Implicit Representations of Grammaticality in Language Models

Yingshan Susan Wang Linlu Qiu Zhaofeng Wu Roger P. Levy Yoon Kim
Massachusetts Institute of Technology
susanw26@mit.edu

Abstract

Grammaticality and likelihood are distinct notions in human language. Pretrained language models (LMs), which are probabilistic models of language fitted to maximize corpus likelihood, generate grammatically well-formed text and discriminate well between grammatical and ungrammatical sentences in tightly controlled minimal pairs. However, their string probabilities do not sharply discriminate between grammatical and ungrammatical sentences overall. But do LMs implicitly acquire a grammaticality distinction distinct from string probability? We explore this question through studying internal representations of LMs, by training a linear probe on a dataset of grammatical and (synthetic) ungrammatical sentences obtained by applying perturbations to a naturalistic text corpus. We find that this simple *grammaticality probe* generalizes to human-curated grammaticality judgment benchmarks and outperforms LM probability-based grammaticality judgments. When applied to semantic plausibility benchmarks, in which both members of a minimal pair are grammatical and differ in only plausibility, the probe however performs worse than string probability. The English-trained probe also exhibits nontrivial cross-lingual generalization, outperforming string probabilities on grammaticality benchmarks in numerous other languages. Additionally, probe scores correlate only weakly with string probabilities. These results collectively suggest that LMs acquire to some extent an implicit grammaticality distinction within their hidden layers.[1]

1 Introduction

In human language, a sentence's *grammaticality*—whether it conforms to the rules of language—is distinct from its *likelihood*—whether it is likely to occur in naturalistic text. Consider the following pair of examples from Levy et al. (2025) (itself adapted from Bock and Miller (1991)):

(1) *The key to the cabinets are on the table.

(2) The key to the cabinets was so thoroughly rusted that it was impossible to fit into the keyhole.

The first sentence violates the rule of English that

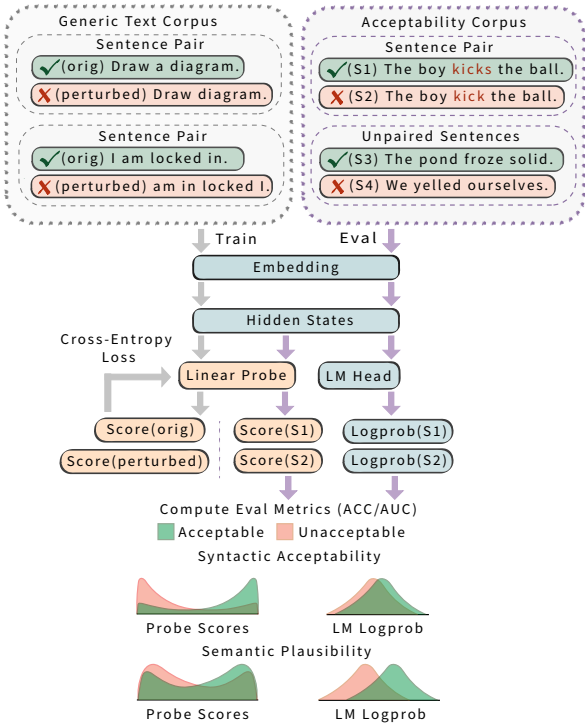


Figure 1: The grammaticality probe is **a linear classifier over LM hidden states**, trained on a synthetic dataset created from perturbing sentences from a generic text corpus. For evaluation, we measure how well probe scores and LM logprob discriminate between grammatical and ungrammatical sentences on acceptability-judgment corpora. As a baseline, we also evaluate the grammaticality probe on semantic plausibility datasets. **The probe is more discriminative for syntactic acceptability than LM logprobs, whereas LM logprobs are more sensitive to semantic plausibility.**

[1]The code for this project is available at https://github.com/SusanWYS/grammaticality_probe.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15035–15055
July 2-7, 2026 ©2026 Association for Computational Linguistics

the number of the verb *are* should match that of the singular head noun of the subject *key*, and is therefore ungrammatical. However, this type of error is common when a plural noun (*cabinets*) intervenes before the verb, so the sentence is almost certainly more likely to be used than the second sentence, which is grammatical but expresses a complex and unusual meaning. Insofar as this *grammaticality–likelihood distinction*—a consequence of the broader *competence–performance distinction*—is present in humans, the extent to which LMs are able to capture this distinction bears directly on their viability as candidate computational models of human language processing and acquisition—a matter of considerable ongoing debate in linguistics and cognitive science (Piantadosi, 2023; Katzir, 2023; Millière, 2024; Fox and Katzir, 2024; Futrell and Mahowald, 2025, *i.a.*).

LMs are at their core statistical systems that assign probabilities to strings, and thus naturally capture a notion of likelihood—they are literally trained to maximize the likelihood of the training data. However, their training data inevitably contain noise arising from (amongst others) *performance* factors, and sentences such as (1) occur frequently in naturalistic text. Thus, LMs do not (nor are they designed to) categorically assign lower probabilities to ungrammatical sentences than grammatical ones; indeed, Levy et al. (2025) note that GPT-2 assigns 146 trillion times more probability to (2) than to (1), and prior studies find that LM probabilities are generally poor at distinguishing grammatical from ungrammatical sentences (Leivada et al., 2024a,b, 2025). Hu et al. (2025) further show that string probabilities alone are theoretically inappropriate quantities for extracting grammaticality judgments under a simple generative model of language. Nevertheless, LM probabilities have been shown to be effective at adjudicating grammaticality across *pairs* of sentences that differ from one another minimally (Marvin and Linzen, 2018; Futrell et al., 2019; Warstadt et al., 2020; Hu et al., 2020, *i.a.*), indicating that while the grammaticality–likelihood distinction is not explicitly baked into LMs, they acquire behavioral generalizations that indicate some knowledge of a grammar. But as Leivada et al. (2024a) note, this type of grammaticality assessment based on minimal pairs is different from the standard by which we assess grammaticality judgments in humans.

However, while string probabilities from LMs are not designed to (only) encode grammaticality, in neural LMs these string probabilities are the result of internal computations performed by the model. In the present work we thus explore the hypothesis that LMs, through large-scale training, learn to *implicitly* encode grammaticality judgments within their hidden representations.[2] A standard approach to testing such hypotheses is through supervised probes, wherein a simple model (e.g., a linear classifier) is trained to predict some phenomena of interest (Alain and Bengio, 2017; Belinkov, 2022). In the case of grammaticality judgments, this might involve training a probe on acceptability judgment datasets on human-curated datasets such as CoLA (Warstadt et al., 2019) or BLiMP (Warstadt et al., 2020). However, such datasets are often small and costly to collect, and thus probes trained on them could overfit; ideally, such datasets should be used as held-out sets for evaluation.

We propose an approach for learning a grammaticality probe without relying on human-annotated data. We create a synthetic dataset of "good" and "bad" sentences by applying noise to sentences in a generic text corpus, where we randomly insert, delete, or shuffle tokens to create (mostly) ungrammatical examples. We find that a linear probe trained on this dataset can outperform probability-based grammaticality judgments in both minimal-pair- and non-minimal-pair-based grammaticality judgment benchmarks. The probe surprisingly exhibits cross-lingual generalization, i.e., a probe trained on English data can be used for grammaticality judgment on other languages, and outperforms probability-based judgments. The outperformance of probes over probabilities is flipped when distinguishing between semantically plausible/implausible sentences, and the probe scores are found to weakly correlate with string probabilities.

In arguing against LMs as theories of human linguistic cognition, Katzir (2023) state that the grammaticality–likelihood distinction is "entirely foreign" in LMs. Our results however suggest that this may not necessarily be the case, and that the

[2] *Metalinguistic* judgments, wherein an LM is explicitly asked to answer whether a sentence is grammatical or not, provides another perspective on the grammaticality–likelihood distinction in LMs. While this approach ostensibly mirrors how we might extract grammaticality judgments from adult native speakers, Hu and Levy (2023) observe that metalinguistic judgments can underestimate a model's linguistic capability. Moreover, this type of assessment assumes that an LM is able to perform basic question-answering out of the box, which may not be the case for many LMs of interest (e.g., smaller LMs that have not been instruction-tuned). Hence, this work does not use it as a major baseline.

Perturbation	Original	Perturbed
Insertion	Analysts agreed. "They had to do it." She bursts into tears and walks away.	Democratization Analysts constructed agreed. "They had to Krenz do it." She bursts into tears fringe and walks away.
Deletion	What is his usual food, Nelly? Be a good lad; and I'll do for you. They allowed this country to be credible.	What his , Nelly? Good lad; I'll you. They allowed to be credible.
Local shuffle	It was jolly of you to make up your mind to come. Now we shall have some discussion, we'll see to that. Have you been reading Spencer?	It was of you to make jolly up your mind to come. Now have discussion shall we some, we'll see to that. you reading Spencer Have been?

Table 1: Original sentences and their perturbed counterparts for three perturbation types.

internal representations of LMs may encode a notion of grammaticality that is distinct from string probabilities.[3]

2 A Grammaticality Judgment Probe

To explore whether grammaticality is implicitly encoded in the hidden states of LMs, we train a probe on a dataset of (mostly) grammatical and (mostly) ungrammatical sentences obtained from applying noise to a generic text corpus.

2.1 Generation of Contrastive Training Set

The training set of the probe comprises sentences with synthetically-derived binary acceptability labels. We sample 50,000 sentences from the Penn Treebank (Marcus et al., 1994) and Project Gutenberg (GenRM, 2025) to form a generic text dataset as grammatical examples. To generate negative ungrammatical examples, we use three simple perturbation functions based on prior work (Cotterell et al., 2018; Cao et al., 2020; Mitchell and Bowers, 2020; Kallini et al., 2024):

- **Insertion:** inserts 1–5 random tokens from the entire corpus at random position in the sentence.
- **Deletion:** randomly removes 1–5 text tokens.
- **Local shuffle:** randomly permutes a contiguous 5-token window.

See Table 1 for examples of perturbations and §A for generation details. We perturb every sentence with a (uniformly) randomly chosen noising function. Sentences that cannot be perturbed are filtered out (e.g., three-word sentences assigned to local shuffle). We label originals as grammatical and perturbed sentences as ungrammatical. 80% of the data is used as train set and 20% as dev set for hyperparameter tuning.

The data collection is not perfect: some original sentences may be ungrammatical, and our perturbations may introduce semantic implausibility instead of syntactic errors. To validate our data generation pipeline, we use an LLM to judge the sentence acceptability of 5,000 randomly sampled sentence pairs, confirming that 93.72% of the perturbed subset is indeed ungrammatical (§A). Furthermore, natural ungrammatical sentences are rarely simple insertion, deletion, or shuffle variants of grammatical text. Demonstrating that a synthetically-trained probe can generalize to real grammaticality benchmarks would thus be a significant result.

2.2 Linear Probe

Given an LM $\mathcal{M}$ and a sentence-label pair (s_i, y_i) with sentence $s_i \in \Sigma^*$ and label $y_i \in \{0, 1\}$, let $h_i := \mathcal{M}(s_i)$ be the hidden states taken at the last token (generally a punctuation token) from an LM $\mathcal{M}$ applied on sentence s_i. Here h_i is either the hidden states of a single layer or all layers concatenated, a design choice we will specify in the experimental setup. We train a logistic classifier with ℓ_2 regularization:

$$-\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|s_i) + \alpha\|w\|_2^2. \quad (1)$$

where $\log p(y_i|s_i)$ is $y_i \log(\sigma(w \cdot h_i + b)) + (1 - y_i)\log(1 - \sigma(w \cdot h_i + b))$ and α is tuned as a hyperparameter on the dev set. We also experiment with ℓ_1 regularization in our later experiments.

3 Experimental Setup

3.1 Models

Our experiments use the following open-weight base models: OLMo-2-7B (OLMo et al., 2025)

[3] Of course, there still remain many unaddressed arguments against LMs as models of human language processing.

and OLMo-3-7B (Olmo et al., 2025), Llama-3.2-1B and Llama-3.1-8B (Grattafiori et al., 2024), and Gemma-2-2B and Gemma-2-9B (Gemma Team, 2024). We do not evaluate on instruction-tuned models because instruction tuning alters the next-token distribution in task- and alignment-specific ways, which would confound our analysis.

3.2 Evaluation Metrics

Let $f(s)$ be a score for sentence s obtained from the probe or (as our baseline) the string probabilities from an LM.[4] We consider two metrics, depending on whether the benchmark is based on minimal pairs or not:

Minimal pairs. Given a grammaticality judgment benchmark such as BLiMP which provides pairs of sentences (s_i, s_i') that differ from one another minimally, we compute the accuracy (**ACC**):

$$ACC = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(f(s_i) > f(s_i')).$$

where s_i is the grammatical one.

Standalone acceptability. On benchmarks such as CoLA which just provide a sentence and its label (s_i, y_i), using pure accuracy would require setting a threshold on $f(s)$ for deciding when something is considered grammatical or not. For a binary classifier, a natural threshold would be 0.5, but the threshold for LM probabilities can be challenging to determine. Accuracy is also not always appropriate in cases where the positive rate differs from 50%, which is indeed the case in some of our benchmarks. We thus instead compute area under the ROC curve (**AUC**):

$$AUC = \frac{1}{n_+ n_-}\sum_{s\in P}\sum_{s'\in N}\Big(\mathbb{1}\left(f(s) > f(s')\right) + \frac{1}{2}\mathbb{1}\left(f(s) = f(s')\right)\Big).$$

Here $P = \{s_i : y_i = 1\}$ and $N = \{s_i : y_i = 0\}$, and we further have $n_+ = |P|$ and $n_- = |N|$. Intuitively, AUC quantifies how consistently a random acceptable s sentence receives a higher score than a random unacceptable sentence s', and provides a more granular window into a classifier's performance than pure accuracy. We compute AUC on minimal-pair benchmarks as well.

[4]For string probabilities we use the length-normalized cumulative logprobs, $\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(w_t|w_{<t})$, which was found to perform better than cumulative probabilities.

3.3 Evaluation Benchmarks

Our primary experiments test on three acceptability datasets in English: BLiMP, CoLA, and SyntaxGym (Warstadt et al., 2020, 2019; Gauthier et al., 2020). For cross-lingual generalization, we also test on six multilingual acceptability sets: Swedish (ScaLA (sv)) (Nielsen, 2023), Dutch (BLiMP-NL) (Suijkerbuijk et al., 2025), Italian (ItaCoLA) (Trotta et al., 2021), Russian (RuCoLA) (Mikhailov et al., 2022), Japanese (JCoLA) (Someya et al., 2024), and Chinese (SLING) (Song et al., 2022). We compute the evaluation metric ACC on minimal-pair data only: BLiMP, BLiMP-NL, and SLING. All other datasets are single sentences with binary acceptability labels, for which we only calculate AUC. Dataset statistics are listed in Table 5 and examples from the different benchmarks are shown in Table 8 of the appendix.

In addition to benchmarks which primarily test for grammaticality, we also evaluate our probe on three semantic plausibility benchmarks collectively referred to as "plausibility sets" (Kauf et al., 2023; Fedorenko et al., 2020; Vassallo et al., 2018; Ivanova et al., 2021): see Table 7 of the appendix for details. LM probabilities entangle grammaticality with likelihood, and thus their performance on such benchmarks have been shown to be quite high (Leivada et al., 2025; Hu et al., 2025). A grammaticality probe that is mostly sensitive to syntax should ideally have low performance on these plausibility benchmarks.

4 Implicit Representations of Grammaticality in Language Models

Can probes trained on unsupervised, synthetic sentence pairs generalize to out-of-distribution human-labeled linguistic acceptability benchmarks? For our main results, we train grammaticality probes layer-by-layer and select the best layers as well as the regularization parameters α based on the synthetic dev set (see §D for training details).[5]

As shown in Figure 2a, our probes surpass the performance of probability-based method in both the minimal-pair setting (BLiMP) and the unpaired setting (BLiMP, CoLA, SyntaxGym), suggesting that grammaticality to an extent is implicitly captured by the hidden states of LMs. See Table 10 in

[5]We find that the middle layers generally perform best. The performance by layer for all models is shown in Figure 6 of the appendix. Figure 7 compares the performance of selected ℓ_2-probes on the held-out dev set against LM logprob.

the appendix for results broken down by different grammaticality judgment categories.[6]

4.1 Grammaticality or Semantic Plausibility?

One reason why string probabilities are inappropriate for adjudicating grammaticality is that they entangle grammaticality with other aspects of language such as meaning (Hu et al., 2025). Such entanglement means that string probabilities can distinguish between semantically plausible/implausible pairs quite well, often better than grammatical/ungrammatical pairs (Leivada et al., 2025; Kauf et al., 2024). Is our probe also similarly entangling grammaticality with plausibility? The results of our probe on the semantic plausibility benchmarks, shown in Figure 2b, suggest otherwise. We find that our probe's performance on semantic plausibility sets is substantially lower than string probabilities, suggesting that the LMs have potentially learned grammaticality representations that are distinct from semantic plausibility.

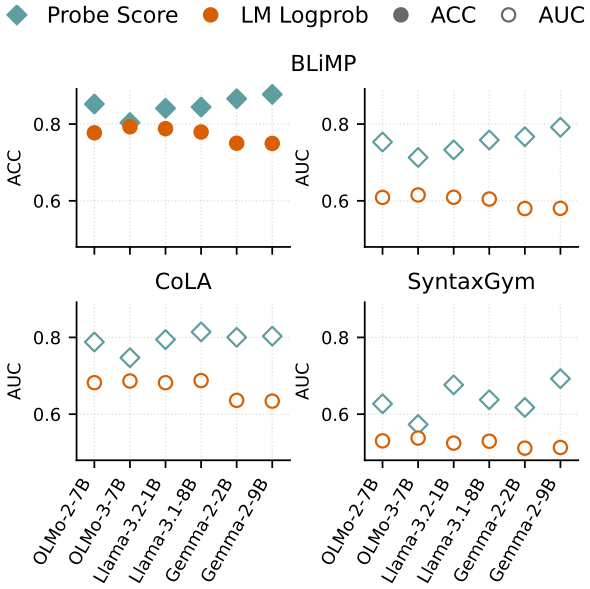
4.2 Cross-lingual Generalization

Given that LMs are implicitly capturing grammaticality within the hidden layers, are these features specialized to English, or are they language-agnostic? We next test whether our probes, trained on English data, zero-shot generalize to other languages. We study languages spanning multiple families, ranging from those typologically closer to English (Swedish and Dutch) to more distantly related ones (Italian and Russian), and finally to typologically distant languages (Japanese and Chinese). The results for Llama and Gemma models are listed in Table 2.[7] Overall, the English-trained probes exhibit nontrivial cross-lingual generalization, and generally outperforms string probability-based grammaticality judgments across languages.

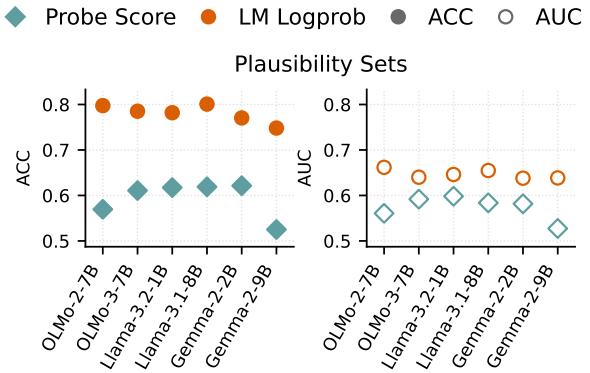
We further test the probes trained on non-English data and find less impressive transfer results, which we report in §M. This observation could potentially be explained by the disproportionate scale of English relative to other languages in the pretraining data, and confirms previous findings of an English-dominant representation space in pretrained models (Wendler et al., 2024).

[6] As a supervised baseline we also train probes on BLiMP and evaluate them on CoLA, SyntaxGym, and our synthetic data. The probe trained on specific violations (agreement errors, island effects, etc.) successfully detects crude perturbations (insertions, deletions, shuffles). See §E.

[7] We do not test OLMo models on multilingual sets since they have limited multilingual capabilities.



(a) **Probe scores outperform LM string logprob baselines on grammaticality acceptability judgment datasets** (ACC and AUC on BLiMP; AUC on all datasets).



(b) Probe scores underperform LM logprob on semantic plausibility benchmarks, suggesting that **probes are more selectively tuned to syntactic acceptability**.

Figure 2: Comparing probe scores and string logprob baselines on grammaticality and semantic plausibility benchmarks. We also compute their 95% confidence intervals in the appendix (§G).

4.3 Localization to Select Neurons

We have observed that grammaticality judgments can be probed out from LM representations and can generalize across languages. Is this phenomenon represented in a distributed manner, or can it be localized to a small number of neurons? We train an ℓ_1-regularized linear probe (i.e., LASSO (Tibshirani, 1996)) on the concatenation of the hidden states from all layers, where we vary the regularization strength until we reach the desired percentage of nonzero weights. We target nonzero rates of $\{0.01\%, 0.05\%, 0.1\%, 0.5\%\}$. We then retrain ℓ_2-regularized probes on top of the LASSO-selected neurons (see §D for details). As a baseline, we also

Model	Method	Swedish	Dutch		Italian	Russian	Japanese	Chinese	
		ScaLA (sv)	BLiMP-NL		ItaCoLA	RuCoLA	JCoLA	SLING	
		AUC	AUC	ACC	AUC	AUC	AUC	AUC	ACC
Llama-3.2-1B	LM Logprob	0.62	0.58	**0.75**	0.58	0.44	0.57	0.55	0.59
	Probe Score	**0.68**	**0.59**	0.69	**0.59**	**0.59**	0.57	**0.57**	**0.63**
Llama-3.1-8B	LM Logprob	0.65	0.62	**0.84**	0.61	0.46	0.59	0.57	0.65
	Probe Score	**0.74**	**0.65**	0.77	**0.65**	**0.58**	**0.63**	**0.60**	**0.69**
Gemma-2-2B	LM Logprob	0.64	0.61	**0.82**	0.55	0.46	0.59	0.57	0.61
	Probe Score	**0.69**	**0.62**	0.73	**0.62**	**0.61**	**0.61**	**0.63**	**0.73**
Gemma-2-9B	LM Logprob	0.66	0.63	**0.86**	0.56	0.47	0.59	0.58	0.62
	Probe Score	**0.83**	**0.73**	0.84	**0.70**	**0.65**	**0.70**	**0.67**	**0.75**

Table 2: Evaluation results of grammaticality probes and LM logprob on multilingual acceptability benchmarks, where the grammaticality probes are trained on synthetic English training data and tested zero-shot on the benchmark in each language. **The probes generally match or surpass the performance of logprob on syntactic acceptability judgments across languages.** Bold numbers indicate better performance.

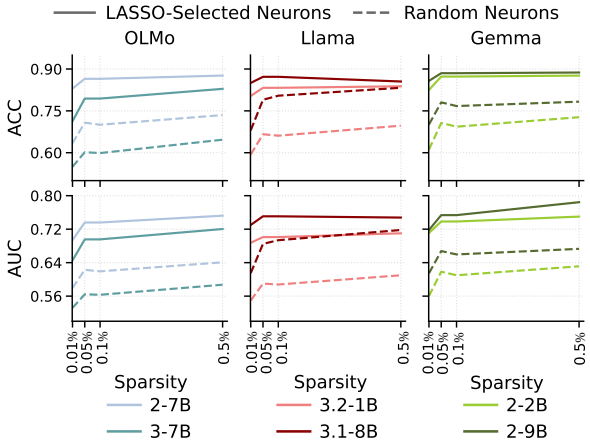


Figure 3: The evaluation results of LASSO probes on BLiMP. **Grammaticality signals are captured by very small (even random) subsets of neurons.** The performance gap between LASSO-selected and random neurons increases at higher sparsity.

train ℓ_2 probes on random subsets of neurons of the same size across 30 seeds and report the average performance.

The nontrivial performance of the random-neuron probes suggests that grammaticality signals are distributed across most layers (consistent with the by-layer ℓ_2 probe in Figure 6). Nevertheless, a smaller subset of neurons carries a disproportionately rich share of this signal. We visualize the results on BLiMP here, and the test results for the remaining benchmarks are shown in Figure 13 and Figure 14. As shown in Figure 3, even with 0.01% of the neurons (which corresponds to about 10 neurons; see neuron counts in Figure 9) the probes work well, suggesting that grammaticality can potentially be localized to a handful of neurons.

5 Revisiting Grammaticality vs. String Probabilities

In the previous section (§4), we showed that grammaticality probes outperform string probabilities at predicting grammaticality. Are these scores distinct from string probabilities? If so, then this may provide the basis for a grammaticality–likelihood distinction in LMs.

As a first step, we visualize the distribution of LM logprob and probe scores for BLiMP in Figure 4, where we show the probe scores in probability space for easier visualiation. The distributions for the other two English benchmarks are shown in Figure 10 and Figure 11. As noted in previous work (Leivada et al., 2025), LM logprob fails to reliably distinguish grammatical from ungrammatical sentences; the grammaticality probe scores however achieve relatively good separation. We then compute the Spearman's correlation between LM logprob and log probe scores (Table 3). We observe only a moderate correlation, which further suggests that the probe scores are capturing distinct information and are not merely a reparameterization of the model's output probabilities.[8]

The observations above motivate a natural question: do string probabilities contain useful extra signal for acceptability beyond the probes, or is the logprob information already implicit in the representations the probe uses?

[8]We also compute the Pearson's correlation between the length normalized logprob and log probe scores, whose results are in Table 14, confirming that the two variables are not strongly linearly correlated.

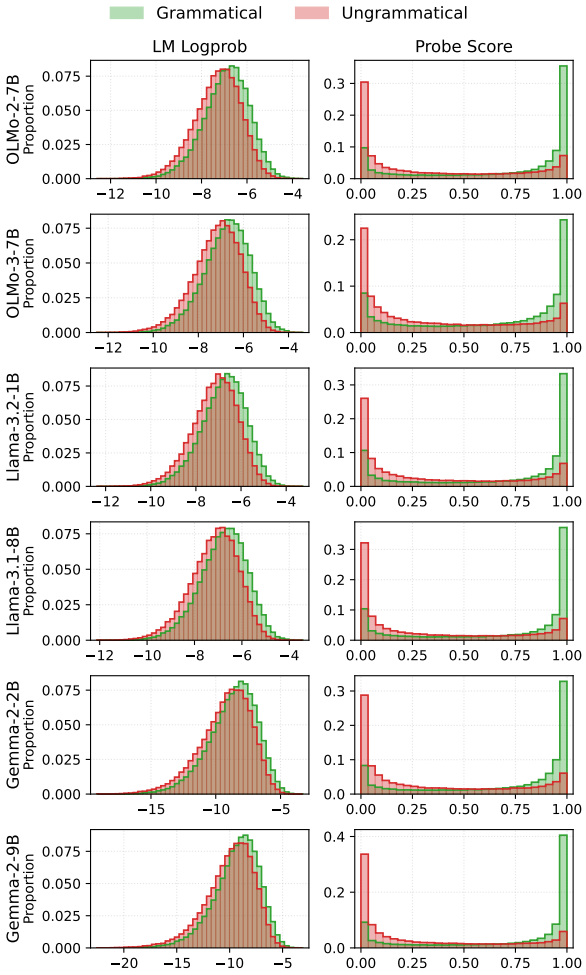


Figure 4: Distributions of LM logprob and probe scores on BLiMP. **Probe scores better separate grammatical and ungrammatical sentences than LM logprob.**

Models	Corr. (logprob, probe score) BLiMP	CoLA	SyntaxGym
Llama-3.2-1B	0.31	0.40	0.089
Llama-3.1-8B	0.27	0.43	0.28
Gemma-2-2B	0.27	0.35	0.24
Gemma-2-9B	0.23	0.37	0.28
OLMo-2-7B	0.23	0.34	0.30
OLMo-3-7B	0.30	0.47	0.23

Table 3: Spearman's correlation evaluates the extent to which two variables exhibit a monotonic relationship. **Correlations between logprob and probe scores are only moderate, suggesting that a higher LM logprob does not always correspond to a higher probe score.**

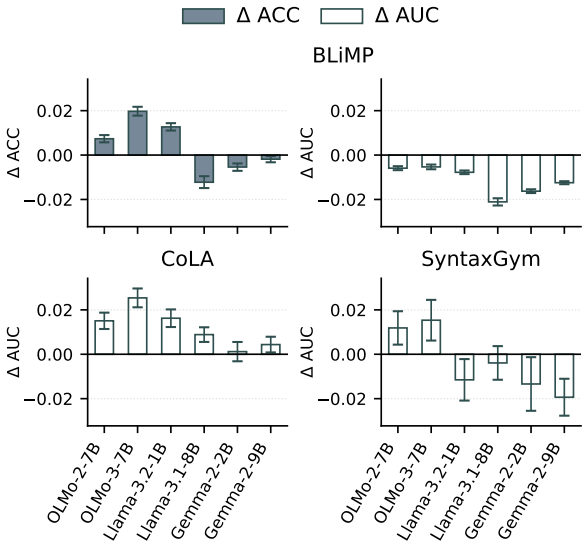


Figure 5: Performance difference (Δ) between ℓ_2 probes augmented with logprob as an input feature and the baseline ℓ_2 probes. Error bars denote 95% confidence intervals. **The string probabilities add little complementary signals to original probe scores.**

5.1 String Probabilities as Additional Features in the Probe

We train a probe with the same setup as in the previous section but with logprob added as an extra feature (see §D for details). Figure 5 reports the performance change when augmenting the ℓ_2 probe with length-normalized logprob (Δ = augmented − baseline). Across BLiMP, CoLA, and SyntaxGym. We do not observe a consistent improvement across model families or datasets. Overall, these results suggest that probabilities provide little complementary signal once the probe already has access to the selected-layer hidden states.

5.2 Probing for String Probabilities

The results of probes trained with logprobs as an extra predictor suggest that likelihood-related information that is relevant for grammaticality may already be encoded in the representations used by the probe. We now move on to a more direct test: can logprobs be recovered from the hidden states?

We investigate this under two setups: a *per-token* approach utilizing the hidden state at each time step, and a *last-token-only* approach using the last-token hidden state. Both setups predict the length-normalized cumulative log-probability. The ridge regression probe is trained on non-perturbed sentences from the generic text corpus. For evaluation, we pool BLiMP, CoLA, and SyntaxGym into a single held-out dataset (see §D).

As shown in Table 4, ridge regression achieves moderately high R^2 when predicting prefix logprob from per-token hidden states and the last-toke hidden states. The gap between the two setups may partially be explained by the lower variance of the length-normalized prefix logprob at the final token

Models	Hidden states → logprob (R^2)	
	Per Token	Last Token Only
Llama-3.2-1B	0.55	0.59
Llama-3.1-8B	0.67	0.65
Gemma-2-2B	0.73	0.68
Gemma-2-9B	0.56	0.66
OLMo-2-7B	0.59	0.58
OLMo-3-7B	0.69	0.51

Table 4: Ridge regression R^2 for predicting length-normalized cumulative logprob from token-level hidden states. The per-token setups consider all time steps, while the last-token-only setup only uses final tokens. **The moderately strong R^2 confirms that substantial information of prefix logprob can be recovered from token-level hidden states.**

versus across all tokens (§ L), which can naturally limit the achievable R^2. The strong out-of-domain performance of the surprisal probe suggests that token-level hidden states retain substantial information about cumulative string probabilities.

6 Related Work

LMs as models of human linguistic cognition. Skeptics of LMs as models of human language learning argue that LMs statistically mimic behaviors but do not possess the underlying structural rules of human cognition (Bender et al., 2021; Katzir, 2023; Cowen, 2023; Chomsky et al., 2023). Proponents, on the other hand, suggest that though LMs are not a complete theory of language, they nonetheless offer rich insights into the science of language (Portelance and Jasbi, 2024; Futrell and Mahowald, 2025).

Central to the debate is inductive bias, i.e., priors brought by the learning systems beyond the training data distribution (Mitchell, 1980; Goyal and Bengio, 2022). Existing work has broadly investigated the inductive bias of LMs in learning syntactic principles, unnatural word orders, and (human) impossible languages (Wilcox et al., 2018; Mitchell and Bowers, 2020; Kallini et al., 2024; Xu et al., 2025). In the same spirit, we treat LMs as a useful testbed for studying human-like linguistic behaviors and investigate the extent to which their representations encode syntactic knowledge.

LM acceptability judgments. Can language models trained with distributional objectives achieve human-level linguistic competence without strong linguistic inductive biases? One way to assess LM syntactic knowledge is metalinguistic acceptability judgment: given a sentence, models can be fine-tuned or few-shot prompted to determine if it is grammatically acceptable (Warstadt et al., 2019). This approach is straightforward but assumes LMs faithfully follow instructions and have metalinguistic knowledge of what grammaticality means. Though metalinguistic prompting is not a major baseline in this work, we obtain and present the relevant results in §N.

An alternative is to use LM string probability as a proxy for acceptability scores. Marvin and Linzen (2018) introduced minimal pairs—grammatical/ungrammatical sentences differing by small, targeted syntactic modifications—to test whether LMs assign higher probability to the grammatical variant, an approach now known as targeted syntactic evaluation (Warstadt et al., 2020; Jumelet et al., 2025). However, probability-based acceptability scoring is sensitive to non-syntactic factors (e.g., input length, unigram frequency) and degrades outside pairwise comparisons (Sinha et al., 2023; Tjuatja et al., 2025; Hu et al., 2025). To address these limitations, our work shifts from using string probability to computing acceptability scores from LM hidden states.

Representations of syntactic information. One way to study the internal mechanisms of LMs is probing: training a low-capacity model (often linear or a small multilayer perceptron) to predict supervised-task features from an network's hidden states (Alain and Bengio, 2017; Belinkov, 2022). LMs have been shown to acquire nontrivial abstractions of syntactic dependencies, part-of-speech tags, parse trees, and incremental parse states, etc. (Lepori and McCoy, 2020; Belinkov et al., 2017; Hewitt and Manning, 2019; Eisape et al., 2022). Though non-linear probes enable more flexible decoding of target features (White et al., 2021; Chen et al., 2021), most prior work assumes that linguistic structures are linearly decodable from LM hidden states (Park et al., 2025). Adopting this premise, we train linear probes to demonstrate that *grammaticality*, a sentence-level syntactic property, is linearly extractable from LM representations.

7 Conclusion

We propose a simple approach to probe for representations of grammaticality in pretrained LMs. The trained probes selectively attend to grammaticality instead of semantic plausibility. We observe

zero-shot cross-lingual transfer: a probe trained exclusively on English generalizes to a broad spectrum of typologically diverse languages. Finally, we find that that probe scores do not simply recapture probability. These results collectively suggest that despite being trained without built-in linguistic knowledge, LMs learn nontrivial representations of grammaticality that is distinct from output likelihoods. With the acknowledgment that LMs are fundamentally different systems from humans, our work invites further investigation into potential usages of LMs as models for linguistic processing and learning.

Limitations

Our synthetic data creation is inherently limited. While we label the generic texts as grammatical, it is possible that they have built-in grammar errors. On the other hand, our synthetic perturbations may produce semantically implausible sentences rather than ungrammatical ones. Therefore, our training set are only pairs of *mostly* grammatical/ungrammatical sentences.

Our work mainly concerns the notion of grammaticality in natural languages. It is likely that LMs also implicitly encode the syntactic acceptability in formal languages, but testing this hypothesis is outside the scope of this work. Any theory work in formal language learning is also beyond the empirical nature of this work.

Our work only examines the representation of LMs at their last checkpoints and does not include any learning-dynamics results to study *when* LMs develop the notion of grammaticality. Future work could investigate if grammaticality can be learned from human-plausible amount of training data.

Ethics Statement

Although we draw analogies to human cognition (namely the competence–performance gap) and discuss the implications of LMs for linguistic cognition, we refrain from making claims about human language acquisition. This work restricts its scope to LMs as models of linguistic theories. Any conclusion of human linguistic processing derived from this work is unwarranted and strongly advised against

Acknowledgements

This study was supported in part by the MIT-IBM Watson AI Lab. We thank the reviewers for the thorough reviews and constructive feedback. We thank Heidi Lei and Xiaoman Delores Ding for their constructive comments. Yingshan Susan Wang was funded by the MIT Undergraduate Research Opportunity Program.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

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

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive Psychology*, 23(1):45–93.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4798–4808, Online. Association for Computational Linguistics.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing BERT in hyperbolic spaces. In *International Conference on Learning Representations*.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. The false promise of ChatGPT. *The New York Times*. Opinion.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, Louisiana. Association for Computational Linguistics.

Tyler Cowen. 2023. Noam chomsky on language, left libertarianism, and progress. Conversations with Tyler, Episode 182. Podcast.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Evelina Fedorenko, Idan Asher Blank, Matthew Siegelman, and Zachary Mineroff. 2020. Lack of selectivity for syntax relative to word meanings throughout the language network. *Cognition*, 203:104348.

Danny Fox and Roni Katzir. 2024. Large language models and theoretical linguistics. *Theoretical Linguistics*, 50:71 – 76.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *Behavioral and Brain Sciences*, page 1–98.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

GenRM. 2025. Gutenberg DPO (gutenberg-dpo-v0.1-jondurbin). Hugging Face Datasets. License: CC BY 4.0. Duplicated from `jondurbin/gutenberg-dpo-v0.1`. Accessed: 2026-01-04.

Anirudh Goyal and Yoshua Bengio. 2022. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266):20210068.

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.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jennifer Hu and Roger P. Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jennifer Hu, Ethan Gotlieb Wilcox, Siyuan Song, Kyle Mahowald, and Roger P. Levy. 2025. What can string probability tell us about grammaticality? *Preprint*, arXiv:2510.16227.

Anna A. Ivanova, Zachary Mineroff, Vitor Zimmerer, Nancy Kanwisher, Rosemary Varley, and Evelina Fedorenko. 2021. The language network is recruited but not required for nonverbal event semantics. *Neurobiology of Language*, 2(2):176–201. PMID: 37216147; PMCID: PMC10158592.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2025. Multiblimp 1.0: A massively multilingual benchmark of linguistic minimal pairs. *Preprint*, arXiv:2504.02768.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Roni Katzir. 2023. Why large language models are poor theories of human linguistic cognition: A reply to piantadosi. *Biolinguistics*.

Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna A Ivanova. 2024. Log probabilities are a reliable estimate of semantic plausibility in base and instruction-tuned language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 263–277, Miami, Florida, US. Association for Computational Linguistics.

Carina Kauf, Anna A. Ivanova, Giulia Rambelli, Emmanuele Chersoni, Jingyuan Selena She, Zawad Chowdhury, Evelina Fedorenko, and Alessandro Lenci. 2023. Event knowledge in large language models: The gap between the impossible and the unlikely. *Cognitive Science*, 47(11):e13386.

Evelina Leivada, Vittoria Dentella, and Fritz Günther. 2024a. Evaluating the language abilities of large

language models vs. humans: Three caveats. *Biolinguistics*.

Evelina Leivada, Fritz Günther, and Vittoria Dentella. 2024b. Reply to hu et al.: Applying different evaluation standards to humans vs. large language models overestimates ai performance. *Proceedings of the National Academy of Sciences of the United States of America*, 121:e2406752121.

Evelina Leivada, Raquel Montero, Paolo Morosi, Natalia Moskvina, Tamara Serrano, Marcel Aguilar, and Fritz Günther. 2025. Large language model probabilities cannot distinguish between possible and impossible language.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roger Levy, Yoon Kim, and Danny Fox. 2025. The science of language in the era of generative ai. *An MIT Exploration of Generative AI*.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Vladislav Mikhailov, Tatiana Shamardina, Max Ryabinin, Alena Pestova, Ivan Smurov, and Ekaterina Artemova. 2022. RuCoLA: Russian corpus of linguistic acceptability. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5207–5227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Raphaël Millière. 2024. Language models as models of language.

Jeff Mitchell and Jeffrey Bowers. 2020. Priorless recurrent networks learn curiously. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5147–5158, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tom M. Mitchell. 1980. The need for biases in learning generalizations. Technical report, Rutgers University, New Brunswick, NJ.

Dan Nielsen. 2023. ScandEval: A benchmark for Scandinavian natural language processing. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 185–201, Tórshavn, Faroe Islands. University of Tartu Library.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2025. Olmo 3. *Preprint*, arXiv:2512.13961.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 24 others. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*.

Steven T Piantadosi. 2023. Modern language models refute chomsky's approach to language. *From fieldwork to linguistic theory: A tribute to Dan Everett*, 15:353–414.

Eva Portelance and Masoud Jasbi. 2024. The roles of neural networks in language acquisition. *Language and Linguistics Compass*, 18(6):e70001.

Koustuv Sinha, Jon Gauthier, Aaron Mueller, Kanishka Misra, Keren Fuentes, Roger Levy, and Adina Williams. 2023. Language model acceptability judgements are not always robust to context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6043–6063, Toronto, Canada. Association for Computational Linguistics.

Taiga Someya, Yushi Sugimoto, and Yohei Oseki. 2024. JCoLA: Japanese corpus of linguistic acceptability. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9477–9488, Torino, Italia. ELRA and ICCL.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, and Mohit Iyyer. 2022. SLING: Sino linguistic evaluation of large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4606–4634, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michelle Suijkerbuijk, Zoë Prins, Marianne de Heer Kloots, Willem Zuidema, and Stefan L. Frank. 2025. Blimp-nl: A corpus of dutch minimal pairs and acceptability judgments for language model evaluation. *Computational Linguistics*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society series b-methodological*, 58:267–288.

Lindia Tjuatja, Graham Neubig, Tal Linzen, and Sophie Hao. 2025. What goes into a LM acceptability judgment? rethinking the impact of frequency and length. 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 2173–2186, Albuquerque, New Mexico. Association for Computational Linguistics.

Daniela Trotta, Raffaele Guarasci, Elisa Leonardelli, and Sara Tonelli. 2021. Monolingual and cross-lingual acceptability judgments with the italian cola corpus. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Paolo Vassallo, Emmanuele Chersoni, Enrico Santus, Alessandro Lenci, and Philippe Blache. 2018. Event Knowledge in Sentence Processing: A New Dataset for the Evaluation of Argument Typicality. In *LREC 2018 Workshop on Linguistic and Neurocognitive Resources (LiNCR)*, Miyazaki, Japan.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. 2021. A non-linear structural probe. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 132–138, Online. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. 2025. Can language models learn typologically implausible languages? *Preprint*, arXiv:2502.12317.

A Contrastive Training Sets

Generic text corpus. To build a generic English text corpus, we combine raw text from two sources: (i) Penn Treebank (PTB; Wall Street Journal text-only splits) (Marcus et al., 1994), and (ii) the `GenRM/gutenberg-dpo-v0.1-jondurbin` dataset derived from Project Gutenberg books (GenRM, 2025). For the Gutenberg Hugging Face dataset, we use only the `chosen` field, which contains the original (human-written) book chapter texts.

Sentence extraction and sampling. We pool all strings (contiguous text blocks) from PTB and Gutenberg and shuffle the strings to interleave inputs from the two domains. We then segment each string into sentences using the PySBD segmenter. If a text block contains quoted spans, we extract the texts inside the quotes and re-segment them (discarding the non-quoted portions). We keep the first 50,000 segmented sentences.

Vocabulary construction. We construct a vocabulary for the `insertion` perturbation. Given the 50,000 sentences, we extract tokens using RegEx to match contiguous alphabetic spans (`[A-Za-z]+`) and map the found tokens to a Python set.

Insertion. The first $\lfloor 50,000/3 \rfloor$ sentences are chosen for `insertion`. We tokenize each sentence with spaCy's English tokenization. We then sample a number $k \in \{1, \ldots, 5\}$, choose k random token boundaries in the sentence (including sentence beginning and end), and insert k vocabulary tokens sampled uniformly with replacement.

Deletion. The next $\lfloor 50{,}000/3 \rfloor$ sentences are assigned to `deletion` and tokenized with spaCy. For each sentence, we sample $k \in \{1, \ldots, 5\}$, delete k uniformly chosen alphabetic tokens (`token.is_alpha`) if possible and skip otherwise.

Local shuffle. The remaining sentences are assigned to `local shuffle` and tokenized with spaCy. For each sentence with at least 5 tokens, we select a random contiguous 5-token window and randomly permute those 5 tokens. If the sentences have fewer than 5 tokens, we skip them.

Finally, we return all the perturbed sentences with their original counterparts in pairs.

Validation of Synthetic Data. We used Claude Opus 4.6 as an acceptability judge on a random sample of 5,000 original and 5,000 perturbed sentences from the synthetic dataset:

- Original ("good") sentences: 93.72% judged acceptable.
- Perturbed ("bad") sentences: 6.28% judged acceptable.

The LLM-as-judge results validate that our perturbation pipeline reliably geneates ungrammatical sentences.

B Evaluation Datasets

The specific statistics of syntactic acceptability evaluation datasets are listed in Table 5. All of them are publicly available on Hugging Face.

SyntaxGym is grouped by sentence conditions. We label conditions as acceptable vs. unacceptable using:

> **Acceptable:** np_match, vp_match, that_nogap, what_subjgap, what_gap, neg_pos, neg_neg, match_sing, match_plural, no-sub_no-matrix, sub_matrix.
> **Unacceptable:** np_mismatch, vp_mismatch, what_nogap, that_subjgap, what_matrixgap, that_matrixgap, that_gap, pos_pos, pos_neg, mismatch_sing, mismatch_plural, sub_no-matrix, no-sub_matrix.

All semantic plausibility benchmarks come from Kauf et al. (2023), and its three datasets are adapted from previous studies (Fedorenko et al., 2020; Vassallo et al., 2018; Ivanova et al., 2021). See Table 7 for relevant information.

C Language Model Details

All inference is run on local H100 and A100 clusters using hugginngface APIs. The model-specific statistics are given in Table C.

D Probe Training Details

All logistic probes are trained using SnapML's `LogisticRegression` (GPU backend), with `dual=False`, `max_iter=1000`, `random_state=0`, `fit_intercept=True`. All ridge regressions are trained with SnapML's `LinearRegression` (GPU backend) with `fit_intercept=True`. All probing experiments could be completed in fewer than 15 hours, with the LASSO probe's neuron-selection process accounting for the majority (about 8 hours).

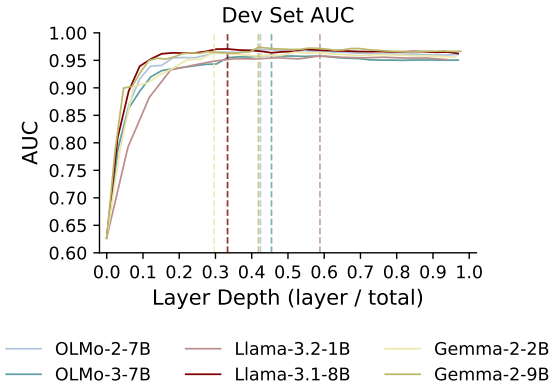


Figure 6: By-layer ℓ_2-probe AUC on the dev set. Vertical lines indicate the best layers selected for evaluation (Llama-3.2-1B: 10, Llama-3.1-8B: 11, Gemma-2-2B: 8, Gemma-2-9B: 18, OLMo-2-7B: 14, OLMo-3-7B: 15).

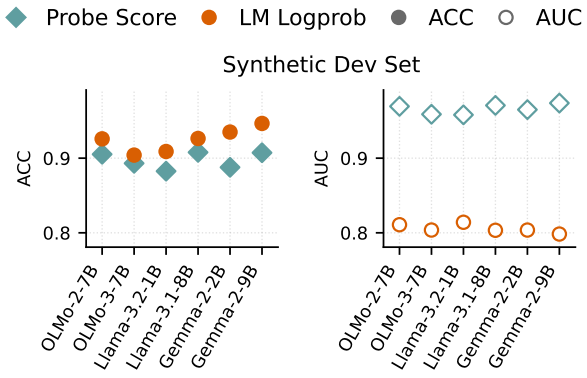


Figure 7: Comparing the performance of the best ℓ_2-probes and LM Logprob on heldout in-domain dev set.

ℓ_2 probe. For each layer, we train an ℓ_2-regularized logistic regression probe on the last-token hidden-state vector, using an 80/20 train–dev split with feature normalization. We preprocess the hidden states by computing per-dimension mean and standard deviation on the training split and z-scoring both train and dev features with these statistics. We sweep the regularization strength over a log-scale grid, $\alpha \in \{2^s, 2^{s+1}, \ldots, 2^e\}$ for $s = -2$, $e = 5$, and select the best hyperparameter per layer by maximizing validation AUC. We choose the layer whose probe achieves the highest AUC on the dev set (Figure 6).

LASSO probe. We train an ℓ_1-regularized logistic regression probe (LASSO) on concatenated last-token hidden states of all model layers. We tune ℓ_1 penalty strength to achieve a target sparsity level. For each target fraction $p \in \{0.01\%, 0.05\%, 0.1\%, 0.5\%\}$, we set the desired number of active neurons to $k = \lceil pD \rceil$ where D is the total num-

Dataset	Language	# Sentences	% Grammatical	Minimal Pairs	References
BLiMP	English	134,000	50.0	✓	Warstadt et al. (2020)
CoLA	English	10,657	70.4	✗	Warstadt et al. (2019)
SyntaxGym	English	2,412	49.0	✗	Gauthier et al. (2020)
ScaLA (sv)	Swedish	10,762	50.0	✗	Nielsen (2023)
BLiMP-NL	Dutch	18,000	50.0	✓	Suijkerbuijk et al. (2025)
ItaCoLA	Italian	8,776	84.4	✗	Trotta et al. (2021)
RuCoLA	Russian	13,445	57.6	✗	Mikhailov et al. (2022)
JCoLA	Japanese	9,154	81.9	✗	Someya et al. (2024)
SLING	Chinese	80,000	50.0	✓	Song et al. (2022)

Table 5: Overview of linguistic acceptability datasets for evaluation.

Model	Hugging Face ID	Hidden Dim	# Layers	# Training Tokens
OLMo-2-7B	`allenai/OLMo-2-1124-7B`	4096	32	4T
OLMo-3-7B	`allenai/Olmo-3-1025-7B`	4096	32	6T
Llama-3.2-1B	`meta-llama/Llama-3.2-1B`	2048	16	9T
Llama-3.1-8B	`meta-llama/Llama-3.1-8B`	4096	32	15T
Gemma-2-2B	`google/gemma-2-2b`	2304	26	2T
Gemma-2-9B	`google/gemma-2-9b`	3584	42	8T

Table 6: Statistics of selected OLMo, Llama, and Gemma models.

Dataset	Language	# Sentences	% Plausible	Minimal Pairs	References
Dataset 1	English	1,564	50.0	✓	Kauf et al. (2023); Fedorenko et al. (2020)
Dataset 2	English	790	50.0	✓	Kauf et al. (2023); Vassallo et al. (2018)
Dataset 3	English	76	50.0	✓	Kauf et al. (2023); Ivanova et al. (2021)

Table 7: Overview of semantic plausibility datasets for evaluation.

Dataset	Structure	Condition / Label	Example Sentence
BLiMP	Minimal Pair (Matched)	Acceptable	The cats licked themselves.
		Unacceptable	The cats licked itself.
CoLA	Individual Sentences	Acceptable	The book was written by John.
		Unacceptable	Books were sent to each other by the students.
SyntaxGym	Test Suite (Grouped by Item)	Cond. A	The farmer near the clerks knows many people.
		Cond. B	The farmer near the clerks know many people.
		Cond. C	The farmers near the clerk knows many people.
		Cond. D	The farmers near the clerk know many people.
Plausibility Dataset 1	Minimal Pair (Matched)	Plausible	The actor won the award.
		Implausible	The actor won the battle.

Table 8: Examples from different evaluation datasets.

ber of neurons and adaptively adjust the ℓ_1 penalty strength until the fitted probe has k' non-zero coefficients for $|k - k'| \leq 0.05k$. We then treat the non-zero dimensions as the selected neuron set and refit an ℓ_2-regularized logistic regression probe restricted to these neurons. For this refit, we sweep $\alpha \in \{2^s, 2^{s+1}, \ldots, 2^e\}$ with $s = -2$ and $e = 5$, and select the best α by validation AUC. As a baseline, we repeat the ℓ_2 refit on 30 random subsets of k' neurons and report the average performance.

Probes augmented with probabilities. For each sentence x, we compute the LM's length normalized string probabilities $\tilde{\ell}(x)$. We then concatenate this scalar to the representation vector used by the probe, yielding an augmented feature vector $z = [h; \tilde{\ell}]$, where h denotes the best-layer hidden-state features chosen in §D. We train the linear probe exactly as in §D, treating the probability feature as an additional input dimension.

Probing for probabilities. We train a ridge regression probe to predict the model's token-level probability from hidden states. Concretely, from the forward pass of the *original sentences* in the generic text corpus, we collect pairs $(h_t, \tilde{\ell}_t)$, where h_t is the the best layer hidden states and $\tilde{\ell}_t$ is the length-normalized cumulative logprob at time t. We z-score h_t using per-dimension mean and standard deviation computed on the training split. We use an 80/20 train–dev split and sweep the ℓ_2 penalty over $\alpha \in \{2^s, 2^{s+1}, \ldots, 2^e\}$ with $s = -2$, $e = 5$, selecting the best α by minimizing dev MSE. In addition, we train a last-token-only variant of the probe by restricting the input features and labels to time $t = T$ for each sentence.

E Performance of Supervised Probes

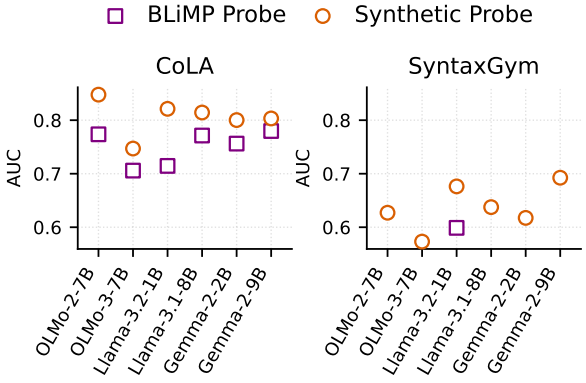


Figure 8: Comparisons of unsupervised (trained on synthetic data) and supervised probes (trained on BLiMP).

	BLiMP Probe → Synthetic Set	
Models	**AUC**	**ACC**
Llama-3.2-1B	0.97	0.99
Llama-3.1-8B	0.98	0.99
Gemma-2-2B	0.97	0.99
Gemma-2-9B	0.98	0.99
OLMo-2-7B	0.98	0.99
OLMo-3-7B	0.97	0.99

Table 9: Probe trained on BLiMP achieves near perfect performance on our synthetic data.

To establish a supervised baseline, we utilize the same selected layers (Figure 6) to train probes on BLiMP, subsequently evaluating them on CoLA, SyntaxGym, and our synthetic data.

F ℓ_2 Probe Performance on BLiMP by Linguistic Term

Linguistic Term	Method	OLMo-2-7B		OLMo-3-7B		Llama-3.2-1B		Llama-3.1-8B		Gemma-2-2B		Gemma-2-9B	
		AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
Anaphor Agreement	LM Logprob	0.69	0.99	0.69	0.99	0.68	0.99	0.69	0.98	0.67	0.99	0.65	0.99
	Probe Score	0.85	0.96	0.81	0.90	0.80	0.95	0.89	0.96	0.78	0.93	0.93	0.98
Argument Structure	LM Logprob	0.60	0.70	0.62	0.73	0.62	0.73	0.61	0.72	0.57	0.66	0.57	0.65
	Probe Score	0.76	0.81	0.74	0.79	0.80	0.85	0.78	0.82	0.80	0.84	0.80	0.85
Binding	LM Logprob	0.62	0.81	0.61	0.78	0.61	0.78	0.62	0.81	0.58	0.78	0.58	0.80
	Probe Score	0.71	0.82	0.65	0.75	0.67	0.80	0.73	0.84	0.69	0.79	0.77	0.87
Control / Raising	LM Logprob	0.64	0.79	0.65	0.81	0.65	0.81	0.64	0.79	0.61	0.76	0.60	0.74
	Probe Score	0.72	0.83	0.70	0.77	0.72	0.83	0.71	0.80	0.77	0.88	0.76	0.86
Determiner-Noun Agreement	LM Logprob	0.67	0.91	0.67	0.93	0.67	0.92	0.66	0.91	0.64	0.93	0.63	0.93
	Probe Score	0.91	0.97	0.81	0.90	0.91	0.96	0.94	0.98	0.97	0.99	0.96	0.99
Ellipsis	LM Logprob	0.63	0.84	0.63	0.87	0.63	0.89	0.63	0.86	0.59	0.84	0.59	0.82
	Probe Score	0.69	0.78	0.63	0.71	0.65	0.77	0.69	0.78	0.64	0.71	0.67	0.74
Filler Gap	LM Logprob	0.58	0.72	0.60	0.77	0.59	0.78	0.57	0.73	0.54	0.69	0.54	0.67
	Probe Score	0.71	0.81	0.68	0.76	0.67	0.76	0.74	0.83	0.75	0.87	0.73	0.83
Irregular Forms	LM Logprob	0.68	0.90	0.73	0.96	0.70	0.92	0.70	0.92	0.66	0.90	0.71	0.96
	Probe Score	0.89	0.98	0.85	0.94	0.87	0.98	0.88	0.97	0.91	1.00	0.92	0.97
Island Effects	LM Logprob	0.61	0.72	0.62	0.72	0.59	0.68	0.59	0.70	0.55	0.64	0.55	0.65
	Probe Score	0.73	0.80	0.70	0.74	0.63	0.70	0.71	0.74	0.71	0.79	0.72	0.78
NPI Licensing	LM Logprob	0.58	0.71	0.58	0.72	0.58	0.71	0.56	0.68	0.57	0.67	0.59	0.71
	Probe Score	0.79	0.91	0.77	0.88	0.75	0.87	0.70	0.80	0.78	0.90	0.81	0.91
Quantifiers	LM Logprob	0.63	0.75	0.58	0.64	0.64	0.76	0.63	0.75	0.57	0.66	0.56	0.62
	Probe Score	0.64	0.72	0.63	0.70	0.70	0.80	0.64	0.75	0.63	0.71	0.68	0.84
Subject-Verb Agreement	LM Logprob	0.60	0.79	0.64	0.88	0.61	0.82	0.61	0.82	0.60	0.83	0.61	0.83
	Probe Score	0.77	0.91	0.76	0.88	0.82	0.95	0.84	0.94	0.85	0.97	0.86	0.95

Table 10: BLiMP results broken down by different categories. We report the performance of best-layer ℓ_2 probe scores and LM logprob.

G ℓ_2-Probe Confidence Intervals

Models	Methods	BLiMP		CoLA	SyntaxGym
		AUC	ACC	AUC	AUC
Llama-3.2-1B	LM Logprob	[0.61, 0.61]	[0.79, 0.79]	[0.67, 0.69]	[0.50, 0.55]
	Probe Score	[0.73, 0.74]	[0.84, 0.84]	[0.78, 0.80]	[0.66, 0.70]
Llama-3.1-8B	LM Logprob	[0.60, 0.61]	[0.78, 0.78]	[0.68, 0.70]	[0.51, 0.55]
	Probe Score	[0.76, 0.76]	[0.84, 0.85]	[0.81, 0.82]	[0.61, 0.66]
Gemma-2-2B	LM Logprob	[0.58, 0.58]	[0.75, 0.75]	[0.62, 0.65]	[0.49, 0.54]
	Probe Score	[0.77, 0.77]	[0.86, 0.87]	[0.79, 0.81]	[0.59, 0.64]
Gemma-2-9B	LM Logprob	[0.58, 0.58]	[0.75, 0.75]	[0.62, 0.65]	[0.49, 0.54]
	Probe Score	[0.79, 0.79]	[0.88, 0.88]	[0.79, 0.81]	[0.67, 0.71]
OLMo-2-7B	LM Logprob	[0.61, 0.61]	[0.77, 0.78]	[0.67, 0.69]	[0.51, 0.55]
	Probe Score	[0.75, 0.76]	[0.85, 0.85]	[0.78, 0.80]	[0.60, 0.65]
OLMo-3-7B	LM Logprob	[0.61, 0.62]	[0.79, 0.80]	[0.68, 0.70]	[0.52, 0.56]
	Probe Score	[0.71, 0.72]	[0.80, 0.81]	[0.74, 0.76]	[0.55, 0.60]

Table 11: 95% confidence intervals of ℓ_2-probe performance metrics (AUC/ACC) for BLiMP, CoLA, and SyntaxGym.

Models	Methods	ScaLA (sr)	BLiMP-NL		ItaCoLA
		AUC	AUC	ACC	AUC
Llama-3.2-1B	LM Logprob	[0.61, 0.63]	[0.57, 0.59]	[0.74, 0.76]	[0.56, 0.59]
	Probe Score	[0.67, 0.69]	[0.58, 0.60]	[0.68, 0.70]	[0.57, 0.61]
Llama-3.1-8B	LM Logprob	[0.64, 0.66]	[0.61, 0.63]	[0.84, 0.85]	[0.60, 0.63]
	Probe Score	[0.73, 0.75]	[0.65, 0.66]	[0.76, 0.77]	[0.64, 0.67]
Gemma-2-2B	LM Logprob	[0.63, 0.65]	[0.60, 0.62]	[0.81, 0.83]	[0.53, 0.57]
	Probe Score	[0.68, 0.70]	[0.62, 0.63]	[0.72, 0.74]	[0.60, 0.63]
Gemma-2-9B	LM Logprob	[0.65, 0.67]	[0.62, 0.63]	[0.85, 0.87]	[0.54, 0.58]
	Probe Score	[0.82, 0.84]	[0.73, 0.74]	[0.83, 0.85]	[0.69, 0.72]

Table 12: 95% confidence intervals of ℓ_2-probe performance metrics (AUC/ACC) for ScaLA (sv) (Swedish), BLiMP-NL (Dutch), and ItaCoLA (Italian).

Models	Methods	RuCoLA	JCoLA	SLiNG	
		AUC	AUC	AUC	ACC
Llama-3.2-1B	LM Logprob	[0.43, 0.45]	[0.55, 0.58]	[0.55, 0.56]	[0.58, 0.59]
	Probe Score	[0.58, 0.60]	[0.55, 0.59]	[0.57, 0.57]	[0.62, 0.63]
Llama-3.1-8B	LM Logprob	[0.45, 0.47]	[0.57, 0.61]	[0.56, 0.57]	[0.65, 0.66]
	Probe Score	[0.56, 0.58]	[0.61, 0.64]	[0.60, 0.60]	[0.68, 0.69]
Gemma-2-2B	LM Logprob	[0.45, 0.47]	[0.58, 0.61]	[0.57, 0.58]	[0.61, 0.62]
	Probe Score	[0.60, 0.62]	[0.59, 0.62]	[0.63, 0.64]	[0.73, 0.74]
Gemma-2-9B	LM Logprob	[0.46, 0.48]	[0.58, 0.61]	[0.57, 0.58]	[0.61, 0.62]
	Probe Score	[0.64, 0.66]	[0.68, 0.71]	[0.67, 0.67]	[0.75, 0.76]

Table 13: 95% confidence intervals of ℓ_2-probe performance metrics for RuCoLA (Russian), JCoLA (Japanese), and SLiNG (Chinese).

H By-Layer Distributions of LASSO-Selected Neurons

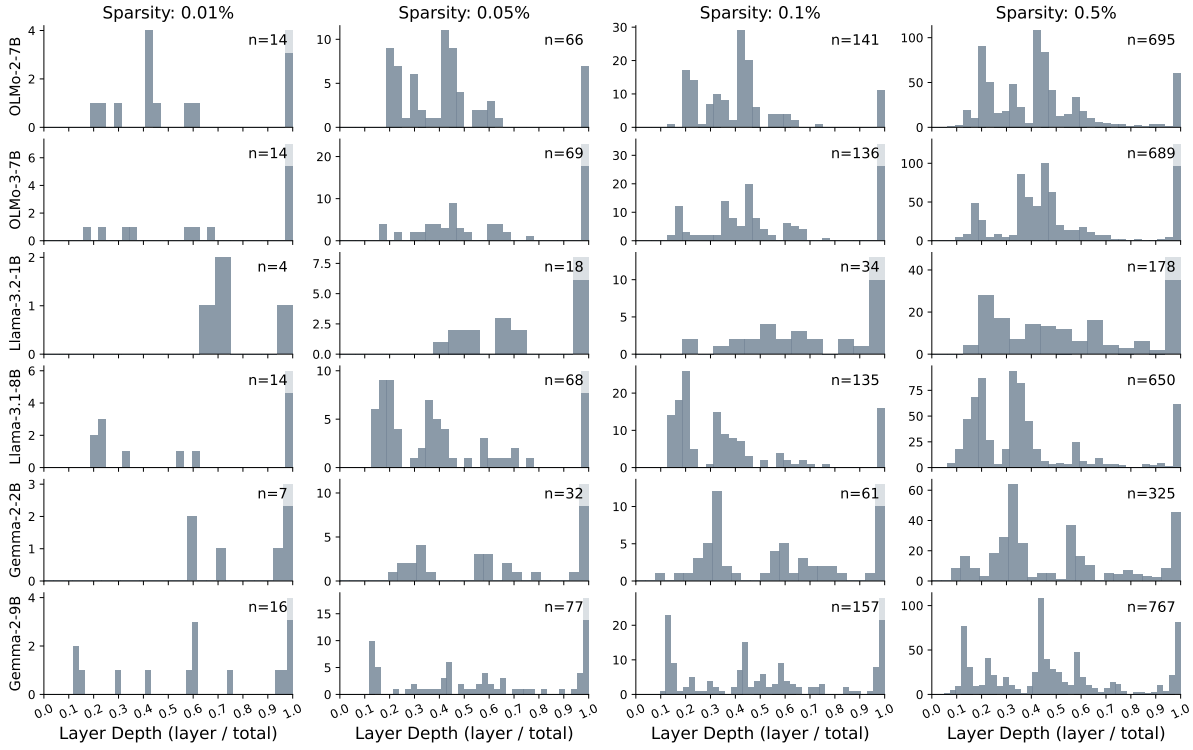


Figure 9: By-layer distributions of LASSO-selected neurons. The important neurons are distributed across many layers, with high concentration in the last layer.

I More Distributions of LM Logprob and ℓ_2 Probe Scores

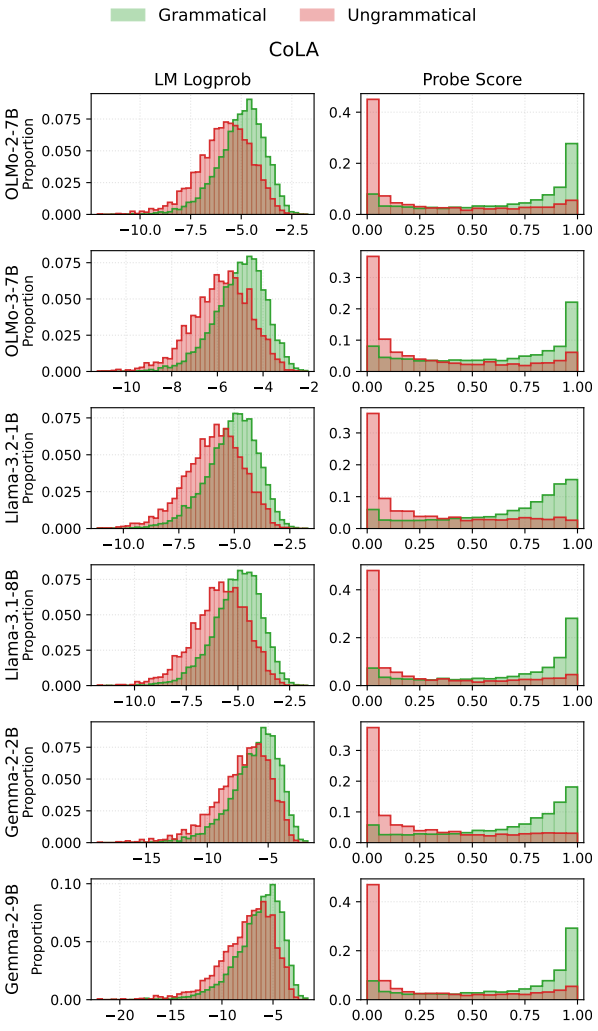


Figure 10: Distributions of LM logprob and probe scores on CoLA.

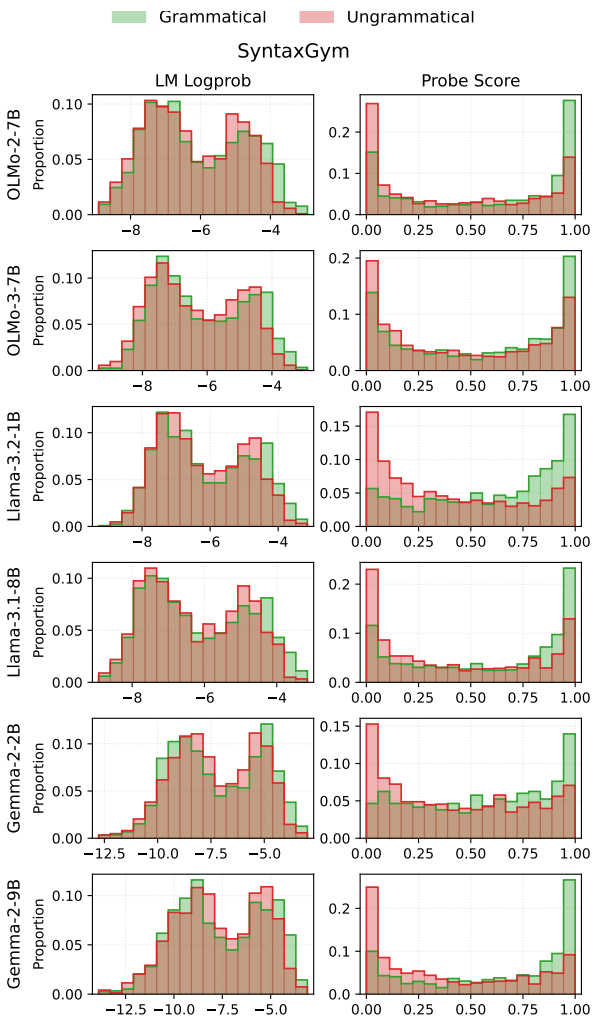


Figure 11: Distributions of LM logprob and probe scores on SyntaxGym.

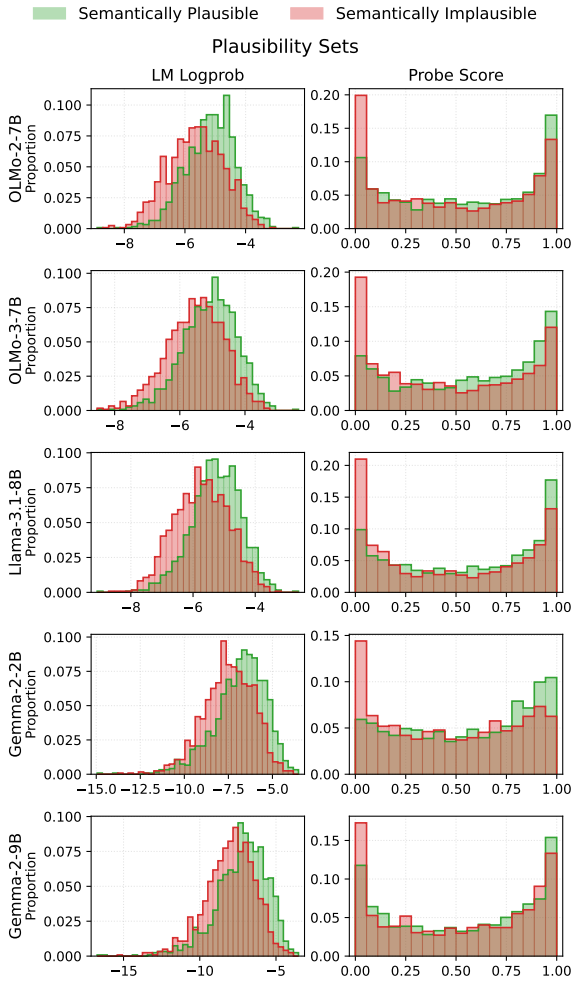


Figure 12: Distributions of LM logprob and probe scores on Plausibility Sets.

J LASSO Probe Performance on All Acceptability Datasets

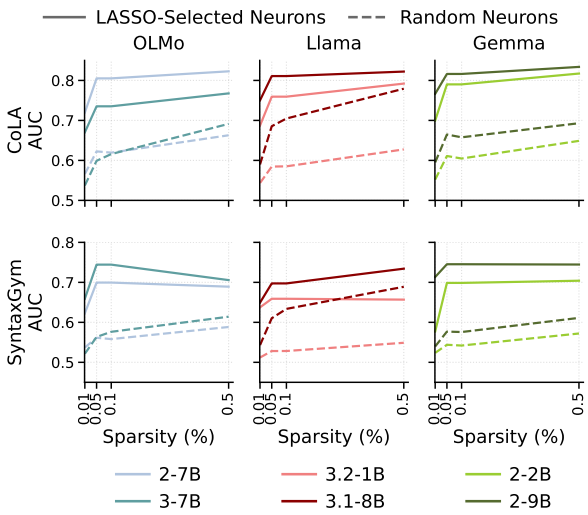


Figure 13: LASSO results on CoLA and SyntaxGym.

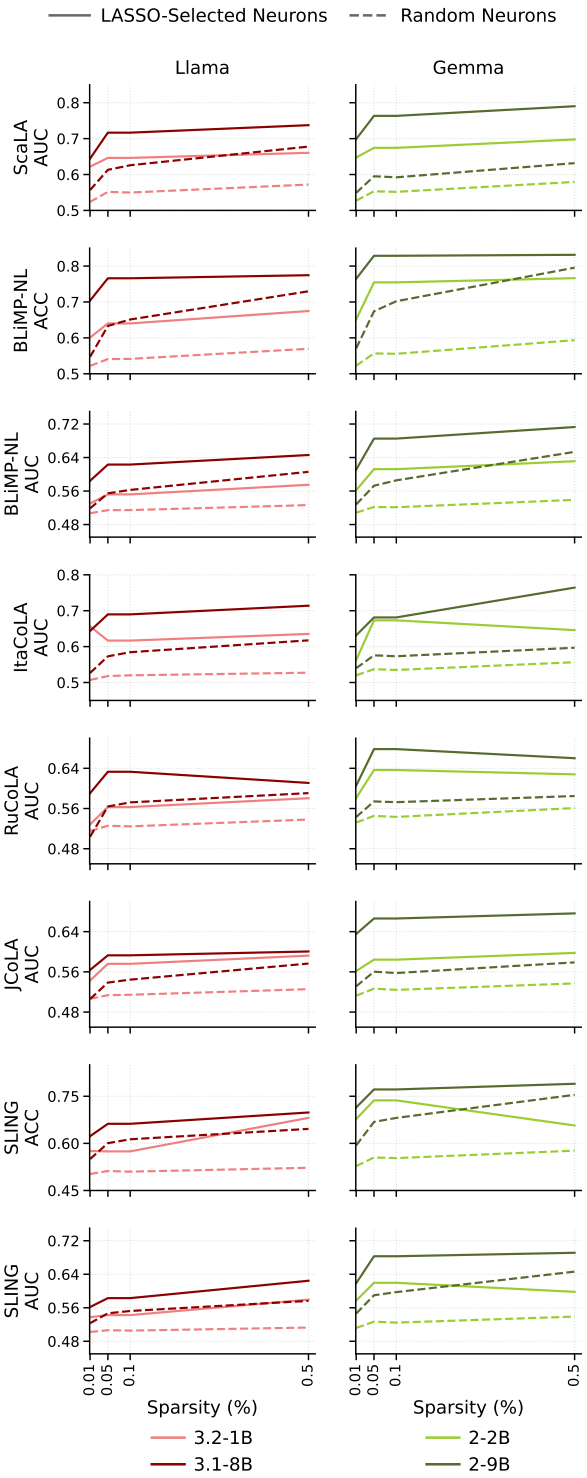


Figure 14: LASSO probe results on multilingual acceptability datasets for Llama and Gemma models.

K Pearson's Correlation of Length Normalized Logprob

	Corr. (logprob, log probe score)		
Models	**BLiMP**	**CoLA**	**SyntaxGym**
Llama-3.2-1B	0.24	0.34	0.07
Llama-3.1-8B	0.22	0.36	0.23
Gemma-2-2B	0.22	0.29	0.18
Gemma-2-9B	0.18	0.29	0.26
OLMo-2-7B	0.19	0.30	0.26
OLMo-3-7B	0.25	0.40	0.21

Table 14: Pearson correlation coefficient of LM logprob and log probe scores. **Correlation strengths are moderate, suggesting that probe scores are not simply recapturing the LM logprob.**

L Variance of Length Normalized Cumulative Logprob

	Train Set Logprob Variance (σ^2)	
Models	**Per Token**	**Last Tok Only**
Llama-3.2-1B	1.99	1.32
Llama-3.1-8B	2.32	1.45
Gemma-2-2B	5.12	2.22
Gemma-2-9B	6.64	2.75
OLMo-2-7B	2.18	1.49
OLMo-3-7B	2.23	1.36

Table 15: Logprob variance of original (assumed grammatical) sentences in the synthetic training corpus. Values are provided for the per-token and last-token-only configurations used in the ridge regression probes.

	Eval Set Logprob Variance (σ^2)	
Models	**Per Token**	**Last Tok Only**
Llama-3.2-1B	2.83	1.32
Llama-3.1-8B	2.43	1.29
Gemma-2-2B	11.97	4.15
Gemma-2-9B	16.41	5.46
OLMo-2-7B	2.65	1.35
OLMo-3-7B	1.33	2.77

Table 16: Logprob variance of sentences in the pooled evaluation sets (BLiMP, CoLA, and SyntaxGym). Values are provided for the per-token and last-token-only setups.

N Syntactic Acceptability Judgments by Metalinguistic Prompting

To compare best-layer ℓ_2 probes against the metalinguistic syntactic judgments, we employ few-shot prompting with the template shown in Figure 15. For a dataset of N sentences, Let $\hat{y}_i \in \{0, 1\}$ denote the binary prediction for sentence x_i generated either via the metalinguistic prompt or the ℓ_2 probe (setting 0.5 as the probe score threshold). The non-pairwise accuracy is Accuracy $= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i)$, where $y_i \in \{0, 1\}$ is the ground-truth binary label.

As seen in Figure 16, our probes surpass the performance of metalinguistic judgments across all datasets for all models. Note that smaller models (Llama-3.2-1B and Gemma-2-2B) may not have much capability to learn in context by fewshot examples, which could explain their near random metalinguistic judgment performance on BLiMP.

O Statements of AI Usage

Coding We use Codex to generate codes for producing the plots. We have manually checked the AI-generated codes to ensure correctness and consistency with our setups.

Writing We use ChatGPT and Gemini for grammar check

M Cross-lingual Transfer of Non-English Trained Probes for Llama-3.2-1B

		Test Dataset (AUC)					
		Swedish	Dutch	Italian	Russian	Japanese	Chinese
Language	**Train Dataset**	**ScaLA (sv)**	**BLiMP-NL**	**ItaCoLA**	**RuCoLA**	**JCoLA**	**SLING**
Swedish	ScaLA (sv)	—	0.58	0.59	0.53	0.52	0.56
Dutch	BLiMP-NL	0.60	—	0.53	0.50	0.50	0.55
Italian	ItaCoLA	0.58	0.53	—	0.51	0.53	0.49
Russian	RuCoLA	0.57	0.54	0.53	—	0.52	0.49
Japanese	JCoLA	0.54	0.54	0.51	0.56	—	0.54
Chinese	SLING	0.54	0.53	0.49	0.52	0.52	—

Table 17: Cross-lingual transfer AUC for Llama-3.2-1B. Each row indicates the training language and each column indicates the test language. Transfer from non-English languages is less effective than from English (Table 2), consistent with the hypothesized English-dominant representation space of the model.

Prompt Template for Metalinguistic Grammaticality judgments

```
Determine if the following sentences are grammatical.

Sentence: The boy kick the ball.
Grammatical: No.

Sentence: That you are back surprised me.
Grammatical: Yes.

Sentence: The story goes on and on.
Grammatical: Yes.

Sentence: Last night I was ever drunk.
Grammatical: No.

Sentence: {Input_Sentence}
Grammatical: {Prediction}
```

Figure 15: The metalinguistic prompting template. The model correctly answers the prompt if it outputs the corresponding grammaticality prediction of the input sentence ("Yes." or "No."). A prediction is classified as "Yes." if its conditional probability exceeds that of "No."; otherwise, it is classified as "No."

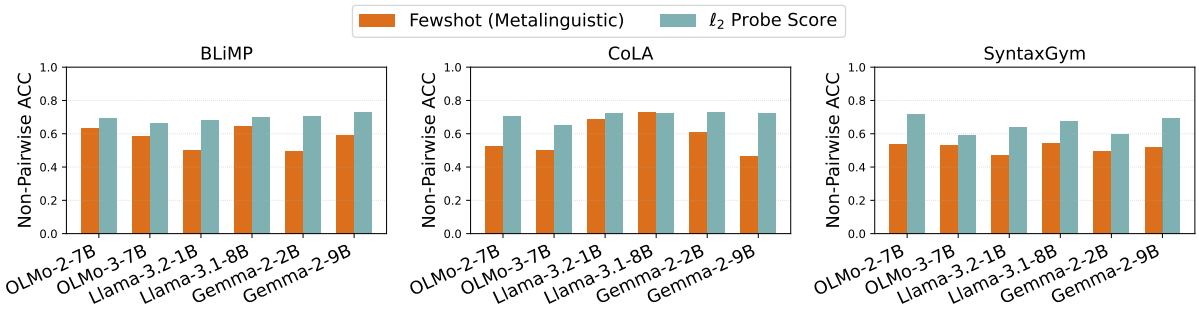


Figure 16: Nonpairwise accuracies of metalinguistic fewshot prompting and ℓ_2 probes. The random baseline is 0.5.