# What Makes Two Language Models Think Alike?

**Anonymous ACL submission**

## Abstract

Do architectural differences significantly affect the way models represent and process language? We propose a new approach, based on metric-learning encoding models (MLEMs), as a first step to answer this question. The approach provides a feature-based comparison of how any two layers of any two models represent linguistic information. We apply the method to BERT, GPT-2 and Mamba. Unlike previous methods, MLEMs offer a *transparent* comparison, by identifying the specific linguistic features responsible for similarities and differences. More generally, the method uses formal, symbolic descriptions of a domain, and use these to compare neural representations. As such, the approach can straightforwardly be extended to other domains, such as speech and vision, and to other neural systems, including human brains.

## 1 Introduction

Marr's hierarchy proposes a structured approach for describing information-processing systems using three levels (Figure 1; Marr, 2010): (1) computational, (2) algorithmic, and (3) implementational. The computational level defines the problem and the system's goals. For example, a goal of a system could be to compute the sum of two numbers. The algorithmic level addresses the strategies used to solve the problem, detailing the step-by-step processes involved. For instance, one algorithm could involve digit-by-digit addition starting from the least significant digit, while another could involve repeated counting. There is therefore a one-to-many relationship between the computational and algorithmic levels (plain arrows) Finally, the implementational level concerns the physical realization of the system, such as how algorithms are executed within the brain's neural architecture or a computer's hardware. Similarly, there's a one-to-many relation between the algorithmic and implementational levels (dashed arrows).

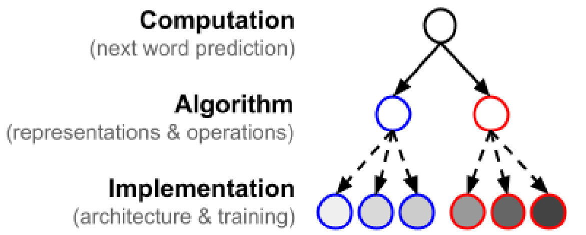


Figure 1: **Marr's levels of analysis**. While language models may share the same computational goal (top level, next-word prediction), their architectures could differ substantially (bottom level). They therefore may or may not develop the same representations and algorithms (middle level) to perform the task.

Language models can be described along Marr's three levels. At the computational level, most language models are trained on a next-word prediction task (but see, e.g., Gloeckle et al., 2024, for multi-token prediction). At the implementational level, different architectures (RNNs, Transformers, SSMs, etc.) can be implemented differently onto hardware. These architectural differences might lead to variations at the algorithmic level, despite sharing the same computational problem, or, conversely, they might converge on a similar algorithmic solution. In this work we ask whether language models with different architectures represent and process language in the same way?

To quantify this, one can start by computing second-order isomorphism between model representations of words (Shepard and Chipman, 1970), or use methods such as Canonical Correlation Analysis (CCA, Wu et al., 2020). Similarity can thus be computed for any pair of layers from any pair of models. However, similarity measures do not provide an *explanation* for what makes the representations and processing of text appear similar or dissimilar across models and layers.

To address this, we propose a novel approach based on metric-learning encoding models

(MLEMs, Jalouzot et al., 2024), which explains model similarity by identifying the linguistic features that underlie it. We illustrate the approach using three different types of neural architectures: encoder-based Transformer, decoder-based Transformer, and Mamba, quantifying their similarity and providing a linguistic-feature-based comparison for each pair of model layers.

Overall, metric-learning encoding models use existing theoretical descriptions as a grid of analysis of models and model comparisons. This approach can thus naturally be extended beyond text to domains such as speech and vision. They can then use any symbolic theory there to compare any two neural models, including artificial neural models (different architectures or different instantiations of the same one) as well as human and non-human animal brains.

## 2 Related Literature

In previous work to quantify similarity between representations of two neural systems, a central approach is based on *second-order isomorphism* (Shepard and Chipman, 1970). Second-order isomorphism suggests that while the representations of two systems belong to different spaces, the similarity between them can be quantified by comparing the pairwise distances within each neural space, thus 'second-order' similarity. Second-order isomorphism has been used to compare representations of two artificial neural networks (Laakso and Cottrell, 2000; Mehrer et al., 2020), or of two brains, where it is also known as Representational Similarity Analysis (RSA; Kriegeskorte et al., 2008; Abnar et al., 2019).

Several other similarity measures between representations of different models have been proposed in previous work, including linear regression (Adriana et al., 2015), canonical correlation analysis (CCA; Raghu et al., 2017; Morcos et al., 2018; Wu et al., 2020; Belinkov and Glass, 2018), statistical shape analysis (Williams et al., 2024), functional behaviors on downstream tasks (e.g., Alain and Bengio, 2018), and Dynamic Similarity Analysis (DSA, Ostrow et al., 2023). However, such measures do not directly provide an explanation for *why* two neural systems converge or differ in the way they represent information.

Recently, metric-learning encoding models (MLEMs; Jalouzot et al., 2024) have been proposed as a method to examine the types of information that predict neural distances between representations within a single neural system. MLEMs have shown their ability to identify which linguistic features most strongly predict neural distances in various layers of a model. Here, we leverage MLEMs to study the similarity between representations in two different language models. This approach offers a feature-based comparison of how two models represent linguistic information, and thereby explains the underlying factors driving the similarities and differences.

## 3 General Setup

**Language Models** We investigated the similarities between three different types of models: (1) GPT-2 (Radford et al., 2019), a **decoder-based Transformer**, (2) BERT (Devlin et al., 2018), an **encoder-based Transformer**, and (3) Mamba, an architecture based on a **state-space model** (Gu and Dao, 2023). We collected representations from each layer of the models for every word in our controlled dataset (see below). For words that are split into multiple tokens, we used the representation of the final token.

**Probing Data** To study the neural encoding of linguistic features, we created a dataset, which contains a list of sentences and their corresponding list of linguistic features. Sentences and features were generated using a custom grammar to cover central linguistic features, such as grammatical number, gender or tense (Table S1).

**Metric-Learning Encoding Models (MLEMs)** Metric-Learning Encoding Models (Jalouzot et al., 2024) start from the assumption that to effectively capture multivariate, distributed neural encoding of linguistic information, one should model *distances* among neural representations rather than individual activations of single units. Given a set of inputs (e.g., words), where each is represented along a set of features (e.g., tense, gender), the goal is to learn a *metric function* (aka, a distance function), which is defined over pairs of inputs and computed based on the features of the inputs only. The optimal such metric function is the one that minimizes the differences between the modeled distances among the inputs and the empirical (neural) ones (Figure S6). This optimal metric can be derived using standard metric-learning methods (Kulis et al., 2013).

## 4 Results

**Feature-Importance Profiles** To quantify similarity among models, we first ask which linguistic features best explain neural distances in each layer of a language model. For this, we computed *feature importance (FI)* based on Metric-Learning Encoding Models. That is, for each layer of a given language model, we computed which linguistic features (tense, grammatical number, etc.) predict neural distances among representations of words in the dataset. Specifically, we computed FI as the average decrease in Spearman correlation score of the trained MLEM on a left-out dataset when permuting a feature. We highlight several main observations in the results (Figure 3): first, part-of-speech is the dominant linguistic feature across layers of Transformer-based models. However, for Mamba, it is so only for the first and last layer. In Mamba, we observe a significant increase in the importance of word position at around layer 10 of the model. Finally, we note that the importance of the grammatical number feature tends to decrease from early to later layers in all models.

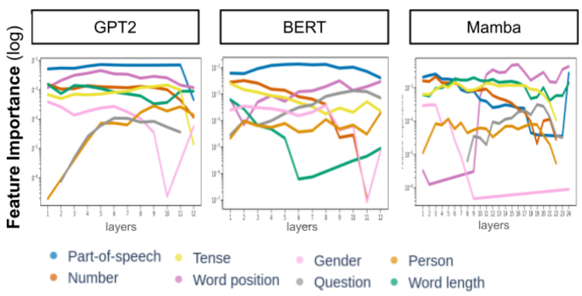


Figure 3: **Feature Importance Profiles**. The relative importance of linguistic features varies across layers and models

**Feature-Based Similarity among Language Models** We next asked, which language models most resemble each other in the way they represent linguistic information? We compared two approaches: *feature-based* and *feature-agnostic* similarity measures. For the former, we computed feature-based similarity based on the FI profiles from the MLEMs. Specifically, for each pair of layers, we computed the Kendall correlation coefficient, which quantifies to what extent the same linguistic features are dominant in both layers. Since linguistic feature with low importance (e.g., near zero) are not predictive of neural distances, it is desired that they will have a small effect on the similarity measure for two models. We therefore quantified similarity with a *weighted* version of Kendall correlation, which weighs feature importance based on their rank (Vigna, 2015). For comparison, for the feature-agnostic similarity measure, we followed a standard RSA approach (Kriegeskorte et al., 2008). Specifically, for each layer of a model, we first computed a dissimilarity matrix (DSM) for all words in the dataset. That is, for each layer, we computed the Euclidean distance among all pairs of stimuli presented to the model. Then, given two DSMs of two different layers, we computed the Spearman correlation between the upper triangles of the DSMs.

Figure 2 shows the resulting feature-based (Panel A) and the feature-agnostic (Panel B) matrices. To further visualize the results, the corresponding plots show each model layer in a shared 2D space, optimally preserving layer-wise similarity using Multi-Dimensional Scaling (MDS, Kruskal, 1964) analyses. Overall, the feature-based and feature-agnostic approaches agree on model simi-

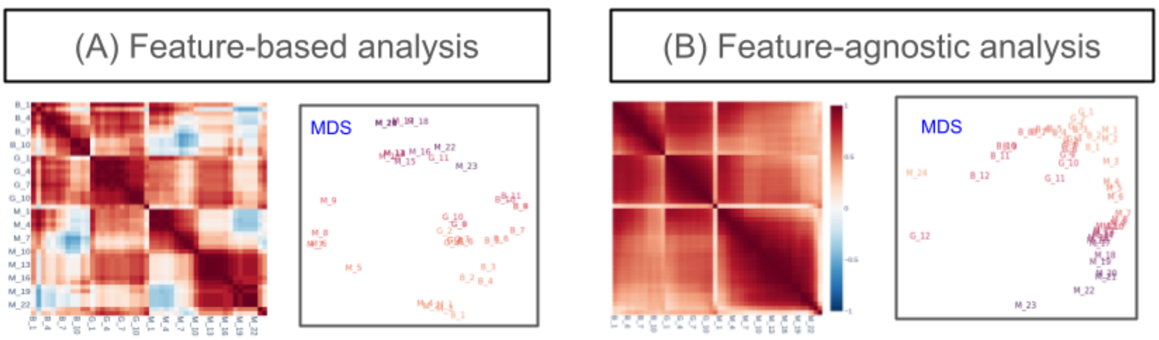


Figure 2: **Model Similarity.** (A) *Feature-based* similarity matrix corresponding to the pairwise correlations between feature-importance values. (B) *Feature-agnostic* similarity matrix based on raw Euclidean distances between word embeddings. The Multi-Dimensional Scaling representations of these distances are represented for both types of analyses (**B** stands for BERT, **G** for GPT2, and **M** for Mamba).

larity ($\rho_{Spearman} = 0.69$). However, feature-based similarity highlights specific differences between and across models. For example, for Mamba, a block structure appears, separating low and high layers of the model. The FI profiles for Mamba explains this difference, given the sudden increase in FI of word position at around layer 10 of the model. This increase in FI is apparent when visualizing all word representations in the model for different layers. Indeed, word position strongly separates word representations at higher but not lower layers (Figure 3). This illustrates the importance of a feature-based compared to feature-agnostic theory in explaining similarity, as we further investigate next.

**What Makes Two Language Models Think Alike?** With the feature-based similarity approach, we are able to answer the question - why two model layers are similar or different in the way they represent linguistic information? Figure 4 illustrates this by contrasting feature importance of different pairs of layers - one with high ($\tau_{weighted} = 0.77$) and the other with low similarity ($\tau_{weighted} = -0.24$). These examples were chosen based on the minimal and maximal values of the similarity matrix (Figure 2A). For the case of high similarity (between GPT2 layer 8 and BERT layer 4), FI values of the two layers largely agree, lying on the diagonal of the scatter. In particular, the most dominant feature in both layers is the same - part-of-speech (PoS). In accordance, the corresponding MDS plots (to the left and top of the scatter), show that word representations (color dots) are, indeed, well separated with respect to part-of-speech (see legend). In contrast, for the case of low similarity (between Mamba-layer-8 and BERT-layer-4), FI values are mostly off diagonal, in particular the most dominant one for PoS. In accordance, the corresponding MDS plot for Mamba does not separate word representations as well as in the two other cases.

## 5 Summary and Conclusions

While language models share the same computational task (next-word prediction), architectural difference might lead to differences in how different models represent and process language. Here, we presented a new approach to quantify such differences. We illustrated the approach with three types of architecture, showing its utility in quantifying model similarity and, importantly, explaining it.

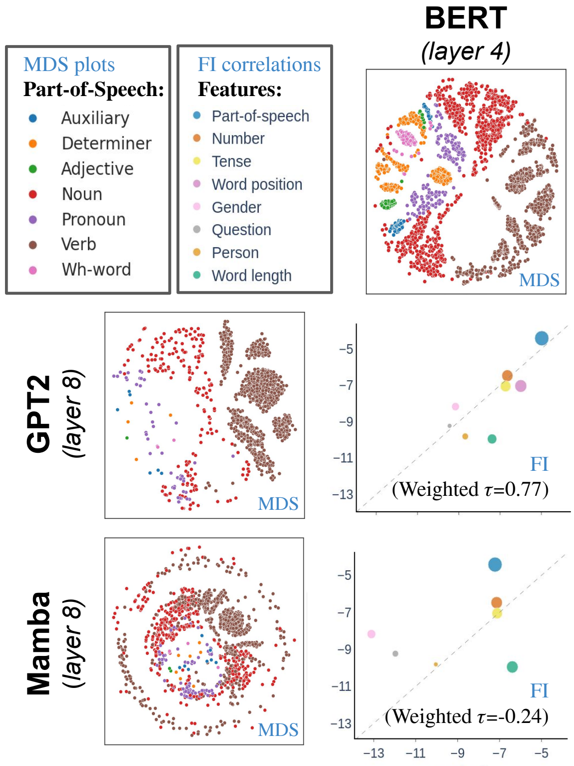


Figure 4: **Illustrating** how model/layers represent linguistic features. MDS plots of the representations, and pairwise comparison of the Feature Importance profiles.

For all pairs of model layers, we identified which linguistic features dominate word representations and whether they are the same or not across models, as illustrated for the case of part-of-speech. Together, this shows the utility of feature-based approaches to study model similarity, providing theory-based explanations for why two models converge or diverge in the way they process natural language text. This approach could naturally be extended to other domains, such as speech and vision. And it can be applied to compare neural systems, including artificial neural networks as we did here as well as human brains.

## Limitations

For simplicity, when computing feature importance, we assumed that there are no interactions among linguistic features in predicting neural distances among sentence representations (i.e., assuming a diagonal weight matrix). However, such interactions are common in many problems, including in language. The framework of MLEMs allows a straightforward way to introduce interactions, while, in contrast to other approaches (such as RSA), it preserves the metric property of the learned distances. Also, in MLEMs, we have only included features that we consider essential to the list of words in the dataset (tense, grammatical number, etc.), as they were created by contrasting these dimensions. Future work can explore more exhaustive lists of features to describe and contrast words, as well as larger datasets and the introduction of possible interactions among features.

## References


Samira Abnar, Lisa Beinborn, Rochelle Choenni, and Jelle Zuidema. 2019. Blackbox meets blackbox: Representational similarity and stability analysis of neural language models and brains. In *Proceedings of the ACL-Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 191–203.

Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. 2015. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2(3):1.

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes. *Preprint*, arXiv:1610.01644.

Yonatan Belinkov and James R. Glass. 2018. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Louis Jalouzot, Robin Sobczyk, Bastien Lhopitallier, Jeanne Salle, Nur Lan, Emmanuel Chemla, and Yair Lakretz. 2024. Metric-learning encoding models identify processing profiles of linguistic features in bert's representations. *Preprint*, arXiv:2402.11608.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. 2008. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2.

Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Brian Kulis et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364.

Aarre Laakso and Garrison Cottrell. 2000. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76.

David Marr. 2010. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. 2020. Individual differences among deep neural network models. *Nature communications*, 11(1):5725.

Ari S. Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5732–5741, Red Hook, NY, USA. Curran Associates Inc.

Mitchell Ostrow, Adam Eisen, Leo Kozachkov, and Ila Fiete. 2023. Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. In *Advances in Neural Information Processing Systems*, volume 36, pages 33824–33837. Curran Associates, Inc.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Roger N Shepard and Susan Chipman. 1970. Second-order isomorphism of internal representations: Shapes of states. *Cognitive Psychology*, 1(1):1–17.

Sebastiano Vigna. 2015. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*, pages 1166–1176.

Alex H. Williams, Erin Kunz, Simon Kornblith, and Scott W. Linderman. 2024. Generalized shape metrics on neural representations. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

# Appendices

## A The Probing Dataset

The probing dataset contains a list of sentences and their corresponding list of linguistic features. Sentences and features were generated using a custom grammar to cover central linguistic features, such as grammatical number, gender or tense, as well as confounding factors, such as word position. Table S1 shows several sentence examples, and the marking of features for each word.

To secure a clean interpretation of the relative contributions of the different features, we checked for correlations between linguistic features. Figure S5 shows the pairwise Pearson correlations among all features in the dataset.

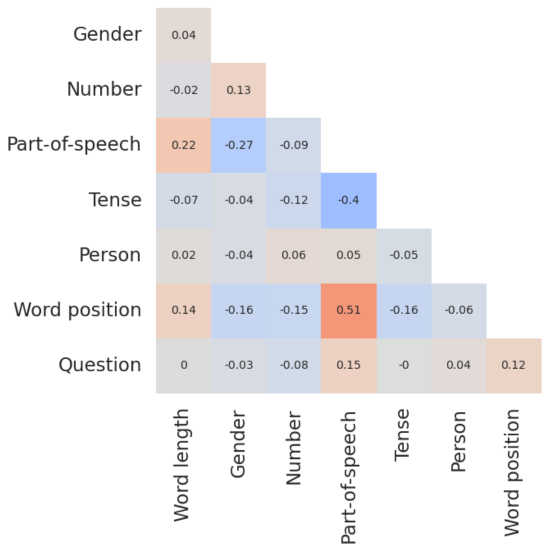


Figure S5: **Pairwise Pearson correlations among all linguistic features in the probing dataset.**

## B Metric Learning Encoding Models (MLEMs)

Following Jalouzot et al. (2024), we provide a formal description of a Metric-Learning Encoding Model: Consider a set of $N$ sentences, each characterized by a set of (linguistic) features $\mathcal{F}$. MLEMs compute two types of pairwise distances. First, *pairwise neural distances* $D^{\mathcal{N}}$ (right branch in Fig. S6), which are computed based on standard distance (e.g. Euclidean or cosine distance) between the neural responses of a set of units (e.g. a layer) for any two sentences.

Second, *pairwise feature distances* $D^{\mathcal{F},W}$ are computed as follows. First, *feature difference vectors* are computed, which indicate on which features two sentences differ: $\Delta(s_i, s_j) = (\mathbb{1}_{f(s_i)\neq f(s_j)})_{f\in\mathcal{F}}$. Then, feature distances are computed using a standard bi-linear form parameterized by a symmetric positive definite matrix $W \in \mathbb{M}_n^+$:

$$\left(D_{ij}^{\mathcal{F},W}\right)^2 = \Delta(s_i, s_j)^T W \Delta(s_i, s_j)$$

MLEMs, as metric-learning methods, optimize $W$ to bring the pairwise feature distances as close as possible to the neural ones, across all $(i, j)$ pairs of stimuli:

$$W^* = \underset{W\in\mathbb{M}_n^+}{\text{argmin}} \sum_{i<j} \left( \left(D_{ij}^{\mathcal{F},W}\right)^2 - \left(D_{ij}^{\mathcal{N}}\right)^2 \right)^2 + \lambda ||W||_2^2$$

When $W$ is assumed to be diagonal (with no interaction terms), the optimization problem simplifies to a least-squares problem, and the symmetric positive definite constraint transforms into a non-negativity constraint on the diagonal elements.

**Model Training and Evaluation** As in Jalouzot et al. (2024), for simplicity, we focused on the diagonal case of $W$ and trained a standard Ridge model with a non-negativity constraint on the parameters. The regularization parameter $\alpha$ was optimized using nested cross-validation (CV; $\alpha \in 10^{[-4,4]}$; To facilitate $\alpha$ optimization across all models, target values were min-max scaled into $[0, 1]$). We evaluated the model using the Spearman correlation score $\rho$ and report the average across CV splits. This score only assesses the similarity between the ranks of the predictions and those of the ground-truth. We chose this score as it is independent

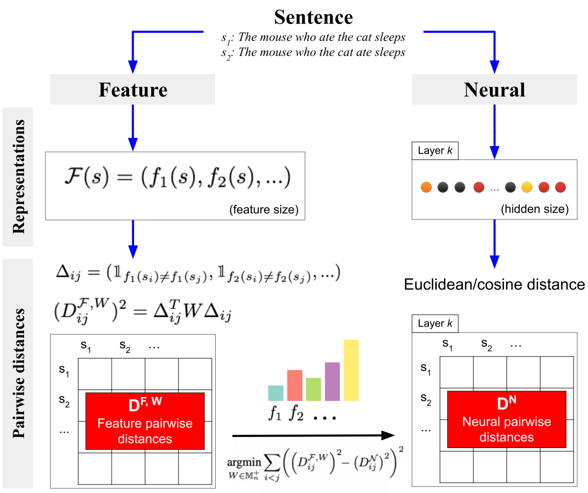


Figure S6: **A Metric-Learning Encoding Model**: MLEMs determine the relative importance of features by identifying the optimal alignment between distances in feature space and neural space.

| word | Word length | Gender | Number | PoS | Tense | Person | Word position | Question |
|---|---|---|---|---|---|---|---|---|
| the | 3 | NaN | NaN | Det | NaN | NaN | 0 | False |
| woman | 5 | female | singular | Noun | NaN | 3 | 1 | False |
| plays | 5 | Nan | singular | Verb | present | 3 | 2 | False |
| no | 2 | NaN | NaN | Det | NaN | NaN | 0 | False |
| prince | 6 | male | singular | Noun | NaN | 3 | 1 | False |
| sings | 5 | NaN | singular | Verb | present | 3 | 2 | False |
| I | 1 | NaN | singular | Pronoun | NaN | 1 | 0 | False |
| vanished | 8 | NaN | NaN | Verb | past | NaN | 1 | False |
| do | 2 | NaN | singular | Auxiliary | present | NaN | 0 | True |
| you | 3 | NaN | NaN | Pronoun | NaN | 2 | 1 | True |
| sing | 4 | NaN | NaN | Verb | present | NaN | 2 | True |
| Mary | 4 | female | singular | Noun | NaN | 3 | 0 | False |
| fell | 4 | NaN | NaN | Verb | past | NaN | 1 | False |
| which | 5 | NaN | NaN | Wh-word | NaN | NaN | 0 | True |
| men | 3 | male | plural | Noun | NaN | 3 | 1 | True |
| sneezed | 7 | NaN | NaN | Verb | past | NaN | 2 | True |

Table S1: **Examples from the probing dataset**

of the scale of the data (unlike the Mean Squared Error) and it cannot be arbitrarily negative when the estimator is very bad (unlike the coefficient of determination $R^2$).