

Model Internal Sleuthing: Finding Lexical Identity and Inflectional Morphology in Modern Language Models

Michael Li[†] Nishant Subramani[†]

[†]Carnegie Mellon University - Language Technologies Institute
 {ml6, nishant2}@cs.cmu.edu

Abstract

Large transformer-based language models dominate modern NLP, yet our understanding of how they encode linguistic information is rooted in studies of early models like BERT and GPT-2. To better understand today’s language models, we investigate how 25 models - from classical architectures (BERT, DeBERTa, GPT-2) to modern large language models (Pythia, OLMo-2, Gemma-2, Qwen2.5, Llama-3.1) - represent lexical identity and inflectional morphology across six typologically diverse languages. Using linear and non-linear classifiers trained on hidden activations, we predict word lemmas and inflectional features layer by layer. We find that models concentrate lexical information linearly in early layers and increasingly nonlinearly in later layers, while keeping inflectional information uniformly accessible and linearly separable throughout. Additional experiments probe the nature of these encodings: attention and residual analyses examine where within layers information can be recovered, steering vector experiments test what information can be functionally manipulated, and intrinsic dimensionality analyses explore how the representational structure evolves across layers. Remarkably, these encoding patterns emerge across all models we test, despite differences in architecture, size, and training regime (pretrained and instruction-tuned variants). This suggests that, even with substantial advances in LLM technologies, transformer models organize linguistic information in similar ways, indicating that these properties are important for next token prediction and are learned early during pretraining. Our code is available at https://github.com/ml5885/model_internal_sleuthing

1 Introduction

Large transformer-based language models (LMs) are widely used for tasks such as text generation, question answering, and code completion (Workshop, 2023; Groeneveld et al., 2024; Grattafiori

et al., 2024; Hui et al., 2024) However, how these models internally represent linguistic information remains an active research area. Prior work suggests a hierarchical organization where different layers specialize in capturing distinct levels of linguistic structure, from surface features to syntax and semantics (Jawahar et al., 2019; Tenney et al., 2019; Rogers et al., 2020).

But these studies focus only on first-generation LMs such as BERT and GPT-2 (Devlin et al., 2019; Radford et al., 2019). Since then, language technology has transformed dramatically - today’s models are far larger, trained on much more data, and adapted through extensive post-training procedures (Brown et al., 2020; Groeneveld et al., 2024; Lambert et al., 2025). We ask: how do modern LMs encode linguistic structure, and how do these representations differ from those in earlier models?

Consider the words *walk*, *walked*, *jump*, and *jumped*. Do language models group words by shared meaning (*walk*, *walked*) or by shared grammar (*walked*, *jumped*)? More broadly, where and how do LMs encode a word’s lemma (its lexical identity) and its grammatical form (inflectional morphology)?

To answer this, we train classifiers to predict either a word’s lemma or its inflection (*e.g.*, tense, number, degree) from hidden state activations, using annotated sentences from Universal Dependencies corpora across six languages: English, Chinese, German, French, Russian, and Turkish (Nivre et al., 2016). We evaluate classifiers on models spanning diverse architectures and sizes. We find that:

1. Lexical information is predominantly encoded in early layers, while inflectional morphology is distributed more uniformly across layers.
2. These layer-wise distributions hold across all models and languages, despite large-scale changes in architecture, amount of pretraining, and post-training procedures.

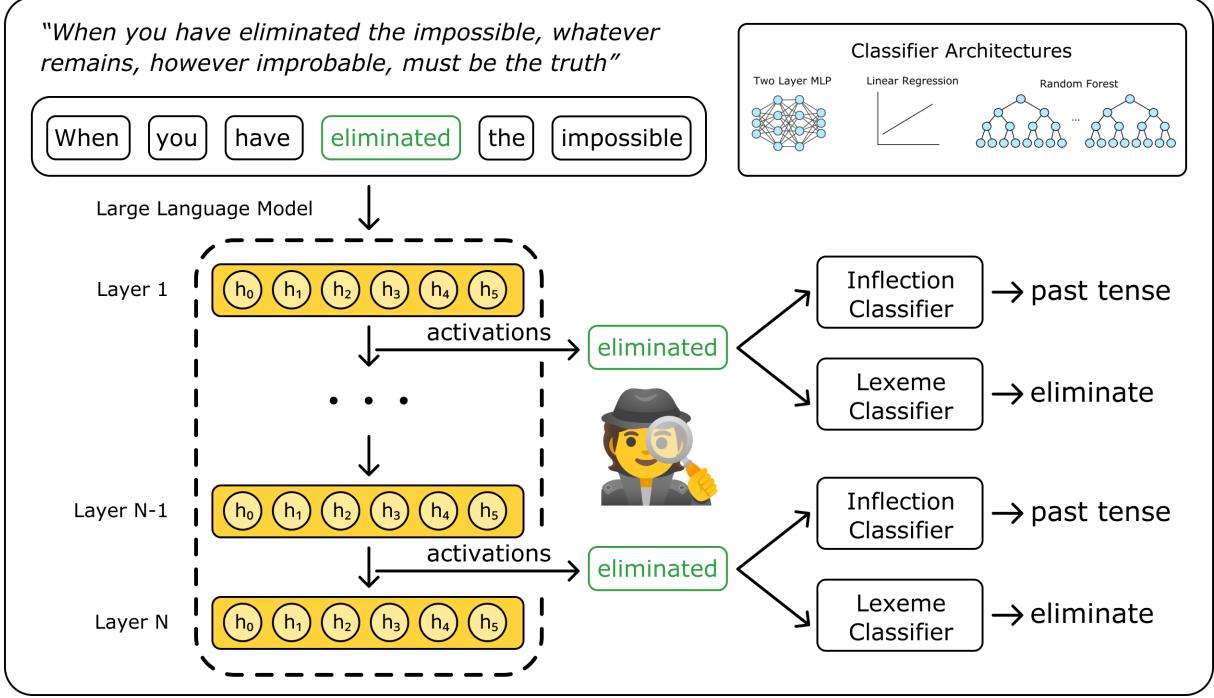


Figure 1: Overview of our classifier methodology. We extract hidden state activations from each model layer for target words and train two classifiers: one predicting inflection and another predicting lemma. We evaluate three classifier architectures (linear regression, MLP, and random forest) for each task to deduce information accessibility and linear separability across the layers of the language model. Performance differences between linear and non-linear classifiers reveal the encoding structure of linguistic information.

3. The intrinsic dimensionality of representations generally declines with layer depth, though in some models the lowest point is in the middle layers - where just one or two dimensions capture nearly all the variance.

2 Model Internal Sleuthing

To study how language models encode linguistic information, we train simple probabilistic classifiers on the models’ internal representations or activations.¹ Specifically, we use the residual stream output of each layer. By observing whether classifiers can recover word-level properties like lemma or inflection from a given representation, we can infer which layers, if any, linguistic information is present, and how directly (*e.g.*, linearly or not). High classifier performance at a given layer suggests that the relevant information is *encoded and accessible* at that layer. Additionally, comparing performance across layers tells us how lexical and inflectional information is distributed throughout the model.

Figure 1 illustrates our approach. For each layer

of each model, we extract hidden state activations for a target word and use them as input features to train specialized classifiers that predict either inflection or lemma. In our experiments, we train three different classifiers: linear regression, two-layer multi-layer perceptron (MLP), and random forest.

2.1 Linear Regression Classifier

Consistent with best practices for probing (Hewitt and Liang, 2019; Liu et al., 2019), we use simple linear regression classifiers. These models are solved via closed-form ridge regression (Hastie et al., 2009), using the equation:

$$W = (X_{\text{train}}^T X_{\text{train}} + \lambda I)^{-1} X_{\text{train}}^T Y_{\text{train}} \quad (1)$$

where $X_{\text{train}} \in \mathbb{R}^{m \times d}$ contains d -dimensional representations for each of the m examples in the training set, $Y_{\text{train}} \in \mathbb{R}^{m \times c}$ is the one-hot encoded matrix of training labels for c classes, λ controls the strength of L2 regularization, and $W \in \mathbb{R}^{d \times c}$ is the learned weight matrix of the classifier. Test predictions are given by:

$$\hat{Y}_{\text{test}} = X_{\text{test}} W \quad (2)$$

¹We use representations, activations, and neurons interchangeably.

2.2 Random Forest Classifier

Random forests provide a complementary approach to linear models by handling complex feature interactions and decision boundaries. Unlike linear models that assume information is linearly separable, random forests combine an often large number of decision trees via model averaging to robustly identify patterns (Breiman, 2001).

Since our classification tasks involve a large number of classes (thousands of unique lemmas and multiple inflection categories), we use a one-vs-all approach. For each class j , we train a binary random forest classifier that distinguishes class j from all other classes. Each binary classifier outputs a probability score $s_j(x) \in [0, 1]$ representing the confidence that input x belongs to class j . The final prediction is assigned based on the maximum score across all classes:

$$\hat{y} = \operatorname{argmax}_{j \in \{1, 2, \dots, c\}} s_j(x) \quad (3)$$

where $s_j(x)$ is the probability that the trained binary classifier assigns to class j for the input x .²

2.3 MLP Classifier

To test for more general non-linearity, we train a simple two-layer MLP with ReLU activation, defined as:

$$\hat{Y} = \operatorname{softmax}(\operatorname{ReLU}(X_{\text{train}} W_1) W_2) \quad (4)$$

where $W_1 \in \mathbb{R}^{d \times h}$ and $W_2 \in \mathbb{R}^{h \times c}$ are the weight matrices and $h = 64$ is the hidden layer size (Rosenblatt, 1958).³ Two-layer MLPs with ReLU activation are universal function approximators, capable of approximating any continuous function to arbitrary precision given sufficient width (Hornik et al., 1989).

Comparing performance across these three classifier types provides insights into how information is encoded: if linear classifiers perform comparably to more complex models, information is likely linearly encoded; if random forests or MLPs significantly outperform linear models, information is non-linearly encoded (Belinkov and Glass, 2019).

2.4 Training and Evaluation

We stratify all datasets into train, validation and test splits. We train classifiers using the training

²We use scikit-learn’s implementation (Pedregosa et al., 2011) with the `predict_proba()` method to obtain probability scores.

³Bias terms are omitted for brevity.

Model	Parameters	Pretraining Data	Layers
Encoder-only			
BERT Base	110M	12.6B tokens ¹	12
BERT Large	340M	12.6B tokens ¹	24
DeBERTa V3 Large	418M	32B tokens ¹	24
Decoder-only			
GPT 2 Small	124M	8B tokens ¹	12
GPT 2 Large	708M	8B tokens ¹	36
GPT 2 XL	1.5B	8B tokens ¹	48
Goldfish English 1000mb	124M	200M tokens	12
Goldfish Chinese 1000mb	124M	200M tokens	12
Goldfish German 1000mb	124M	200M tokens	12
Goldfish French 1000mb	124M	200M tokens	12
Goldfish Russian 1000mb	124M	200M tokens	12
Goldfish Turkish 1000mb	124M	200M tokens	12
Pythia 6.9b	6900M	300B tokens	32
Pythia 6.9b Tulu	6900M	300B tokens	32
OLMo 2 7B	7300M	4T tokens	32
OLMo 2 7B Instruct	7300M	4T tokens	32
Gemma 2 2B	2610M	2T tokens	26
Gemma 2 2B Instruct	2610M	2T tokens	26
Qwen2.5 1.5B	1540M	18T tokens	28
Qwen2.5 1.5B Instruct	1540M	18T tokens	28
Qwen2.5 7B	7620M	18T tokens	28
Qwen2.5 7B Instruct	7620M	18T tokens	28
Llama 3.1 8B	8000M	15T tokens	32
Llama 3.1 8B Instruct	8000M	15T tokens	32
Encoder-Decoder			
mT5-base	580M	1T tokens	12

Table 1: Overview of models used in this study.

¹Converted from GB to tokens using the approximation that 1GB of data is approximately 200M tokens in English (Chang et al., 2024).

set, select hyperparameters using the validation set, and evaluate on the held-out test set using accuracy and macro F1. For linear regression, we use ridge regularization with $\lambda = 0.01$. For random forests, we tune the number of trees (50, 100, 200) and maximum depth (5, 10, 20, None) using grid search. For MLPs, we use a hidden layer size of 64, learning rate of 0.001, weight decay of 0.01, and train for up to 100 epochs with early stopping based on validation loss. We solve equation (1) in closed-form to identify $W \in \mathbb{R}^{d \times c}$, and learn W_1 and W_2 for the MLP via stochastic gradient descent using the AdamW optimizer on cross-entropy loss (Kingma and Ba, 2014).⁴

Linear Separability Gap To quantify whether a layer’s information is linearly encoded or not, we define the *linear separability gap* as the difference in accuracy between non-linear and linear models. A larger gap indicates the signal is buried in non-linear structure, while a small gap suggests it is directly accessible. In practice, this gap is typically between -0.3 and 0.3, though theoretically it can range from -1 to 1.

⁴We use weight decay for regularization for MLP models.

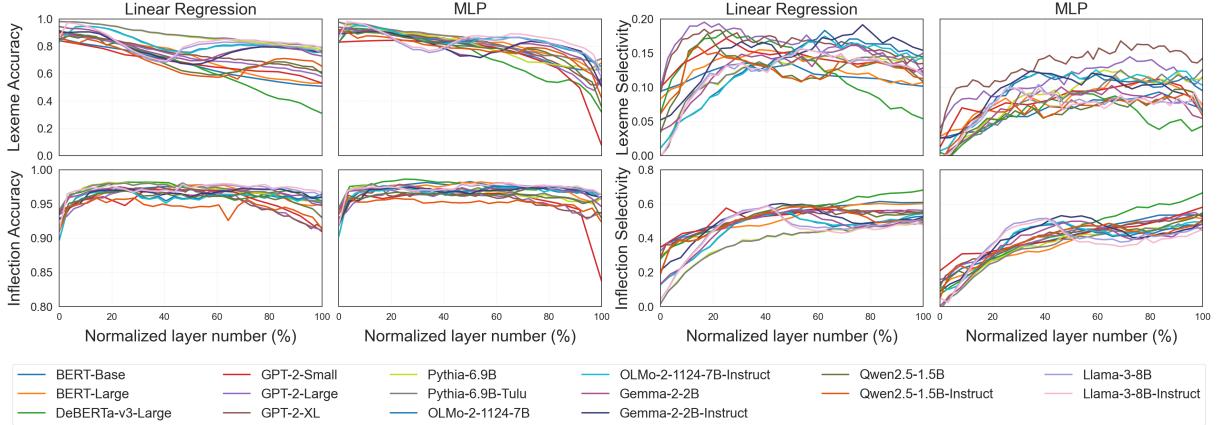


Figure 2: Linguistic accuracy and classifier selectivity across model layers for English. The first two columns show lexeme (top) and inflection (bottom) prediction accuracy using Linear Regression (left) and MLP (right) classifiers. The next two columns show classifier selectivity (difference between linguistic and control task accuracy) for the same tasks and classifiers. Higher selectivity indicates better generalization rather than memorization. Each line represents a different model. Multilingual results are shown in Figure 3.

3 Experiments

Using the classifier methodology introduced in §2, we describe the components of our experimental setup: the datasets, model suite, and procedure for extracting token-level representations.

3.1 Datasets

We use Universal Dependencies (UD) corpora across six typologically diverse languages: English, Chinese, German, French, Russian, and Turkish (Nivre et al., 2016). For each language, we select the following UD treebanks: Georgetown University Multilayer (GUM) Corpus for English (Zeldes, 2017), Google Stanford Dependencies (GSD) for Chinese, German, and French (McDonald et al., 2013; Guillaume et al., 2019), SynTagRus for Russian (Drogaanova et al., 2018), and IMST for Turkish (Sulubacak et al., 2016).

These corpora provide rich morphological annotations across various genres and linguistic phenomena. Each dataset contains sentences with target words annotated for their lemma (base form) and inflectional features (grammatical categories such as tense, number, case, etc.).

Formally, a data point is a tuple (S, i, ℓ, c) where $S = \{w_0, w_1, \dots, w_n\}$ is a sentence, i is the index of the target word w_i within the sentence, ℓ is the lemma of w_i , and c is the inflection category of w_i .

The languages represent diverse morphological typologies, which allows us to test whether our findings generalize across different linguistic structures.

For each language, we split the dataset into training, validation, and test sets with a 70-10-20 ratio using stratified sampling based on inflection labels to ensure balanced representation across all grammatical categories.⁵

3.2 Models

We study a diverse set of pretrained transformer language models spanning different architectures, sizes, and training regimes. Table 1 lists all models used in this study, see Table 6 for the HuggingFace identifiers.

For English, we evaluate all models listed in Table 1 (excluding the non-English Goldfish models). For the five non-English languages (Chinese, German, French, Russian, Turkish), we focus on models that have explicit multilingual training: the Goldfish monolingual models trained specifically for each target language (Chang et al., 2024), multilingual Qwen2.5 variants that include these languages in their training data, and the multilingual mT5-base model (Xue et al., 2021). This ensures that we evaluate models on languages they were trained on while maintaining sufficient coverage.

The Goldfish models are particularly valuable for cross-linguistic comparison as they use identical architectures (124M parameters, 12 layers) and training procedures while being trained monolingually on the same amount of data (200M tokens) for each language.

⁵See Appendix B for complete details including dataset statistics, tokenization information, and visualizations for all languages

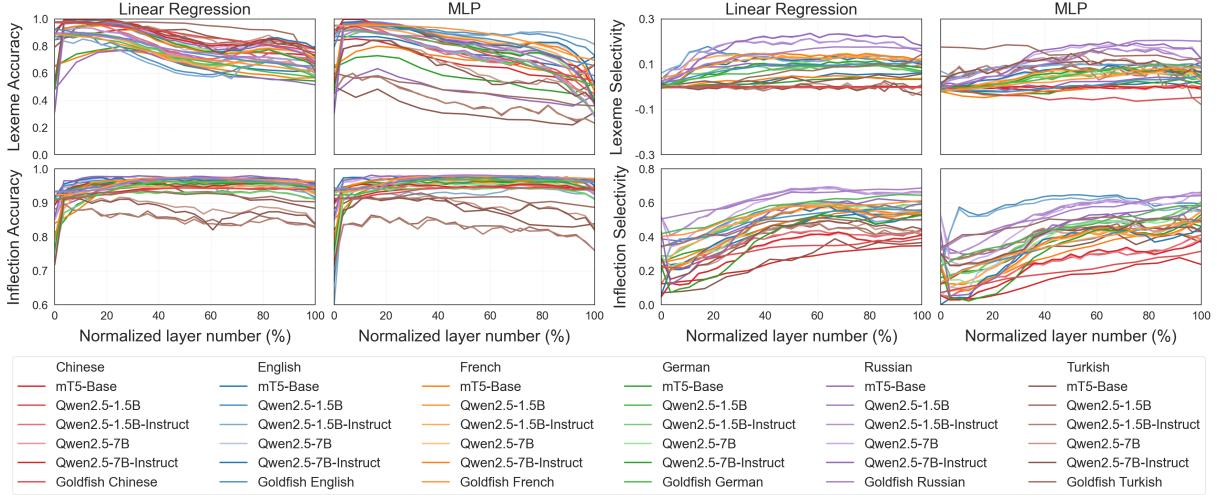


Figure 3: Cross-linguistic patterns in linguistic accuracy and classifier selectivity. The first two columns show lexeme (top) and inflection (bottom) prediction accuracy using Linear Regression (left) and MLP (right) classifiers. The third column shows inflection prediction with Random Forest classifiers (lemma prediction is computationally infeasible due to the large number of classes). The rightmost columns show classifier selectivity for lexeme (top) and inflection (bottom) tasks. Each line represents a different model-language combination.

3.3 Representation Extraction

For each data point (S, i, ℓ, c) , we tokenize the sentence S using model-specific tokenizers and process it through the model to obtain hidden states from all layers. Specifically, we use representations from the residual stream, which form the basis of our layer-wise analysis of how linguistic information is encoded throughout the model across different languages.⁶

4 Results

We present our classifier results for both lexical identity and inflectional morphology across different classifier architectures and languages. Our analysis encompasses 19 models for English, and extends to five additional languages using 6 multilingual and monolingual models to examine cross-linguistic patterns.

4.1 Lexical Identity

Across all architectures and languages, lexical information is most accessible in early layers and becomes progressively harder to extract in deeper layers.

Linear Regression Classifiers Linear regression experiments show lemma prediction performance consistently declining from high accuracy (0.8–1.0) in initial layers toward final layers (Figures 2

⁶For target words split into multiple subword tokens, we use the representation of the last subword token to represent the entire word (Devlin et al., 2019).

and 3, top left). The decline varies across models: DeBERTa-v3-Large shows the steepest drop from 0.9 to 0.3, while Pythia-6.9B maintains relatively strong performance in deeper layers.

Cross-linguistic analysis reveals variation in this pattern. Turkish models demonstrate the most dramatic decline, falling from 0.95 to 0.25, while Russian models maintain higher accuracy throughout, declining only to 0.6–0.8. Chinese models show intermediate performance, consistent with the language’s analytic structure and limited morphological complexity.

MLP Classifiers Two-layer MLPs achieve higher overall accuracies and show less severe decline compared to linear regression (Figures 2 and 3, top middle). The Pythia family maintains the strongest performance across all layers, while GPT-2, BERT, and Qwen2.5 variants show the largest improvements from linear to MLP classifiers.

MLPs provide improvements across all languages, with Turkish showing the greatest benefit from non-linear classification. This indicates that lexical information in morphologically complex languages becomes increasingly encoded in non-linear ways in deeper layers.

Random Forest Classifiers Random forest classifiers for lemma prediction were not evaluated due to computational constraints arising from thousands of unique lemma classes.

4.2 Inflectional Morphology

Inflectional morphology shows markedly different encoding patterns. Morphological information remains stable throughout the network while lexical information shows layer-wise variation.

Linear Regression Classifiers Inflectional morphology prediction maintains consistently high performance (0.9-1.0) across all layers and models (Figures 2 and 3, bottom left). Accuracy curves show minimal degradation from early to late layers. Inflectional morphology classification exhibits consistent performance across architectures, unlike lexical prediction where models show substantial variation.

This consistency extends across multilingual evaluation. English, German, French, and Russian achieve high accuracy (0.90-1.0) across all classifier types. Turkish consistently underperforms relative to other languages (0.85-0.92 accuracy), reflecting the challenges posed by agglutinative morphological systems where inflectional features combine through complex suffix patterns.

Analysis of specific morphological categories shows that comparative and superlative adjective forms are most challenging across all tested languages. Detailed error analysis for English appears in Appendix F.

MLP Classifiers MLPs show nearly identical patterns to linear regression for inflectional morphology, maintaining accuracy between 0.9-1.0 across all layers (Figures 2 and 3, bottom middle). The lack of improvement from additional non-linear capacity indicates that inflectional morphology is encoded linearly throughout transformer networks.

Random Forest Classifiers Random forest classifiers for inflectional morphology achieve lower overall accuracy (0.65-0.90) with a declining trend across layers. Russian and German perform best while Turkish remains most challenging. Complete results appear in Appendix E.

4.3 Classifier Error Analysis

To further understand these results, we conduct a comprehensive error analysis. Examining classifier performance across different lexeme categories and inflection types shows a frequency effect: linguistic forms that appear more often in training data achieve higher classification accuracy. Detailed breakdowns by category (available in Ap-

pendix F) show that this frequency dependence applies to both lexical and morphological classification, with the effect more pronounced for lexical identity where rare words consistently prove most challenging to classify.

5 Analysis

Our experiments demonstrate clear patterns in how transformer models encode linguistic information. We organize our analysis around five key aspects: cross-linguistic encoding patterns, generalization versus memorization, representation dimensionality, attention mechanisms, and model-level findings.⁷

Cross-Linguistic Encoding Patterns We observe consistent and distinct encoding behaviors for lexical and inflectional information across all six languages. Lexical information is most accessible in early layers (0-2) and then declines, while inflectional information remains uniformly high across all layers (Figures 2 and 3). This separation holds for morphologically complex languages like Turkish and Russian, extending the hierarchical view of transformer representations as a general property rather than an English-specific artifact.

The magnitude varies significantly across languages. Turkish shows the steepest lexical accuracy decline (0.95 -> 0.25) and consistently lower performance, reflecting challenges from its agglutinative morphological system, consistent with Ismayilzada et al. (2025) who found LLMs struggle with morphological compositional generalization in Turkish and Finnish. Russian maintains higher lexical accuracy while achieving the highest inflectional selectivity, while Chinese shows intermediate performance across both.

Generalization vs. Memorization Different types of linguistic information require different levels of classifier complexity and show distinct patterns of generalization versus memorization. We quantify classifier complexity using the linear separability gap from §2 and assess generalization through control tasks following Hewitt and Liang (2019), where words are randomly but consistently mapped to arbitrary labels.

⁷Unless otherwise noted, our analyses focus on English. This is because most models we evaluate are either trained exclusively on English or do not offer full multilingual support. Only a few models we use - Qwen2.5, mT5, and the monolingual Goldfish variants - support non-English inputs.

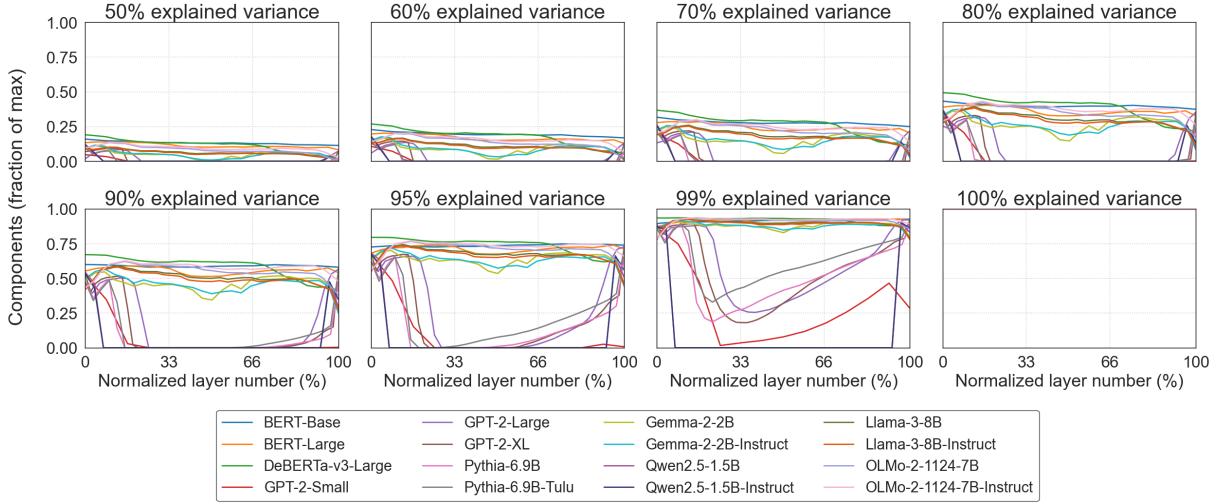


Figure 4: Fraction of maximum components needed to achieve different explained variance thresholds across layers. Each subplot shows a different variance threshold (50%-100%), with lines colored by model. Note that 100% explained variance always requires the full component set. Models fall into two groups: in one, dimensionality declines gradually; in the other, the middle layers require only one component. See additional plots in Figure 17.

Cross-linguistic analysis shows consistent model-family patterns in the linear separability gap (Figure 16). For inflectional features, mT5 and Goldfish models show slight positive gaps (0.01–0.05) across most languages, indicating modest benefits from non-linear classification. Qwen2.5 variants show slight negative to near-zero gaps, suggesting that linear classifiers are sufficient or even superior for inflectional morphology in these models. For lexical features, all model families exhibit negative gaps across all languages, with the effect most pronounced in early layers and gradually diminishing toward later layers. The negative gaps range from -0.2 to -0.5, suggesting that linear regression consistently outperforms MLPs for lexical classification.

Control task analysis demonstrates a clear distinction between these information types. Inflection prediction shows strong *classifier selectivity* (0.4–0.6) across most languages, with Russian and German achieving the highest values due to their systematic inflectional patterns. This high selectivity indicates that models learn generalizable morphological rules rather than memorizing arbitrary mappings. Lemma prediction universally shows near-zero selectivity across all languages, indicating that lexical retrieval relies entirely on memorization rather than systematic patterns.⁸

This distinction aligns with the notion that mor-

phological rules are systematic and generalizable (*e.g.*, adding *-ed* to form past tense) while lexical identity is item-specific. It also connects to previous work on the increasing contextualization of representations in later layers of transformer models (Ethayarajh, 2019). Subramani et al. (2022) similarly found that steering vectors become less effective in later layers and that representations reflect semantic similarity less in later layers.

Representation Dimensionality To better understand how representations are structured across layers, we measure intrinsic dimensionality using PCA (Table 2 and Figure 4) (Ansuini et al., 2019). Lower intrinsic dimensions indicate that most variance in the representations can be captured by fewer principal components, suggesting the information is compressed into specific directions in latent space.

Models cluster into two groups: some show gradual dimensionality decline (BERT, DeBERTa, Gemma, Llama 3.1), while others have middle layers where a single component explains most variance (GPT-2, Qwen2.5, Pythia, OLMo-2).

These compressed middle layers align with maximal inflectional separability and minimal lexical separability. Later layers expand dimensionality, for generation, but retain linearly accessible inflectional information.

Low intrinsic dimensionality has practical implications: steering vectors can more effectively manipulate model behavior by targeting dominant directions. Compressed middle-layer representa-

⁸This is perhaps unsurprising given the nature of the lemma task: it involves thousands of unique classes, many of which appear infrequently in the training data.

Model	d_{model}	ID ₅₀			ID ₇₀			ID ₉₀		
		First	Mid	Final	First	Mid	Final	First	Mid	Final
BERT-Base	768	123	100	88	244	212	192	461	451	446
BERT-Large	1024	138	105	85	286	226	208	567	527	554
DeBERTa-v3-Large	1024	196	133	29	377	299	113	688	635	423
GPT-2-Small	768	37	1	1	152	1	1	402	1	3
GPT-2-Large	1280	24	1	95	172	1	284	583	1	726
GPT-2-XL	1600	113	1	118	340	1	356	838	1	914
Pythia-6.9B	4096	391	1	96	865	1	517	1952	1	1925
Pythia-6.9B-Tulu	4096	390	1	244	862	1	832	1949	1	2292
OLMo-2-7B	4096	404	310	41	833	896	299	1772	2279	1550
OLMo-2-7B-Instruct	4096	404	358	111	833	974	567	1772	2361	1964
Gemma-2-2B	2304	216	8	11	505	130	70	1129	794	611
Gemma-2-2B-Instruct	2304	222	22	8	520	198	57	1153	899	572
Qwen-2.5-1.5B	1536	184	1	9	399	1	50	835	1	452
Qwen-2.5-1.5B-Instruct	1536	184	1	11	394	1	70	820	1	533
Llama-3.1-8B	4096	373	240	35	789	727	187	1722	2051	1119
Llama-3.1-8B-Instruct	4096	372	215	31	788	664	181	1722	1957	1093

Table 2: Number of principal-component axes required to reach 50% (ID₅₀), 70% (ID₇₀) and 90% (ID₉₀) explained variance in the first, middle and last layers of each model.

tions support both efficient computation and effective intervention.

Attention Head Analysis To probe how attention mechanisms contribute to linguistic encoding, we analyzed attention head outputs alongside residual stream representations, using averaged activations across attention heads at each layer.

We find that lexical information shows a pronounced drop in attention outputs during middle layers before partially recovering, while the residual stream maintains higher accessibility. For inflectional morphology, both attention outputs and the residual stream maintain high accuracy, though residual streams consistently outperform attention outputs in middle layers.

Notably, we observe that attention heads encode inflectional information more strongly than lexical information, consistent with findings that attention heads primarily encode contextual knowledge while MLPs store parametric knowledge (Subramani et al., 2022). See Appendix C for complete results.

Training Dynamics Despite differences in architecture and training data, all models show consistent behaviors, suggesting these are properties of transformer architectures rather than training artifacts. To test this theory, we examined how these structures develop during pre-training by training classifiers at intermediate model checkpoints for

both OLMo-2 and Pythia (Figures 18 and 19 in Appendix F). Lexeme classification reaches peak accuracy early in training before slowly declining, while inflectional accuracy remains high and stable across both checkpoints and layer depth.

This result indicates that models establish the separation between lexical and inflectional information early and maintain this structure throughout training. Taken together with our model architecture findings, we find that separating lexical and inflectional information facilitates effective next-token prediction.

Steering Experiments We conducted steering vector experiments to validate the functional importance of these representations. Results show mean changes of 0.9-1.0 in classifier predictions when applying inflectional steering vectors. However, sensitivity varies: some models maintain robust effectiveness across layers while others show pronounced dips around 10% layer depth. This variation is explained by our intrinsic dimensionality findings - specifically, the Qwen2.5 models, which show the most dramatic steering effectiveness drops (from 1.0 to 0.6-0.7), are precisely those with extreme middle-layer compression requiring only 1 component for 50-99% variance. Complete experiment details and results appear in Appendix D.

6 Related Work

Probing for linguistic information Probing studies typically use supervised classifiers to predict linguistic properties from model representations (Alain and Bengio, 2017; Adi et al., 2017). Extensive work has established that early transformer models (BERT, GPT-2) learn hierarchical linguistic structures, with different layers specializing in different information types: lower layers capture surface features and morphology, middle layers encode syntax, and upper layers represent semantics and context (Jawahar et al., 2019; Tenney et al., 2019; Rogers et al., 2020). More relevant to our work, Vulić et al. (2020) found that lexical information concentrates in lower layers, while Ethayarajh (2019) showed that representations become increasingly context-specific in higher layers.

Activation steering Beyond probing, recent work has explored manipulating model behavior by intervening on internal representations. This includes steering vectors (Subramani et al., 2022), inference-time interventions (Li et al., 2023), representation editing (Meng et al., 2022), sparse autoencoders for feature discovery (Bricken et al., 2023), and causal mediation analysis (Vig et al., 2020). While these methods typically evaluate changes in model outputs, our steering experiments focus on measuring representational changes. See Appendix A for more detailed discussion.

7 Conclusion

In this paper, we examine how 25 transformer models represent lexical identity and inflectional morphology. We train classifiers on layer-wise activations to predict word lemmas and inflectional features, using control tasks across six typologically diverse languages to distinguish generalization from memorization.

We find that lexical information is primarily encoded in earlier layers and becomes progressively non-linear deeper in the network. In contrast, inflectional information remains linearly accessible across all layers. These patterns are consistent across architecture, model size, and training procedures, and cross-linguistic evaluation confirms that they represent general properties of transformer models rather than language-specific artifacts. This finding extends previous observations about layer-wise linguistic specialization in early transformer models like BERT.

Our selectivity analyses indicate that models store lexical relationships through memorization, whereas they learn more systematic, context-independent patterns for inflectional morphology. This trend holds consistently across languages, though Turkish, with its agglutinative morphology, poses particular challenges, and Russian yields the highest accuracy.

Further investigation into attention heads shows stronger encoding of inflectional information compared to lexical information. Additionally, steering vector experiments demonstrate that it is possible to manipulate inflectional representations effectively, with varying success depending on layer position and intrinsic dimensionality. Analysis of pretraining checkpoints reveals that these representational patterns emerge early and remain stable, suggesting that they are likely structural properties rather than artifacts of extensive training.

These observations are robust across a wide range of architectures, model sizes - from 124M to 8B parameters - and training corpus sizes from 200M to 18T tokens. This consistency implies that this organization of linguistic information is essential for both next token prediction during pre-training and complex instruction-following. Our work highlights that certain aspects of how these models process language remain remarkably consistent, even as model architectures and training methods rapidly advance.

8 Limitations

Representation Extraction for Decoder Models Our current approach for extracting word representations from decoder-only models uses the final subword token. This assumption is an intuitive and natural choice, but may not be optimal for all architectures and models. Future work could develop better extraction methods that account for subword tokenization effects and leverage attention patterns to create more accurate word-level representations.

Form and Function in Inflection Some languages contain cases where different grammatical functions share the same surface form (e.g., infinitive and non-past verb forms in English). We do not explicitly examine these cases in our classification experiments, but these ambiguities create opportunities to better examine how models separate form from function across languages.

Indirect Nature of Classifiers While our classifier methodology follows established best practices (Hewitt and Liang, 2019; Liu et al., 2019), we only detect correlations in hidden activations, not causal mechanisms.

Scope of Steering Experiments Our steering vector experiments measure changes in classifier performance rather than downstream model outputs. Evaluating effects on actual model generation would require more complex experimental designs to control for confounding factors and ensure that observed changes result from the intended representational modifications rather than other influences.

References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. *Fine-grained analysis of sentence embeddings using auxiliary prediction tasks*. In *5th International Conference on Learning Representations (Conference Track)*.
- Guillaume Alain and Yoshua Bengio. 2017. *Understanding intermediate layers using linear classifier probes*. In *5th International Conference on Learning Representations (Workshop Track)*.
- Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. 2019. *Intrinsic dimension of data representations in deep neural networks*. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yonatan Belinkov and James Glass. 2019. *Analysis methods in neural language processing: A survey*. *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Leo Breiman. 2001. *Random forests*. *Mach. Learn.*, 45(1):5–32.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosematicity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. [Https://transformer-circuits.pub/2023/monosemantic-features/index.html](https://transformer-circuits.pub/2023/monosemantic-features/index.html).
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. *Language models are few-shot learners*. *Preprint*, arXiv:2005.14165.
- Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2024. *Goldfish: Monolingual language models for 350 languages*. *Preprint*, arXiv:2408.10441.
- Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. *Sparse autoencoders find highly interpretable features in language models*. *Preprint*, arXiv:2309.08600.
- Thao Anh Dang, Limor Raviv, and Lukas Galke. 2024. *Tokenization and morphology in multilingual language models: A comparative analysis of mt5 and byt5*. *Preprint*, arXiv:2410.11627.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian ud treebanks. In *Proceedings of the 17th international workshop on treebanks and linguistic theories (tlt 2018)*, volume 155, pages 53–66. Linköping University Electronic Press Linköping, Sweden.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. *Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals*. *Transactions of the Association for Computational Linguistics*, 9:160–175.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. [Https://transformer-circuits.pub/2021/framework/index.html](https://transformer-circuits.pub/2021/framework/index.html).
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

- Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. 2021. **Causal abstractions of neural networks**. In *Advances in Neural Information Processing Systems*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. **The llama 3 herd of models**. *Preprint*, arXiv:2407.21783.
- Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Author, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. **OLMo: Accelerating the science of language models**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.
- Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. 2019. **Conversion et améliorations de corpus du français annotés en Universal Dependencies [conversion and improvement of Universal Dependencies French corpora]**. *Traitement Automatique des Langues*, 60(2):71–95.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2 edition. Springer, New York, NY, USA.
- John Hewitt and Percy Liang. 2019. **Designing and interpreting probes with control tasks**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. **Multilayer feedforward networks are universal approximators**. *Neural Networks*, 2(5):359–366.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. **Editing models with task arithmetic**. In *The Eleventh International Conference on Learning Representations*.
- Mete Ismayilzada, Defne Ciri, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuvan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke Van Der Plas. 2025. **Evaluating morphological compositional generalization in large language models**. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1270–1305, Albuquerque, New Mexico. Association for Computational Linguistics.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. **What does BERT learn about the structure of language?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. **Tulu 3: Pushing frontiers in open language model post-training**. *Preprint*, arXiv:2411.15124.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. **Inference-time intervention: Eliciting truthful answers from a language model**. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. **Linguistic knowledge and transferability of contextual representations**. In *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)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ryan McDonald, Joakim Nivre, Yvonne Quirkbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. **Universal Dependency annotation for multilingual parsing**. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.
- Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. **Locating and editing factual associations in GPT**. In *Advances in Neural Information Processing Systems*.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic̆, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman.

2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).
- nostalgebraist. 2020. interpreting gpt: the logit lens.
- Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. *Preprint*, arXiv:2312.06681.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Nishant Subramani, Jason Eisner, Justin Svegliato, Benjamin Van Durme, Yu Su, and Sam Thomson. 2025. MICE for CATs: Model-internal confidence estimation for calibrating agents with tools. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12362–12375, Albuquerque, New Mexico. Association for Computational Linguistics.
- Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.
- Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan. The COLING 2016 Organizing Committee.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.
- Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.
- Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.
- BigScience Workshop. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

A Additional Related Work

A.1 Advanced Probing Methodologies

Beyond standard linear probes, there are many sophisticated approaches to understand model representations. Amnesic probing (Elazar et al., 2021) removes specific information from representations to test whether it’s necessary for downstream tasks. Minimum description length probes (Voita and Titov, 2020) balance probe complexity with performance to avoid overfitting. Causal probing (Geiger et al., 2021) aims to establish causal rather than merely correlational relationships between representations and linguistic properties. Recently, Subramani et al. (2025) find that decoding from activations directly using the Logit Lens can be used to learn confidence estimators for tool-calling agents (nostalgebraist, 2020).

A.2 Model Manipulation and Steering

Steering vectors demonstrate that specific directions in activation space correspond to high-level behavioral changes (Subramani et al., 2022). Building on this, Panickssery et al. (2024) achieves behavioral control by adding activation differences between contrasting examples. Li et al. (2023) introduce inference-time intervention, a method that shifts model activations during inference across limited attention heads to control model behavior. While these methods operate in activation space, task vectors enable arithmetic operations on model capabilities by manipulating weight space (Ilharco et al., 2023).

Recent work has also examined how multilingual models like mT5 and ByT5 encode morphological information differently across languages (Dang et al., 2024), finding that tokenization strategies significantly impact morphological representation quality, particularly for morphologically rich languages.

A.3 Feature Discovery and Mechanistic Interpretability

Recent work has explored using sparse autoencoders to discover latent features (Cunningham et al., 2023; Bricken et al., 2023), providing clearer targets for steering and interpretation. Mechanistic interpretability approaches aim to reverse-engineer the algorithms learned by neural networks (Elhage et al., 2021). Representation editing directly modifies model weights to alter specific factual associations (Meng et al., 2022). These methods comple-

ment classification-based approaches by identifying the underlying structure of learned representations and their functional significance.

B Dataset Statistics

This section provides statistics and visualizations for the datasets and models used in our experiments across all six languages. Only words containing alphabetic characters and apostrophes were considered.

B.1 English Dataset Details

For the English GUM corpus specifically, the data covers three main syntactic categories: nouns (49.5%), verbs (31.2%), and adjectives (19.4

Figure 5a shows the distribution of word categories in the English dataset, and Figure 5b presents the distribution of inflection categories.

B.2 Tokenization Statistics

An important consideration for our analysis is how different models tokenize the words in our dataset. Table 5 shows tokenization statistics across the models we analyze. Encoder-only models like BERT and DeBERTa tend to split words into more tokens than decoder-only models like GPT-2 and Qwen2, which may affect how information is encoded across layers.

B.3 Effects of Tokenization

Tokenization is an essential component of language modeling. To test how tokenization influences our findings, we use analogy completion tasks in English (*e.g.*, *man:king::woman:?*) and compare two approaches: averaging subtoken embeddings after standard tokenization and summing embeddings from whole-word tokens.

For each approach, we perform vector arithmetic on word representations (*e.g.*, *king - man + woman*). We measure performance by ranking all vocabulary words by cosine similarity to the resulting representation, and observe how highly the expected word (*e.g.*, *queen*) ranks, with a lower rank indicating better performance.

Whole-word representations markedly outperform averaged subtokens across all models (Figure 6), implying that linguistic regularities are primarily stored in whole-word embeddings rather than compositionally across subtokens. Despite tokenization effects, our classifier results show consistent patterns across models using different tok-

Language	Total Words	Unique Lemmas	Unique Forms	Inflection Types	Sentences	Avg. Length
English	54,816	7,848	11,720	8	8,415	6.5
Chinese	44,166	11,184	11,237	4	7,892	5.8
German	84,710	24,140	31,890	9	9,234	7.3
French	115,847	13,804	24,485	6	8,765	6.6
Russian	193,320	20,943	59,830	8	10,234	7.1
Turkish	20,881	3,776	11,680	7	6,789	6.4

Table 3: Dataset statistics across all six languages. Russian has the largest dataset and the highest number of unique forms, reflecting its rich inflectional morphology. Turkish has the fewest total words and lemmas, while Chinese has the fewest inflection types.

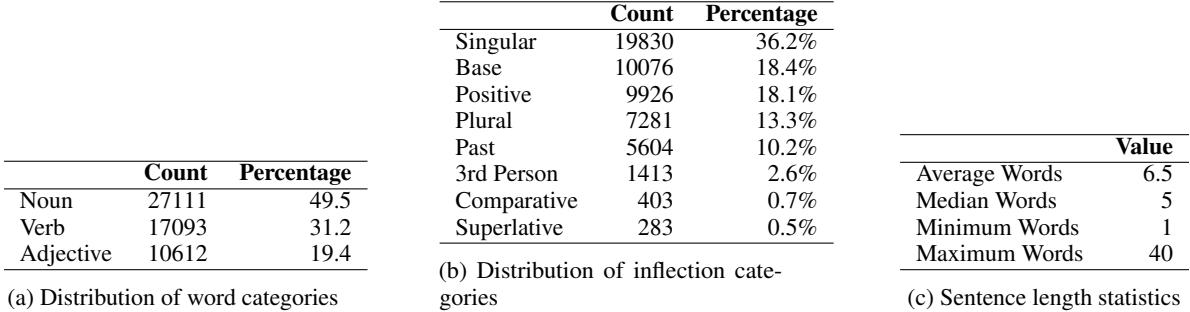


Figure 5: Distribution statistics for the English dataset

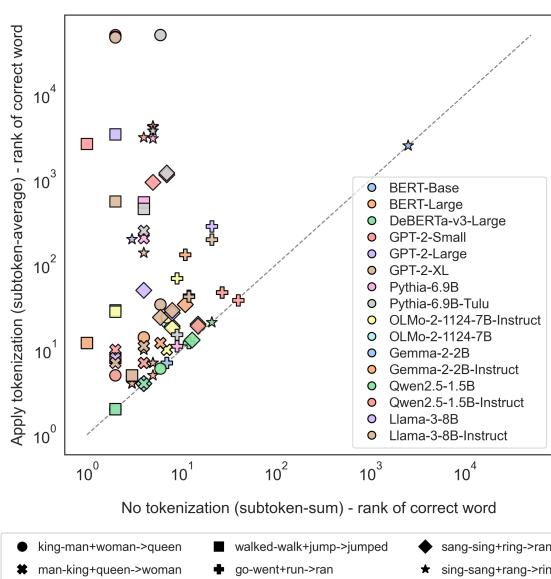


Figure 6: Effect of tokenization strategy on analogy completion rank. Each point corresponds to a model (color) and analogy (shape). The x-axis is the rank using whole-word representations. The y-axis is the rank using tokenized representations. Here, rank means the position of the expected word when all vocabulary words are sorted by similarity to the resulting embedding from vector arithmetic; lower is better. Points above the gray $y=x$ line mean tokenization hurts performance.

Model	Tokenizer Type
BERT Base/Large	WordPiece
DeBERTa V3 Large	SentencePiece
GPT-2 variants	BPE
Pythia variants	BPE
OLMo 2 variants	BPE (tiktoken)
Gemma 2 variants	SentencePiece
Qwen 2.5 variants	Byte-level BPE
Llama 3.1 variants	BPE (tiktoken)

Table 4: Tokenization strategies used by different model families. BPE means byte-pair encoding.

enizers (see Table 4), indicating robust encoding of lexical and morphological information.

C Attention Head Analysis

We conducted additional experiments analyzing attention head outputs alongside residual stream representations to understand how different components of transformer models contribute to linguistic encoding.

C.1 Methodology

We averaged activations across all attention heads at each layer for Qwen2.5-1.5B and Qwen2.5-1.5B-Instruct models using the English dataset. We then trained linear regression,

Model	Avg. tokens per word	Med. tokens per word	Max tokens per word	Percent multitoken
BERT variants	1.11	1.0	6.0	6.95
DeBERTa-v3-large	1.03	1.0	4.0	2.2
GPT-2 variants	1.52	1.0	5.0	42.25
Pythia-6.9B variants	1.48	1.0	5.0	39.1
OLMo2-7B variants	1.43	1.0	4.0	35.9
Gemma2-2B variants	1.19	1.0	4.0	16.55
Qwen2.5-1.5B variants	1.43	1.0	4.0	35.9
Llama-3.1-8B variants	1.43	1.0	4.0	35.85

Table 5: Tokenization statistics across different models (English only). Most models have an average of 1.0-1.5 tokens per word and a median of 1, indicating that most words are tokenized as a single unit. However, there is variation in the proportion of words split into multiple tokens. Decoder-only models (e.g., GPT-2, Pythia, Qwen2, LLaMA) split 35-42% of words, while BERT and DeBERTa variants split fewer words (2-7%). Maximum tokens per word range from 4 to 6 across all models.

Model	HuggingFace ID
BERT-Base	bert-base-uncased
BERT-Large	bert-large-uncased
DeBERTa-v3-Large	microsoft/deberta-v3-large
mT5-base	google/mt5-base
GPT-2-Small	openai-community/gpt2
GPT-2-Large	openai-community/gpt2-large
GPT-2-XL	openai-community/gpt2-xl
Pythia-6.9B	EleutherAI/pythia-6.9b
Pythia-6.9B-Tulu	allenai/open-instruct-pythia-6.9b-tulu
OLMo-2-1124-7B	allenai/OLMo-2-1124-7B
OLMo-2-1124-7B-Instruct	allenai/OLMo-2-1124-7B-Instruct
Gemma-2-2B	google/gemma-2-2b
Gemma-2-2B-Instruct	google/gemma-2-2b-it
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B
Qwen2.5-1.5B-Instruct	Qwen/Qwen2.5-1.5B-Instruct
Qwen2.5-7B	Qwen/Qwen2.5-7B
Qwen2.5-7B-Instruct	Qwen/Qwen2.5-7B-Instruct
Llama-3.1-8B	meta-llama/Llama-3.1-8B
Llama-3.1-8B-Instruct	meta-llama/Llama-3.1-8B-Instruct
Goldfish English	goldfish-models/goldfish_eng_latn_1000mb
Goldfish Chinese	goldfish-models/goldfish_zho_hans_1000mb
Goldfish German	goldfish-models/goldfish_deu_latn_1000mb
Goldfish French	goldfish-models/goldfish_fra_latn_1000mb
Goldfish Russian	goldfish-models/goldfish_rus_cyrl_1000mb
Goldfish Turkish	goldfish-models/goldfish_tur_latn_1000mb

Table 6: Canonical HuggingFace model IDs used to load models in our study.

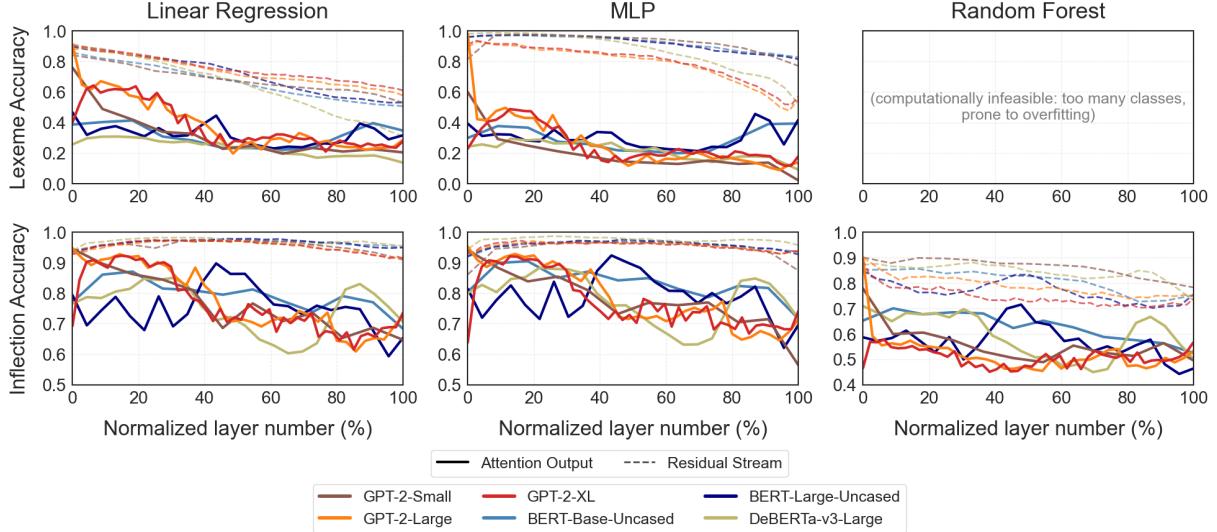


Figure 7: Linguistic accuracy for attention head outputs (solid lines) versus residual stream representations (dashed lines) across BERT and GPT-2 model families. Top row shows lexeme accuracy, bottom row shows inflection accuracy. Columns represent different classifier types. Attention outputs show pronounced drops in lexical accuracy during middle layers, while residual streams maintain higher performance. Both streams maintain high inflectional accuracy, though residual streams consistently outperform attention outputs.

MLP, and random forest classifiers on both attention head outputs and residual stream representations to compare their encoding patterns.

C.2 Results

The attention analysis reveals distinct encoding patterns between attention heads and residual streams. For lexical information, residual streams maintain relatively high accuracy (0.6-0.9) with gradual decline, while attention outputs show sharp drops to 0.2-0.4 in middle layers before modest recovery. For inflectional morphology, both components maintain high accuracy (0.7-1.0), though residual streams consistently outperform attention outputs, particularly in middle layers.

Selectivity analysis confirms that attention heads encode inflectional information more than lexical information, with inflection selectivity reaching 0.4-0.5 while lexeme selectivity remains near zero. This supports the hypothesis that attention mechanisms primarily handle contextual relationships rather than parametric lexical knowledge.

D Steering Vector Analysis

We conducted steering vector experiments to test whether inflectional representations can be functionally manipulated and to understand model sensitivity to activation interventions.

D.1 Methodology

For each inflectional category, we computed steering vectors as:

$$\mathbf{s}_i = \mu_i - \lambda \cdot \frac{1}{|C|-1} \sum_{j \in C, j \neq i} \mu_j \quad (5)$$

We tested multiple values of λ (1, 5, 10, 20, 100) and measured the impact on MLP classifier performance when adding these steering vectors to existing activations for 1000 test words. We evaluated both mean probability change and prediction flip rate across all models.

D.2 Results

Steering vector analysis demonstrates that inflectional representations can be effectively manipulated across most models and layers. The majority of models show robust steering effectiveness (>0.95 mean probability change and flip rate) throughout all layers, indicating that inflectional information is functionally accessible and modifiable.

However, we observe model-specific sensitivity patterns. The Qwen2.5 variants show pronounced vulnerability in middle layers (around 10% depth), where steering effectiveness drops to 0.6-0.7 before recovering in later layers. This sensitivity pattern aligns with our intrinsic dimensionality findings, where these same models show extreme compres-

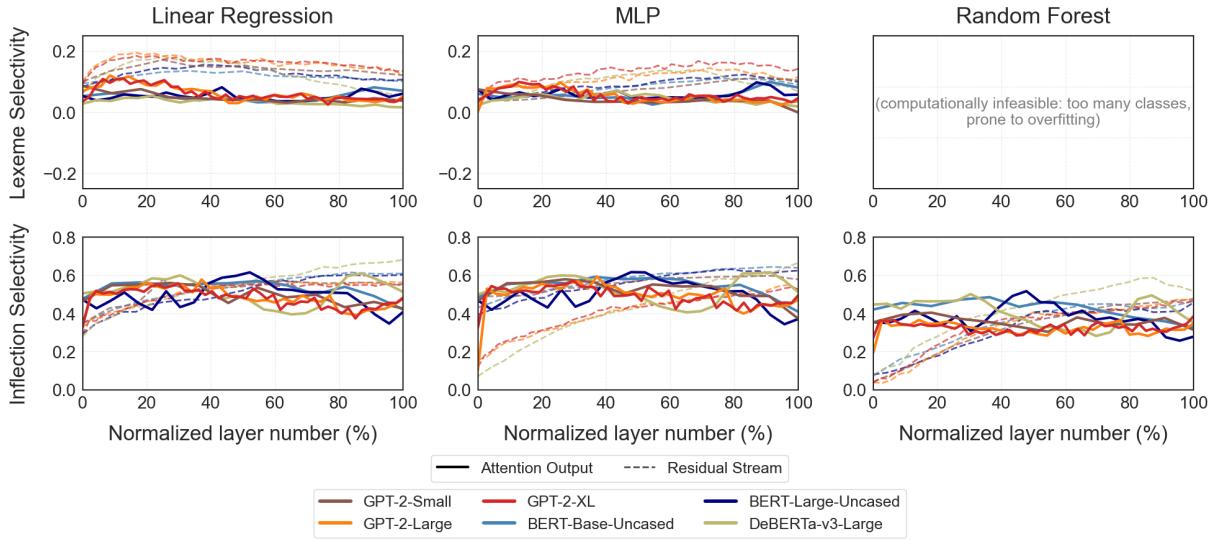


Figure 8: Classifier selectivity for attention head outputs (solid lines) versus residual stream representations (dashed lines) across BERT and GPT-2 model families. Top row shows lexeme selectivity, bottom row shows inflection selectivity. Attention outputs show near-zero lexeme selectivity throughout all layers, while inflection selectivity reaches moderate levels (0.4–0.5) in both streams, with residual streams showing slightly higher values.

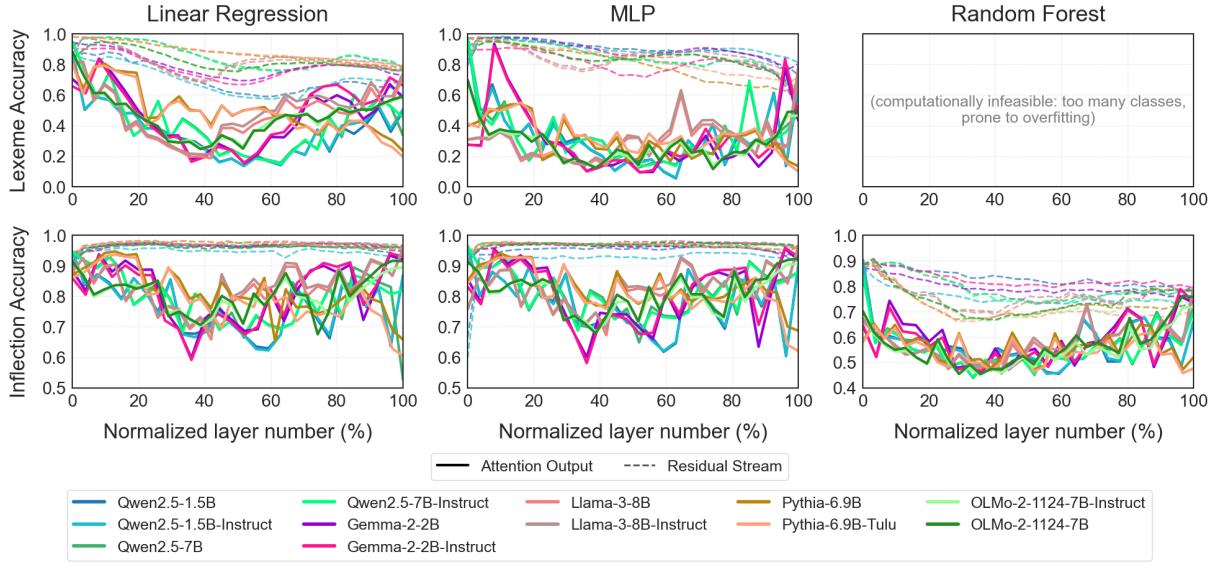


Figure 9: Linguistic accuracy for attention head outputs (solid lines) versus residual stream representations (dashed lines) across contemporary model families. Top row shows lexeme accuracy, bottom row shows inflection accuracy. The pattern of attention outputs showing lower lexical accuracy in middle layers is consistent across all model families, while inflectional accuracy remains high in both streams.

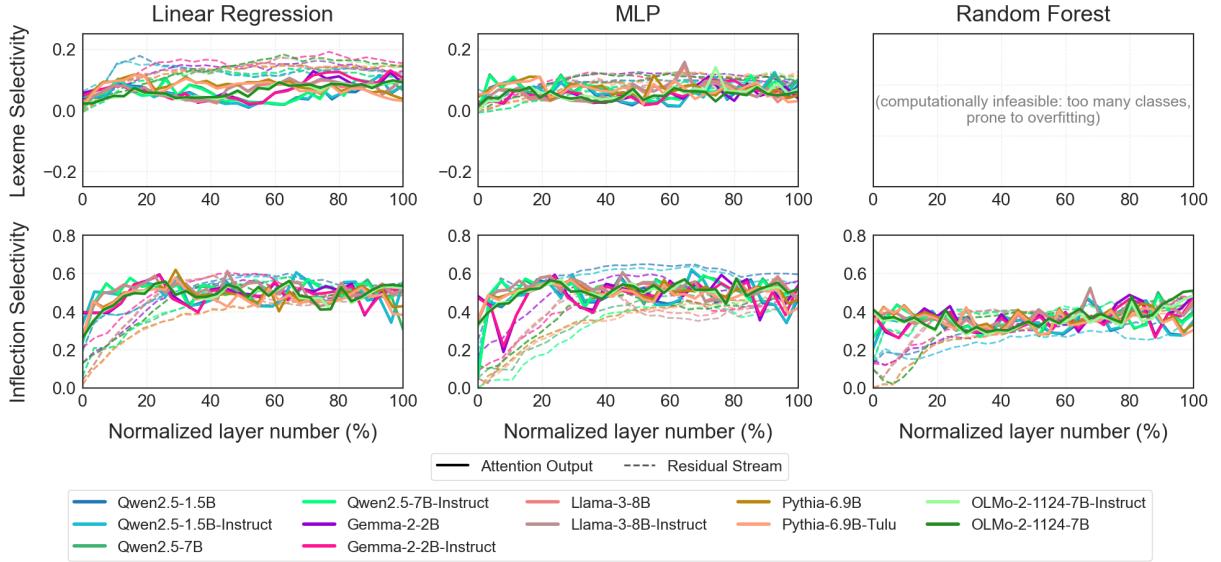


Figure 10: Classifier selectivity for attention head outputs (solid lines) versus residual stream representations (dashed lines) across contemporary model families. The selectivity patterns mirror those seen in BERT and GPT-2 families, with attention outputs maintaining near-zero lexeme selectivity and moderate inflection selectivity across all models.

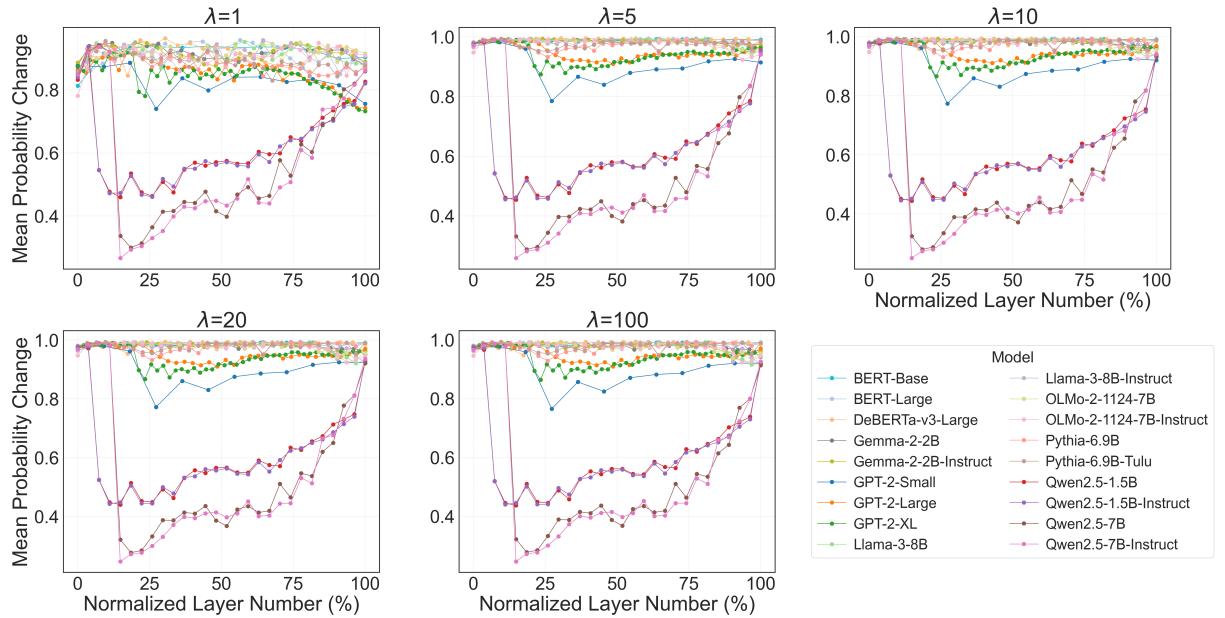


Figure 11: Mean probability change for inflection prediction when applying steering vectors across different λ values. Five panels show results for $\lambda \in \{1, 5, 10, 20, 100\}$. All models start with high effectiveness ($\approx 0.9-1.0$) at layer 0. Most models maintain stable performance, but Qwen2.5 variants show pronounced sensitivity dips around 10% layer depth before recovering. Higher λ values increase steering effectiveness while preserving the overall pattern.

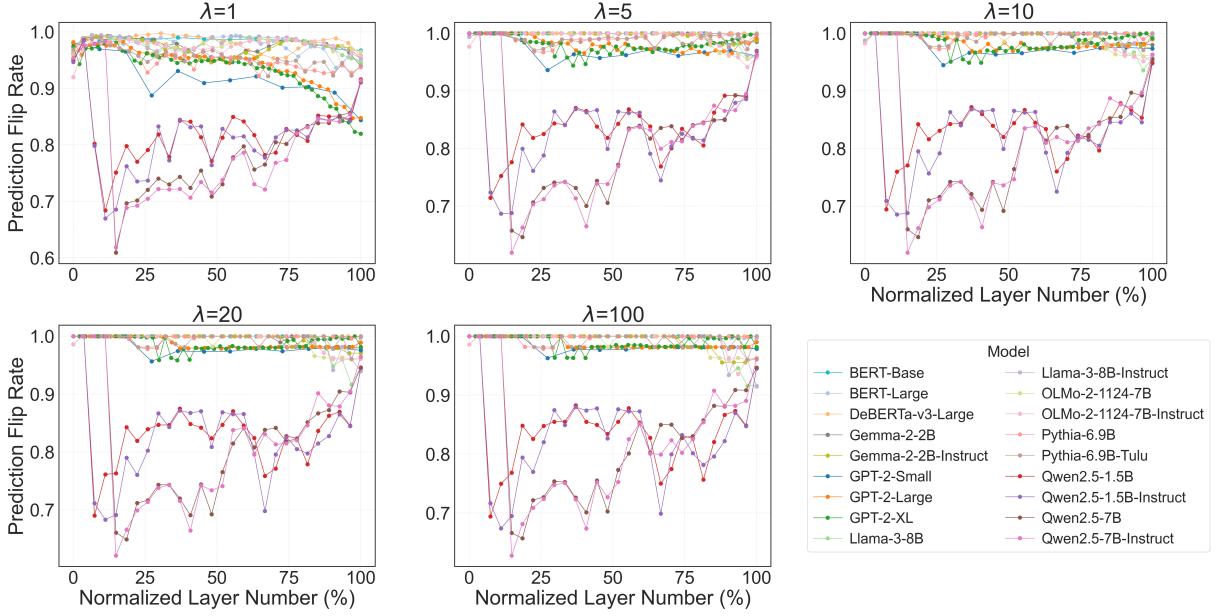


Figure 12: Prediction flip rate when applying steering vectors across different λ values. The flip rate patterns mirror the probability change results, with most models maintaining high rates (0.98-1.0) throughout all layers. Qwen2.5 variants show characteristic V-shaped dips to $\approx 0.60-0.70$ around 10% layer depth. The consistency across λ values suggests that steering effectiveness depends more on model architecture than intervention strength.

sion in middle layers (requiring only 1 component for 50-99% variance).

The consistency of steering effectiveness across different λ values suggests that the underlying representational structure, rather than intervention magnitude, determines steering success. This supports our conclusion that models with compressed middle-layer representations are more susceptible to activation interventions.

E Random Forest Analysis

We evaluated Random Forest classifiers as an additional non-linear baseline to complement our linear regression and MLP analyses. Random forests provide a different approach to capturing non-linear patterns through ensemble methods and feature interactions.

E.1 English Results

Lexical Identity Random Forest classifiers for lemma prediction were not comprehensively evaluated due to computational constraints arising from thousands of unique lemma classes. The one-vs-all training approach required for multi-class classification with Random Forests becomes prohibitively expensive with such a large number of classes.

Inflectional Morphology Random Forest classifiers for inflectional morphology show markedly

different patterns compared to linear regression and MLP classifiers (Figure 13, bottom left). While linear and MLP classifiers maintain consistently high accuracy (0.9-1.0) across all layers, Random Forests exhibit lower overall accuracy (0.65-0.90) and a declining trend across layers. This contrasts sharply with the flat, high-performance curves observed for linear methods.

Selectivity Analysis Random Forest selectivity for inflectional morphology (Figure 13, bottom right) shows moderate values (0.2-0.4) but with more variability across models compared to linear methods. This suggests that while Random Forests can generalize beyond memorization, they are less consistent in their ability to extract the underlying morphological patterns compared to simpler linear approaches.

E.2 Cross-Linguistic Patterns

The multilingual Random Forest analysis (Figure 14) reveals consistent cross-linguistic patterns while highlighting language-specific challenges.

Cross-Linguistic Performance Random Forest classifiers for inflectional morphology show the same language ordering observed with linear and MLP classifiers: Russian and German achieve the highest accuracy (0.80-0.90), English and French show intermediate performance (0.70-0.85), Chi-

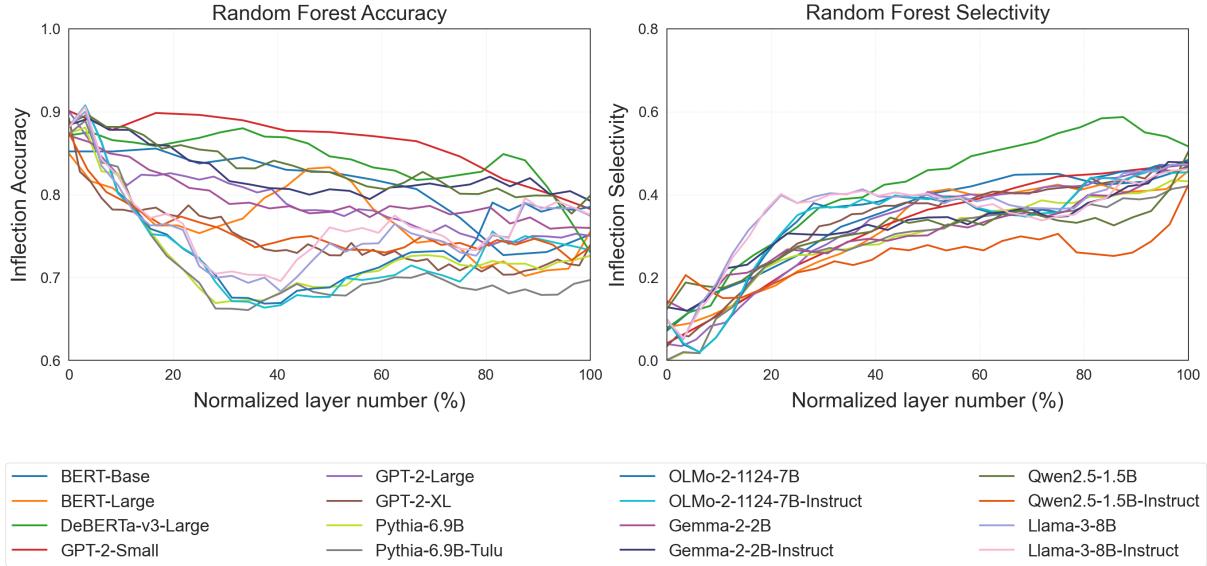


Figure 13: Random Forest classifier performance across model layers for English. The left column shows accuracy for lexeme (top) and inflection (bottom) prediction. The right column shows classifier selectivity (difference between linguistic and control task accuracy) for the same tasks. Each line represents a different model.

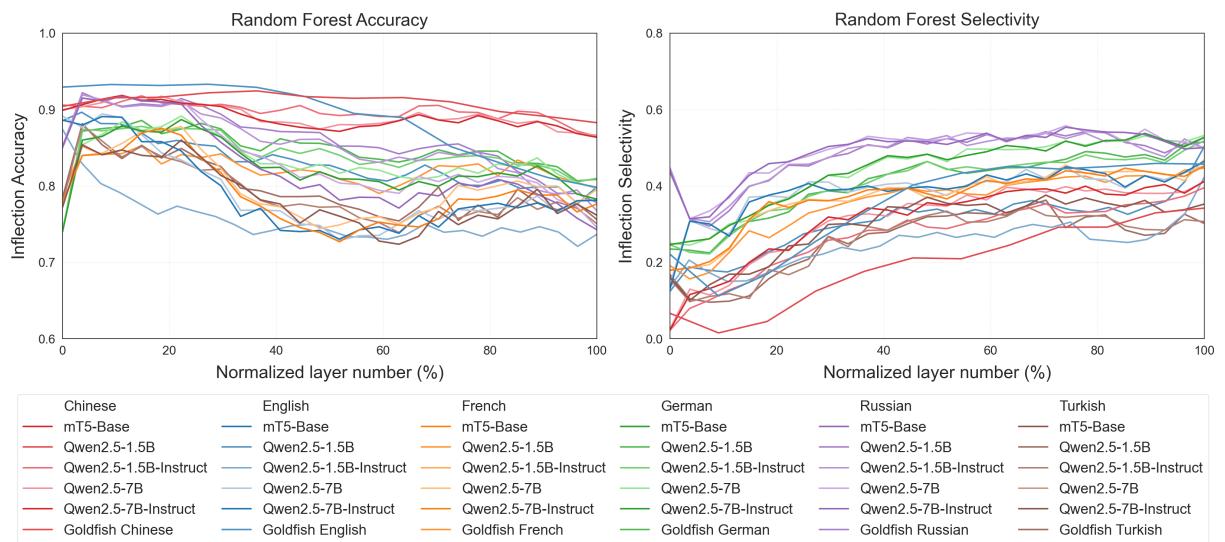


Figure 14: Random Forest classifier performance across languages and model layers. The left column shows accuracy for inflection prediction (lemma prediction omitted due to computational constraints with thousands of classes). The right column shows classifier selectivity.

Model	3rd person (n=249)	Base (n=1,833)	Comparative (n=76)	Past (n=1,003)	Plural (n=1,247)	Positive (n=1,785)	Singular (n=3,587)	Superlative (n=52)
BERT-Base	0.960	0.965	0.817	0.967	0.983	0.946	0.971	0.759
BERT-Large	0.956	0.964	0.861	0.968	0.982	0.950	0.971	0.768
DeBERTa-v3-Large	0.938	0.974	0.831	0.961	0.986	0.954	0.977	0.706
GPT-2-Small	0.828	0.958	0.840	0.956	0.974	0.941	0.964	0.754
GPT-2-Large	0.812	0.958	0.826	0.951	0.975	0.936	0.967	0.792
GPT-2-XL	0.817	0.959	0.813	0.948	0.977	0.940	0.968	0.788
Pythia-6.9B	0.886	0.972	0.904	0.964	0.989	0.957	0.977	0.907
Pythia-6.9B-Tulu	0.899	0.973	0.909	0.967	0.989	0.956	0.976	0.910
OLMo-2-1124-7B	0.938	0.968	0.902	0.972	0.981	0.923	0.966	0.888
OLMo-2-1124-7B-Instruct	0.927	0.967	0.896	0.971	0.981	0.923	0.965	0.872
Gemma-2-2B	0.901	0.968	0.797	0.969	0.986	0.947	0.974	0.833
Gemma-2-2B-Instruct	0.913	0.966	0.863	0.973	0.988	0.938	0.972	0.872
Qwen2.5-1.5B	0.856	0.950	0.802	0.942	0.972	0.919	0.957	0.688
Qwen2.5-1.5B-Instruct	0.774	0.954	0.647	0.945	0.972	0.921	0.965	0.630

Table 7: Breakdown of inflection classification accuracy by morphological feature for each model using linear regression classifiers (English). Inflections are grouped by their morphological features (*e.g.*, Past, Plural, Comparative). For each group, the reported accuracy is the average of accuracies from classifiers trained at each model layer. All accuracy values are on a 0–1 scale. Comparative and superlative forms consistently show the lowest accuracy across all models, reflecting the challenges of these less frequent morphological categories.

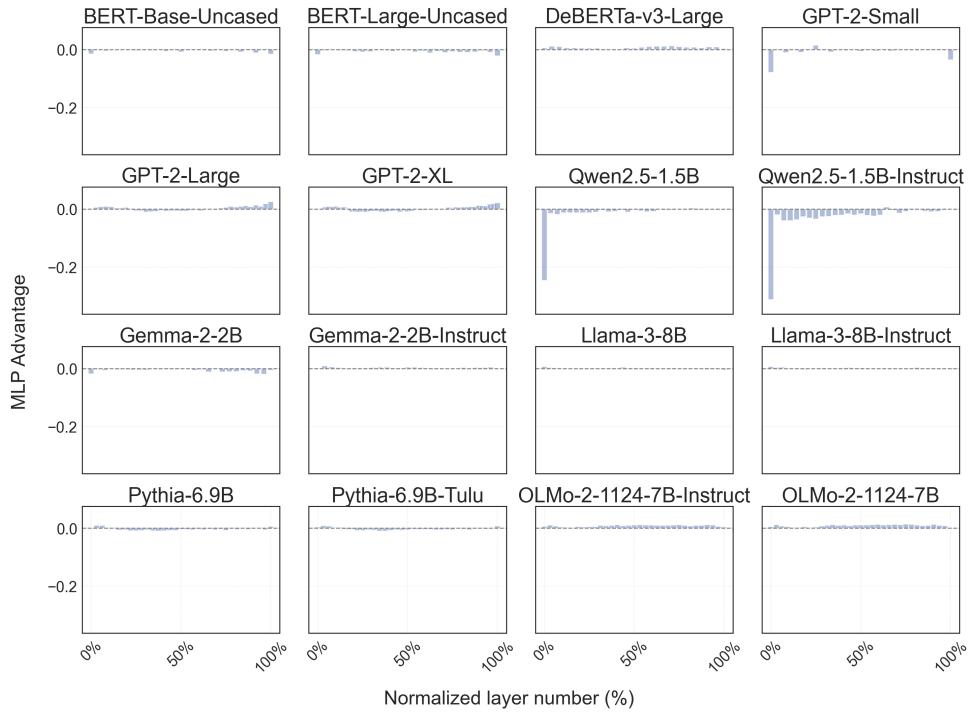
nese maintains moderate accuracy (0.65–0.80), and Turkish consistently underperforms (0.55–0.75). This consistency across classifier types confirms that the observed patterns reflect genuine differences in how linguistic information is encoded rather than classifier-specific artifacts.

Selectivity Patterns Random Forest selectivity patterns are similar to those of linear classifiers but with greater variance. Russian and German maintain the highest selectivity (0.3–0.5), while Turkish shows the lowest (0.1–0.3). This consistency suggests that the underlying representational structure, rather than classifier choice, drives the observed cross-linguistic differences.

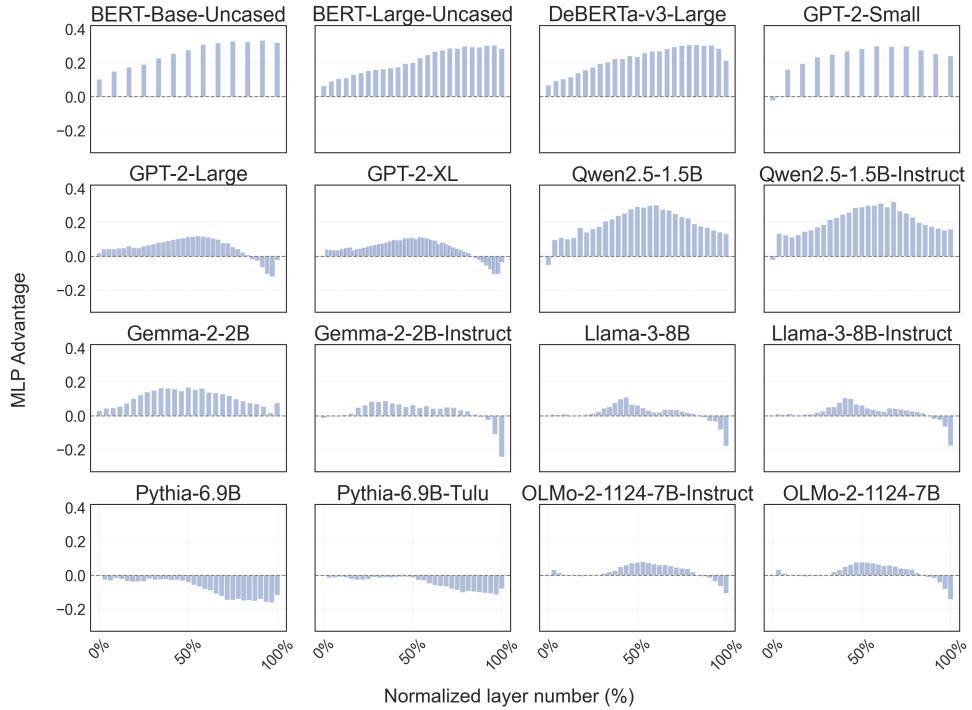
F Classifier Error Analysis

We conducted a detailed error analysis of our classifiers to better understand their performance across different morphological features and languages. See Table 7 through Table 25 for the full results.

Linear Separability Gap



(a) Linear separability gap for inflection prediction



(b) Linear separability gap for lemma prediction

Figure 15: Performance advantage of MLP classifiers over linear classifiers (in percentage points) across model layers for English. The linear separability gap measures how much a non-linear transformation improves classifier performance compared to a simple linear mapping. For inflection prediction, the gap is consistently minimal (mostly within ± 0.02 percentage points) and sometimes negative, indicating that inflectional features are primarily encoded in a linear fashion throughout the network. By contrast, the linear separability gap for lemma prediction is relatively large (0.1–0.3 percentage points) and positive across most models

Cross-Linguistic Linear Separability Gap

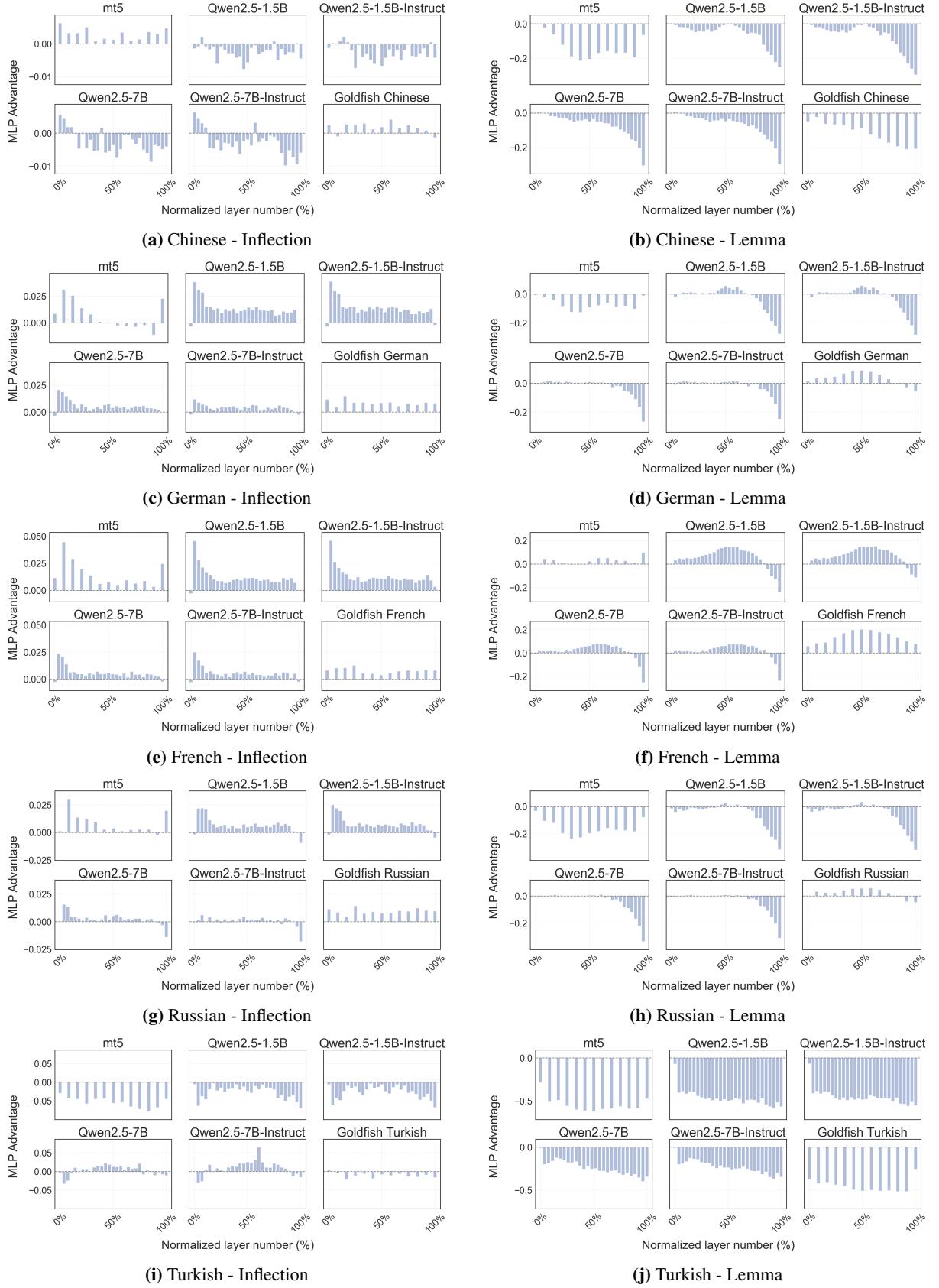


Figure 16: Cross-linguistic linear separability gap showing performance advantage of MLP classifiers over linear classifiers across model layers for five additional languages. For inflectional features, mT5 and Goldfish models show slight positive gaps (indicating modest benefits from non-linear classification), while Qwen2.5 variants show slight negative gaps (indicating linear classifiers are sufficient or superior). For lexical features, all models show negative gaps that are most pronounced in early layers, suggesting that linear regression with regularization consistently outperforms MLPs for lexical classification across all model families and languages.

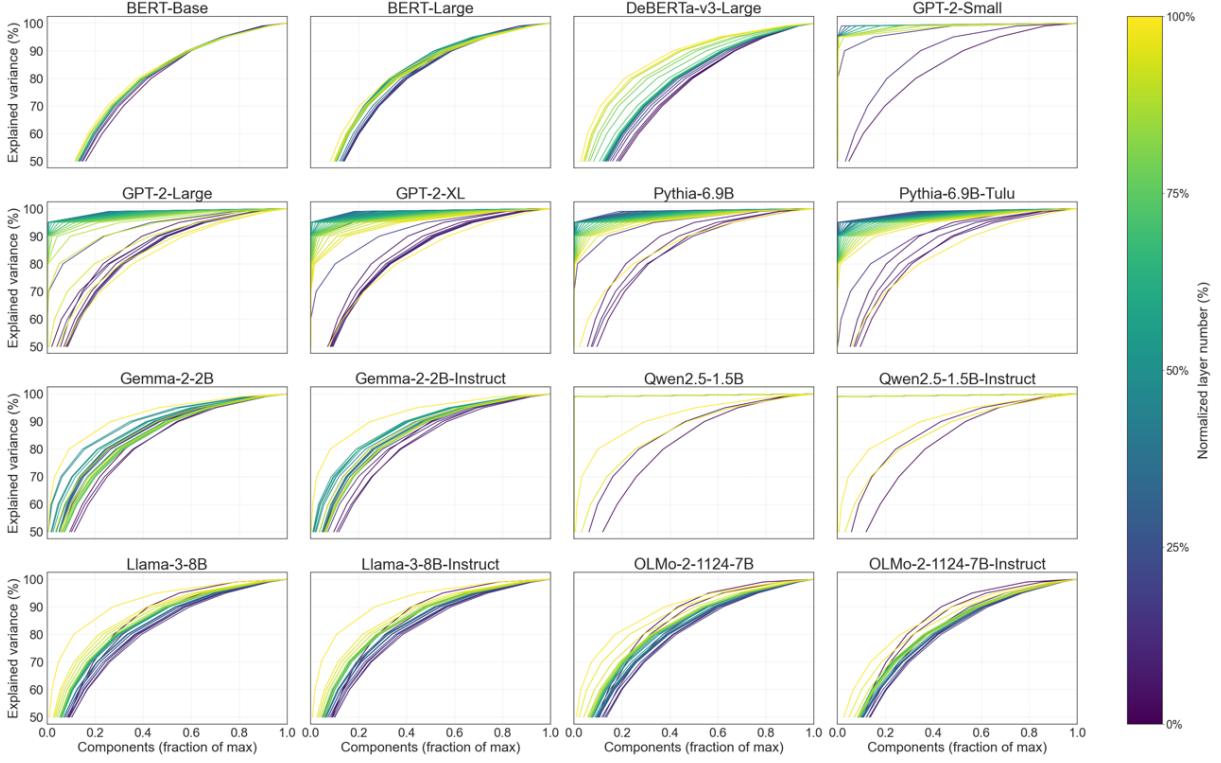


Figure 17: Intrinsic dimensionality curves for all models for English. Each subplot shows the relationship between the percentage of maximum PCA components (x-axis) and the percentage of explained variance (y-axis) across different layers. The color gradient from purple (early layers, 0%) to yellow (late layers, 100%) indicates the relative layer depth within each model. Models like BERT, Gemma, and Llama show similar compression patterns, while GPT-2 variants, Qwen and Pythia exhibit opposite trends in their middle layers.

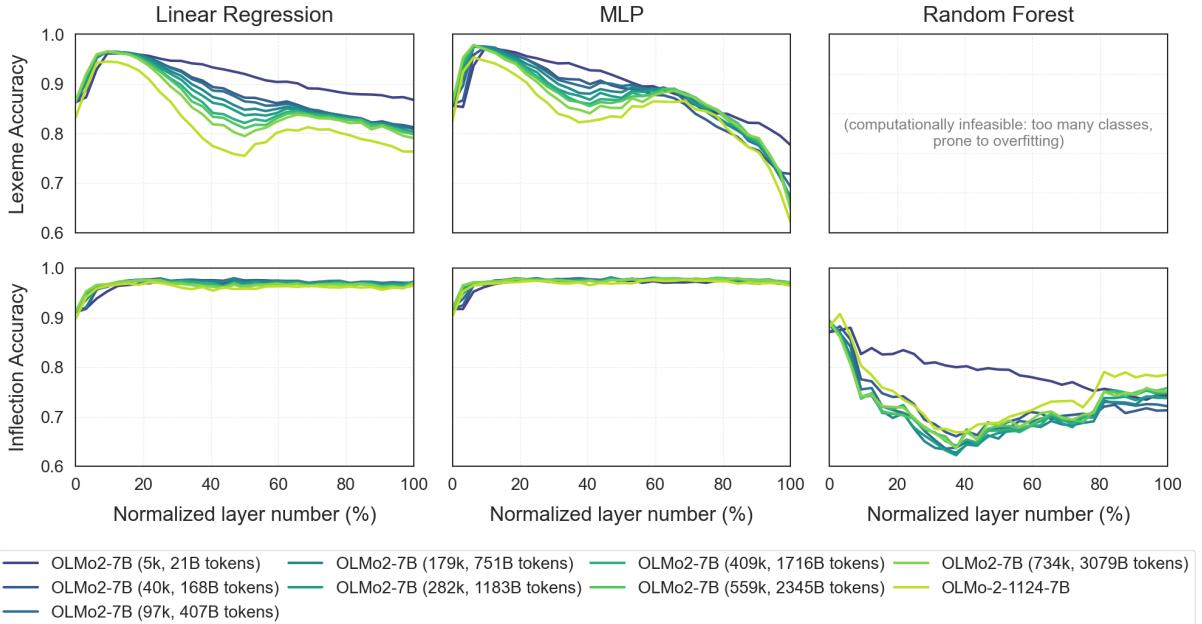


Figure 18: Lexeme (top row) and inflection (bottom row) prediction accuracy across normalized layer number for OLMo-2-7B checkpoints at various pretraining steps (5k-734k steps) for English. The full model is 928k steps. Checkpoints are on a color gradient from brightest (earliest) to darkest (latest). Early checkpoints exhibit higher lexeme accuracy than later ones, while inflectional accuracy remains flat across layers and checkpoints.

Model	3rd person (n=249)	Base (n=1,833)	Comparative (n=76)	Past (n=1,003)	Plural (n=1,247)	Positive (n=1,785)	Singular (n=3,587)	Superlative (n=52)
BERT-Base	0.973	0.969	0.910	0.972	0.989	0.959	0.974	0.939
BERT-Large	0.967	0.970	0.910	0.973	0.988	0.961	0.975	0.931
DeBERTa-v3-Large	0.954	0.976	0.925	0.966	0.989	0.962	0.979	0.867
GPT-2-Small	0.921	0.963	0.928	0.952	0.972	0.930	0.963	0.870
GPT-2-Large	0.857	0.962	0.872	0.955	0.976	0.942	0.967	0.854
GPT-2-XL	0.921	0.963	0.928	0.952	0.972	0.930	0.963	0.870
Pythia-6.9B	0.932	0.972	0.921	0.961	0.982	0.949	0.971	0.886
Pythia-6.9B-Tulu	0.948	0.974	0.932	0.964	0.983	0.949	0.971	0.897
OLMo-2-1124-7B	0.957	0.968	0.926	0.966	0.989	0.949	0.973	0.905
OLMo-2-1124-7B-Instruct	0.939	0.967	0.903	0.967	0.988	0.949	0.973	0.873
Gemma-2-2B	0.913	0.967	0.863	0.968	0.990	0.950	0.976	0.907
Gemma-2-2B-Instruct	0.930	0.970	0.878	0.975	0.989	0.946	0.974	0.906
Qwen2.5-1.5B	0.882	0.948	0.822	0.943	0.974	0.927	0.957	0.736
Qwen2.5-1.5B-Instruct	0.808	0.953	0.697	0.947	0.974	0.930	0.965	0.682

Table 8: Breakdown of inflection classification accuracy by morphological feature for each model using MLP classifiers (English). Inflections are grouped by their morphological features (*e.g.*, Past, Plural, Comparative). For each group, the reported accuracy is the average of accuracies from classifiers trained at each model layer. All accuracy values are on a 0–1 scale. MLP classifiers provide modest improvements over linear regression, particularly for comparative and superlative forms, though the relative ordering across morphological features remains consistent.

Model	Noun (n=1,739)	Verb (n=641)	Adjective (n=641)	Adverb (n=23)	Pronoun (n=1)	Preposition (n=1)	Conjunction (n=1)	Interjection (n=1)	Other (n=9)
BERT-Base	0.636	0.737	0.609	0.805	0.292	0.000	0.585	0.000	0.902
BERT-Large	0.684	0.777	0.653	0.826	0.580	0.154	0.662	0.065	0.897
DeBERTa-v3-Large	0.592	0.737	0.585	0.723	0.440	0.077	0.438	0.081	0.866
GPT-2-Small	0.631	0.789	0.612	0.813	0.542	0.000	0.415	0.033	0.896
GPT-2-Large	0.691	0.810	0.688	0.847	0.853	0.174	0.267	0.115	0.912
GPT-2-XL	0.713	0.827	0.708	0.847	0.724	0.222	0.311	0.241	0.899
Pythia-6.9B	0.856	0.926	0.836	0.926	0.938	0.443	0.566	0.488	0.934
Pythia-6.9B-Tulu	0.864	0.930	0.843	0.930	0.923	0.514	0.651	0.476	0.936
OLMo-2-1124-7B	0.798	0.875	0.794	0.913	0.697	0.339	0.363	0.495	0.913
OLMo-2-1124-7B-Instruct	0.798	0.868	0.792	0.902	0.606	0.339	0.331	0.495	0.910
Gemma-2-2B	0.757	0.869	0.736	0.876	0.667	0.179	0.205	0.288	0.891
Gemma-2-2B-Instruct	0.749	0.844	0.742	0.872	0.620	0.137	0.152	0.247	0.912
Qwen2.5-1.5B	0.652	0.801	0.650	0.828	0.526	0.082	0.223	0.068	0.867
Qwen2.5-1.5B-Instruct	0.642	0.800	0.632	0.831	0.544	0.082	0.245	0.068	0.877
Llama-3.1-8B	0.776	0.882	0.771	0.887	0.831	0.286	0.396	0.321	0.911
Llama-3.1-8B-Instruct	0.796	0.892	0.788	0.896	0.908	0.300	0.443	0.357	0.917

Table 9: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model using linear regression classifiers (English). Lexemes are grouped by their POS tags (*e.g.*, Noun, Verb, Adjective). For each group, the reported accuracy is the average of accuracies from classifiers trained at each model layer. All accuracy values are on a 0–1 scale. Performance varies significantly with frequency: frequent categories like nouns and verbs achieve higher accuracy, while infrequent categories like pronouns and prepositions show lower performance due to limited training examples.

Model	Noun (n=1,739)	Verb (n=641)	Adjective (n=641)	Adverb (n=23)	Pronoun (n=1)	Preposition (n=1)	Conjunction (n=1)	Interjection (n=1)	Other (n=9)
BERT-Base	0.775	0.831	0.748	0.873	0.458	0.125	0.756	0.267	0.898
BERT-Large	0.813	0.863	0.785	0.884	0.540	0.231	0.725	0.323	0.897
DeBERTa-v3-Large	0.689	0.803	0.682	0.802	0.700	0.115	0.662	0.242	0.861
GPT-2-Small	0.678	0.792	0.665	0.765	0.042	0.000	0.610	0.000	0.830
GPT-2-Large	0.754	0.837	0.755	0.827	0.347	0.188	0.596	0.385	0.871
GPT-2-XL	0.774	0.844	0.771	0.827	0.561	0.232	0.561	0.431	0.860
Pythia-6.9B	0.774	0.856	0.768	0.862	0.554	0.229	0.528	0.310	0.868
Pythia-6.9B-Tulu	0.818	0.880	0.803	0.887	0.554	0.343	0.613	0.381	0.889
OLMo-2-1124-7B	0.818	0.877	0.828	0.896	0.727	0.290	0.734	0.505	0.885
OLMo-2-1124-7B-Instruct	0.822	0.874	0.829	0.897	0.667	0.306	0.750	0.473	0.886
Gemma-2-2B	0.763	0.860	0.763	0.881	0.574	0.125	0.443	0.182	0.880
Gemma-2-2B-Instruct	0.777	0.846	0.785	0.882	0.580	0.137	0.400	0.299	0.875
Qwen2.5-1.5B	0.747	0.838	0.742	0.811	0.228	0.131	0.628	0.164	0.857
Qwen2.5-1.5B-Instruct	0.749	0.840	0.738	0.818	0.211	0.098	0.564	0.123	0.860
Llama-3.1-8B	0.798	0.879	0.807	0.886	0.800	0.214	0.679	0.393	0.882
Llama-3.1-8B-Instruct	0.824	0.893	0.826	0.895	0.831	0.257	0.689	0.429	0.887

Table 10: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model using Multi-Layer Perceptron (MLP) classifiers (English). Lexemes are grouped by their POS tags (*e.g.*, Noun, Verb, Adjective). For each group, the reported accuracy is the average of accuracies from classifiers trained at each model layer. All accuracy values are on a 0–1 scale. MLP classifiers provide consistent improvements over linear regression across all POS categories, though the frequency-dependent performance patterns persist.

Model	Linear Regression				MLP			
	Positive (n=300)	Base (n=2,074)	Plural (n=3)	Singular (n=3,947)	Positive (n=300)	Base (n=2,074)	Plural (n=3)	Singular (n=3,947)
mT5-Base	0.739	0.913	0.436	0.962	0.783	0.919	0.231	0.961
Qwen2.5-1.5B	0.785	0.929	0.034	0.969	0.801	0.924	0.092	0.967
Qwen2.5-1.5B-Instruct	0.779	0.925	0.034	0.964	0.803	0.923	0.057	0.967
Qwen2.5-7B	0.824	0.937	0.310	0.970	0.828	0.929	0.310	0.969
Qwen2.5-7B-Instruct	0.819	0.936	0.299	0.970	0.823	0.928	0.276	0.969
Goldfish Chinese	0.793	0.912	0.000	0.958	0.816	0.915	0.000	0.957

Table 11: Breakdown of inflection classification accuracy for each model by inflection type using Linear Regression and Multi-Layer Perceptron (MLP) classifiers (Chinese). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Noun (n=1,179)	Verb (n=564)	Adjective (n=108)	Adverb (n=22)	Preposition (n=20)	Other (n=50)
mT5-Base	0.838	0.828	0.786	0.762	0.920	0.726
Qwen2.5-1.5B	0.810	0.797	0.746	0.715	0.872	0.699
Qwen2.5-1.5B-Instruct	0.813	0.799	0.748	0.713	0.873	0.700
Qwen2.5-7B	0.887	0.882	0.846	0.847	0.915	0.817
Qwen2.5-7B-Instruct	0.886	0.877	0.843	0.835	0.913	0.811
Goldfish Chinese	0.883	0.878	0.845	0.875	0.954	0.858

Table 12: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Linear Regression classifiers (Chinese). Lexemes are grouped by their POS tags (*e.g.*, Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Noun (n=1,179)	Verb (n=564)	Adjective (n=108)	Adverb (n=22)	Preposition (n=20)	Other (n=50)
mT5-Base	0.698	0.712	0.564	0.571	0.884	0.569
Qwen2.5-1.5B	0.748	0.761	0.658	0.668	0.826	0.669
Qwen2.5-1.5B-Instruct	0.735	0.745	0.643	0.643	0.814	0.655
Qwen2.5-7B	0.815	0.826	0.749	0.745	0.848	0.750
Qwen2.5-7B-Instruct	0.815	0.822	0.747	0.734	0.845	0.744
Goldfish Chinese	0.766	0.771	0.647	0.621	0.912	0.682

Table 13: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Multi-Layer Perceptron (MLP) classifiers (Chinese). Lexemes are grouped by their POS tags (e.g., Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=417)	3rd person (n=517)	Positive (n=1,720)	Past (n=839)	Plural (n=1,076)	Superlative (n=52)	Singular (n=3,197)	Comparative (n=141)
mT5-Base	0.908	0.941	0.940	0.960	0.882	0.572	0.962	0.636
Qwen2.5-1.5B	0.849	0.889	0.922	0.914	0.888	0.657	0.953	0.796
Qwen2.5-1.5B-Instruct	0.844	0.887	0.922	0.910	0.889	0.659	0.952	0.795
Qwen2.5-7B	0.892	0.922	0.939	0.947	0.909	0.826	0.962	0.878
Qwen2.5-7B-Instruct	0.915	0.934	0.945	0.962	0.924	0.866	0.968	0.909
Goldfish German	0.938	0.941	0.955	0.979	0.916	0.542	0.968	0.708

Table 14: Breakdown of inflection classification accuracy for each model by inflection type using Linear Regression classifiers (German). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=417)	3rd person (n=517)	Positive (n=1,720)	Past (n=839)	Plural (n=1,076)	Superlative (n=52)	Singular (n=3,197)	Comparative (n=141)
mT5-Base	0.921	0.945	0.948	0.959	0.884	0.723	0.967	0.770
Qwen2.5-1.5B	0.890	0.915	0.930	0.940	0.897	0.831	0.958	0.892
Qwen2.5-1.5B-Instruct	0.888	0.914	0.930	0.938	0.898	0.825	0.957	0.897
Qwen2.5-7B	0.912	0.932	0.944	0.956	0.913	0.868	0.964	0.924
Qwen2.5-7B-Instruct	0.925	0.941	0.950	0.966	0.928	0.901	0.970	0.936
Goldfish German	0.947	0.957	0.964	0.978	0.923	0.817	0.970	0.896

Table 15: Breakdown of inflection classification accuracy for each model by inflection type using Multi-Layer Perceptron (MLP) classifiers (German). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Linear Regression				MLP			
	Noun (n=1,262)	Verb (n=395)	Adjective (n=406)	Other (n=12)	Noun (n=1,262)	Verb (n=395)	Adjective (n=406)	Other (n=12)
mT5-Base	0.685	0.662	0.568	0.750	0.611	0.602	0.486	0.723
Qwen2.5-1.5B	0.743	0.725	0.715	0.775	0.721	0.700	0.687	0.711
Qwen2.5-1.5B-Instruct	0.740	0.722	0.715	0.766	0.722	0.698	0.687	0.704
Qwen2.5-7B	0.821	0.809	0.808	0.829	0.795	0.786	0.783	0.814
Qwen2.5-7B-Instruct	0.815	0.803	0.803	0.821	0.795	0.785	0.782	0.813
Goldfish German	0.720	0.747	0.701	0.769	0.758	0.772	0.742	0.769

Table 16: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Linear Regression and Multi-Layer Perceptron (MLP) classifiers (German). Lexemes are grouped by their POS tags (e.g., Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=688)	3rd person (n=776)	Positive (n=1,833)	Past (n=857)	Plural (n=1,457)	Singular (n=5,169)
mT5-Base	0.934	0.912	0.879	0.908	0.954	0.970
Qwen2.5-1.5B	0.933	0.858	0.896	0.903	0.958	0.967
Qwen2.5-1.5B-Instruct	0.930	0.852	0.893	0.898	0.958	0.966
Qwen2.5-7B	0.955	0.918	0.918	0.931	0.965	0.975
Qwen2.5-7B-Instruct	0.951	0.913	0.915	0.928	0.964	0.974
Goldfish French	0.942	0.955	0.937	0.930	0.968	0.976

Table 17: Breakdown of inflection classification accuracy for each model by inflection type using Linear Regression classifiers (French). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=688)	3rd person (n=776)	Positive (n=1,833)	Past (n=857)	Plural (n=1,457)	Singular (n=5,169)
mT5-Base	0.957	0.937	0.911	0.935	0.957	0.977
Qwen2.5-1.5B	0.954	0.905	0.914	0.925	0.965	0.968
Qwen2.5-1.5B-Instruct	0.954	0.902	0.911	0.924	0.965	0.968
Qwen2.5-7B	0.966	0.936	0.930	0.937	0.970	0.976
Qwen2.5-7B-Instruct	0.962	0.931	0.926	0.934	0.970	0.975
Goldfish French	0.974	0.967	0.945	0.942	0.973	0.979

Table 18: Breakdown of inflection classification accuracy for each model by inflection type using Multi-Layer Perceptron (MLP) classifiers (French). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Linear Regression				MLP			
	Noun (n=1,496)	Verb (n=406)	Adjective (n=358)	Other (n=15)	Noun (n=1,496)	Verb (n=406)	Adjective (n=358)	Other (n=15)
	0.708	0.577	0.605	0.799	0.755	0.560	0.636	0.820
mT5-Base	0.754	0.725	0.673	0.824	0.807	0.765	0.751	0.853
Qwen2.5-1.5B	0.750	0.718	0.671	0.820	0.824	0.776	0.768	0.869
Qwen2.5-1.5B-Instruct	0.840	0.814	0.764	0.869	0.856	0.825	0.794	0.884
Qwen2.5-7B	0.833	0.805	0.758	0.860	0.851	0.818	0.792	0.883
Goldfish French	0.749	0.758	0.661	0.811	0.894	0.869	0.813	0.888

Table 19: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Linear Regression and Multi-Layer Perceptron (MLP) classifiers (French). Lexemes are grouped by their POS tags (e.g., Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=690)	3rd person (n=456)	Positive (n=1,192)	Past (n=455)	Plural (n=1,333)	Superlative (n=3)	Singular (n=3,316)	Comparative (n=23)
mT5-Base	0.930	0.978	0.975	0.957	0.877	0.000	0.977	0.799
Qwen2.5-1.5B	0.925	0.946	0.974	0.938	0.923	0.015	0.966	0.835
Qwen2.5-1.5B-Instruct	0.924	0.943	0.974	0.934	0.921	0.015	0.966	0.817
Qwen2.5-7B	0.949	0.966	0.979	0.958	0.948	0.094	0.977	0.872
Qwen2.5-7B-Instruct	0.951	0.974	0.980	0.970	0.948	0.080	0.980	0.918
Goldfish Russian	0.940	0.950	0.976	0.931	0.921	0.000	0.976	0.867

Table 20: Breakdown of inflection classification accuracy for each model by inflection type using Linear Regression classifiers (Russian). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=690)	3rd person (n=456)	Positive (n=1,192)	Past (n=455)	Plural (n=1,333)	Superlative (n=3)	Singular (n=3,316)	Comparative (n=23)
mT5-Base	0.959	0.978	0.969	0.966	0.904	0.000	0.978	0.849
Qwen2.5-1.5B	0.952	0.955	0.972	0.948	0.933	0.089	0.970	0.899
Qwen2.5-1.5B-Instruct	0.950	0.954	0.973	0.947	0.933	0.089	0.969	0.911
Qwen2.5-7B	0.963	0.964	0.978	0.960	0.951	0.246	0.979	0.910
Qwen2.5-7B-Instruct	0.961	0.970	0.978	0.966	0.949	0.126	0.980	0.924
Goldfish Russian	0.965	0.972	0.978	0.948	0.943	0.000	0.977	0.934

Table 21: Breakdown of inflection classification accuracy for each model by inflection type using Multi-Layer Perceptron (MLP) classifiers (Russian). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Linear Regression				MLP			
	Noun (n=982)	Verb (n=333)	Adjective (n=275)	Other (n=4)	Noun (n=982)	Verb (n=333)	Adjective (n=275)	Other (n=4)
mT5-Base	0.660	0.614	0.542	0.648	0.492	0.484	0.387	0.426
Qwen2.5-1.5B	0.777	0.712	0.759	0.720	0.712	0.696	0.716	0.647
Qwen2.5-1.5B-Instruct	0.772	0.704	0.756	0.720	0.710	0.689	0.717	0.643
Qwen2.5-7B	0.854	0.790	0.843	0.812	0.798	0.794	0.813	0.749
Qwen2.5-7B-Instruct	0.845	0.778	0.835	0.807	0.794	0.785	0.809	0.744
Goldfish Russian	0.795	0.723	0.764	0.676	0.810	0.776	0.759	0.657

Table 22: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Linear Regression and Multi-Layer Perceptron (MLP) classifiers (Russian). Lexemes are grouped by their POS tags (e.g., Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=154)	3rd person (n=51)	Positive (n=401)	Past (n=168)	Plural (n=33)	Singular (n=632)
mT5-Base	0.860	0.911	0.928	0.966	0.837	0.952
Qwen2.5-1.5B	0.808	0.802	0.721	0.928	0.861	0.892
Qwen2.5-1.5B-Instruct	0.809	0.817	0.720	0.941	0.878	0.899
Qwen2.5-7B	0.865	0.879	0.810	0.966	0.903	0.909
Qwen2.5-7B-Instruct	0.850	0.874	0.796	0.960	0.886	0.900
Goldfish Turkish	0.847	0.915	0.880	0.964	0.872	0.963

Table 23: Breakdown of inflection classification accuracy for each model by inflection type using Linear Regression classifiers (Turkish). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Base (n=154)	3rd person (n=51)	Positive (n=401)	Past (n=168)	Plural (n=33)	Singular (n=632)
mT5-Base	0.755	0.760	0.848	0.922	0.515	0.949
Qwen2.5-1.5B	0.770	0.767	0.667	0.919	0.765	0.914
Qwen2.5-1.5B-Instruct	0.762	0.757	0.662	0.917	0.766	0.913
Qwen2.5-7B	0.853	0.845	0.791	0.956	0.875	0.937
Qwen2.5-7B-Instruct	0.845	0.844	0.786	0.956	0.875	0.932
Goldfish Turkish	0.832	0.879	0.870	0.957	0.834	0.957

Table 24: Breakdown of inflection classification accuracy for each model by inflection type using Multi-Layer Perceptron (MLP) classifiers (Turkish). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

Model	Linear Regression				MLP			
	Noun (n=221)	Verb (n=53)	Adjective (n=104)	Other (n=13)	Noun (n=221)	Verb (n=53)	Adjective (n=104)	Other (n=13)
	mT5-Base	0.866	0.823	0.921	0.955	0.215	0.421	0.374
Qwen2.5-1.5B	0.834	0.805	0.866	0.877	0.307	0.439	0.449	0.693
Qwen2.5-1.5B-Instruct	0.816	0.791	0.860	0.874	0.305	0.439	0.448	0.691
Qwen2.5-7B	0.871	0.850	0.900	0.904	0.595	0.625	0.695	0.809
Qwen2.5-7B-Instruct	0.850	0.823	0.883	0.885	0.579	0.613	0.678	0.800
Goldfish Turkish	0.929	0.904	0.940	0.969	0.386	0.550	0.477	0.808

Table 25: Breakdown of lexeme classification accuracy by Part of Speech (POS) for each model, using Linear Regression and Multi-Layer Perceptron (MLP) classifiers (Turkish). Lexemes are grouped by their POS tags (e.g., Noun, Verb, Adjective). Accuracies are calculated over all examples for a given group across all layers. Counts (n) are derived from a single representative layer for each group. All accuracy values are on a 0–1 scale.

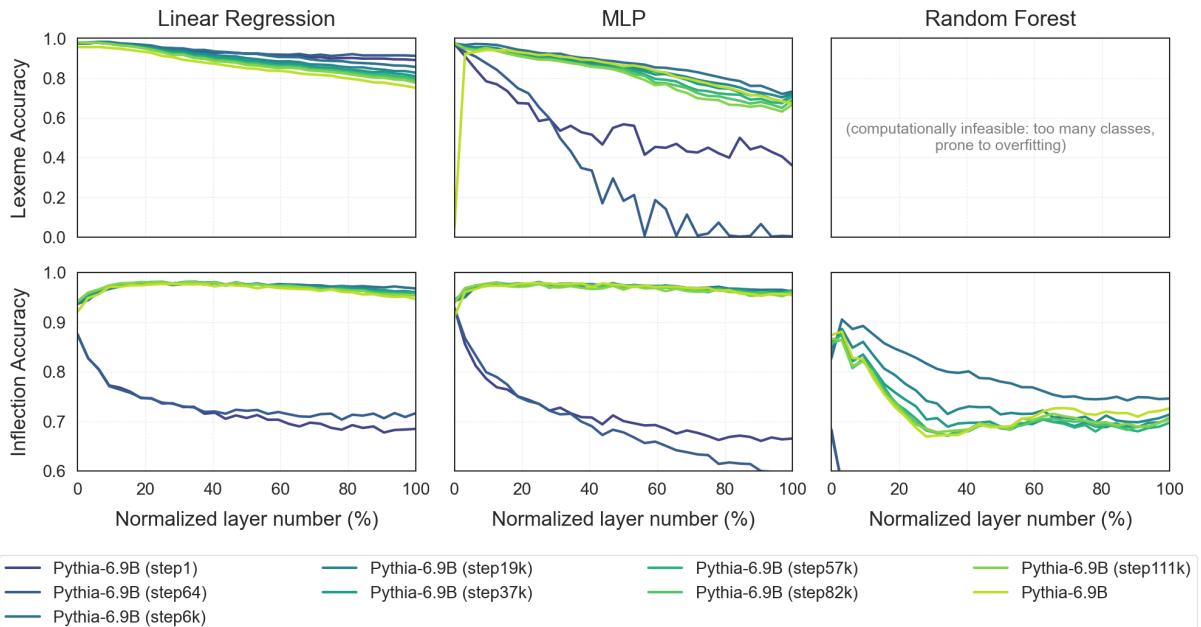


Figure 19: Lexeme (top row) and inflection (bottom row) prediction accuracy across normalized layer number for Pythia-6.9B checkpoints at various pretraining steps (1–111k steps) for English. The full model is 143k steps. Checkpoints are on a color gradient from brightest (earliest) to darkest (latest). Lexeme accuracy declines both with deeper layers and with more training, whereas inflectional accuracy stays uniformly high across all layers and checkpoints.