increasingly used as complementary *interpretability* tools, alongside probing methods, to gain insight into how information is linguistically organized and transformed inside neural language models [34,35]. In our work, we extend this framework to examine how data ordering reshapes the internal structure of representations throughout pretraining, which, to our knowledge, has not been previously studied.

Motivated by the *manifold hypothesis*, we treat contextual representations as lying near a lower-dimensional (possibly curved) manifold embedded in a high-dimensional space. Under this view, different orderings may not only change *what* the model learns, but also *how* representations populate and traverse this manifold over training.

To characterize these effects, we adopt two complementary perspectives. The first focuses on *isotropy*, which measures how uniformly representations occupy their underlying manifold and serves as a diagnostic for geometric degeneracy or representational collapse, while recognizing that excessive uniformity can also obscure meaningful structure. The second examines *intrinsic dimensionality*, which estimates the effective number of degrees of freedom used by a representation and provides insight into how abstraction, compression, and specialization evolve across layers and training stages. To make this analysis robust to both globally linear structure and locally non-linear geometry, we compute intrinsic dimensionality using both *linear* and *nonlinear* (manifold-/neighborhood-based) estimators.

3.2.4. Downstream tasks

We consider the following tasks, selected to be consistently applicable to both encoder-only and decoder-only architectures in order to maintain a unified evaluation framework and ensure direct comparability of results:

- **Part-of-Speech Tagging**: the assignment of syntactic category labels to individual tokens within a sentence.
- **Human Complexity Judgment**: the prediction of a complexity score assigned by raters in a sentence reading task designed to elicit the human perception of linguistic complexity.
- **Sentiment Analysis**: the classification of sentence-level sentiment polarity (positive, negative, neutral, or mixed), testing semantic and affective aspects of meaning.

Together, these tasks provide a controlled way to examine how curriculum learning shapes linguistic representations and transfer behavior. Part-of-speech tagging primarily tests syntactic knowledge, human complexity judgment assesses sensitivity to structural and readability-related cues, and sentiment analysis captures higher-level semantic and affective information.

In what follows, we detail the concrete datasets, metrics, and evaluation protocols used to implement our approach. All experiments use the **same pre-training data, model sizes, and optimization settings, differing only in the ordering of the training examples** according to the defined curricula. This design provides a controlled basis for comparing learning dynamics and final model behavior.

4. Experimental setting

4.1. Data

This section describes the datasets used in our experimental framework, including the corpus employed for language model pre-training

and the datasets used for evaluation. The pre-training data are shared across all curricula, model architectures, and training runs, ensuring that observed differences in learning dynamics arise solely from variations in data ordering. Evaluation datasets are used consistently across checkpoints and curricula to assess optimization behavior, representation geometry, linguistic probing performance, and downstream task generalization.

4.1.1. Pre-training dataset

All models are pre-trained on a subset of the Italian Wikipedia dump from October 2021. To keep experiments computationally tractable while preserving linguistic diversity, we sample ten million sentences from the full corpus, corresponding to approximately 240 million tokens and slightly less than half of the original content. Sampling is performed at the sentence level in order to preserve the natural distribution of topics and writing styles present in Wikipedia.

The raw Wikipedia dump is processed using the `Wikiextractor` library [36], which performs standard text normalization, including the removal of HTML and markup tags, tables, lists, and other non-linguistic artefacts such as URLs and code snippets. The extracted text is then segmented into sentences using the Italian pipeline of UDPipe [37]. Following segmentation, sentences shorter than six tokens or longer than sixty tokens are filtered out to remove degenerate or excessively long inputs.

This processed corpus constitutes the base pre-training dataset used across all experiments. For curricula based on external readability metrics (Gulpease and Read-IT), sentences for which the corresponding metric yields missing or invalid values are excluded. After this additional filtering, the Gulpease-based curriculum comprises 9,589,538 sentences, while the Read-IT-based curriculum comprises 9,647,949 sentences.

4.1.2. Evaluation datasets

We employ multiple evaluation datasets, each selected to probe specific aspects of model behavior. These datasets are used consistently across checkpoints and curricula, with their role depending on the evaluation setting (perplexity, probing, downstream tasks, or representation geometry).

Wikipedia in-domain test set We construct a held-out in-domain evaluation set consisting of 10,000 sentences sampled from the same Italian Wikipedia dump used for pre-training, with no overlap with the training data. This dataset is used as an in-domain reference for evaluating language modeling performance and representation properties. Specifically, the Wikipedia test set is used for computing (pseudo-)perplexity at each checkpoint, and analyzing the geometry of the embedding space. For perplexity evaluation, we restrict computation to a fixed subset of 1000 sentences sampled from this in-domain Wikipedia test set for computational efficiency.

Universal Dependencies Treebank (ISDT) As an out-of-domain evaluation dataset, we use the Italian Stanford Dependency Treebank (ISDT) [38], a subsection of the Italian Universal Dependencies treebanks. ISDT consists primarily of newspaper text and provides gold-standard morphological and syntactic annotations. This dataset serves two purposes. First, it is used as an out-of-domain benchmark for perplexity evaluation, allowing us to assess generalization beyond Wikipedia-style text. Second, it is used for linguistic probing experiments, where gold annotations are required to extract reliable target labels. For the perplexity evaluation we sample again 1000 sentences, while for probing, we use a subset of 8000 sentences as training set and 2000 sentences as test set.

Part-of-Speech Tagging For downstream POS tagging, we use the dataset from the 2009 *EVALITA* POS Tagging shared task [39]. The training set consists of newspaper articles from the online edition of *La Repubblica*, containing 108,875 word forms across 3719 sentences. The test set is drawn from the Italian Wikipedia and includes 5014 word forms across 149 sentences. The data were annotated using a combination of automatic and manual procedures, followed by human revision.

Human Complexity Judgment To evaluate sentence-level complexity prediction, we use the Italian subset of the dataset introduced by Brunato et al. [40], comprising 1123 sentences. Sentences are drawn from Italian Universal Dependencies treebanks and are uniformly distributed in length between 10 and 35 tokens. Each sentence is annotated with 20 human ratings on a 1–7 Likert scale, collected via crowdsourcing. The task is to predict the average complexity score assigned to each sentence.

Sentiment Analysis For sentiment analysis, we use the *SENTIPOLC* dataset [41], which consists of Italian tweets annotated for subjectivity, polarity, and irony. We focus exclusively on polarity, modeled as a multi-label classification task with two binary labels indicating the presence of positive and negative sentiment. The dataset contains 7410 training tweets and 2000 test tweets.

4.2. Model architectures

We employ two transformer-based architectures of comparable size to study the effects of curriculum learning under different language modeling objectives. The first model follows the encoder-only architecture of BERT [7], while the second adopts a decoder-only, GPT-2-style architecture [8]. Both models are configured to have approximately the same number of parameters (about 65 million), enabling a controlled comparison across architectures.

The two architectures share the same core configuration, comprising eight transformer layers, eight attention heads, a hidden dimension of 512, an intermediate feed-forward dimension of 2048, and a vocabulary size of 30,000 tokens.

To account for variability due to stochastic initialization, all experiments are conducted using **three different random initializations of the model weights** for each architecture and training configuration, and results are reported across these runs. Tokenization follows the standard choices associated with each model family: a WordPiece tokenizer for the encoder-based model and a byte-level BPE tokenizer for the decoder-based model. In both cases, tokenizers are trained from scratch on the pre-training corpus to ensure consistency between the vocabulary and the training data.

4.3. Pre-training setup and training protocol

Each model is pre-trained using the standard language modeling objective associated with its architecture. Encoder-based models are trained with the *Masked Language Modeling* (MLM) objective [7], while decoder-based models are trained with the *Causal Language Modeling* (CLM) objective [8].

For the encoder-based model, we adopt MLM without the auxiliary Next Sentence Prediction (NSP) objective. The NSP component, originally introduced in early versions of BERT, was later shown to provide limited benefits for language understanding and has been omitted in subsequent variants [42]. Under the MLM objective, a fixed proportion of input tokens (15% in our setup) is selected for prediction, with selected tokens replaced by a special `[MASK]` symbol during training.

For the decoder-based model, we employ the standard autoregressive CLM objective [8], in which the model is trained to predict each token conditioned on all preceding tokens in the sequence.

As described in Section 3.1, each model is trained under multiple data ordering strategies, including three complexity-based curricula (Sentence Length, Gulpease, and Read-IT), their inverted counterparts, and five random orderings with corresponding inverse versions. This design allows us to isolate the effect of training order on optimization dynamics and representation learning, independently of data content or model architecture.

All models are trained for three epochs using the AdamW optimizer, with a learning rate of 10^{-4}, weight decay of 0.01, and an effective batch size of 256. Training is implemented using the Hugging Face `Transformers` library.

Importantly, training is performed at the *sentence level*: each sentence constitutes an independent training instance, and no concatenation across document boundaries is applied. This choice ensures that

the sequence in which sentences are presented during training exactly reflects the curriculum ordering, without confounding effects from context impact or document-level structure.

Checkpointing All models are trained for three full passes over the pretraining corpus. To enable a detailed analysis of learning dynamics, models are saved at multiple checkpoints throughout pre-training. Since curriculum-induced effects are often most pronounced during the early stages of optimization, checkpoints are saved more densely at the beginning of training and progressively more sparsely as training advances. Concretely, checkpoints are saved every 400 training steps (approximately 100k sentences) up to step 4,000, corresponding to roughly the first 1M sentences, and every 4000 steps thereafter until the end of training. This checkpointing strategy provides fine-grained temporal resolution during early learning while maintaining coverage of later training phases.

4.4. Evaluation measures

Intrinsic Performance

We evaluate intrinsic language modeling performance via *pretraining loss* and *perplexity-based measures*. We track the training loss associated with each model's pretraining objective, averaged over batches and recorded at regular checkpoints to analyze learning dynamics under different data ordering strategies.

For the decoder-only model, perplexity is computed in the standard way, i.e. computing the probability assigned to each token conditioned on its leftward context. As perplexity is not directly defined for masked language models, which do not represent a left-to-right factorization over entire sequences, for the encoder-based model, we compute pseudo-perplexity following [42]; this metric approximates perplexity by iteratively masking each token in the input sequence and estimating its conditional probability given the remaining context.

Both perplexity (for the decoder model) and pseudo-perplexity (for the encoder model) are computed at each training checkpoint on an in-domain test set derived from Wikipedia, as well as on an out-of-domain test set based on the UD treebank.

Linguistic Probing

The probing-based evaluation comprises a suite of sentence-level experiments designed to assess the linguistic information encoded in model representations. We considered 60 linguistic features spanning lexical, morphological, and syntactic properties of text; the full list of features is reported in Appendix A (Table A.1). These features are extracted from the UD treebank using the Profiling-UD tool [43].

Probing is conducted using a standard diagnostic setup in which the pretrained model parameters are kept frozen and lightweight predictors are trained on top of the learned representations. For each training checkpoint and for each layer of the model, we train an independent Ridge regression model to predict the value of a single linguistic feature from sentence representations. Sentence-level representations are obtained by mean pooling the token-level hidden states produced by the model. For each considered feature, model's performance is evaluated in terms of Spearman's rank correlation coefficient between the gold and predicted value. This setup allows us to analyze how linguistic information is distributed across layers and how it evolves over the course of training under different data ordering strategies.

Geometry of the Embedding Space We track isotropy and intrinsic dimensionality using three complementary metrics. For isotropy, we rely on *IsoScore* (which ranges from 0 to 1, where a value of 1 indicates perfect isotropy) that provides a single normalized scalar measuring how uniformly a point cloud utilizes the ambient space. This score was introduced as a robust alternative to commonly used isotropy proxies for contextual embeddings [44].

As regards intrinsic dimensionality, we measure linear dimension with *PCA@99*, defined as the smallest number of principal components needed to explain 99% of the variance. This PCA-spectrum summary is computationally efficient and closely connected to post-processing approaches that remove dominant components or whiten representations to mitigate anisotropy [45,46]. Additionally, we include the nonlinear intrinsic-dimension estimator TwoNN [47,48], which infers dimensionality from local nearest-neighbor distance ratios and can capture curved, nonlinearly embedded manifolds that linear projections may miss. TwoNN/ID-based geometric analyses have been used to study latent representations under deep clustering and their impact on clustering performance [49,50]. Both these intrinsic dimensionality metrics can range from 0 to the dimension of the ambient space.

Taken together, *IsoScore*, *PCA@99*, and *TwoNN* jointly characterize directional concentration and effective degrees of freedom in representation manifolds across layers and checkpoints. All scores are computed on random subsamples of 10,000 sentences per checkpoint, a common practical regime where intrinsic-dimension estimates for high-dimensional LM representations empirically stabilize while keeping computation tractable [51,52]. Appendix D provides further details on these stability considerations.

Downstream Task Evaluation

The performance on the downstream tasks is evaluated by fine-tuning task-specific prediction heads on top of the pretrained models at each checkpoint. The datasets used for downstream evaluation, together with their corresponding training and test splits, are described in Section 4.1.2.

For each downstream task, models are fine-tuned for 10 epochs using a fixed learning rate of 5×10^{-5}, which was sufficient to ensure stable convergence across tasks and checkpoints. To ensure comparability across training stages and curricula, the same fine-tuning configuration is applied at every checkpoint. Performing separate hyperparameter optimization for each checkpoint and task would be computationally prohibitive and is therefore not considered.

4.5. Baselines

Our study primarily focuses on comparing curriculum-based data orderings against non-curriculum training. Accordingly, our main baselines consist of models trained with randomly shuffled data, which provide a natural reference for isolating the effects of structured ordering. In addition, since our evaluations track model behavior throughout pre-training, each model can be directly compared to its own early-stage checkpoints.

Code to reproduce our experiments is available at the following link: https://github.com/lucadinidue/curriculum_learning.

5. Results

In this section, we present the results of our experiments on how different curriculum strategies affect model performance and learning dynamics. Our evaluation proceeds in a progressively higher-level fashion. We begin with intrinsic metrics, which directly reflect the pre-training process itself, i.e. training loss and model perplexity (Section 5.1). We then move one step up to intermediate behavioral evaluations, where we examine how pre-training order shapes the emergence of linguistic knowledge through probing analyses (Section 5.2) and examining the geometry of their learned representation spaces (Section 5.3). Finally, we consider task-level performance, evaluating the practical impact of curricula on downstream NLP tasks (Section 5.4).

For clarity and robustness, we aggregate results across model initialization and data-ordering seeds rather than reporting each run separately. Throughout the figures and tables, results for each curriculum are averaged over three initialization seeds. The random baselines, instead, are aggregated over 3 initialization seeds $\times$ 5 random orderings $\times$ 2 (standard and anti-curriculum variants), for a total of 30 models. When shown, error bars denote the standard deviation across all corresponding runs (i.e., across seeds and, for the random baselines, orderings).

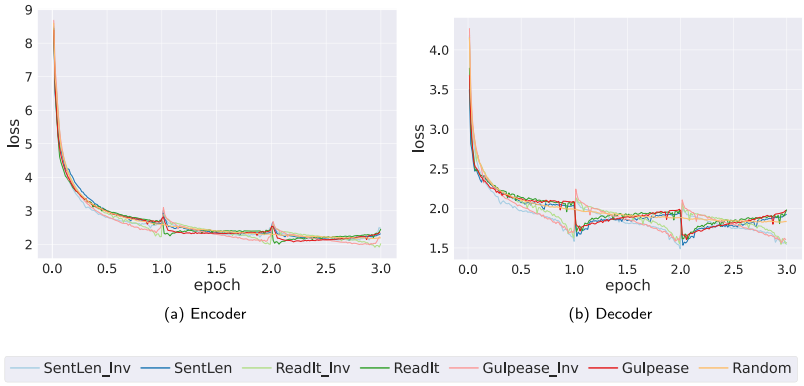


Fig. 1. Language modeling loss across pre-training epochs for encoder and decoder models under different curricula.

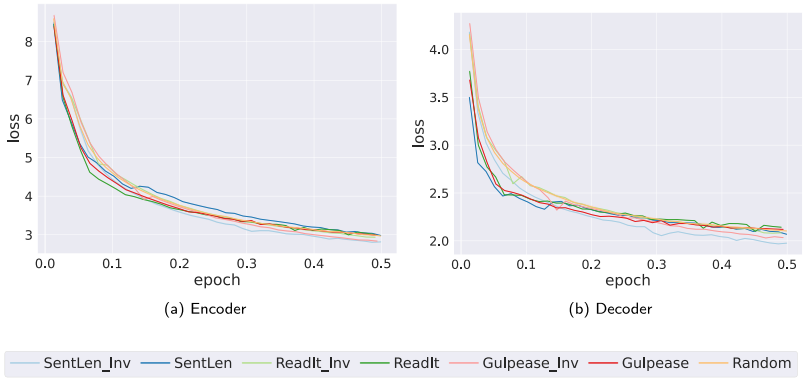


Fig. 2. Early-stage language modeling loss during pre-training.

Table 1
Average training time (hh:mm) for encoder and decoder models across curricula.

Curriculum	Encoder	Decoder
Random	7:20	6:21
SentLen	5:18	4:27
SentLen_Inv	5:18	4:27
Gulpease	5:39	5:08
Gulpease_Inv	5:43	5:10
ReadIt	6:59	5:36
ReadIt_Inv	6:40	5:37

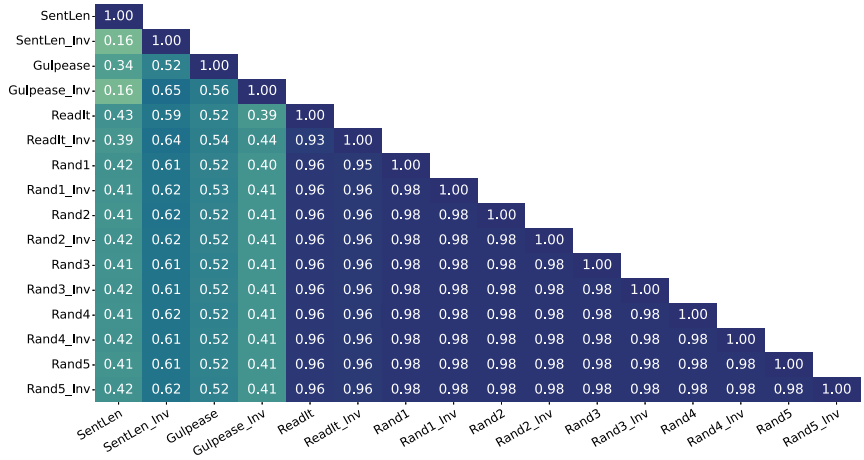
This aggregation reduces the impact of stochastic variation and enables more reliable comparisons across curricula.

Table 1 reports the average training time for each curriculum, alongside the average training time obtained with random orderings. The first result we observe is that curriculum-based orderings are consistently faster than the random orderings. In particular, curricula that are more directly dependent on sentence length yield the largest speed-ups (the faster are curricula based on sentence length, followed by those based on Gulpease and then on ReadIt). This is due to the fact that sorting data by length results in batches containing sentences of equal or very similar size, thereby reducing the amount of padding within each batch. This leads to fewer unnecessary computations and, consequently, lower overall training time.

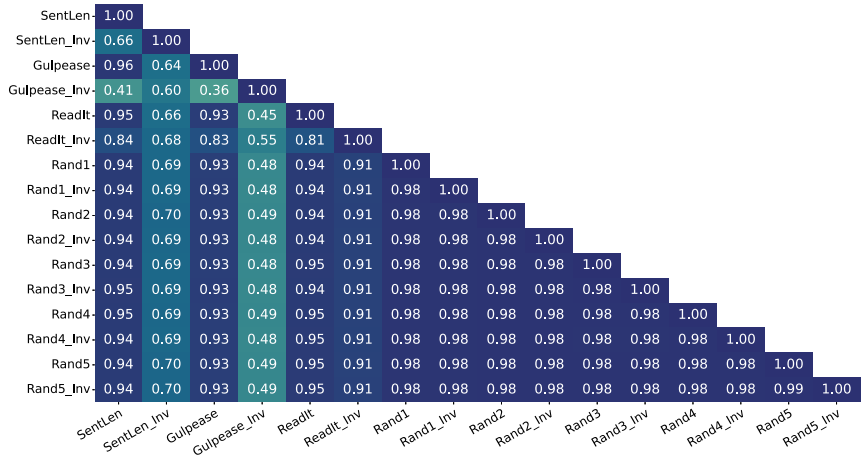
5.1. The effect of data ordering on model's intrinsic metrics.

Loss Fig. 1 shows the evolution of the pre-training loss across epochs for all models. For the encoder models (left), we observe that the loss of curriculum-trained models exhibits pronounced peaks at the beginning of each new epoch. Although the three complexity measures capture different linguistic aspects, they all induce orderings in which early and late portions of the data differ in sentence length, lexical composition, and syntactic structure. As a result, the model is exposed to markedly different distributions across epoch boundaries. The abrupt transition between these distributions, whether from simpler to more structured sentences or the opposite in the inverted curricula, creates a sudden shift in the type of input the model is required to process and is likely responsible for the temporary degradation observed in the masked language modeling objective. A smaller loss increase is also noticeable at the end of each epoch, which may further explain the sharper peaks at epoch restarts. This behavior could stem from outlier sentences located at the extremes of the complexity scales used to define the curricula. Although we excluded sentences with scores outside the metric's valid range, some residual outliers remain. Highly complex sentences can include artifacts such as residual tables or other non-natural language segments, while very short sentences (around six words) may lack sufficient context to constitute realistic utterances. Moreover, both very short and very complex sentences make the masked language modeling objective inherently more challenging, as limited or overly intricate context reduces the predictability of masked tokens.

For the decoder models (Fig. 1 right), we observe a clear divergence between curricula and anti-curricula. In the curricula, the loss sharply decreases at the beginning of each epoch and then gradually increases, whereas in the anti-curricula the opposite pattern emerges: an initial loss increase followed by a decrease. While in causal language modeling the loss is naturally affected by sentence length (since it is computed as the sum of token-level losses) it is in practice dominated by the magnitude of each token's contribution rather than by the number of tokens alone. This indicates that the model's behaviour is driven primarily by sentence-level complexity, and the fact that the same pattern appears consistently across all three curricula supports this interpretation.



(a) Encoder - Wikipedia



(b) Decoder - Wikipedia

Fig. 3. Pairwise Spearman rank correlations between sentence-level perplexity scores assigned by models trained with different pre-training data orderings on Wikipedia.

Finally, Fig. 2 provides a closer view of the initial training phase. During the first few training steps, curriculum-trained models exhibit a steeper decrease in training loss, with an inversion point occurring around 15% of the training process. This trend suggests that curricula may facilitate faster initial adaptation to the training distribution, likely because the early stages present sentences that are easier for the model to predict, thereby easing the optimization of the language modeling objective. It is worth noting that this loss is computed on the training data itself, and therefore reflects optimization dynamics rather than generalization ability. The latter is better captured by the perplexity measured on held-out data, which we discuss in the following subsection.

Key Findings: Complexity-based curricula introduce strong distributional shifts at epoch boundaries, resulting in pronounced loss peaks. For the encoder models, these peaks arise from abrupt changes in sentence characteristics, whereas for the decoder models the loss closely follows variations in sentence difficulty. Curricula also lead to a faster initial reduction in loss, indicating an early optimization advantage.

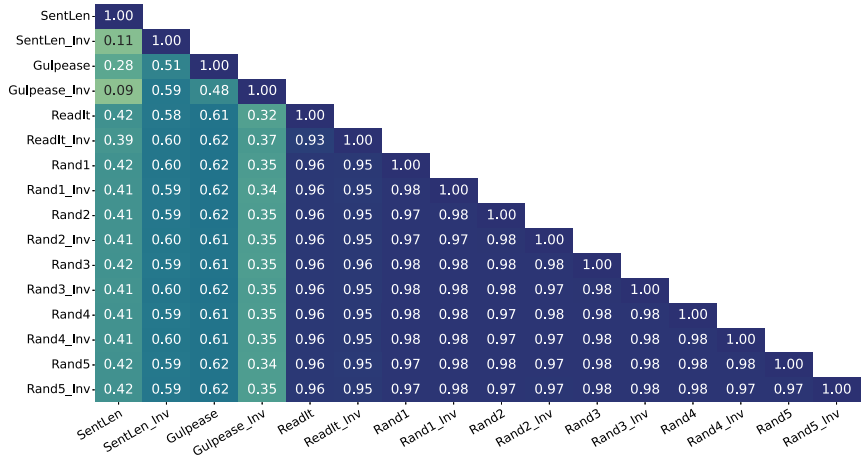
Perplexity The second intrinsic evaluation concerns model perplexity (pseudo-perplexity for encoder-based models). To obtain a broader and more robust assessment, we compute perplexity on two evaluation sets of 1000 sentences each: an in-domain Wikipedia test set and a sample drawn from the UD-Treebank, both described in Section 4.1.2. Since our goal is to assess whether models undergoing different pretraining curricula are similarly perplexed, rather than to compare absolute perplexity values, we rely on Spearman rank correlation.

The results reported here are obtained by comparing how model rank sentences according to perplexity score at the end of the pretraining. A complementary, fine-grained analysis of perplexity trajectories throughout training is provided in Appendix B.

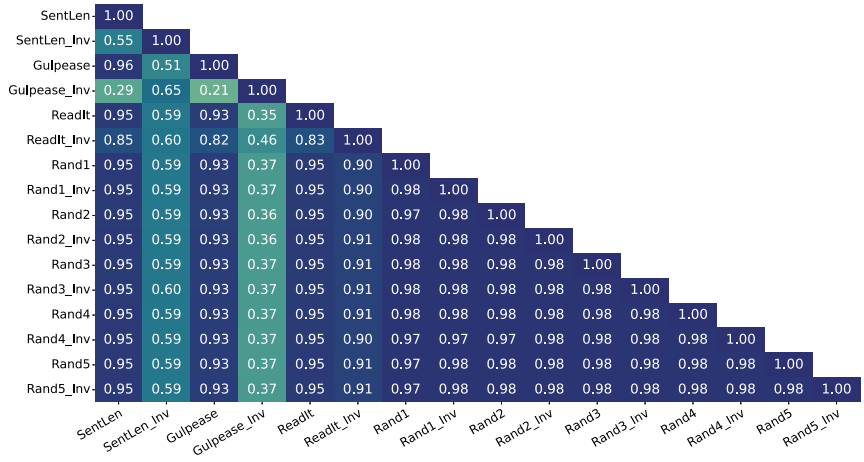
Fig. 3 shows the pairwise correlations between models' (pseudo-)perplexities on the Wikipedia dataset, and Fig. 4 reports the same analysis for the Treebank dataset.[1]

When comparing different data orderings, both encoder- and decoder-based models exhibit very high correlations among models trained with different random orderings. This indicates that stochastic variation alone leads to highly consistent sentence-level perplexity rankings, providing a stable reference against which curriculum-induced effects can be assessed.

[1] All reported correlations are statistically significant at $p < 0.05$.



(a) Encoder - UD Italian Treebank



(b) Decoder - UD Italian Treebank

Fig. 4. Pairwise Spearman rank correlations between sentence-level perplexity scores assigned by models trained with different pre-training data orderings on the UD Treebank.

When turning to curriculum-based orderings, however, a clear divergence emerges between encoder- and decoder-based architectures.

Encoder-based models display substantial sensitivity to data ordering. With the exception of the Read-It-based curricula, which induce pseudo-perplexity patterns closely aligned with both in-domain and out-of-domain random baselines, most curriculum-trained encoders show markedly lower correlations with other models. This sensitivity is especially pronounced when comparing curricula with their corresponding anti-curricula. The lowest correlations are observed for Sentence Length versus Sentence Length Inverted and for Sentence Length versus Gulpease Inverted, indicating that curricula emphasizing opposing or misaligned notions of surface complexity lead to strongly divergent pseudo-perplexity rankings. These effects are further amplified in the out-of-domain setting.

A different pattern emerges for decoder-based models. Across both in-domain and out-of-domain evaluations, forward curricula yield perplexity rankings that remain highly similar to those produced by random orderings, with correlation coefficients consistently above 0.93. Divergences from this pattern are almost exclusively associated with anti-curricula. Models trained with Sentence Length Inverted and Gulpease Inverted show noticeably lower correlations, while the effect for Read-It Inverted is present but considerably weaker. Interestingly, Gulpease Inverted produces lower correlations than Sentence Length Inverted, suggesting that the magnitude of the effect is not straightforwardly related to the reliance on surface-level features of the curriculum function. For decoder models, the lowest correlations are observed when comparing Gulpease Inverted with its forward counterpart and with Sentence Length-based curriculum.

For both architectures, correlations computed on the in-domain dataset are slightly higher than those obtained out of domain, although the overall qualitative patterns remain unchanged.

Key Findings: Data ordering can influence the language modeling behavior learned during pre-training with different random data orderings lead to highly consistent sentence-level perplexity rankings across both encoder- and decoder-based models. Curriculum-based orderings can alter these rankings, particularly when the ordering reverses a complexity signal, with encoder models showing greater sensitivity than decoder models. Overall, the impact depends on the specific ordering function used, rather than on the presence of an ordering itself.

5.2. *The effect of data ordering on model's linguistic competence*

The linguistic probing evaluation aimed to assess how different training data ordering affects the acquisition of linguistic competence over

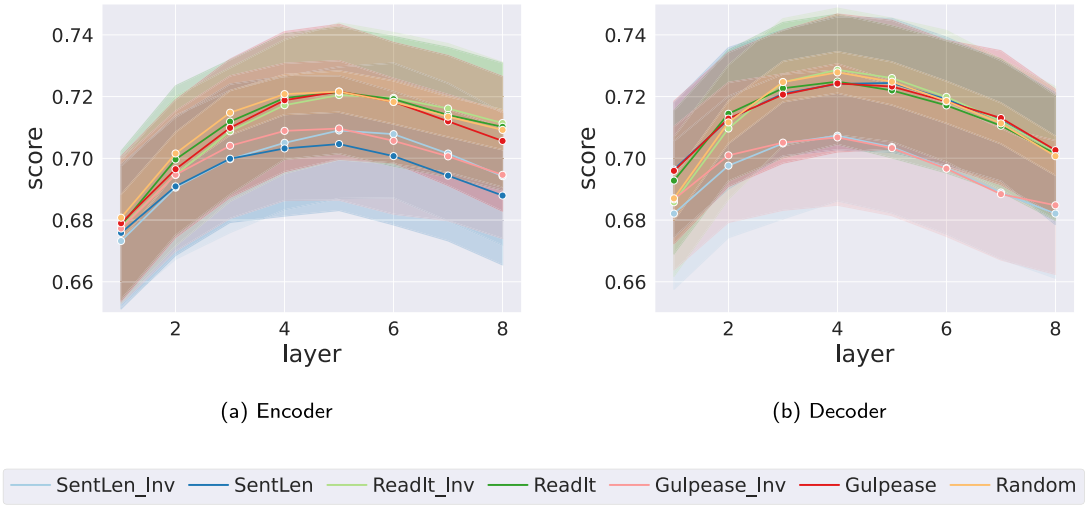


Fig. 5. Layer-wise probing performance averaged across all linguistic features at the end of pre-training.

time (i.e. across pre-training steps) and across model layers. The evaluation considered both learning performance and learning speed.

As noted earlier, our probing suite covers 60 linguistic features, making it impractical to display full learning curves for every layer, checkpoint, and feature combination.

As a global summary of how training curricula affect the models' linguistic competence, Fig. 5 reports the average probing performance across layers in the final checkpoint[2] Overall, the two architectures show remarkably similar trends, both in absolute performance and in their middle-layer peak.

Across curricula, Gulpease, ReadIt, and ReadIt-Inverted consistently achieve performance levels comparable to the random baselines. In contrast, Gulpease-Inverted and Sentence-Length-Inverted yield noticeably lower scores for both architectures. The only clear divergence between encoder and decoder concerns the Sentence-Length curriculum: while it is the worst-performing curriculum for the encoder, it is among the best-performing ones for the decoder. This divergence suggests that sentence length interacts differently with encoder and decoder learning dynamics. For the encoder, a strict length-based ordering may over-constrain early training, exposing the model to structurally homogeneous inputs and limiting syntactic variability at a stage where bidirectional context promotes rapid abstraction. In contrast, the autoregressive decoder may benefit from shorter sequences early on, as they simplify next-token prediction and stabilize initial learning.

Fig. 6 reports the average probing performance at the last layer across pre-training checkpoints, offering insight into how linguistic capabilities evolve under different data orderings. From now on, evaluations performed during pre-training are reported as a function of the number of training tokens processed, rather than training steps. This choice ensures that comparisons across curricula capture differences in learning dynamics instead of discrepancies in token exposure, since different data orderings can result in substantially different numbers of processed tokens at the same training step.

For both encoder and decoder models, curriculum-based orderings exhibit systematic fluctuations in probing performance around epoch boundaries.

Across models, two distinct behaviors emerge. The first and most prominent pattern appears after an initial convergence phase marked by a sharp increase in probing performance. In this case, performance drops toward the end of each epoch and then recovers at the beginning of the subsequent one. This behavior characterizes complex-to-simple orderings and is most clearly observed for Sentence Length Inverted, while it is weaker for Gulpease Inverted and only marginally visible for Read-It Inverted. The magnitude of the effect appears to depend on the extent to which the sorting function relies on surface-level features.

The second pattern, which is generally less pronounced, applies to simple-to-complex curricula and shows the opposite trend. After convergence, probing performance tends to increase toward the end of an epoch and slightly decrease at the beginning of the next. As before, the magnitude of this effect reflects the granularity of the underlying ordering criterion.

Overall, these dynamics suggest that models tend to achieve higher probing performance during training stages in which they are exposed to more complex sentences. The only exception to these trends is again the forward Sentence Length encoder model, which instead clearly follows the same dynamics observed for anti-curriculum orderings.

Finally, these dynamics are consistently observed across all three training epochs, although their magnitude diminishes as training progresses, suggesting a gradual stabilization of linguistic representations.

Overall, these analyses indicate that ordering pre-training data by complexity does not systematically improve linguistic competence. In several cases, curriculum-based models show drops in probing performance at epoch boundaries, pointing to non-trivial interactions between data ordering and representation learning dynamics.

5.2.1. *Linguistic features analysis*

Beyond aggregate probing performance, we conduct a fine-grained analysis to determine whether specific linguistic phenomena consistently benefit from particular curriculum strategies. This allows us to distinguish between general improvements in linguistic encoding and targeted gains that reflect the structural biases introduced by different data orderings.

To this end, we smooth probing scores by averaging them across consecutive checkpoints and layers. For the early training phase, scores are averaged over the first ten checkpoints, while for the late phase we average over five checkpoints, reflecting the sparser checkpoint sampling at later stages. This yields early and late probing scores for each model-feature pair.

We then compute z-scores for curriculum-trained models relative to the corresponding random-order baselines. A linguistic feature is considered *supported* by a curriculum if all three training seeds achieve a

[2] Appendix C provides feature-wise versions of the plots discussed above, along with heatmaps reporting the corresponding probing scores for the models' final layer.

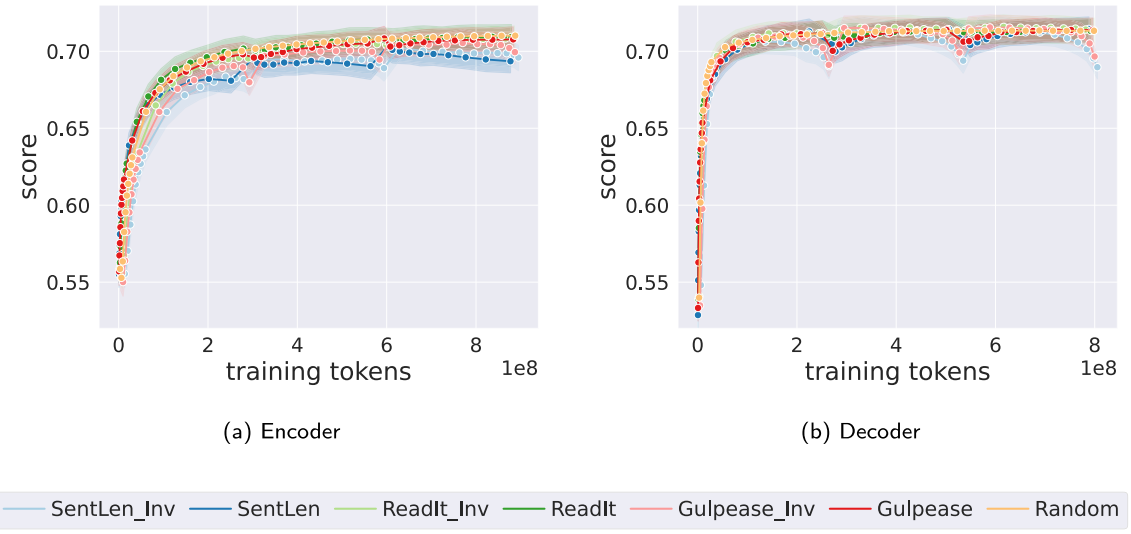


Fig. 6. Evolution of probing performance during pre-training, averaged across all linguistic features and model layers.

Table 2
Number of training tokens processed up to the tenth checkpoint for each curriculum.

Curriculum	Encoder	Decoder
SentLen	8,714,225	6,508,263
Gulpease	11,893,668	9,491,966
ReadIt	17,920,663	15,056,076
Avg Random	30,637,915	27,559,828
ReadIt_Inv	42,129,884	38,824,053
Gulpease_Inv	47,090,681	43,905,879
SentLen_Inv	60,022,260	55,883,352

z-score greater than 2, indicating performance consistently at least two standard deviations above the random baseline.

Figs. 7–9 report the z-scores of curriculum-supported features. Specifically, they show early-supported features for the encoder, early-supported features for the decoder, and late-supported features for both encoder and decoder, respectively.

Across architectures, the Sentence Length Inverted curriculum leads to earlier improvements in the encoding of a broad set of linguistic features spanning multiple levels of linguistic description. This effect is consistent with the fact that, at early checkpoints, this curriculum exposes the model to a larger number of tokens (see Table 2) and to sentences with greater linguistic variability, particularly in terms of syntactic structure and verbal morphology, i.e. properties that are especially rich in a morphologically complex language such as Italian. The strongest gains are observed for sentence-length-related feature (*n_token*) and for properties that depend directly on it, such as the longest dependency link (*max_link_length*) and the average length of syntactic chains headed by a preposition (*n_prepositional_chains*).

Interestingly, the second most effective curriculum for the encoder in the early phase is Read-IT, despite exposing the model to substantially fewer tokens than Sentence Length Inverted within the first ten checkpoints. Unlike length-based ordering, Read-IT relies on a composite complexity score that captures interactions among multiple linguistic features. This suggests that, even with limited early exposure, the curriculum may prioritize sentences that promote the early emergence of specific syntactic phenomena, such as clausal modifiers or nominal heads (*dep_dist_acl*) and the presence of explicit subjects (*dep_dist_nsubj*).

Conversely, this pattern does not hold for the decoder-based model. In the early stages of training, the decoder is most strongly affected by curricula that, despite relying on different complexity ordering functions, expose the model to a large volume of training data, namely the inverted curricula. Among these, Read-It Inverted exhibits a more pronounced effect than Gulpease Inverted.

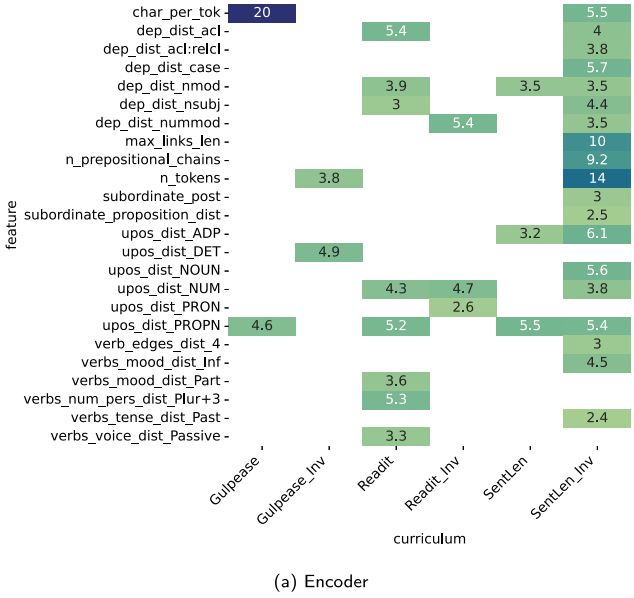
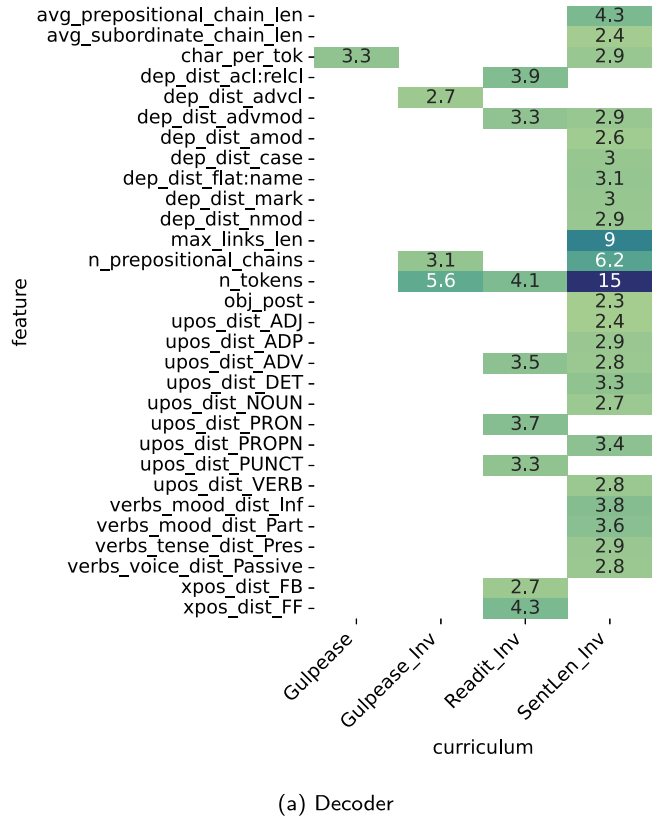


Fig. 7. Z-scores of encoder probing performance relative to random-order baselines, averaged across layers, seeds, and the first ten checkpoints. Only features with z≥2 for all seeds of at least one curriculum are shown.

Fig. 9 highlights one of the most informative findings of this analysis: linguistic features that benefit from curriculum learning at the end of training tend to align closely with the specific data ordering function used during pretraining. A clear example is the encoding of token length (*char_per_tok*), which is consistently supported by both Gulpease and Gulpease Inverted curricula for both encoder and decoder models. In decoder models, this feature also shows weaker but systematic gains under Sentence Length-based curricula, plausibly reflecting correlations between token length and sentence length in the training data.

A similar alignment is observed for sentence length (*n_tokens*), which is supported by Sentence Length curricula in both architectures and by Sentence Length Inverted in the encoder. Sentence Length ordering also benefits structurally related features such as *max_links_len* and, in en-



(a) Decoder

Fig. 8. Z-scores of decoder probing performance relative to random-order baselines, averaged across layers, seeds, and the first ten checkpoints. Only features with z≥2 for all seeds of at least one curriculum are shown.

coder models, *n_prepositional_chains*. On the decoder side, the Read-It curriculum supports *upos_dist_PRON*, suggesting that readability-based orderings can influence sensitivity to specific morphosyntactic patterns.

Overall, despite some architecture-specific differences, the same linguistic features tend to benefit from the same curricula across models. This consistency indicates that late-stage curriculum effects primarily reflect the linguistic properties emphasized by the chosen data ordering strategy.

For a more in-depth view of model performance during training on these features, averaged across layers, Figs. 10 and 11 report probing performance throughout pre-training on the six most strongly supported features for the encoder and the decoder, respectively.[3]

As anticipated from the z-score analysis, both architectures show a clear and consistent positive impact of the Gulpease and Gulpease Inverted curricula on the prediction of the *char_per_tok* feature, with performance exceeding that of all other models by approximately 0.1 points.

A second prominent trend concerns the *n_tokens* feature, which is mirrored, even if with lower magnitude, by the related features *max_links_length* and *n_prepositional_chains*. For the encoder model, nearly all curricula and anti curricula achieve higher average probing scores than random orderings, with the exception of the Gulpease curriculum. The strongest effects are observed for Sentence Length-based curricula, followed by Gulpease-based curricula, again indicating that the magnitude of the effect is closely tied to the dependence of the ordering function on the target feature.

For the decoder model, a similar pattern emerges but with reduced magnitude, which is consistent with the pronounced drops in probing performance observed at epoch boundaries for curriculum based models. Despite this instability, the influence of length related orderings remains clearly visible.

The remaining two features among the most impacted for both architectures show earlier improvements under the Sentence Length Inverted curriculum. However, when aligning checkpoints by the number of seen training tokens, this apparent faster convergence disappears. This observation supports our hypothesis that the early gains of Sentence Length Inverted models are, in some cases, an artifact of having processed more tokens at a given checkpoint rather than a real acceleration of learning of specific linguistic phenomena.

Key Findings: Curriculum learning influences probing performance in a feature-specific and architecture-dependent manner: early gains often reflect increased data exposure rather than faster linguistic learning, especially for decoders, while late-stage improvements consistently align with the linguistic properties emphasized by the curriculum's ordering function. Across both encoder and decoder models, curricula yield the strongest and most persistent benefits for features directly targeted by their complexity criteria, indicating that data ordering shapes representation learning primarily through the linguistic biases it imposes over training.

5.3. The effect of data ordering on the geometry of the embedding space

As stated in Section 4.4, we analyze how curriculum orderings reshape the geometry of the representation space throughout pre-training by tracking three complementary diagnostics across checkpoints: *IsoScore* (global isotropy), PCA@99 (linear effective dimension), and *TwoNN* (nonlinear intrinsic dimension). Fig. 12 reports the evolution across learning time of these three metrics for the *last-layer* sentence representations of the encoder (left) and decoder (right). Our findings reveal three main patterns.

(i) Sorting by surface cues amplifies anisotropy and induces periodic geometry shifts In the encoder, Random ordering and both ReadIt schedules exhibit a smooth, largely monotonic increase in isotropy and converge to the highest IsoScore values (Fig. 12, top-left). In contrast, Sentence Length- and Gulpease-based curricula remain substantially more anisotropic throughout training: IsoScore stays lower and shows larger within-epoch variability (including localized spikes) rather than clean convergence.

A plausible mechanism is that strict ordering by length (or a length-dominated readability proxy) creates long runs of examples with highly homogeneous surface statistics. This reduces local batch diversity, makes successive gradient updates more correlated, and can amplify a small set of dominant directions in representation space.

For the decoder, isotropy increases across all settings and ends in a narrower band than in the encoder, but the same qualitative separation remains (Fig. 12, top-right): Random and ReadIt orderings are the most stable and most isotropic, whereas Sentence Length- and Gulpease-based schedules exhibit pronounced transient isotropy collapses at epoch boundaries. Importantly, the *phase* of these collapses flips with schedule direction. In the forward curriculum, IsoScore typically drops immediately after the epoch reset and then recovers over the epoch. In the inverted curriculum, IsoScore instead jumps up at the reset and then decays within the epoch. This mirrors the epoch-level instabilities observed in the loss/perplexity curves.

Overall, baseline anisotropy is driven primarily by the *degree of local homogeneity* induced by the ordering criterion (length/readability), whereas schedule direction mainly controls the *timing* of the within-epoch oscillations. ReadIt, which mixes multiple difficulty cues, yields more heterogeneous local windows than pure surface sorting and therefore produces trajectories closer to Random.

(ii) Linear effective dimension covaries with isotropy in the encoder, but largely decouples in the decoder For the encoder, PCA@99 increases

[3] Due to space limitations, we cannot provide a detailed discussion of all features. Appendix C reports the average layer probing performance for the complete set of probing features.

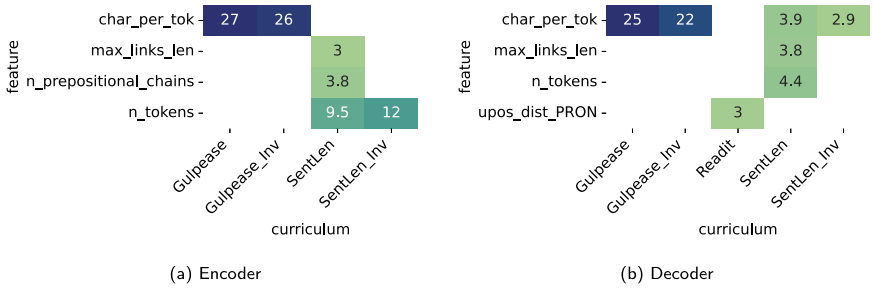


Fig. 9. Z-scores of encoder and decoder probing performance relative to random-order baselines, averaged across layers, seeds, and the last five checkpoints. Only features with z≥2 for all seeds of at least one curriculum are shown.

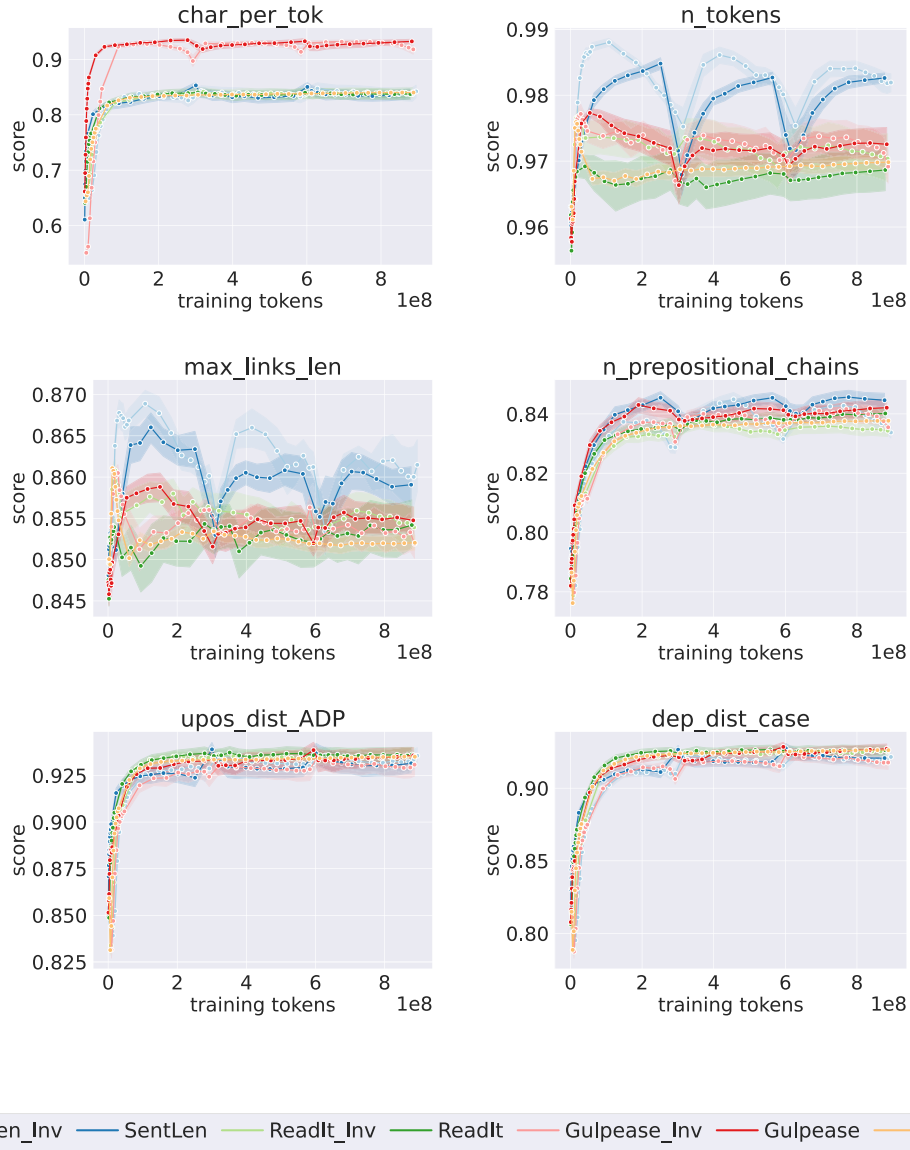


Fig. 10. Evolution of probing performance, averaged across layers, for the six encoder features with the highest z-scores relative to random baselines.

rapidly early in training and then saturates (Fig. 12, middle-left), consistent with an initial expansion into a higher-dimensional linear subspace followed by stabilization. The same curricula that reduce isotropy (Sentence Length and Gulpease, in both directions) also converge to smaller PCA@99 values. This coupling supports the interpretation that anisotropy reflects variance concentrating along a small number of globally dominant directions. In the decoder, by contrast, PCA@99 trajectories largely overlap across curricula (Fig. 12, middle-right). Thus, even when ordering induces sharp isotropy instabilities at epoch transitions, it has a weaker effect on the global eigenvalue spectrum of last-layer decoder representations under autoregressive training.

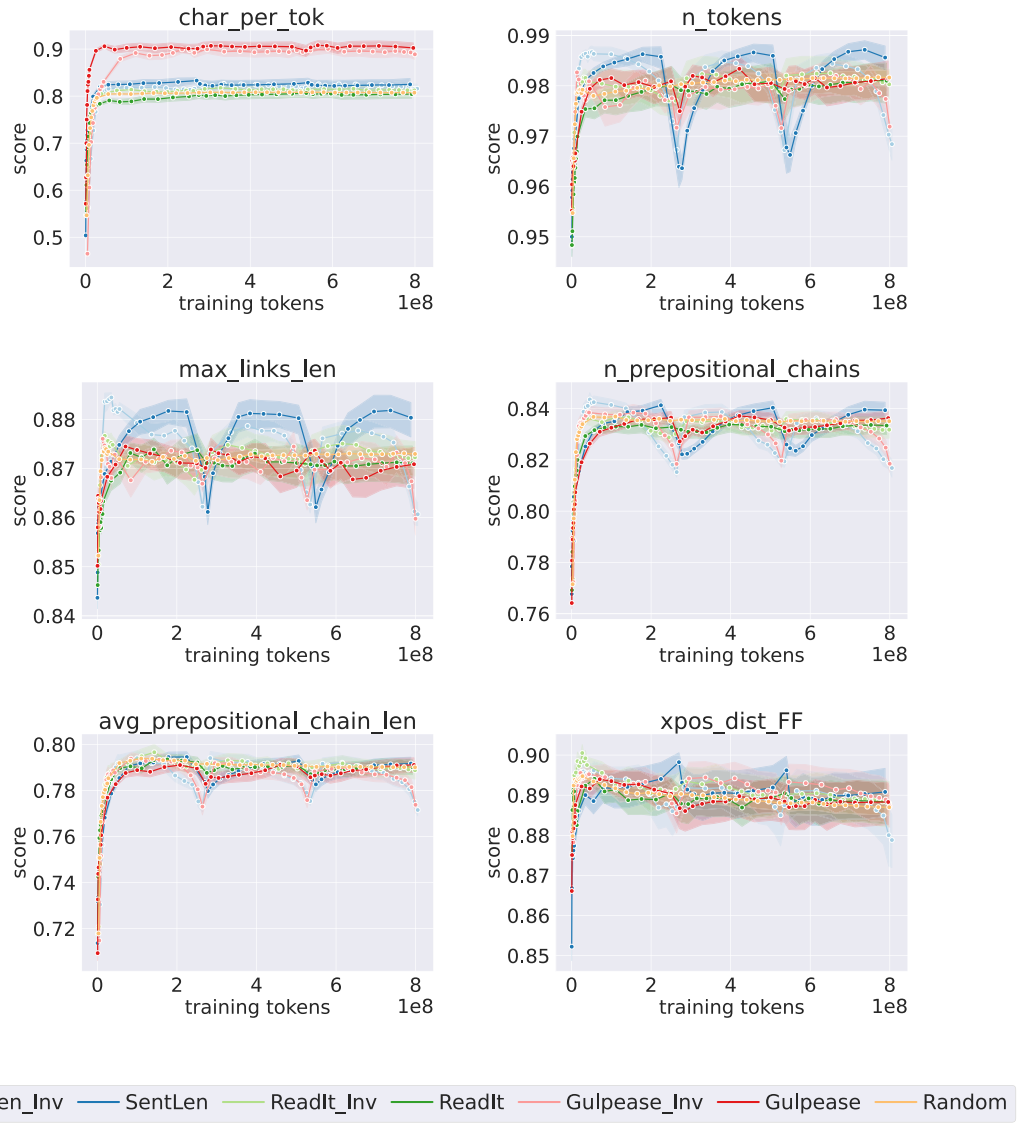


Fig. 11. Evolution of probing performance, averaged across layers, for the six decoder features with the highest z-scores relative to random baselines.

(iii) Nonlinear intrinsic dimension is comparatively robust to data ordering Across both architectures, TwoNN follows a consistent trajectory. Intrinsic dimensionality drops sharply in the earliest phase of training and then settles into a narrow plateau (Fig. 12, bottom row). While small curriculum-specific offsets are visible at early checkpoints, all orderings converge to very similar TwoNN values by the end of pre-training. Taken together with the IsoScore and PCA@99 trajectories, this pattern suggests that curriculum learning mainly affects *global* geometric structure (anisotropy and linear variance concentration), rather than the *local* degrees of freedom of the representation manifold.

A key reason is methodological. IsoScore and PCA@99 are global, covariance-driven diagnostics, and therefore respond strongly when an ordering concentrates variance along a small number of dominant directions. TwoNN, instead, is a *local* intrinsic-dimension estimator that infers dimensionality from the ratio between first- and second-nearest-neighbour distances. Because it relies on relative neighbourhood statistics (and is explicitly designed to reduce sensitivity to density variation and curvature), TwoNN is comparatively insensitive to global rescalings or covariance-spectrum distortions. As a result, curricula can substantially reshape isotropy and linear effective dimension without producing large, sustained changes in TwoNN.

We now examine how these geometric properties evolve across layers in the two models, highlighting their differences. Figs. 13 and 14 extend the analysis along network depth, considering early, intermediate, and late training checkpoints.

For BERT (Fig. 13), models trained with Random and ReadIt-based curricula consistently separate from those trained with Sentence Length- and Gulpease-based curricula across most layers, both in terms of IsoScore and PCA@99. This separation becomes increasingly pronounced at later checkpoints, suggesting that data ordering affects the global covariance structure of internal representations throughout the network, rather than being limited to the output layer. In contrast, TwoNN remains comparatively stable across curricula at all depths. Notably, at the earliest checkpoint, ReadIt (and its inverted variant) exhibits visibly wider across-seed standard deviation bands, indicating greater sensitivity to initialization and early optimization noise under this ordering. A plausible explanation is that ReadIt ranks examples using a composite readability signal (rather than a single proxy such as length), so the “easy” prefix can be less homogeneous and may induce higher gradient variability across seeds before representations stabilize.

The decoder (Fig. 14) exhibits a partially analogous pattern. Curriculum effects are most evident in isotropy across layers, while differ-

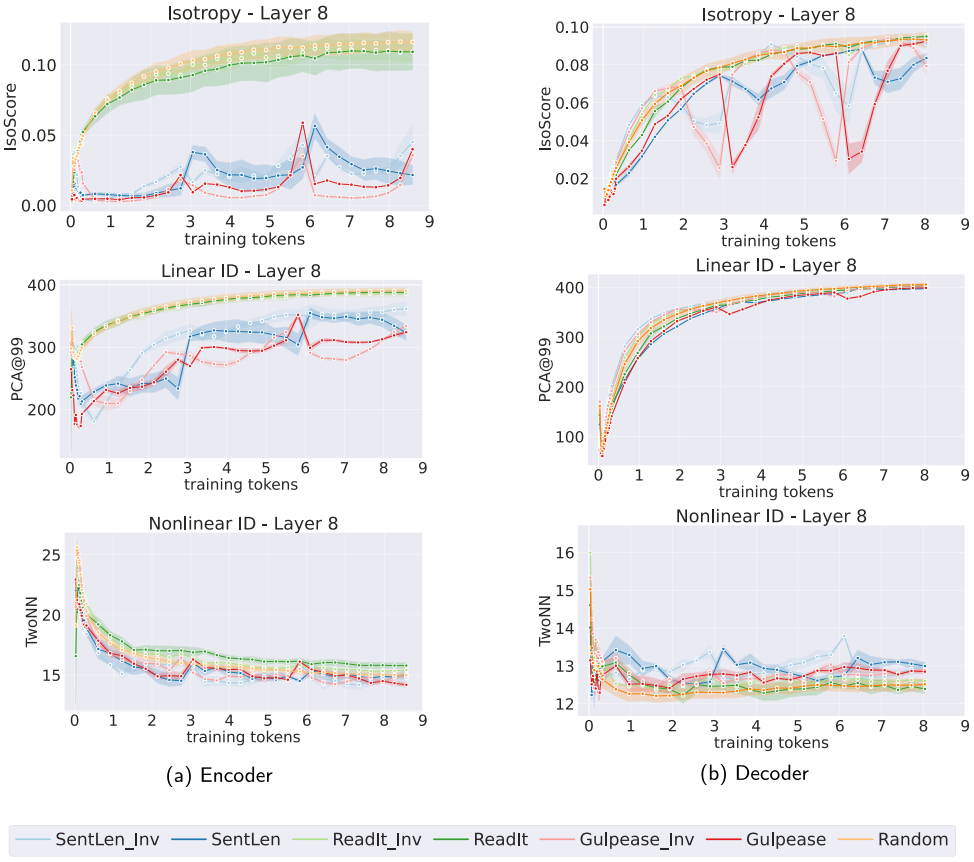


Fig. 12. IsoScore (top), PCA@99 linear intrinsic dimension (middle), and TwoNN (bottom) nonlinear intrinsic dimension (bottom) for encoder (left) and decoder (right) last-layer sentence representations across training checkpoints. Curves show mean over seeds, shaded regions denote across-seed standard deviation.

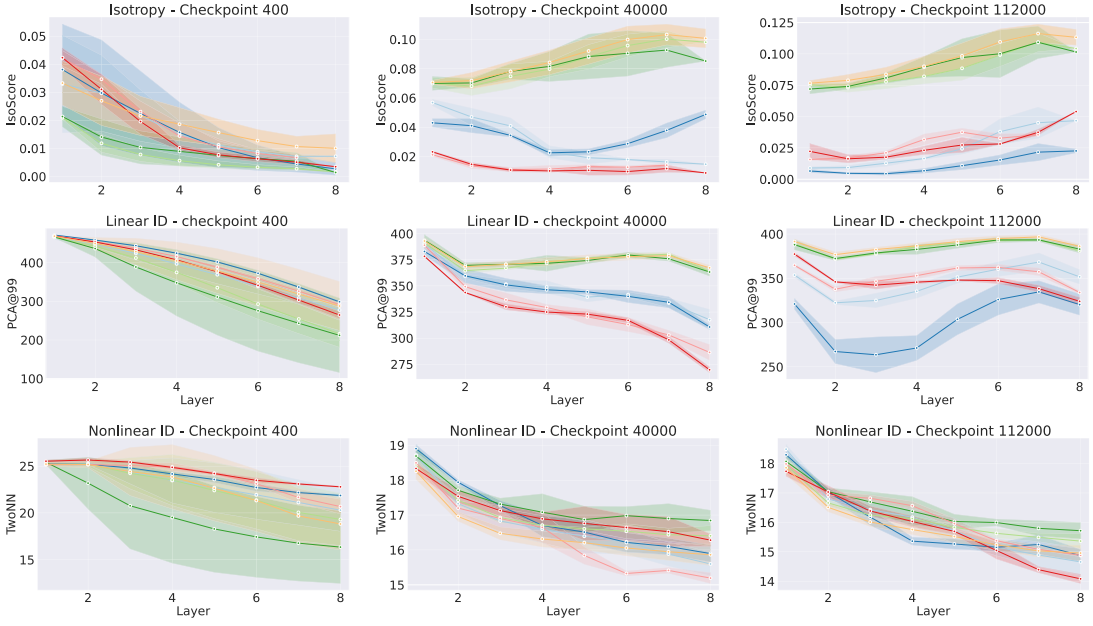


Fig. 13. Layerwise geometry in the encoder model at first, intermediate and last checkpoint. Rows correspond to IsoScore, PCA@99, and TwoNN, columns correspond to checkpoints. Curves show mean over seeds, shaded regions denote across-seed standard deviation.

ences in PCA@99 are more limited and intrinsic dimensionality remains tightly clustered. In addition, the decoder curves generally show smaller across-seed variability (narrower shaded bands) than the encoder. One plausible reason is that autoregressive next-token training provides a dense, consistent supervision signal at every position, whereas masked-LM training predicts only a subset of tokens and introduces additional stochasticity through the masking process, making early training dynamics more seed-sensitive.

Appendix E (Figs. E.1 and E.2) further extends the geometric analysis to all layers. Across much of the model stack, Random and ReadIt-based curricula remain clearly separated from Sentence Length- and Gulpease-based curricula in terms of isotropy and linear intrinsic dimensionality,

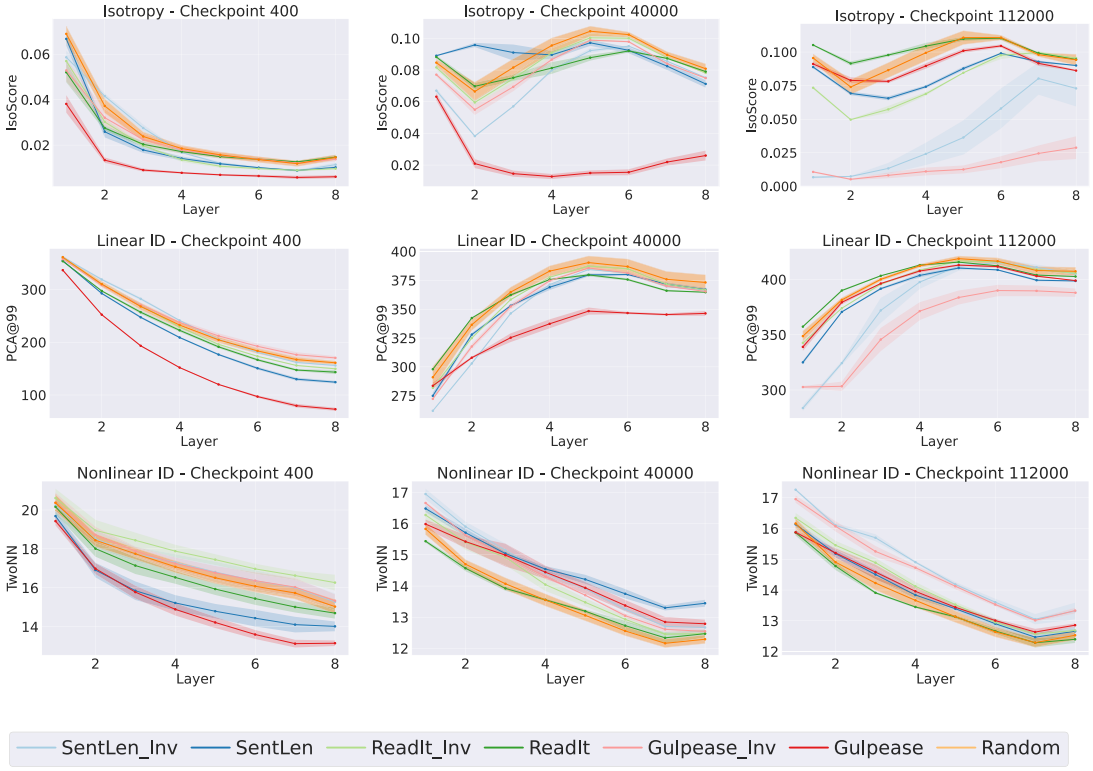


Fig. 14. Layerwise geometry in the decoder model at first, intermediate and last checkpoint.Rows correspond to IsoScore, PCA@99, and TwoNN, columns correspond to checkpoints. Curves show mean over seeds, shaded regions denote across-seed standard deviation.

whereas TwoNN consistently displays more stable behavior across different data orderings.

Key Findings: Curriculum learning reshapes the embedding space mainly by altering its global variance structure, rather than the underlying nonlinear intrinsic dimensionality. Orderings that cluster sentences according to superficial cues, such as sentence length, tend to amplify anisotropy and concentrate variance along a small number of dominant directions, particularly in encoder models. These effects depend more on local homogeneity and epoch-level distribution shifts than on the direction of the curriculum, while intrinsic dimensionality remains largely stable across ordering strategies.

5.4. Downstream-tasks

The last evaluation level on the impact of data ordering considers the model performance on the three downstream tasks described in Section 3.2.4: Part-of-Speech Tagging, Human Complexity Judgement, and Sentiment Analysis.

Before analysing learning trajectories, Table 3 provides an overview of the final performance achieved by each model at the end of pre-training. Overall, the differences across curricula are small, indicating that the final downstream generalization is only marginally affected by data ordering.

Across all downstream tasks and both the encoder and decoder architectures, performance is only weakly influenced by data ordering. In general, curriculum-based models achieve slightly lower scores than those trained with random orderings, although the differences are very small, typically ≈ 0.01 points. In a limited number of cases, curricula lead to marginal improvements over the random baselines, the only notable exception is the Gulpease-inverted curriculum, which reaches the best performance on sentiment analysis, particularly for the decoder model.

We next provide a finer-grained analysis of how models trained under different curricula evolve throughout pre-training. Consistent with the methodology adopted in the previous sections, results are presented as a function of the number of processed training tokens.

Part Of Speech Tagging

Fig. 15 shows the encoder (left) and decoder (right) models' F1-scores. For the encoder, POS tagging is only marginally affected by training-data ordering. During the initial phase of training, some small differences appear. Curriculum-trained models, especially those using the ReadIt ordering, show slightly steeper learning, while the sentence-length curriculum improves briefly but then slows down around the first 50M tokens. Anti-curricula, particularly the sentence-length anti-curriculum, converge more slowly in the early phase. These effects are short termed: after roughly 200M tokens, all variants converge tightly to an F1 of 0.96, with overlapping variance bands indicating no stable advantage for any curriculum.

The decoder follows the same pattern, though early differences are less pronounced. Gulpease-ordered training again yields slightly faster initial gains, but after $\approx$200M tokens all models converge, this time to slightly lower F1 values (0.90-0.92). Variance across seeds is also somewhat higher for the decoder.

Overall, complexity-based ordering can accelerate learning during the earliest training stages, but these benefits vanish once models approach convergence. Final POS-tagging performance is effectively independent of curriculum choice.

Human Complexity Judgment

Fig. 16 reports Mean Absolute Error (top) and Spearman correlation (bottom). For encoder models, performance across curricula is very similar, with maximal differences below 0.02 points. Learning curves are considerably noisier than in POS tagging, and relative rankings between curricula shift repeatedly throughout training. Sentence-length-based curricula are the most unstable and tend to underperform, especially

Table 3
Performance on downstream task resolution at the end of pre-training. **Bold** indicates scores at least 2 standard deviations above the average random score (or below in case of complexity MAE), while *italic* indicates scores at least 2 standard deviations below it (or above in case of complexity MAE).

	COMPLEXITY (MAE) ↓		COMPLEXITY (Spearman)↑		POS TAGGING (F1) ↑		SENTIMENT (Avg F1) ↑	
Curriculum	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
SENTLEN	*0.442*	0.442	*0.819*	0.833	*0.963*	*0.910*	0.745	0.723
SENTLEN_INV	0.422	0.439	0.830	**0.842**	*0.962*	*0.909*	0.739	0.727
GULPEASE	0.428	**0.437**	0.835	0.838	0.966	0.913	*0.733*	0.728
GULPEASE_INV	0.429	0.444	*0.825*	0.837	0.964	*0.911*	**0.755**	**0.742**
READIT	0.426	0.444	0.830	0.830	0.965	0.916	*0.731*	0.719
READIT_INV	0.427	0.440	0.830	**0.839**	*0.962*	0.913	0.740	0.730
RAND1	0.428	0.443	0.835	0.833	0.967	0.914	0.740	0.721
RAND1_INV	0.427	0.447	0.834	0.830	0.966	0.914	0.743	0.709
RAND2	0.430	0.446	0.828	0.835	0.967	0.915	0.744	0.730
RAND2_INV	0.427	0.449	0.833	0.831	0.967	0.916	0.744	0.726
RAND3	0.431	**0.438**	0.831	**0.839**	0.966	*0.911*	0.747	0.725
RAND3_INV	0.428	0.443	0.832	0.835	0.967	0.914	0.748	0.711
RAND4	0.421	0.451	0.837	0.828	0.965	0.914	0.742	0.726
RAND4_INV	0.421	0.446	0.838	0.830	0.964	0.914	0.742	0.725
RAND5	0.432	0.446	0.827	0.832	0.964	0.915	0.740	0.719
RAND5_INV	0.427	0.446	0.833	0.833	0.967	0.913	0.740	0.715

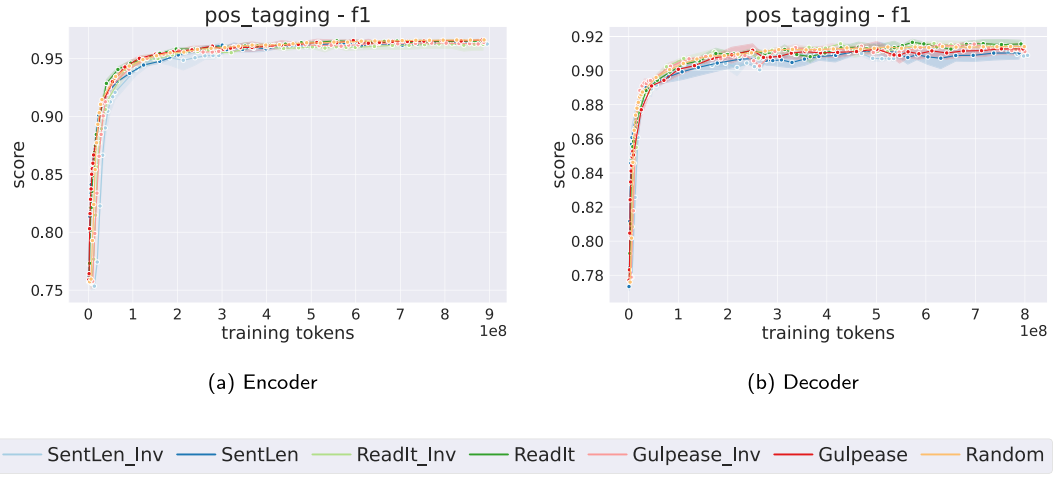


Fig. 15. Evolution of POS tagging performance during pre-training.

in the forward ordering. Readit-based orderings are the most stable, with Gulpease in between. This suggests that stability is influenced by whether the curriculum strongly correlates with sentence length, which may amplify distribution shifts across training phases.

The decoder models yield clearer patterns. Both the Sentence Length and Gulpease curricula show visibly faster convergence and, except for the Readit forward curriculum, remain consistently above the random baseline after the initial training phase. Although absolute gaps remain small (≈0.01), they persist throughout most of training, aside from the two epoch-shift discontinuities. This indicates that for the decoder, curriculum ordering confers a modest but consistent advantage, likely because the curricula reflect human-perceived difficulty signals that align well with the target task.

Sentiment Analysis

Fig. 17 presents the combined F1-score for Positive and Negative classes. Here, identifying meaningful differences between curricula is difficult: pre-training improves performance by only ≈0.04 points, so the overall impact is limited. As in the complexity task, curves fluctuate substantially, making relative rankings unreliable.

For the encoder, curriculum-based orderings tend to perform slightly worse than the random baseline, while anti-curriculum variants match or slightly exceed it, with Gulpease-inverted showing the clearest and most stable improvement.

For the decoder, the picture changes slightly. Except for the Readit curriculum (which struggles to converge in the first third of training) all other curricula perform similarly to or slightly better than the random baseline. Again, the strongest performance comes from the Gulpease-inverted ordering.

Across the three downstream tasks, a coherent pattern emerges regarding the influence of training-data ordering. POS tagging is essentially insensitive to curriculum design. Early-phase differences appear but quickly vanish, and all models converge to nearly identical final performance. In contrast, the Human Complexity Judgement task, especially for the decoder, shows mild but consistent advantages for complexity-based curricula, suggesting that curricula aligned with human notions of difficulty can support faster and slightly more stable acquisition. Sentiment analysis, where pre-training has only a marginal impact overall, does not exhibit a robust curriculum effect. The only exception is the Gulpease-inverted model, which sporadically outperforms the random baselines.

Key Findings: Curriculum learning yields only modest, task-dependent benefits, primarily in cases where the downstream task is conceptually

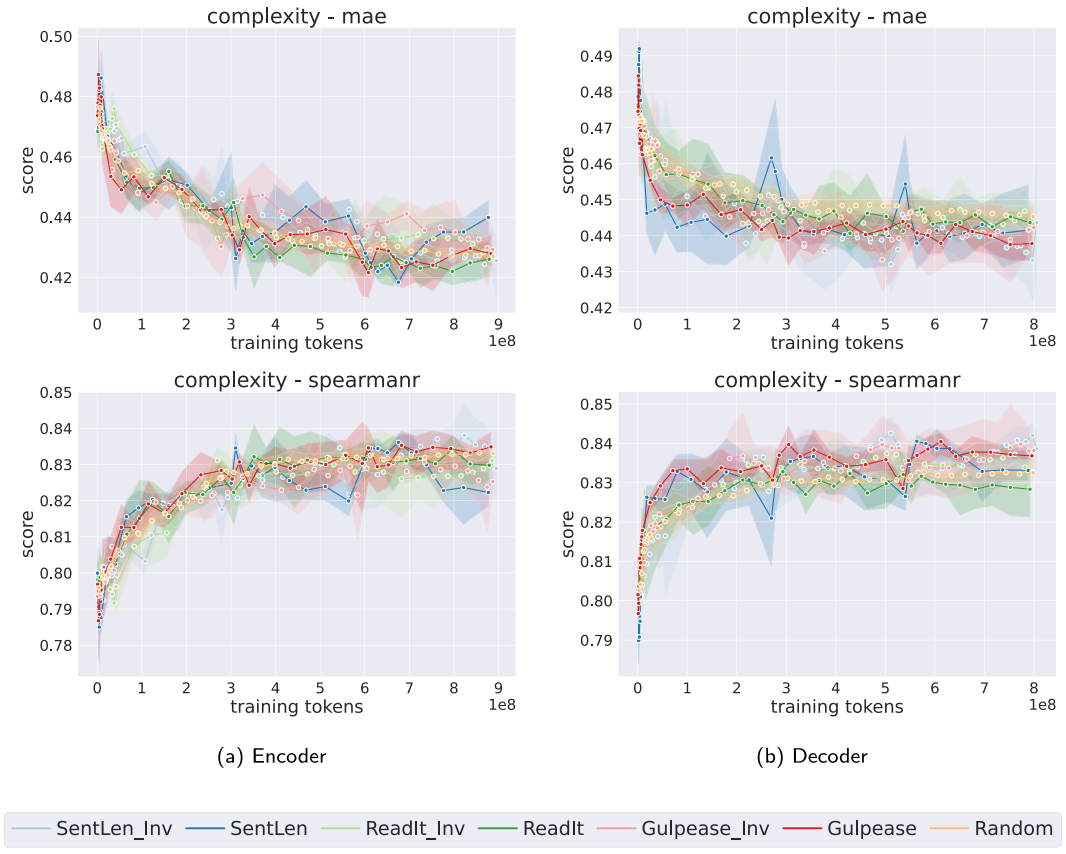


Fig. 16. Evolution of human complexity judgment performance during pre-training.

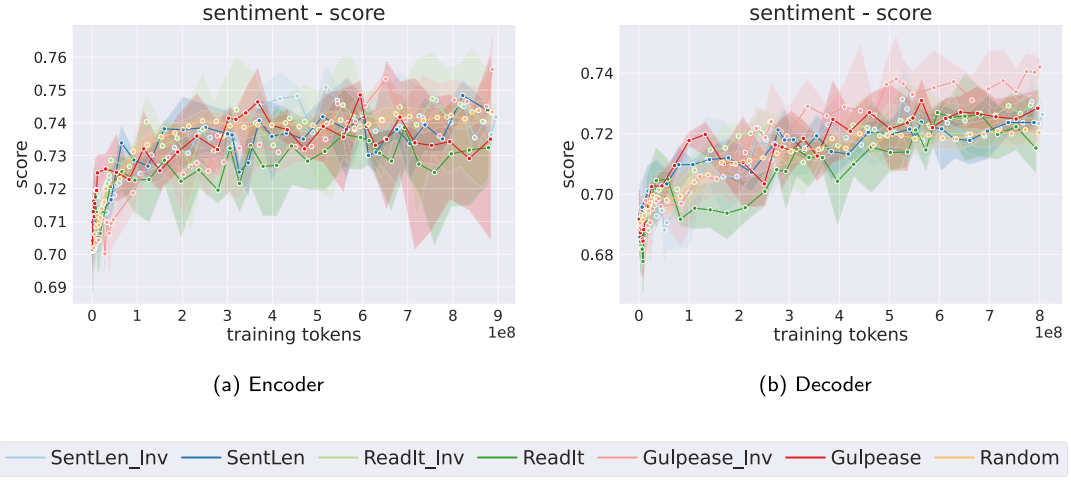


Fig. 17. Evolution of sentiment analysis performance during pre-training.

aligned with the complexity signal used to order the data. Despite substantial differences in training dynamics, all models ultimately converge to largely comparable levels of downstream performance. Any advantages conferred by a particular ordering are subtle, highly task-specific, and considerably smaller than the variance observed across different random seeds.

6. Discussion and conclusion

This work is designed to systematically examine pretraining data ordering as an independent design variable in transformer language models, and to characterize its effects across architectures and evaluation dimensions. Our findings provide a nuanced picture: while curriculum-based orderings do not uniformly improve final task performance, they exert measurable and interpretable effects on optimization dynamics, representation geometry, and the emergence of linguistic structure. Below, we discuss these results in light of the contributions outlined in the Introduction.

Data ordering as an independent factor in pretraining

A central contribution of this study is the isolation of data ordering from other pretraining variables such as data volume, model size, and optimization settings. By holding these factors constant and varying only the sequence in which training examples are presented, we show that data ordering alone meaningfully affects learning dynamics.

Across both encoder and decoder models, curriculum-based orderings produce optimization trajectories that differ from random baselines. In particular, curricula tend to accelerate early loss reduction while introducing instabilities at epoch boundaries, reflecting abrupt distributional shifts in globally sorted data. Because random orderings yield highly consistent behavior across seeds, these differences cannot be attributed to stochastic variation. Instead, they indicate that data ordering shapes the training path itself, even when final performance differences remain limited.

Importantly, these findings position data ordering as an orthogonal axis of pretraining design, distinct from data efficiency. Unlike cognitively motivated setups that reduce or constrain training input, our results demonstrate that even at fixed scale, the temporal organization of data can alter how models learn.

Effects of different curriculum-inspired ordering strategies

Beyond the general effect of ordering, our results show that the nature of the ordering signal is crucial. Not all curricula behave alike.

Orderings based on simple surface-level criteria, such as sentence length or Gulpease readability, produce stronger and more disruptive effects than Read-It, which combines multiple linguistic cues. Length- and Gulpease-based curricula reduce local diversity during training, leading to sharper loss oscillations and greater anisotropy in the embedding space, particularly for encoder models.

Interestingly, reversing a curriculum (i.e., training from hard to easy rather than easy to hard) often has a stronger impact than changing the complexity criterion itself, a pattern that is particularly evident in the linguistic probing results. This suggests that curriculum effects are driven less by an "easy-to-hard" progression and more by temporal clustering and distributional misalignment across training batches. In other words, it is the structure of exposure and not simply the difficulty level that determines how ordering influences learning.

Insights from a multi-dimensional evaluation framework

A key contribution of this work is the use of a multi-dimensional evaluation framework that goes beyond final loss or downstream accuracy.

Intrinsic evaluations (Sections 5.1) reveal that curriculum-based orderings primarily affect learning dynamics rather than end-state generalization. While random orderings lead to highly stable loss and perplexity behavior, curricula introduce distinct optimization trajectories and, in some cases, pronounced instabilities at epoch boundaries. These effects are particularly evident in sentence-level perplexity rankings for encoder models, indicating that data ordering can meaningfully reshape predictive uncertainty even when average perplexity differences remain small.

Probing-based analyses (Section 5.2) further clarify how these dynamics translate into linguistic representations. Early improvements in probing performance, and especially under inverted curricula, are often attributable to greater data exposure rather than faster acquisition of linguistic abstractions, as these gains largely disappear when training progress is aligned by the number of processed tokens. At later stages, however, probing results exhibit a consistent and interpretable pattern: linguistic features that benefit from curriculum learning tend to align closely with the curriculum's ordering signal. Length-based curricula preferentially enhance sentence-length-related and structurally correlated features, while curricula based on more complex readability metrics early promote a better encoding of an heterogeneous set of linguistic phenomena. These feature-specific effects persist across architectures despite modest changes in average probing scores, indicating that curriculum learning selectively biases which linguistic properties are emphasized.

The geometric analysis of representation space (Section 5.3) complements these findings by showing that curriculum learning primarily affects global variance structure rather than the underlying nonlinear dimensionality of representations. Orderings based on shallow complexity cues amplify anisotropy and concentrate variance along a small number of dominant directions, particularly in encoder models, while nonlinear intrinsic dimensionality remains largely stable across curricula. This explains how curricula can induce substantial representational differences without consistently improving downstream task performance.

Finally, downstream evaluations (Section 5.4) show that the effects observed during pretraining only partially transfer to task-level performance. While some curricula yield modest gains on specific tasks, these improvements are neither uniform nor robust across architectures and tasks. This reinforces the broader conclusion that curriculum learning in large-scale pretraining functions less as a general-purpose performance enhancer and more as a mechanism for shaping learning trajectories and representational biases.

Architecture-dependent responses to data ordering

By contrasting encoder-only and decoder-only models under identical training conditions, we uncover systematic architecture-specific sensitivities to data ordering. Encoder models are markedly more sensitive: curriculum-based orderings substantially alter perplexity rankings, probing performance dynamics, and embedding geometry. Decoder models, by contrast, are more robust to forward curricula, with significant effects emerging mainly under anti-curricula.

These differences likely reflect the interaction between curriculum-induced distribution shifts and architectural inductive biases. Bidirectional masked language modeling encourages rapid abstraction over global context, making encoder models more susceptible to homogeneity and abrupt shifts in input statistics. Autoregressive decoders, while affected by curriculum-induced non-stationarity, appear more stable in their final representational structure, particularly in terms of linear intrinsic dimension.

Importantly, these findings suggest that curriculum strategies cannot be assumed to transfer uniformly across architectures. Ordering schemes that are benign or even beneficial for decoders may be disruptive for encoders, and vice versa.

Taken together, our results suggest that curriculum learning in large-scale language model pretraining is best understood as a biasing mechanism rather than a universally beneficial optimization strategy. Data ordering influences which linguistic properties are learned more strongly, how representations are organized, and how stable learning dynamics are, but it does not reliably improve overall language competence or downstream performance.

From a practical perspective, this implies that curriculum design should be guided by specific representational or behavioral goals, rather than adopted as a default training strategy. Moreover, care must be taken to avoid unintended distribution shifts, particularly when using globally sorted curricula over multiple epochs.

7. Limitations and future work

While this work provides a systematic analysis of pretraining data ordering in transformer language models, several limitations remain and suggest directions for future research.

First, our experiments are conducted at a moderate model scale (65M parameters) and within a single linguistic setting (Italian). This design enables dense checkpointing, multiple curricula and random baselines, and detailed probing and geometric analyses under strict factorial control. However, curriculum effects may interact with model scale, training stability, or language-specific distributional properties. Extending the analysis to larger models and to morphologically and typologically diverse languages represents an important direction for assessing the robustness and generality of the observed patterns.

Second, we focus on fixed, global ordering strategies defined prior to training. While this design choice allows us to isolate the effect of data ordering under controlled conditions, it does not capture adaptive or model-driven curricula that evolve in response to model competence or uncertainty. Exploring dynamic ordering strategies, such as those based on model confidence, loss, or representational change, could clarify whether more responsive curricula mitigate some of the instabilities observed here.

Third, our downstream evaluation emphasizes controlled probing and classification-based tasks that can be consistently applied across encoder-only and decoder-only architectures. While this facilitates direct comparison of architecture-dependent curriculum effects, it does not fully assess the impact of data ordering in large-scale generative settings. Evaluating curriculum-based pretraining in tasks such as summarization or translation would help clarify its practical implications for end-to-end generation.

Overall, these limitations point to a broader research agenda in which data ordering is studied alongside scale, adaptivity, and representational analysis. We view the present work as a foundation for such investigations, providing both methodological tools and empirical insights for future exploration.

CRediT authorship contribution statement

Luca Dini: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Lucia Domenichelli:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Dominique Brunato:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization; **Felice Dell'Orletta:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the project "Advancing Italian Language Processing with Small-Scale Training and Preference Modeling" (IsCb8_AILP), funded by CINECA under the ISCRA initiative, for the availability of HPC resources and technical support.

Appendix A. List of linguistic features extracted by profiling-UD

Table Table A.1 reports the set of features extracted with profiling-UD, along with their description, used to conduct probing analyses across the different layers of the models.

Appendix B. Perplexity evolution during pre-training

This appendix provides a more detailed analysis of model perplexity, extending the results presented in Section 5.1. Fig. B.1 shows the evolution of perplexity during pre-training, evaluated on both the in-domain Wikipedia test set and the out-of-domain UD Treebank.

For both models, we observe that the Gulpease curriculum and anti-curriculum exhibit sharp increases in perplexity at epoch boundaries, mirroring the peaks seen in the training loss. In the encoder model, a similar pattern is visible also for the Sentence Length curriculum and anti-curriculum, whereas in the decoder model this effect, although present, is less pronounced. To better visualize the differences among the remaining conditions, Fig. B.2 displays the same curves with the y-axis truncated to retain the 80% of the lowest scores.

Interestingly, the Read-It curriculum and its anti-curriculum behave differently from the others: their perplexity trajectories remain close to those of the random models, with only slightly higher values, particularly for the decoder model. Across all training regimes, models show lower perplexity on the in-domain evaluation set (Wikipedia) and higher perplexity on the out-of-domain set (Treebank). This pattern indicates that data ordering has limited influence on generalization as measured by perplexity. Moreover, the convergence speed of the Read-It curriculum, the only curriculum that reaches stable perplexity values comparable to the random baselines, is also very similar to that of the random model.

Appendix C. Probing results for each linguistic feature

This appendix extends the analysis of Section 5.2 by reporting probing scores separately for each individual linguistic feature.

Figs. C.1 and C.2 show last-layer probing performance at the final training checkpoint for all curriculum-based models, together with the average performance of the random baselines, for the encoder and decoder architectures, respectively. The row labeled `AVG` reports the mean performance across all probing features for each model.

With the exception of the *char_per_tok* feature, on which Gulpease-based models achieve markedly higher scores than all other models, curriculum-based models generally perform comparably to, or slightly worse than, the random baselines (typically by 0.01-0.02 points). For the remaining features, complexity-based pre-training orderings do not substantially alter which linguistic properties are easy or difficult to predict. This is reflected in the heatmaps by the alignment of darker and lighter regions along rows, indicating that each feature gets similar scores across different curricula.

To provide a broader view, Figs. C.3 and C.4 report the same probing results across all model layers, for the encoder and decoder architectures, respectively. These plots make the relative underperformance of certain curricula more apparent. For the encoder, Sentence Length, Sentence Length Inverted, and Gulpease Inverted perform worse than the other curricula. For the decoder, the weakest results are observed for Sentence Length Inverted and Gulpease Inverted.

Overall, the lowest-performing decoder models tend to deviate more strongly from the remaining curricula than is observed for the encoder, suggesting a higher sensitivity of decoder representations to data ordering. Another notable pattern is that Gulpease-based curricula appear to support learning for several linguistic features. Although these effects do not yield strong statistical signals under the z-score criteria, the plots, particularly for the encoder, show multiple features for which Read-It and Read-It Inverted achieve higher scores than the random baselines. These improvements are likely not captured by the z-score analysis because they are not consistent across all layers or not large enough in magnitude.

Finally, Figs. C.5 and C.6 report the average probing performance across layers for each linguistic feature throughout pre-training. In these plots, the weaker curricula are again more clearly identifiable. For the encoder, both Sentence Length-based curricula and Gulpease Inverted exhibit consistently lower performance. For the decoder, all curricula except the two Read-It-based orderings underperform relative to the Random models. This behavior is likely related to the pronounced drops in probing performance observed at epoch boundaries, from which these models do not fully recover before the subsequent epoch begins.

At the same time, this training-time perspective, particularly for the decoder, highlights that Gulpease-based curricula can yield slightly higher performance or faster convergence for some linguistic features. While these effects are modest, they are consistent with the feature-level patterns observed in the layer-wise analyses.

Table A.1
Description of the linguistic features extracted by Profiling-UD.

Feature	Description
`n_tokens`	Total number of tokens.
`char_per_tok`	Average number of characters per word (punctuation excluded).
`lexical_density`	Ratio of content words (NOUN, PROPN, VERB, ADJ, ADV) over the total number of words in a document.
`upos_dist_<UPOS>`	Distribution of the 17 Universal POS tags (e.g., `upos_dist_NOUN`, `upos_dist_VERB`); see https://universaldependencies.org/u/pos/.
`verbs_voice_dist_Active`	Distribution of verbs in the active form.
`verbs_voice_dist_Passive`	Distribution of verbs in the passive form.
`verbs_mood_dist_Ind`	Distribution of indicative mood verbs.
`verbs_mood_dist_Part`	Distribution of participle verbs.
`verbs_mood_dist_Inf`	Distribution of infinitive verbs.
`verbs_tense_dist_Pres`	Distribution of present tense verbs.
`verbs_tense_dist_Past`	Distribution of past tense verbs.
`verbs_tense_dist_PassPross`	Distribution of present perfect simple verbs.
`verbs_num_pers_dist_Sing+3`	Distribution of third-person singular verbs.
`verbs_num_pers_dist_Plur+3`	Distribution of third-person plural verbs.
`avg_token_per_clause`	Average number of tokens per clause.
`avg_links_len`	Average linear distance (in tokens) between a dependent and its head, averaged across all dependency links.
`max_links_len`	Longest distance (in tokens) between a head and its dependent.
`dep_dist_<DEPREL>`	Distribution of the 37 Universal Dependency relations (e.g., `dep_dist_nsubj`, `dep_dist_obj`); see https://universaldependencies.org/u/dep/.
`subj_pre`	Distribution of preverbal subjects.
`subj_post`	Distribution of postverbal subjects.
`obj_post`	Distribution of postverbal objects.
`n_prepositional_chains`	Number of prepositional chains (embedded prepositional complements depending on a noun).
`avg_prepositional_chain_len`	Average length of prepositional chains (number of embedded complements introduced by a preposition that depend on a noun).
`avg_subordinate_chain_len`	Average length of subordinate chains (average number of embedded complements headed by a subordinate clause).
`principal_proposition_dist`	Distribution of main clauses.
`subordinate_proposition_dist`	Distribution of subordinate clauses.
`subordinate_post`	Distribution of subordinate clauses following the main clause.
`verb_edges_dist_<k>`	Distribution of verbs by arity class; arity is the number of dependents for each verbal head.
`avg_verb_edges`	Average verb arity.

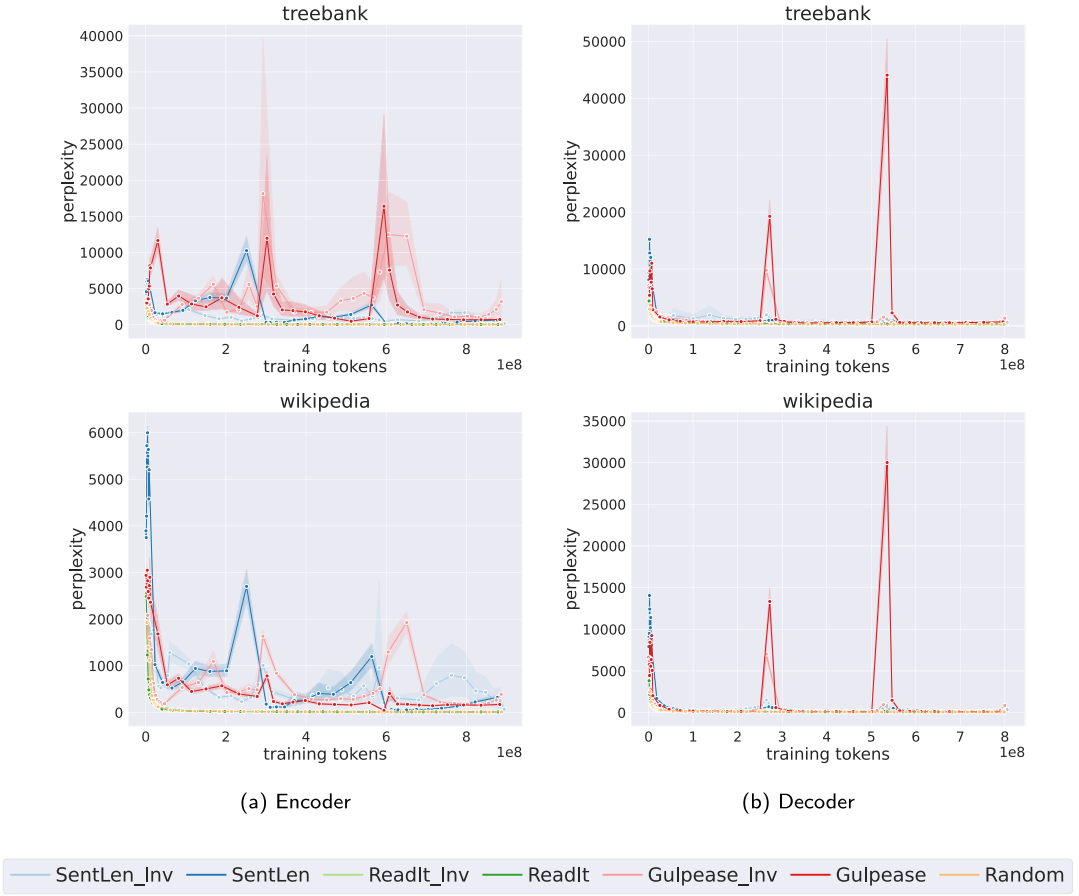


Fig. B.1. Evolution of pseudo-perplexity (encoder) and perplexity (decoder) during pre-training on the UD Treebank and Wikipedia datasets.

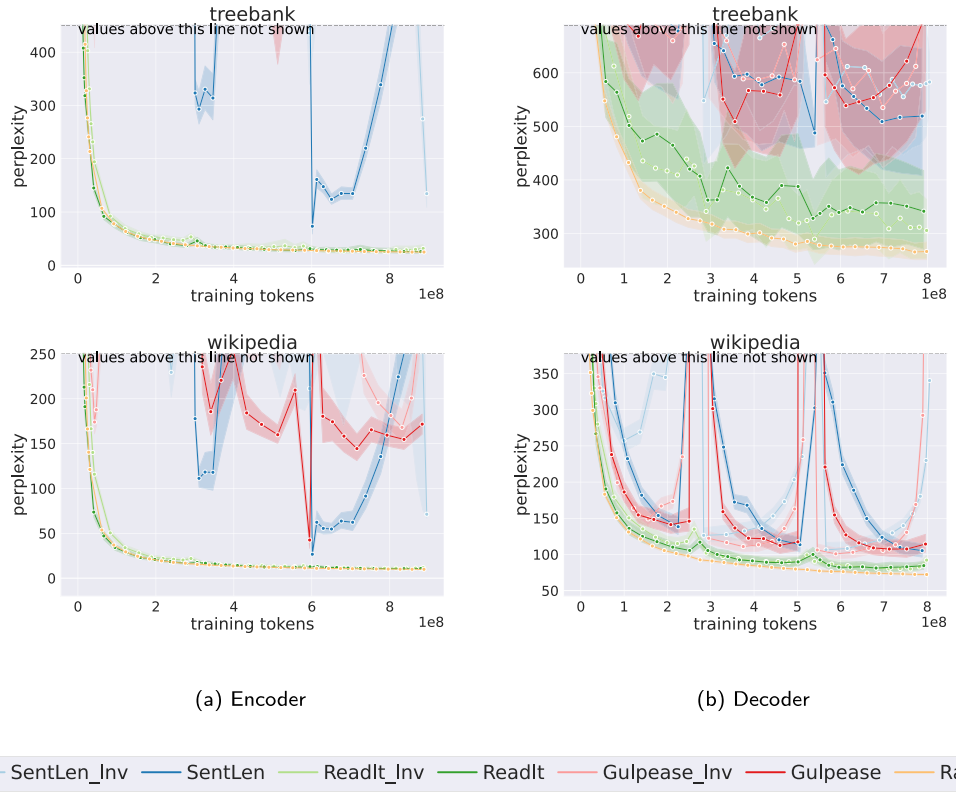


Fig. B.2. Zoomed view of pseudo-perplexity (encoder) and perplexity (decoder) during pre-training, restricted to the lower 80th percentile of values.

Appendix D. Geometric metrics

Geometric scores are computed on a random subsample of (N = 10,000) sentence representations (ambient dimension $d = 768$) for each checkpoint. This choice reflects a standard trade-off in intrinsic-geometry analyses: increasing (N) improves estimator stability but rapidly increases the cost of distance- and covariance-based computations. Intrinsic-dimension surveys explicitly discuss this stability/tractability trade-off and commonly rely on subsampling regimes when operating in high-dimensional settings [51].

For nonlinear intrinsic dimension, we use TwoNN, a nearest-neighbour estimator based on the ratio between first- and second-neighbour distances. Because TwoNN is fundamentally local and scale-dependent, its estimates can vary with sampling density (i.e., "data resolution"). Changing N changes typical neighbourhood radii and may affect the inferred slope. Moreover, in finite-sample, high-dimensional regimes, nearest-neighbour statistics can lead to systematic underestimation effects, as discussed in Facco et al. [47].

In contrast, our linear metric PCA@99 is derived from the global covariance spectrum and tends to stabilize earlier in practice than local kNN-based estimators. More generally, linear "effective dimension" estimators have been argued to be comparatively robust in high-dimensional noisy regimes [53]. Finally, for Isotropy we use IsoScore, which was proposed specifically to provide a reliable scalar measure of how uniformly variance is distributed across dimensions, addressing known brittleness of earlier isotropy proxies [44]. Recent work applying geometric analyses to encoder representations likewise reports that isotropy-based diagnostics are stable and informative for distinguishing isotropic vs. anisotropic point clouds [35].

Appendix E. Geometric scores over layers

Figs. E.1 and E.2 extend the last-layer geometric diagnostics by reporting IsoScore (left), PCA@99 (middle), and TwoNN (right) *at every layer* across training checkpoints. For the encoder, the qualitative picture is consistent throughout the stack: Random and ReadIt orderings rapidly reach higher and comparatively smoother IsoScore trajectories, while Sentence Length and Gulpease curricula produce markedly lower IsoScore values for most of training, punctuated by sharp, synchronized transients that recur at approximately the same checkpoints across layers (notably around epoch boundaries). PCA@99 closely mirrors these effects: the low-IsoScore regimes correspond to a smaller number of principal components explaining 99% of the variance, indicating stronger variance concentration along a limited set of directions. Across depth, the encoder's IsoScore plateaus tend to decrease slightly towards higher layers, suggesting that upper encoder layers become progressively more anisotropic while preserving the same curriculum-induced ordering.

For the decoder, IsoScore increases in all curricula and layers and occupies a narrower final range than in the encoder, but the same qualitative separation remains visible in stability, indeed Random and ReadIt are comparatively smooth, whereas the surface-statistic curricula (Sentence Length and Gulpease, in both forward and inverted variants) show pronounced oscillations and transient collapses around epoch boundaries. By upper decoder layers, PCA@99 curves largely converge across curricula, suggesting that most curriculum-specific geometric effects in the decoder are short-lived and most clearly expressed through isotropy dynamics rather than persistent differences in the final covariance spectrum.

Finally, TwoNN varies comparatively little across curricula in both encoder and decoder (relative to IsoScore/PCA@99). Encoder TwoNN typically exhibits a sharp early drop followed by a plateau, while decoder TwoNN is comparatively flatter after the initial transient. Taken together, this pattern supports the interpretation that curriculum primarily modulates *global variance allocation* (isotropy and PCA-based effective dimension) rather than inducing large, sustained changes in the *local* intrinsic dimension estimated by TwoNN.

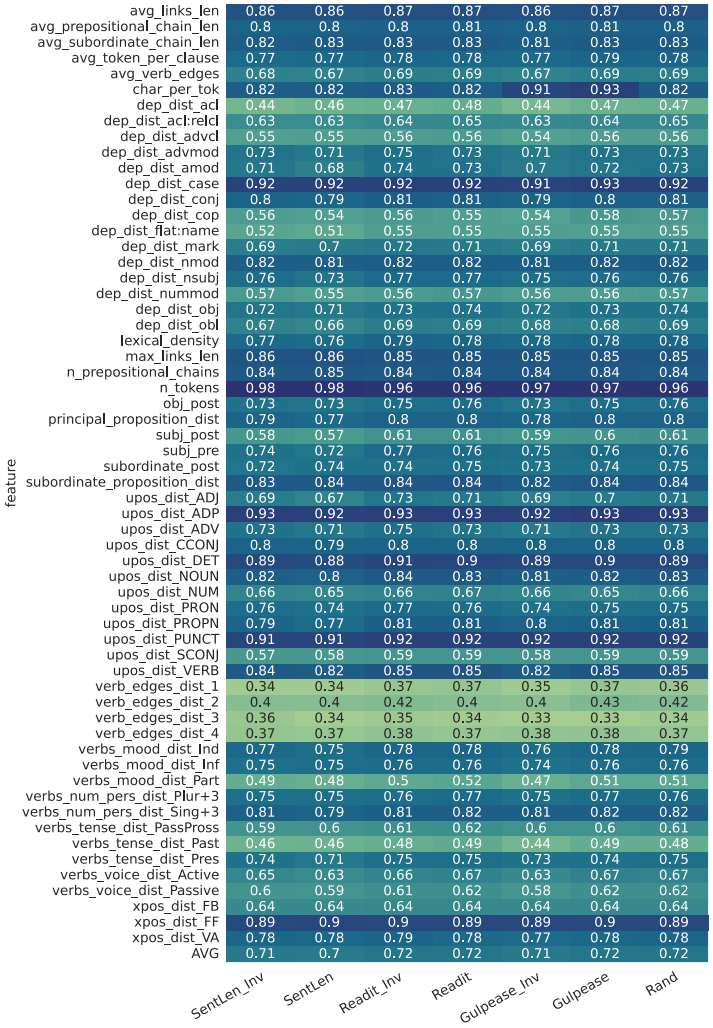


Fig. C.1. Probing performance across all linguistic features at the final layer and final pre-training checkpoint for the encoder model.

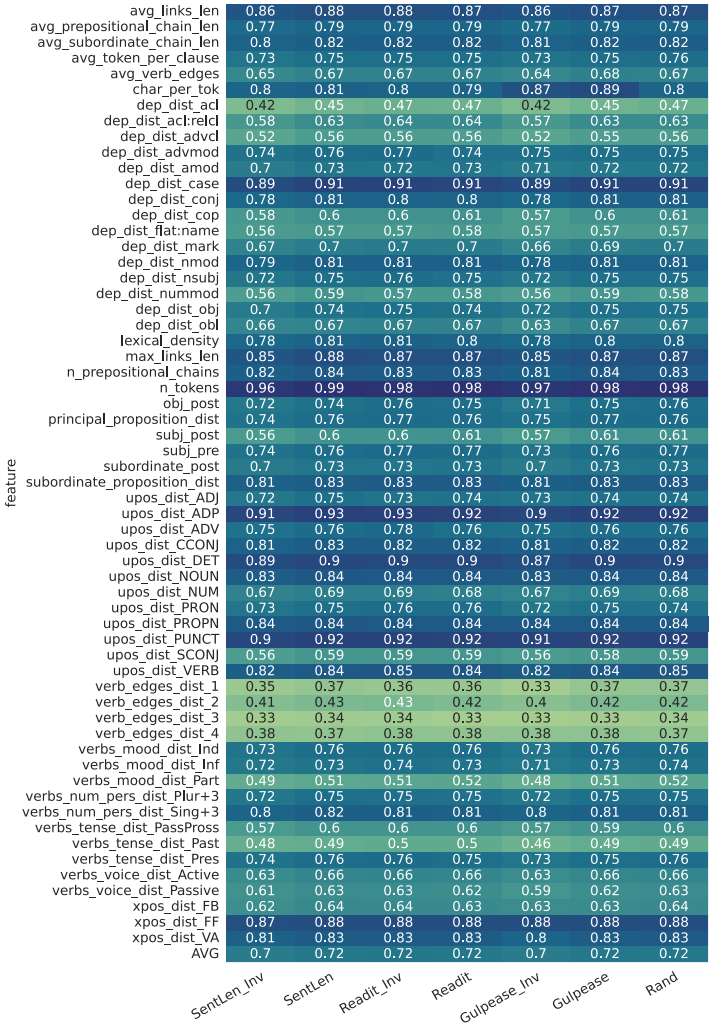


Fig. C.2. Probing performance across all linguistic features at the final layer and final pre-training checkpoint for the decoder model.

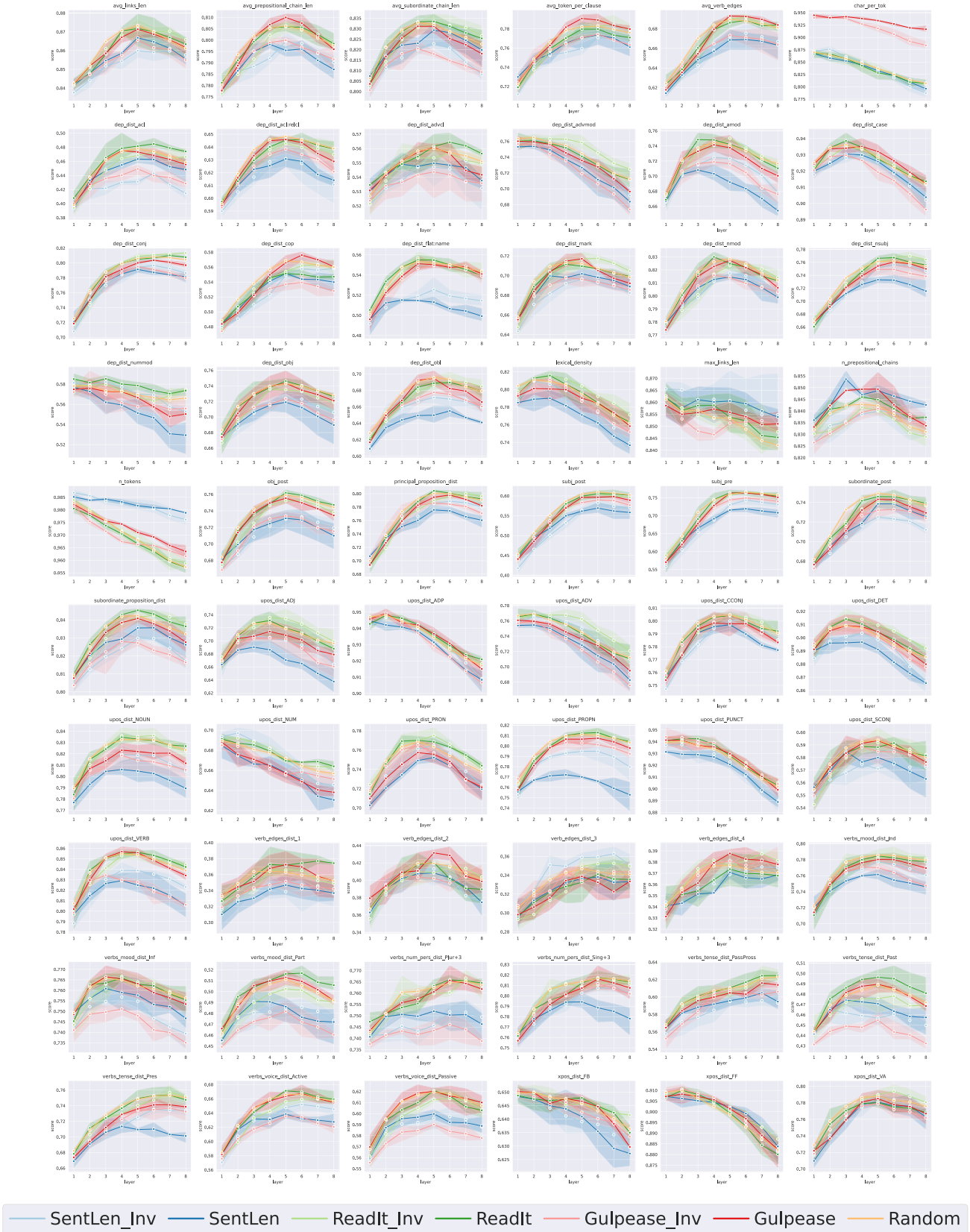


Fig. C.3. Layer-wise probing performance across all linguistic features at the final pre-training checkpoint for the encoder model.

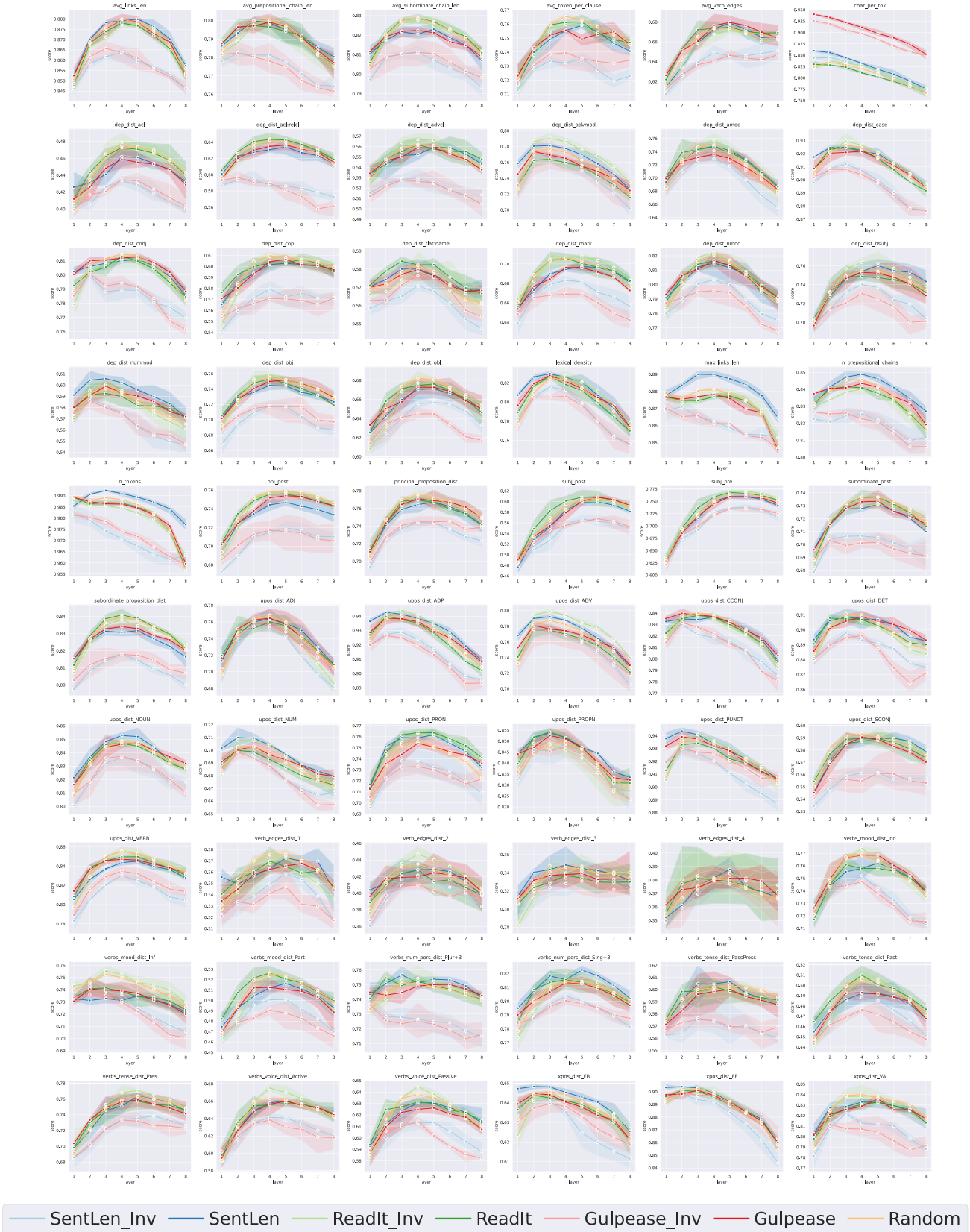


Fig. C.4. Layer-wise probing performance across all linguistic features at the final pre-training checkpoint for the decoder model.

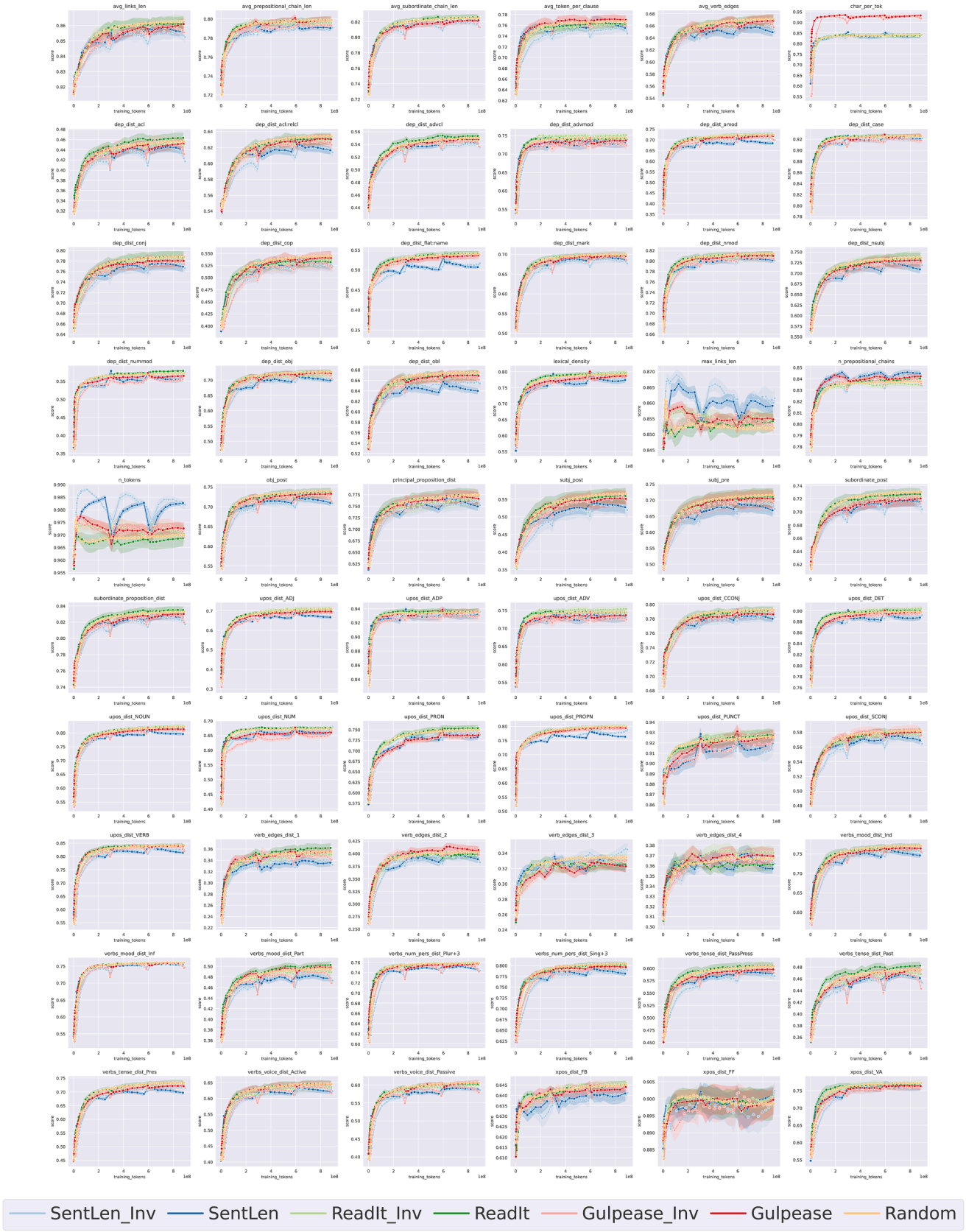


Fig. C.5. Evolution of probing performance during pre-training, averaged across layers, for the encoder model.

Fig. C.6. Evolution of probing performance during pre-training, averaged across layers, for the decoder model.

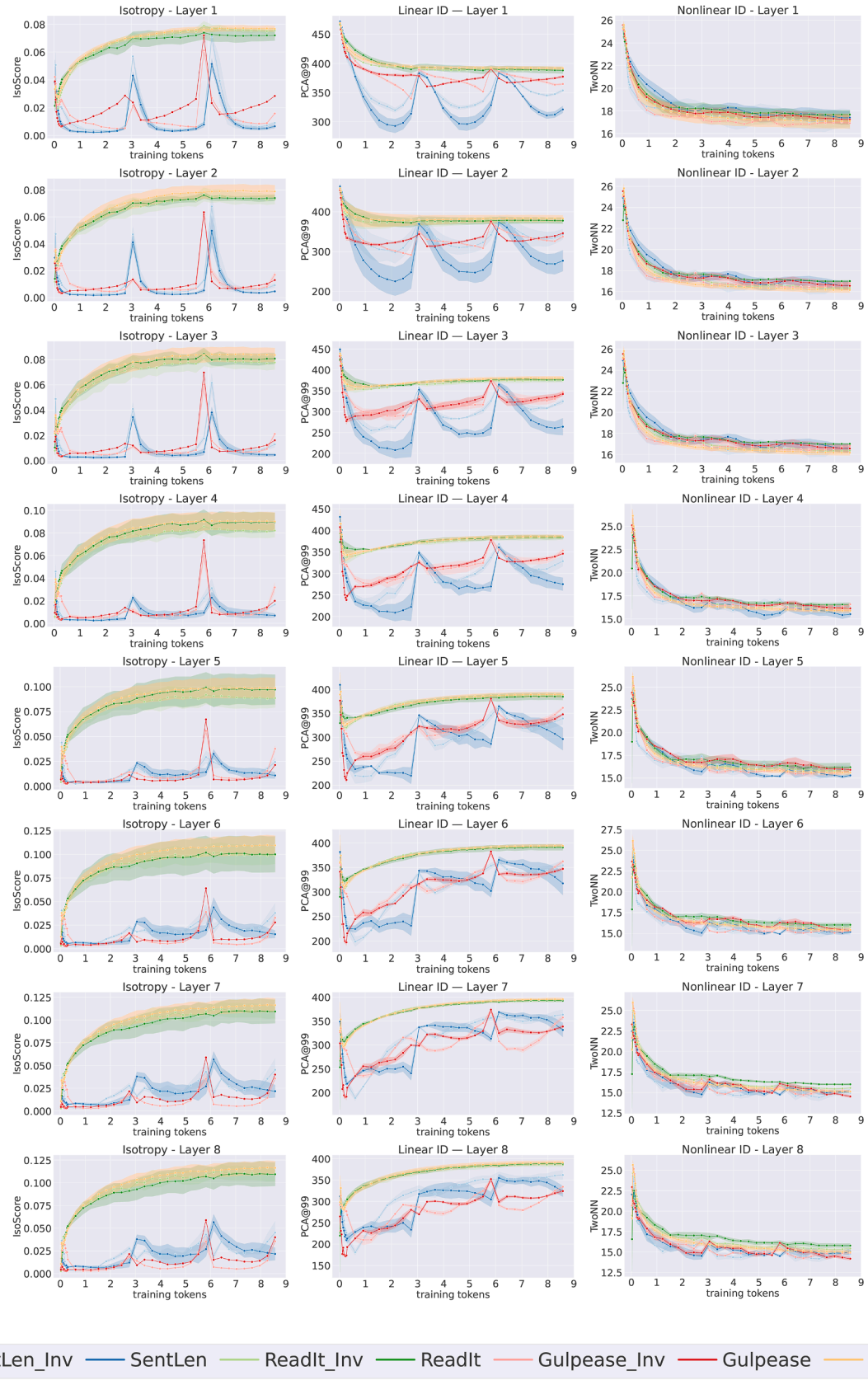


Fig. E.1. Layer-wise geometric scores across all layers at the final pre-training checkpoint for the encoder model. Isotropy, Linear ID and Nonlinear ID respectively on first, second and third column.

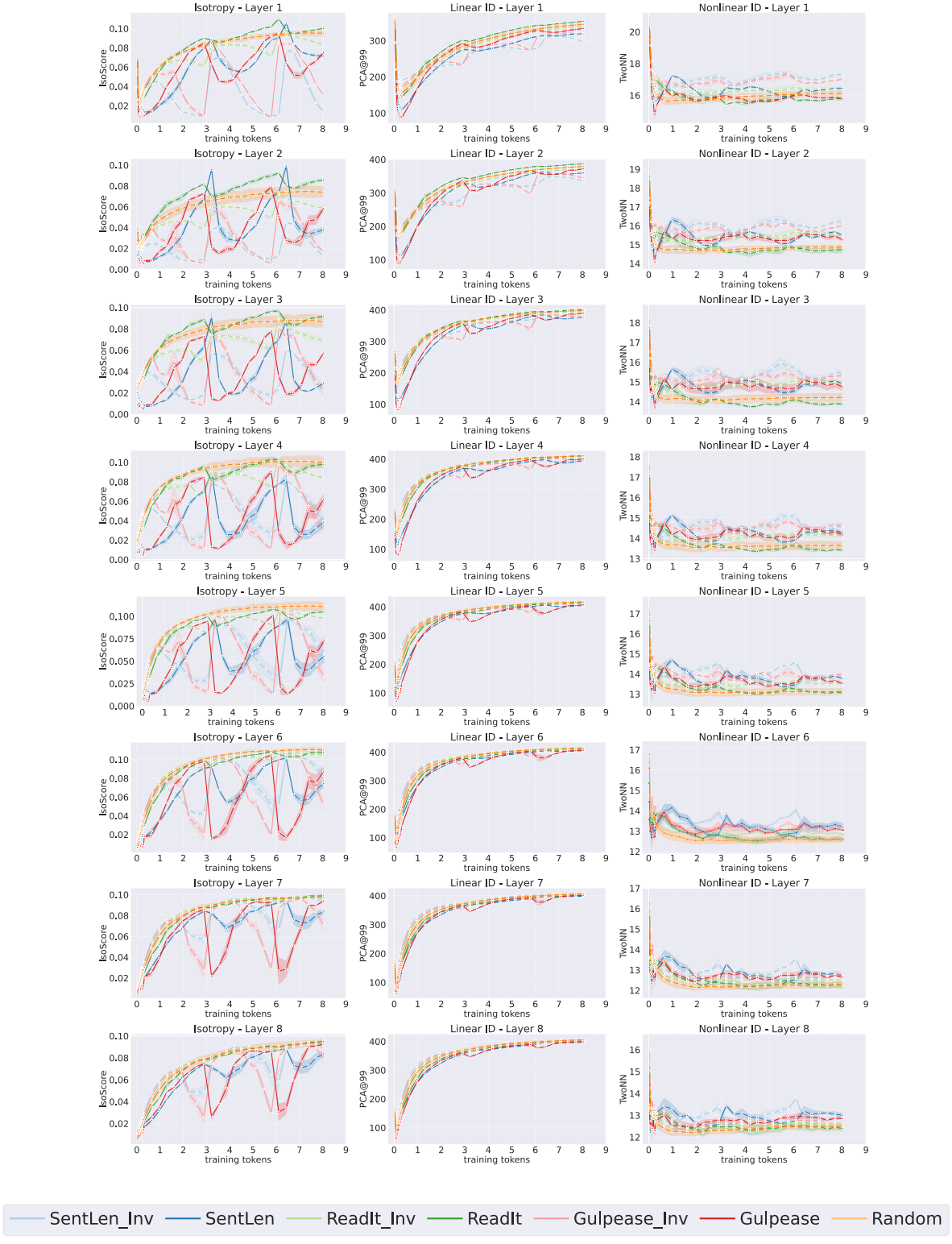


Fig. E.2. Layer-wise geometric scores across all layers at the final pre-training checkpoint for the decoder model. Isotropy, Linear ID and Nonlinear ID respectively on first, second and third column.

References

[1] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 41-48. https://doi.org/10.1145/1553374.1553380

[2] Y. Tsvetkov, M. Faruqui, W. Ling, B. MacWhinney, C. Dyer, Learning the curriculum with Bayesian optimization for task-specific word representation learning, in: K. Erk, N.A. Smith (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 130–139. https://doi.org/10.18653/v1/P16-1013

[3] Y. Zhou, B. Yang, D.F. Wong, Y. Wan, L.S. Chao, Uncertainty-Aware curriculum learning for neural machine translation, in: Annual Meeting of the Association for Computational Linguistics, 2020. https://api.semanticscholar.org/CorpusID:220047761.

[4] M.Y. Hu, A. Chen, N. Saphra, K. Cho, Latent state models of training dynamics, Transactions on Machine Learning Research (2023). , https://openreview.net/forum?id=NE2xXWo0LF.

[5] A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, et al., Findings of the BabyLM challenge: sample-efficient pretraining on developmentally plausible corpora, in: Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, 2023, pp. 1–34.

[6] M.Y. Hu, A. Mueller, C. Ross, A. Williams, T. Linzen, C. Zhuang, R. Cotterell, L. Choshen, A. Warstadt, E.G. Wilcox, Findings of the second babyLM challenge: sample-efficient pretraining on developmentally plausible corpora, in: M.Y. Hu, A. Mueller, C. Ross, A. Williams, T. Linzen, C. Zhuang, L. Choshen, R. Cotterell, A. Warstadt, E.G. Wilcox (Eds.), The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, Association for Computational Linguistics, Miami, FL, USA, 2024, pp. 1–21. https://aclanthology.org/2024.conll-babylm.1/.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, et al., BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[9] J. Mange, Effect of training data order for machine learning, in: 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 406–407. https://doi.org/10.1109/CSCI49370.2019.00078

[10] J.L. Elman, Learning and development in neural networks: the importance of starting small, Cognition 48 (1) (1993) 71–99. https://www.sciencedirect.com/science/article/pii/0010027793900584. https://doi.org/https://doi.org/10.1016/0010-0277(93)90058-4

[11] K. Nagatsuka, C. Broni-Bediako, M. Atsumi, Pre-training a BERT with curriculum learning by increasing block-size of input text, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), INCOMA Ltd., Held Online, 2021, pp. 989–996. https://aclanthology.org/2021.ranlp-1.112/.

[12] L. Ranaldi, G. Pucci, F.M. Zanzotto, Modeling easiness for training transformers with curriculum learning, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, 2023, pp. 937–948. https://aclanthology.org/2023.ranlp-1.101/.

[13] A. Agrawal, S. Singh, L. Schneider, M. Samuels, On the role of corpus ordering in language modeling, in: Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, 2021, pp. 142–154.

[14] R. Diehl Martinez, Z. Goriely, H. McGovern, C. Davis, A. Caines, P. Buttery, L. Beinborn, CLIMB – curriculum learning for infant-inspired model building, in: A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, R. Cotterell (Eds.), Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, Association for Computational Linguistics, Singapore, 2023, pp. 112–127. https://doi.org/10.18653/v1/2023.conll-babylm.10

[15] A. Graves, M.G. Bellemare, J. Menick, R. Munos, K. Kavukcuoglu, Automated curriculum learning for neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1311–1320.

[16] Y. Wan, B. Yang, D.F. Wong, Y. Zhou, L.S. Chao, H. Zhang, B. Chen, Self-paced learning for neural machine translation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 1074–1080. https://doi.org/10.18653/v1/2020.emnlp-main.80

[17] Y. Huang, D. Xiong, IT2ACL Learning easy-to-hard instructions via 2-Phase automated curriculum learning for large language models, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 9405–9421. https://aclanthology.org/2024.lrec-main.822/.

[18] X. Tang, J. Wang, Q. Su, C.-R. Huang, J. Gu, An effective incorporating heterogeneous knowledge curriculum learning for sequence labeling, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2025, pp. 495–503.

[19] X. Chen, J. Lu, M. Kim, D. Zhang, J. Tang, A. Pich'e, N. Gontier, Y. Bengio, E. Kamalloo, Self-evolving curriculum for LLM reasoning, ArXiv abs/2505.14970 (2025). https://api.semanticscholar.org/CorpusID:278782199.

[20] J. Gu, Z. Yu, Data annealing for informal language understanding tasks, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 3153–3159. https://doi.org/10.18653/v1/2020.findings-emnlp.282

[21] X. Zhang, P. Shapiro, G. Kumar, P. McNamee, M. Carpuat, K. Duh, Curriculum learning for domain adaptation in neural machine translation, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1903–1915. https://doi.org/10.18653/v1/N19-1189

[22] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N.A. Smith, Don't stop pretraining: adapt language models to domains and tasks, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

[23] S.M. Xie, S. Santurkar, T. Ma, P.S. Liang, Data selection for language models via importance resampling, Adv. Neural Inf. Process. Syst. 36 (2023) 34201–34227.

[24] L. Ding, L. Wang, X. Liu, D.F. Wong, D. Tao, Z. Tu, Rejuvenating low-frequency words: making the most of parallel data in non-autoregressive translation, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3431–3441. https://doi.org/10.18653/v1/2021.acl-long.266

[25] L. Ding, L. Wang, S. Shi, D. Tao, Z. Tu, Redistributing low-frequency words: making the most of monolingual data in non-autoregressive translation, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2417–2426. https://doi.org/10.18653/v1/2022.acl-long.172

[26] P. Lucisano, M.E. Piemontese, Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana, Scuola e Città 3 (1988) 57–68.

[27] F. Dell'Orletta, S. Montemagni, G. Venturi, READ–IT: assessing readability of Italian texts with a view to text simplification, in: N. Alm (Ed.), Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, pp. 73–83. https://aclanthology.org/W11-2308/.

[28] A. Ettinger, What BERT is not: lessons from a new suite of psycholinguistic diagnostics for language models, Trans. Assoc. Comput. Ling. 8 (2020) 34–48. https://aclanthology.org/2020.tacl-1.3/. https://doi.org/10.1162/tacl_a_00298

[29] Y. Belinkov, Probing classifiers: promises, shortcomings, and advances, Comput. Ling. 48 (1) (2022) 207–219. https://aclanthology.org/2022.cl-1.7/. https://doi.org/10.1162/coli_a_00422

[30] M. Xu, B. Yang, D.F. Wong, L.S. Chao, Multi-view self-attention networks, Knowl. Based Syst. 241 (2022) 108268. https://www.sciencedirect.com/science/article/pii/S0950705122000855. https://doi.org/https://doi.org/10.1016/j.knosys.2022.108268

[31] A. Miaschi, D. Brunato, F. Dell'Orletta, G. Venturi, Linguistic profiling of a neural language model, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 745–756. https://doi.org/10.18653/v1/2020.coling-main.65

[32] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[33] C. Fefferman, S. Mitter, H. Narayanan, Testing the manifold hypothesis, J. Am. Math. Soc. 29 (4) (2016) 983–1049.

[34] J. Mamou, H. Le, M.D. Rio, C. Stephenson, H. Tang, Y. Kim, S. Chung, Emergence of separable manifolds in deep language representations, (2020). arXiv:2006.01095

[35] L. Dini, L. Domenichelli, D. Brunato, F. Dell'Orletta, From human reading to NLM understanding: evaluating the role of eye-tracking data in encoder-based models, in: W. Che, J. Nabende, E. Shutova, M.T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 17796–17813. https://doi.org/10.18653/v1/2025.acl-long.870

[36] G. Attardi, WikiExtractor, 2015, (https://github.com/attardi/wikiextractor).

[37] M. Straka, UDPIPE 2.0 prototype at coNLL 2018 UD shared task, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 197–207. https://doi.org/10.18653/v1/K18-2020

[38] C. Bosco, S. Montemagni, M. Simi, Converting Italian treebanks: towards an Italian stanford dependency treebank, in: A. Pareja-Lora, M. Liakata, S. Dipper (Eds.), Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 61–69. https://aclanthology.org/W13-2308/.

[39] G. Attardi, M. Simi, Overview of the EVALITA 2009 part-of-speech tagging task, in: Proceedings of the Workshop EVALITA 2009, Reggio Emilia, Italy, 2009.

[40] D. Brunato, L. De Mattei, F. Dell'Orletta, B. Iavarone, G. Venturi, Is this sentence difficult? Do you agree?, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2690–2699. https://doi.org/10.18653/v1/D18-1289

[41] F. Barbieri, V. Basile, D. Croce, M. Nissim, N. Novielli, V. Patti, et al., Overview of the evalita 2016 sentiment polarity classification task, in: CEUR Workshop Proceedings, 1749, CEUR-WS, 2016, pp. 1–11.

[42] J. Salazar, D. Liang, T.Q. Nguyen, K. Kirchhoff, Masked language model scoring, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 2699–2712. https://doi.org/10.18653/v1/2020.acl-main.240

[43] D. Brunato, A. Cimino, F. Dell'Orletta, G. Venturi, S. Montemagni, Profiling-UD: a tool for linguistic profiling of texts, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 7145–7151. https://aclanthology.org/2020.lrec-1.883/.

[44] W. Rudman, N. Gillman, T. Rayne, C. Eickhoff, ISOScore: measuring the uniformity of embedding space utilization, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.),

Findings of the Association for Computational Linguistics: ACL 2022, Associationfor Computational Linguistics, Dublin, Ireland, 2022, pp. 3325–3339. https://doi.org/10.18653/v1/2022.findings-acl.262
[45] J. Mu, P. Viswanath, All-but-the-top: simple and effective post-processing for word representations, in: 6th International Conference on Learning Representations, ICLR 2018, 2018.
[46] J. Su, J. Cao, W. Liu, Y. Ou, Whitening sentence representations for better semantics and faster retrieval, (2021). https://doi.org/10.48550/arXiv.2103.15316
[47] E. Facco, M. d'Errico, A. Rodriguez, A. Laio, Estimating the intrinsic dimension of datasets by a minimal neighborhood information, Sci. Rep. 7 (1) (2017) 12140.
[48] A. Ansuini, A. Laio, J.H. Macke, D. Zoccolan, Intrinsic dimension of data representations in deep neural networks, Adv. Neural Inf. Process. Syst. 32 (2019), pp. 6109–6119.
[49] N. Mrabah, M. Bouguessa, R. Ksantini, Escaping feature twist: a variational graph auto-Encoder for node clustering, in: IJCAI, 2022, pp. 3351–3357.
[50] A. Shaheen, N. Mrabah, R. Ksantini, A. Alqaddoumi, Rethinking deep clustering paradigms: self-supervision is all you need, Neural Netw. 181 (2025) 106773.
[51] P. Campadelli, E. Casiraghi, C. Ceruti, A. Rozza, Intrinsic dimension estimation: relevant techniques and a benchmark framework, Math. Problems Eng. 2015 (1) (2015) 759567.
[52] E. Cheng, C. Kervadec, M. Baroni, Bridging information-theoretic and geometric compression in language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 12397–12420. https://doi.org/10.18653/v1/2023.emnlp-main.762
[53] L. Albergante, J. Bac, A. Zinovyev, Estimating the effective dimension of large biological datasets using Fisher separability analysis, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.