Metric Learning Encoding Models: A Multivariate Framework for Interpreting Neural Representations

Anonymous authors
Paper under double-blind review

Abstract

Understanding how explicit theoretical features are encoded in opaque neural systems is a central challenge now common to neuroscience and AI. We introduce Metric Learning Encoding Models (MLEMs) to address this challenge most directly as a metric learning problem: we fit the distance in the space of theoretical features to match the distance in neural space. Our framework improves on univariate encoding and decoding methods by building on second-order isomorphism methods, such as Representational Similarity Analysis, and extends them by learning a metric that efficiently models feature as well as interactions between them. The effectiveness of MLEM is validated through two sets of simulations. First, MLEMs recover ground-truth importance features in synthetic datasets better than state-of-the-art methods, such as Feature Reweighted RSA (FR-RSA). Second, we deploy MLEMs on real language data, where they show stronger robustness to noise in calculating the importance of linguistic features (gender, tense, etc.). MLEMs are applicable to any domains where theoretical features can be identified, such as language, vision, audition, etc. Code will be made available upon publication.

Keywords: Multivariate Encoding; Distributed Neural Representations; Metric Learning; Representational Similarity Analysis

1 Introduction

A central goal in both neuroscience and Artificial Intelligence is to understand how complex, high-dimensional systems like the brain and deep neural networks represent and process information (Belinkov & Glass, 2019; Pohl et al., 2024). These systems excel at complex tasks such as natural language processing but their internal mechanisms are still not well understood and for this reason are often called "black boxes". High-dimensional recordings of neural activity can be obtained from both biological and artificial systems, but interpreting these complex data remains a major challenge.

A powerful strategy for interpreting these systems is to decompose those internal representations into a set of discrete, theoretically-grounded features. For instance, in the domain of language, it is natural to ask how different aspects of language, such as syntax (Pallier et al., 2011; Hewitt & Manning, 2019; Diego-Simón et al., 2024), semantics (Huth et al., 2016; Li et al., 2024), or morphology (Crepaldi et al., 2019) are encoded in neural recordings from the human brain or in the activation patterns of a Large Language Model (LLM). Similarly, in vision, one might investigate how features like object category, shape, or texture are represented in the visual cortex or in a convolutional neural network.

Two general approaches to study the neural encoding of theoretical features can be discerned in the literature: *decoding* and *encoding* models (Kriegeskorte & Douglas, 2019). **Decoding models** (or "probes") attempt to predict features from neural activity (Hupkes et al., 2018; Tenney et al., 2019; Arps et al., 2022). High performance of a decoding model demonstrates that feature-related information is present but this approach is susceptible to confounds. Indeed a feature may be decodable simply because it correlates with another, truly encoded feature (Hewitt & Liang, 2019; Belinkov, 2022; Kumar et al., 2022a). For example, a probe trained to decode tense in a sentence from some representations might succeed not because tense is explicitly

encoded in the representations, but because, e.g., future tense correlates with the superficial presence of "will" in the sentence, which is all the model needs to encode for success without any further level of abstraction. **Encoding models**, conversely, predict neural activity from features (Wehbe et al., 2014; Schrimpf et al., 2020; Caucheteux et al., 2021; Caucheteux & King, 2022; Pasquiou et al., 2022; Oota et al., 2022; Pasquiou et al., 2023; Hosseini et al., 2024). Those models can be fitted on multiple features and confounds and allow for delineating their contribution. However, unlike decoding models, they are often univariate, fitting each neural unit (e.g., neuron or voxel) independently. Therefore, they cannot model distributed representations encoded across populations of units, which is a critical characteristic of neural computation (Georgopoulos et al., 1986; Rumelhart et al., 1986).

Here, we introduce a simple approach, which preserves the best from both worlds, by extending encoding methods to the multivariate case within a metric learning framework (Kulis, 2013). In a nutshell, the method learns an optimal metric over a space of theoretical features to best match the geometry of the neural representation space (Goldstein et al., 2024; Brown & Farivar, 2024). We show that previous methods, such as Representational Similarity Analysis (RSA; Kriegeskorte et al. (2008)) and its feature-reweighted variant (FR-RSA, Jozwik et al. (2017); Storrs et al. (2021); Xu et al. (2021); Kaniuth & Hebart (2022)) can be derived as *special cases* of MLEMs. We call the method Metric Learning Encoding Models, or MLEMs for short. A key innovation of MLEM is its ability to model interactions between features by design, a critical requirement for understanding complex domains like language where features are rarely processed in isolation and are often entangled in a single representation vector (Huang et al., 2021; Kumar et al., 2022b). For instance, in English, the marking of a verb with an "-s" suffix depends on the interaction between two features of its subject: its number (singular) and its person (third). Hence a neural representation of the verb has to jointly encode both features to be used downstream to produce the correct morphological form.

The approach is validated in two settings. First, using simulated data with a known ground-truth, we show that MLEM recovers underlying feature weights more accurately than a strong baseline, FR-RSA. Second, when applied to representations from an LLM processing controlled syntactic stimuli, MLEM proves more robust to noise and converges faster, while identifying similar feature importance profiles to FR-RSA. These improvements in robustness and data efficiency are especially useful for real-world applications, such as analyzing feature encoding in brain data which is often noisy and scarce.

The present study focuses on text stimuli and LLM representations, but the MLEM framework is modality-agnostic. As a second-order isomorphism method, it can be applied to any domain where stimuli can be described by a set of theoretical features and measured in a high-dimensional representation space. This includes any experiment involving visual or auditory stimuli and neural representations can come from various neuroimaging modalities (fMRI, MEG, ECoG, etc.) or even from intermediate layers of artificial neural networks. This work thus provides a versatile tool for uncovering the principles of neural information processing in both artificial and biological systems.

2 Methods and Materials

We formalize multivariate encoding as a metric learning problem. The Metric Learning Encoding Model (MLEM) approach, summarized in Figure 1, consists of four main steps: (A) For a set of stimuli, pairwise distances are computed for different theoretical features, creating a Representational Dissimilarity Matrix (RDM) D^f for each. (B) An RDM D^N is also computed from neural representations of the same stimuli. (C) A metric over the feature space is learned that maximizes the Spearman correlation between feature-based distances and neural distances. This metric is parameterized by a symmetric positive definite matrix W, whose off-diagonal terms explicitly model feature interactions. (D) Permutation Feature Importance is used to assess the contribution of each feature and interaction to the model's predictions. The following sections detail each of these steps, as well as the training procedure, the baseline and datasets used in this study.

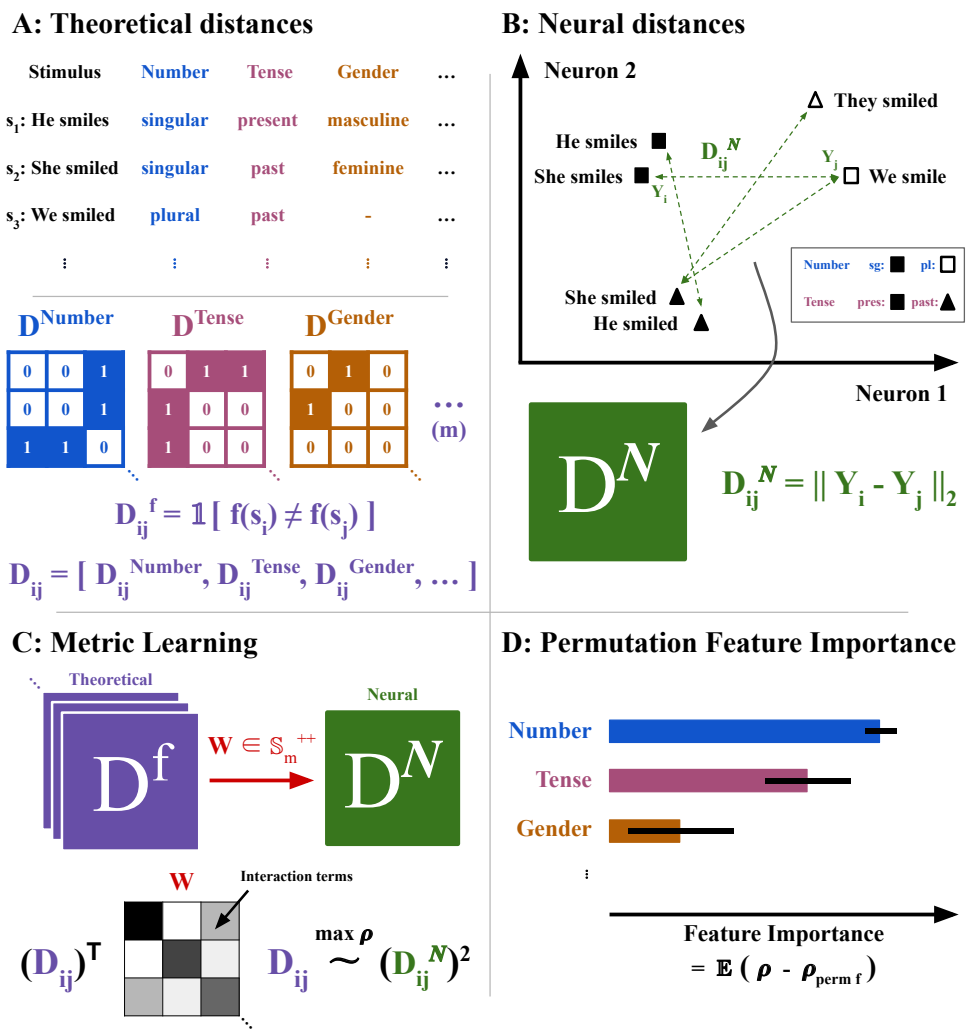


Figure 1: **Overview of the Metric Learning Encoding Models (MLEMs) Approach:** an example from natural language

A: Consider a set of n stimuli (rows; sentences) and m theoretical features (columns; grammatical number, tense, gender, ...) along with their values for each stimulus. In this example, the features are categorical and *feature distance* are defined as whether a pair of stimuli (s_i, s_j) share a given feature f_k: $\mathbb{1}\left[f_k(s_i) \neq f_k(s_j)\right]$. Accordingly, a Representational Dissimilarity Matrix (RDM) is computed for each of the m features ($D^{\text{Number}}, D^{\text{Tense}}, D^{\text{Gender}}, \ldots \in \mathbb{R}_+^{n \times n}$). Elements of the matrix thus indicate whether pairs of stimuli share a given feature. Finally, for each pair of stimuli, let $D_{ij}^F \in \mathbb{R}^m$ be a vector, composed of the (i, j) elements from all RDMs (bottom panel). D_{ij}^F summarizes the *theoretical distance* between two stimuli, with respect to all features.

B: On the other hand, consider n high-dimensional neural representations of the stimuli (e.g. embeddings of LLMs or fMRI images, here represented in 2D). From these representations, one can compute an RDM of *neural distances* D^N (e.g., based on Euclidean distance).

C: The core metric learning component of MLEM involves learning a weighted norm of the feature distance vectors $\|D_{ij}^F\|_W$ to approximate the empirical neural distances D_{ij}^N. This approximation is optimized to maximize the Spearman correlation ρ between the theoretical and neural distances. The norm is parametrized with a learned Symmetric Positive Definite (SPD) matrix $W \in \mathbb{S}_m^{++}$ so as to be a valid metric (distance function). Crucially, the off-diagonal coefficients of W allow for direct modeling of interactions between features (e.g., between grammatical number and gender) since they may have a combined effect on neural representations.

D: Permutation Feature Importance is applied to a fitted MLEM model to assess the importance of each feature and their interactions in predicting neural distances. In practice, the importance is the average decrease in Spearman correlation ρ when the feature values are permuted.

2.1 Metric Learning Encoding Models (MLEM)

Formally, consider n stimuli $(s_i)_{i \leq n}$ (e.g., sentences, images) for which values are available for each of m theoretical features $(f_k)_{k \leq m}$ (e.g., linguistic features, visual categories, confounds). The features can be continuous, discrete, categorical, ordered or not, and even multi-dimensional. Assume that high-dimensional (neural) representations $Y \in \mathbb{R}^{n \times d}$ (e.g., fMRI images, LLM embeddings) are available for each stimulus s_i.

2.1.1 Representational Dissimilarity Matrices (RDMs)

Similarly to RSA, MLEM relies on computing RDMs (pairwise distances) between stimuli. On one hand, in the neural space for a given distance (e.g. Euclidean):

$$D^N_{ij} = \|Y_i - Y_j\|_2$$

Where Y_i is the neural representation of stimulus s_i (e.g. fMRI image, LLM embedding). The choice of Euclidean distance is standard in many representation analysis studies. It assumes that neural representations form a vector space where proximity reflects similarity. While other distance metrics (e.g., cosine distance) could be used, Euclidean distance is a general-purpose choice that is sensitive to both the angle and magnitude of representation vectors.

On the other hand, feature distances for all pairs of stimuli can be computed. This work mainly explored nominal features (unordered and categorical) and therefore defined the distance between two stimuli to 1 if they have different values with respect to a given feature and 0 otherwise. For ordered features, the absolute difference between the values of the two stimuli is used:

$$D^k_{ij} = \begin{cases} |f_k(s_i) - f_k(s_j)| & \text{if } f_k \text{ is ordered} \\ \\ \mathbb{1}\left[f_k(s_i) \neq f_k(s_j)\right] & \text{otherwise} \end{cases}$$

This choice is arbitrary and other distances could be used depending on the nature of the features. For instance, multidimensional features could be used along with the Euclidean distance.

Note that some features might be underspecified, or undetermined, for some stimuli (e.g. gender value in the case of plural pronouns in English; 'we', 'they'..). Here, in practice, this is implemented with NaN values and the distance is set to 0 if either stimulus of a pair is NaN. Here too, other choices can be made depending on one's dataset and goals.

Finally, let D^F_{ij} be the m-dimensional vector of feature distances for a pair of stimuli (s_i, s_j):

$$D^F_{ij} = \left(\; D^{f_1}_{ij} \; , \; D^{f_2}_{ij} \; , \; \ldots \; , \; D^{f_m}_{ij} \; \right)$$

It summarizes the theoretical distance between a pair of stimuli with respect to all features.

2.1.2 Metric learning formulation

The main goal of MLEMs is to learn a metric (distance) function, which approximates the neural distances from the theoretical distances. The metric function is defined over all features, and it is parametrized using a weight matrix W.

We define MLEM as a metric function implemented via a weighted norm $\| \bullet \|_W$. That is, given a vector D^F_{ij} for the theoretical distance between a pair of stimuli, we define the following metric function:

$$\widehat{D^N_{ij}} = \|D^F_{ij}\|_W = \sqrt{(D^F_{ij})^T W (D^F_{ij})}$$

The norm in this metric is parametrized by a matrix W, which weighs each feature and each feature-feature interaction. We optimize W to maximize the Spearman correlation ρ between the empirical neural distances

D^N_{ij} and the modeled ones $\widehat{D^N_{ij}}$:

$$\sup_{W \in \mathbb{S}_m^{++}} \rho_{i<j} \left(\widehat{D^N_{ij}} \ , \ D^N_{ij} \right)$$

Note that the weight matrix needs to be symmetric positive definite $W \in \mathbb{S}_m^{++}$ for the norm to be valid (i.e. to satisfy the properties of a metric, such as positive definiteness and the triangle inequality). This Metric Learning component is the core of MLEM.

The choice of Spearman correlation as the optimization objective is motivated by its robustness. As a rank-based measure, it is invariant to monotonic transformations of the distances and less sensitive to outliers than other objectives like Pearson correlation or Mean Squared Error (MSE). Pearson correlation assumes a linear relationship between the predicted and observed distances, which may be too restrictive. MSE would aim for a direct match in distance values, making the model highly sensitive to the scale of the distances and to outliers. Furthermore, Spearman correlation provides a score between -1 and 1, which is more directly interpretable than unbounded metrics like MSE or R^2 (not lower-bounded). While Kendall's Tau is another robust rank-based correlation, Spearman correlation is often preferred for its computational efficiency on larger datasets. By optimizing for rank correlation, MLEM focuses on preserving the relative arrangement of stimuli in the representation space, which is a core principle of second-order isomorphism methods.

2.2 Model Training

The model is trained using stochastic gradient descent. Further details on the optimization procedure are provided in Section B.

2.2.1 Batch size and breaking free from quadratic complexity

The major limitation of methods relying on pairwise comparisons between stimuli, such as computing RDMs, is the quadratic complexity. However here, the matrix of weights W is optimized through stochastic gradient descent on batches of pairs of stimuli. A training step only requires a subset of the RDMs corresponding to a batch of pairs. This can be computed on the fly from the values of the features $f_k(s_i)$ and the high-dimensional representations Y. Empirically, the optimization of the weight matrix W often converges before the entire RDM has been explored. This removes the need for the computation of all pairwise distances which can be prohibitive for large datasets.

One question remains: how to choose the batch size? Such a parameter should be small for computing efficiency but large enough to provide a good coverage of the values of the theoretical features, in particular the sparse ones and their interactions. In practice we used the smallest batch size that would allow for computing correlations between the features with variability below a given threshold. This criterion ensures that the gradient estimates during optimization are based on a representative sample of feature co-occurrences, which is necessary for stable learning of the interaction terms in W. More details about this procedure can be found in Section C.

2.3 Evaluation and Permutation Feature Importance

Models are evaluated using Spearman correlation ρ (higher is better) between the predicted neural distances $\widehat{D^N_{ij}}$ and the empirical ones D^N_{ij} on pairs of held-out stimuli.

Furthermore, we want to quantify the importance of each feature for the model, defined as how much each feature helps approximate neural distances. To this end, *Permutation Feature Importance* (Breiman, 2001) is used. It measures the importance of a feature as the average degradation in model performance when permuting the values of the feature across samples. This method is more reliable than looking directly at the weights of the model and provides quantities which are straightforward to interpret. For instance, in this setting, an importance of 0.3 for one feature means that the model loses 0.3 of Spearman correlation when essentially destroying this feature. This provides a clear and interpretable measure of importance, unlike metrics such as R^2 or MSE, for which the scale of the importance score is less intuitive.

More formally, assume a set of n input samples $x_i \in \mathbb{R}^d$, with corresponding targets $y_i \in \mathbb{R}^{d'}$, a (trained) model $M : \mathbb{R}^d \to \mathbb{R}^{d'}$ that predicts targets from inputs, and a score function $S : \mathbb{R}^{n \times d'} \times \mathbb{R}^{n \times d'} \to \mathbb{R}$ that evaluates the model performance. Permutation Feature Importance breaks the relationship between an input feature and the target by permuting the feature values across all samples. If S_{baseline} is the score of the model without any permutation:

$$S_{\text{baseline}} = S\Big(M\Big(X\Big), Y\Big)$$

and S_p^k is the score of the model after applying permutation p to the values of feature f_k across all samples:

$$S_p^k = S\left(M\left(X_p^{f_k}\right), Y\right)$$

then the importance of feature f_k is quantified as:

$$\mathbb{E}_p\left[S_{\text{baseline}} - S_p^k\right]$$

In practice, we find low variability across permutations therefore we approximate the expectation over p by averaging over only 10 random permutations for computational efficiency. The Permutation Feature Importance framework is extended to quantify the importance of feature interactions, which correspond to the off-diagonal terms in the weight matrix W. This involves reshaping the input space to treat each interaction as a separate feature. A detailed explanation of this procedure is provided in Section D.

Ablation studies (leave-one-covariate-out, Lei et al. (2018)) are another approach to feature importance which are more computationally expensive since they require retraining the model. They are also unsuitable here since ablating/removing a single feature would require removing its interactions as well to be able to retrain with the parametrization on W. Therefore it would be impossible to isolate the importance of a single feature or an interaction.

However Permutation Feature Importance is prone to false detections in the presence of high correlations between the features (Strobl et al., 2008). Other methods are more robust to this, such as feature grouping (Chamma et al., 2024) and Conditional Permutation Importance (CPI, Strobl et al. (2008)), but they are not applicable here. In the NLP examples, grouping features would lead to groups that are hard to interpret. Regarding CPI, the gist of this method is to permute only the *independent* part of the features with respect to the other ones. To do so, the method relies on training a predictor to reconstruct a feature based on the others and then permuting the residuals (intrinsic information from the feature). In the MLEM framework, since the objective is to model feature interactions, the residuals would be null. Indeed, predicting the interactions between f_1 and f_2 given at least f_1 and f_2 is very easy for a general predictor.

2.4 Baseline: FR-RSA with interactions (FR-RSA-I)

By design MLEM has parameters for each feature interaction (off-diagonal elements of W) and Permutation Feature Importance allows their importance to be quantified. It can be critical to account for feature interactions in domains such as natural language processing. For a fair comparison, an augmented variant of FR-RSA that includes interaction terms between features is considered as a state-of-the-art baseline. We denote it as **FR-RSA-I**. In practice, FR-RSA-I is the same as MLEM without the symmetric positive definite (SPD) constraint on W. It is hypothesized that enforcing this constraint in MLEM is beneficial for a more reliable estimation of W. The model is constrained to be a distance, which should help predict the target distance in neural space more precisely and more sample efficiently. It is also hypothesized that it makes the model more robust to noise and overfitting.

2.5 Datasets

MLEM is evaluated in two setups, using two types of datasets: (1) synthetic datasets and (2) a fully annotated dataset of English sentences for which representations are extracted from a Large Language Model (LLM). The reason to evaluate MLEM on synthetic datasets is that the true effects of features, and their interactions, on neural distances are known. That is, all values of the elements of W are known, to which we refer as the *ground-truths*. This allows assessment of MLEM's ability to recover a known underlying metric. The sentence dataset, on the other hand, provides a more realistic setting to test the method on empirical data.

Table 1: **Relative Clause dataset design.** 2x2 design emphasizing center-embedded vs peripheral and subject vs object relative clauses.

Attachment Site	Relative Clause Type	Template / Example
Center-Embedded	**Subject relative**	*Subj [who verb obj] verb obj.* The woman [who sees the girl] likes the queen.
	Object relative	*Subj [who subj verb] verb obj.* The woman [who the princess sees] admires the actress.
Peripheral (Right-Branching)	**Subject relative**	*Subj verb obj [who verb obj].* The woman sees the princess [who admires the actress].
	Object relative	*Subj verb obj [who subj verb].* The woman admires the actress [who the princess sees].

2.5.1 Simulated datasets

To evaluate MLEM's ability to recover a ground-truth metric over features and their interactions, synthetic datasets are generated with a known ground-truth weight matrix W_{GT}. We consider $n = 256$ stimuli (s_i) and $m = 16$ binary theoretical features (f_k). The features have two categories, "A" and "B", which are randomly assigned to each stimulus, such that $f_k(s_i)$ takes the value "A" with a probability of 0.5 and "B" otherwise. From these features, feature RDMs D^k_{ij} are derived using an indicator distance (1 if feature values differ, 0 otherwise).

To establish a true underlying metric in the feature space, a ground truth symmetric positive definite matrix $W_{GT} \in \mathbb{S}^{++}_m$ is generated using `make_spd_matrix` from `scikit-learn` (Pedregosa et al., 2011) and normalized to have a Frobenius norm of 1 for comparability across datasets. To generate neural representations that reflect this feature space, pairwise neural distances $D^{N_{GT}}_{ij}$ between stimuli s_i and s_j are computed as:

$$D^{N_{GT}}_{ij} = \|D^F_{ij}\|_{W_{GT}} = \sqrt{(D^F_{ij})^T W_{GT} D^F_{ij}}$$

where $D^F_{ij} = [D^f_{ij}]_f$ is the vector of feature distances for the pair of stimuli (s_i, s_j). To create high-dimensional representations, Multidimensional Scaling (MDS) is used to generate $d = 768$-dimensional representations Y whose pairwise distances closely match $D^{N_{GT}}$. To simulate realistic neural variability, Gaussian noise is added to these representations. This noise is scaled by the standard deviation of each dimension of the generated representations and an additional configurable noise level parameter called "Noise level" in the figures. Finally, to obtain the actual neural distances D^N_{ij}, the pairwise Euclidean distances between these noisy, high-dimensional representations are computed.

Models are trained on five datasets generated with different random seeds, using an 80% training and 20% testing split. Metrics are reported as their average (full line) and their standard deviation (shaded area) across the five seeds. Figures display the metrics measured on the test set when applicable.

2.5.2 Relative Clause Sentence Dataset

To test and compare MLEM and FR-RSA-I on real data, we use a dataset of generated sentences emphasizing a specific linguistic structure: relative clauses. Following syntactic theory, relative clauses create sentences that are relatively complex, in particular object relative clauses due to a syntactic movement. We focus are interested in relative clauses since this movement is expected to generate large distances in neural space. We therefore contrast sentence stimuli with and without such syntactic movement. In addition, we mark all sentence stimuli also based on other, presumably simpler, features, such as grammatical number (singular/plural).

A dataset of 7.7k sentences was generated with manually crafted grammars to emphasize the relative clause linguistic structure in a 2x2 design: center-embedded vs peripheral (right-branching) and subject vs object relative clauses. This design is summarized in Table 1. The dataset has 12 features which are displayed in Table S1 along with their values, the distance function used, and a brief description. Feature RDMs are then computed using either the indicator function (categorical values) or the absolute difference (continuous values) depending on the feature.

Permutation Feature Importance is known to be sensitive to high correlations between features. Because importances are harder to interpret under such correlations, the dataset was carefully balanced to mitigate them. As a result, the correlations between the feature RDMs are negligible (c.f. Figure S1).

For this sentence dataset, representations are taken from a well-studied Large Language Model: BERT (Devlin et al., 2019). More precisely, the pre-trained `bert-base-uncased` model from the `transformers` library (Wolf et al., 2020). BERT is fed with the sentences and hidden states are extracted from its 12 layers. These hidden states are averaged over the token dimension to get fixed-size sentence embeddings, which is necessary for the method. These embeddings are then regarded as the neural representations of the sentences for BERT. Neural distances D^N_{ij} are computed as the Euclidean distance between these sentence embeddings. To also evaluate robustness to noise in this more realistic setting, we added artificial Gaussian noise to the BERT representations in a similar way as the simulation. The noise is scaled by the standard deviation of the representations for each dimension and a configurable noise level parameter, called "Artificial Noise level" in the figures.

5-fold cross-validation is applied and metrics are reported as their average (full line) and their standard deviation (shaded area) across the five folds. Figures display the metrics measured on the test set when applicable.

In practice we run the batch size estimation procedure described in Section 2.2.1 and Section B with a conservative set of parameters and obtain a large batch size of ∼20,000 for this dataset for high accuracy.

3 Results

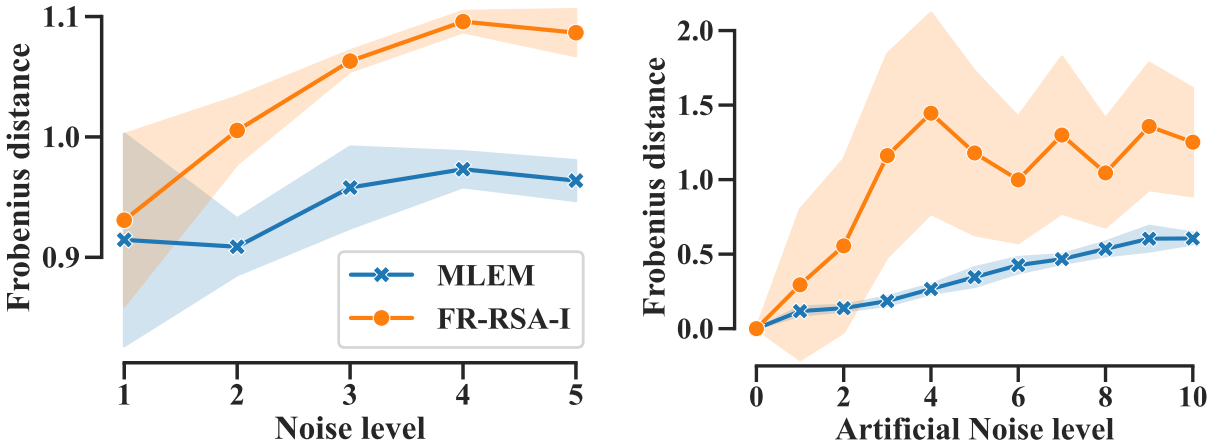
3.1 MLEMs are more accurate and robust to noise compared to the strongest baseline method

A key difference between MLEM and FR-RSA-I is that MLEM enforces the weight matrix W to be symmetric positive definite (SPD), constraining it to define a valid metric. We hypothesize that this constraint provides regularization that leads to more accurate weight recovery and greater robustness to noise, as demonstrated in the following results.

Simulation Results. First, we used simulated datasets where the ground-truth weight matrix W_{GT} is known (see Section 2). Figure 2a shows the Frobenius distance between the estimated weights and the ground-truth weights for both methods across different levels of noise. MLEM consistently achieves a lower Frobenius distance, indicating that its estimated weights are closer to the ground truth and demonstrating its superior accuracy in recovering the true feature geometry.

Experiments with LLMs. Second, to evaluate robustness, we measured how much the estimated weights changed when artificial noise was added to the neural representations from layer 7 of the LLM. Figure 2b shows the Frobenius distance between weights estimated with and without this noise. The weights from MLEM show a smaller distance to their noiseless counterpart across all noise levels, demonstrating the superior stability and robustness of MLEM.

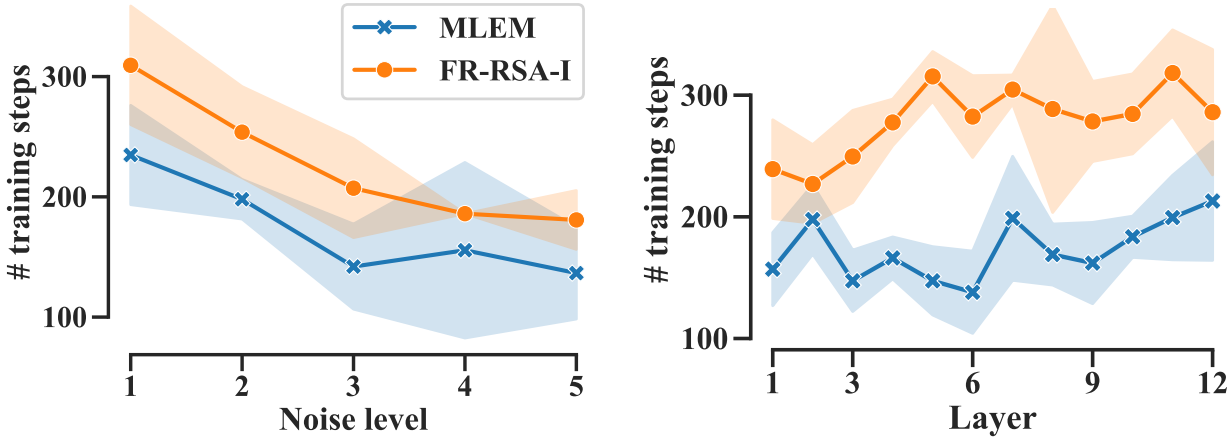
To provide a more detailed view of this robustness, we examined the profiles of learned weights and feature importances under increasing levels of artificial noise (Figures S4 and S5). For both weights and feature importances, the profiles for FR-RSA-I degrade more rapidly than for MLEM, confirming MLEM's superior stability. To quantify the stability of the feature rankings, we used Kendall's weighted τ to compare feature importance profiles at each noise level to the noiseless profile (Figure S6). MLEM maintains a consistently higher correlation, indicating that its feature importance rankings are more reliable in noisy conditions.



(a) Accuracy on simulated data.

(b) Robustness to artificial noise.

Figure 2: **MLEM is more accurate and robust to noise than FR-RSA-I.** Frobenius distance at different noise levels between estimated weights and **(a)** ground-truth weights in synthetic data (accuracy), and **(b)** estimation at noise level 0 on LLM embeddings (robustness). MLEM (blue) consistently achieves lower error than FR-RSA-I (orange). In both plots, the full line represents the average and the shaded area represents the standard deviation across 5 runs.



(a) Simulated data.

(b) Relative clause dataset with LLM embeddings.

Figure 3: **MLEM converges faster than FR-RSA-I.** The plots compare the number of training steps (epochs) of MLEM (blue) and FR-RSA-I (orange). **(a)** On simulated datasets, MLEM converges faster across all noise levels. **(b)** On the relative clause dataset with LLM embeddings, MLEM converges faster for each of the 12 layers. This demonstrates MLEM's superior computational efficiency. The full line represents the average and the shaded area represents the standard deviation across 5 runs.

3.2 MLEMs converge faster than baseline methods

We next compared the convergence speed of MLEMs and FR-RSA-I. Figure 3 shows the number of training steps for both methods on the simulated datasets (Figure 3a) and the relative clause dataset with LLM embeddings (Figure 3b). In both cases, MLEM converges faster than FR-RSA-I. As our approach does not explore the whole RDMs before converging (see Section 2.2.1), this result also means that MLEM is more data efficient.

As shown in Figure S14, MLEM's speed advantage is most pronounced at low artificial noise levels but both methods seem to converge at the same speed when noise increases. However MLEM still shows less variability in the number of training steps across runs.

3.3 Encoding performance is comparable between MLEM and FR-RSA-I

MLEM and FR-RSA-I obtain similar final encoding performance measured by Spearman correlation between predicted and observed neural distances on held-out stimuli (see Figure S9). Therefore MLEM recovers more faithful feature and interactions weights, yields more reliable feature importance profiles, and converges faster, while preserving encoding power.

3.4 MLEMs reveal importance profiles of the theoretical feature across layers of an LLM

To understand how the LLM represents sentences, we train one MLEM per layer and investigate the importance of features and interactions obtained.

For readability, we display a subset of the 12 features from the relative clause dataset. We selected the top features in terms of both highest importance and absolute weight across layers for MLEM and FR-RSA-I, resulting in 4 features and one interaction. Figure 4 shows the profiles of feature importances across layers and Figure S2 shows the absolute weights. The importance curves show interesting patterns. For instance "Relative Clause type" and "Attachment site" achieve their highest scores in the middle layers, consistent with prior work suggesting that LLMs process syntax predominantly in middle layers (in particular Hewitt & Manning (2019) for BERT).

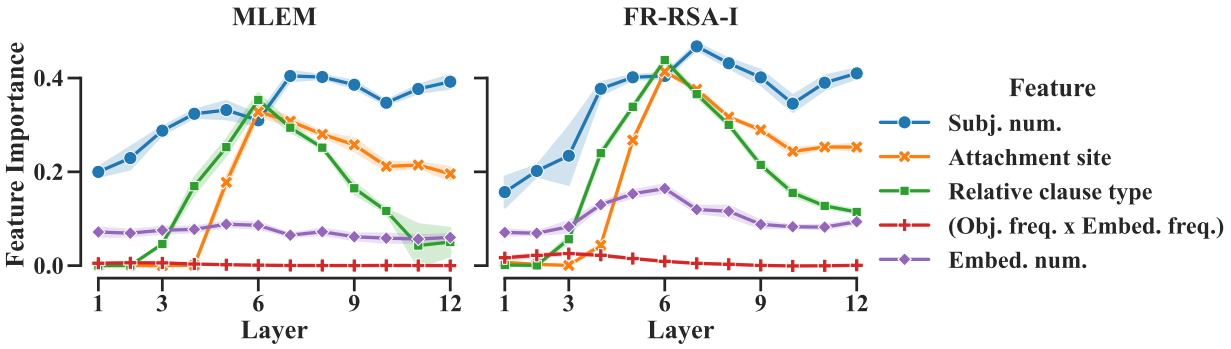


Figure 4: **MLEM and FR-RSA-I identify similar profiles of feature importance across LLM layers.** The plots show the top feature importances for MLEM (left) and FR-RSA-I (right) across the 12 layers of the LLM. Each line represents a feature or interaction, and its y-value indicates its importance at a given layer. Both methods reveal a similar pattern: features related to sentence structure, such as "Relative Clause type" and "Attachment site", gain importance in the middle layers, which is consistent with prior work on LLM's syntactic processing. This suggests that both methods can uncover meaningful linguistic information. The full line represents the average and the shaded area represents the standard deviation across 5 cross-validation folds.

3.5 Feature Importance of MLEMs reveal complex geometrical structures in LLM representations

To verify that importance values correspond to meaningful geometric structures, we visualized the representations in 2D using Multidimensional Scaling (MDS, Kruskal (1964)). The layer-wise feature importance profiles reflect the geometry of the neural representation. Figure 5 shows that a clear clustering based on "Relative Clause type" emerges in the middle layers, where its feature importance peaks in Figure 4. This validates that feature importance profiles produced by MLEM are informative about the geometry of neural representations.

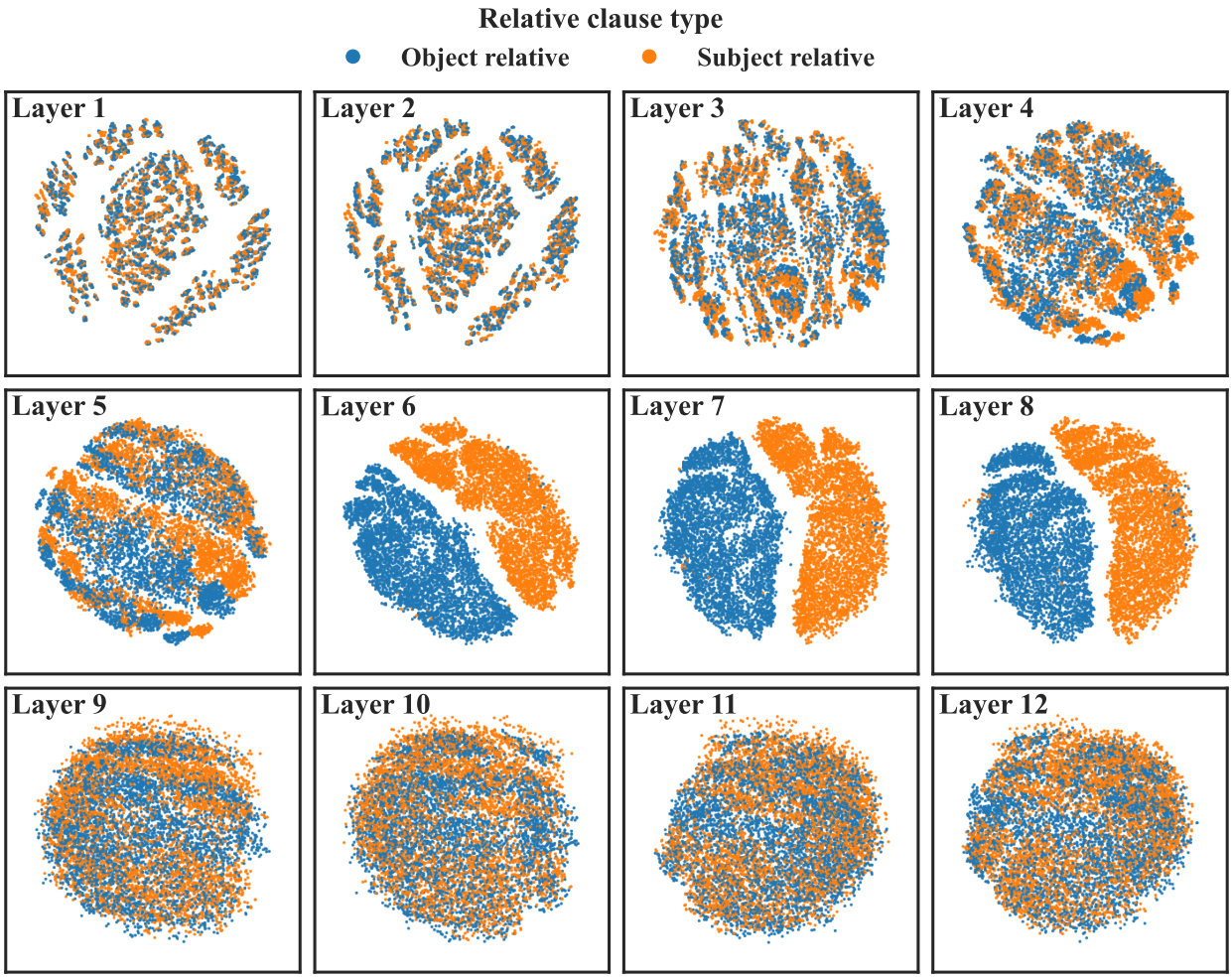


Figure 5: **The geometry of LLM representations reflects syntactic structure, especially in middle layers.** This figure shows 2D MDS visualizations of sentence representations from each of the LLM's 12 layers. Points are colored by the "Relative Clause type" feature (object relative: blue, subject relative: orange). A clear clustering based on this feature emerges in the middle layers, where the two colors become separable. This geometric separation aligns with the peak importance of the "Relative Clause type" feature shown in Figure 4, validating that the feature importances identified by the approach correspond to tangible structures in neural space.

Furthermore, the relative feature importances of different features reflect a nested-structure geometry. Figure 6 demonstrates at the layer 6, the most important features define a hierarchical clustering. The space is first organized by "Relative Clause type," then by "Attachment site," and finally by "Subject number," offering an interpretable visualization of how abstract linguistic properties are hierarchically encoded.

Figure S7 compares MLEM and FR-RSA-I across layers (weights and feature importance agreement), and Figure S8 extends this comparison under artificial noise. Finally, Figure S12 contrasts the multivariate MLEM with unit-wise univariate MLEMs, showing the benefit of modeling distributed representations.

To contextualize MLEM's informativeness against alternative approaches, we compared it with both decoding and multivariate encoding baselines. A standard decoding approach, using linear classifiers to predict features from representations, confirms that feature information is present but reveals little about how representational geometry evolves across layers, as accuracies remain uniformly high (Figure S10). In contrast, a multivariate random forest encoding baseline recovers a feature importance profile similar to MLEM's but cannot model feature interactions by design and lacks the robustness conferred by MLEM's metric learning

framework (Section H). These comparisons underscore MLEM's unique advantage in providing detailed and robust insights into the structure of neural representations.

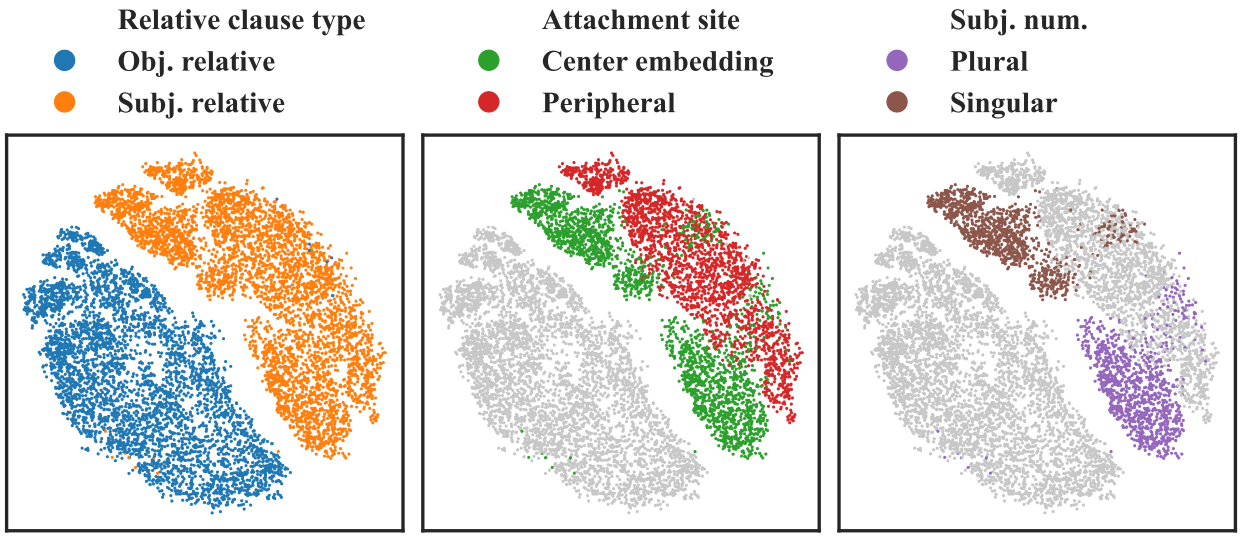


Figure 6: **Top features from MLEM reveal a hierarchical clustering in the space of neural representation.** This figure visualizes the sentence representations from layer 6 of the LLM, demonstrating how the top three features identified by MLEM for this layer organize the space hierarchically. First, representations cluster by "Relative Clause type" (subject relative: blue, object relative: orange). Within the subject-relative cluster, a secondary split emerges based on "Attachment site" (center-embedded: green, peripheral: red). Finally, within the subject-relative, center-embedded cluster, a third level of organization appears based on "Subject number" (plural: purple, singular: brown). This hierarchical structure, discovered using MLEM's feature importances, provides a clear and interpretable view of the geometry of neural representations. See Figure S13 for a complete view.

4 Discussion

This work introduced Metric Learning Encoding Models (MLEM), a multivariate encoding framework that learns an optimal metric over a feature space to model the geometry of neural representations. Experiments on simulated and real-world data demonstrate that MLEM offers several advantages over existing methods like FR-RSA: it is more accurate in recovering ground-truth feature weights (Figure 2a), more robust to noise (Figure 2b), and converges faster (Figures 3a and 3b) which makes it more data-efficient. By combining this with Permutation Feature Importance, MLEM reliably quantifies the contribution of both individual features and their interactions to neural representations, revealing interpretable geometric structures in high-dimensional data (Figures 5 and 6).

4.1 Advantages of multivariate encoding over alternative approaches

MLEM's multivariate approach offers several key advantages over decoding methods and univariate encoding approaches. Decoding models (or "probes"), while demonstrating that feature-related information is present, are susceptible to confounds where a feature may be decodable simply because it correlates with another, truly encoded feature (Hewitt & Liang, 2019; Belinkov, 2022; Kumar et al., 2022a). Moreover, as shown in Figure S10, decoding accuracies remain uniformly high across layers and provide little insight into how representational geometry changes, highlighting a key limitation of this approach.

Univariate encoding approaches, which fit each neural unit independently, cannot model distributed representations encoded across populations of units, a critical characteristic of neural computation (Georgopoulos et al., 1986). The comparison with univariate MLEMs (Figure S12) demonstrates that the multivariate

approach successfully captures geometric structure encoded across multiple units, which individual univariate models cannot access. This is particularly important for understanding complex domains like language, where information is often distributedly encoded across neural populations rather than localized in individual units.

4.2 Modeling feature interactions through metric learning

A central contribution of MLEM is its principled approach to modeling feature interactions through metric learning. Feature-Reweighted RSA (FR-RSA) can be seen as a special case of MLEM where the weight matrix is constrained to be diagonal, corresponding to independent feature contributions without interactions. Even when FR-RSA is artificially extended to capture interactions (FR-RSA-I), it fails to match MLEM's performance because the set of weights is not constrained to form a valid metric. While our analysis of LLM representations did not reveal strong, interpretable interactions, the ability to model them is crucial. Even if interactions are not of primary interest, their inclusion can lead to more accurate estimates of the main effects of individual features, as demonstrated in our simulations.

The symmetric positive definite (SPD) constraint in MLEM ensures that the learned weights define a valid metric, which serves multiple purposes. First, it provides theoretical grounding by guaranteeing that the weighted distance satisfies metric properties such as positive definiteness and the triangle inequality. Second, this constraint functions as a form of regularization, guiding the optimization toward more stable and interpretable solutions. As demonstrated in our results (Figure 2), this constraint is crucial for achieving superior accuracy in recovering ground-truth weights and increased robustness to noise compared to FR-RSA-I. Without this constraint, as in FR-RSA-I, the learned weights may not correspond to a meaningful geometric structure, making the model more prone to overfitting and less robust to noise. The uncoupled estimation of interaction terms in FR-RSA-I can impair their accuracy, as shown in our simulations. The superior performance of MLEM in recovering ground-truth weights and its increased robustness can be attributed to this principled metric learning formulation.

4.3 Robustness to noise in neural data

Neural data, whether from biological systems or artificial networks, inevitably contains noise from various sources including measurement error, biological variability, and computational precision. The robustness analysis (Figures 2b, S4 and S5) demonstrates that MLEM maintains more stable weight estimates and feature importance profiles under increasing noise levels compared to FR-RSA-I. This robustness stems from the SPD constraint, which regularizes the learning process and prevents the model from fitting to spurious noise patterns. Additionally, the use of Spearman correlation as the optimization objective makes the approach less sensitive to outliers and non-linear relationships between feature and neural geometries. This is because Spearman correlation operates on ranks, making it invariant to monotonic transformations of the distances and robust to extreme values. This choice also makes the resulting scores and feature importances more interpretable than alternatives like MSE or R^2. This robustness is crucial for real-world applications where neural measurements are inherently noisy.

4.4 Computational efficiency and practical impact

The computational efficiency of MLEM stems from multiple innovations that make it convenient for both large and small datasets.

First, the use of stochastic gradient descent for the optimization along with the on-the-fly computation of batches of data (Section 2.2.1) avoids the quadratic complexity of computing full RDMs. For instance on the relative clause dataset, MLEM converges in less than 200 steps (Figure 3b). With the estimated batch size of 20,000 pairs, this means it explored at most 4 million of the $\sim$30 million possible pairs, or less than 15% of the full RDMs. This makes MLEM scalable to datasets with thousands or even millions of stimuli for which computing and storing full RDMs would be computationally intractable with traditional RSA-based methods. Furthermore, convergence without needing to see all stimulus pairs makes MLEM more data-efficient. This means that for limited data scenarios, which is common in neuroimaging studies where

acquisition is expensive and time-consuming, MLEM converges to a better solution with the same amount of data.

Second, as demonstrated in Figures 3a and 3b, MLEM converges faster than FR-RSA-I. The sole difference between the two methods is the SPD constraint, hence the speed advantage can be directly attributed to this principled parametrization. By constraining the optimization to the manifold of valid metrics, the SPD parametrization guides gradient descent toward meaningful solutions more efficiently than the unconstrained approach of FR-RSA-I. Therefore the SPD constraint provides an additional performance boost in addition to its regularization benefits.

4.5 Comparing MLEM with similar approaches from the literature

The closest variant of RSA to MLEM is *Representational Similarity Learning* (RSL, Oswal et al. (2016)). RSL is a decoding approach: given human similarity judgments S and neural measurements X, it seeks a sparse and symmetric weight matrix W such that $XWX^T \approx S$. In contrast, MLEM is an encoding approach, neural distances are predicted from feature distances. Furthermore, its focus is on the neural encoding of multiple theoretical features while RSL was applied to decode a single matrix of human similarity judgments.

Another related method is Mixed RSA (Khaligh-Razavi et al., 2017), which combines voxel-receptive-field (RF) modeling with RSA. Mixed RSA first learns a linear mapping from model features to individual voxel responses on a training set, and then computes an RDM from the predicted voxel responses on a test set. This predicted RDM is then compared to the empirical RDM. While both MLEM and Mixed RSA aim to find a better alignment between feature-based and neural representations, they operate at different levels. Mixed RSA is a hybrid approach that first builds a first-order encoding model (features to activations) and then performs a second-order comparison (RSA). In contrast, MLEM is a pure second-order model that directly learns a metric on the feature space to match the geometry of the neural space, without modeling individual neural units. This makes MLEM more directly focused on the representational geometry and less dependent on the quality of individual unit predictions.

A similar metric learning approach was also adopted more practically in the study of phoneme perception, based on behavioral phoneme-confusion data (Lakretz et al., 2018).

Finally, other multivariate encoding approaches include Reduced-Rank Ridge Regression (RRR, Mukherjee & Zhu (2011)) and Back-to-back (B2B) regression (King et al., 2020). Both are first-order encoding models that predict neural activations directly from features, unlike MLEM, which is a second-order model that operates on distances. RRR regularizes the linear mapping from features to neural activations by enforcing a low-rank constraint on the coefficient matrix. This is conceptually different from MLEM's approach of learning a metric on the feature space. B2B regression aims to disentangle correlated features by chaining a decoding model (from neural data to features) with an encoding model (from features back to the decoder's predictions). While it addresses feature collinearity, it does not explicitly model feature interactions or the geometry of the representation space in the way MLEM does.

4.6 Broader impact and future applications

The MLEM framework is modality-agnostic and opens several avenues for future research across multiple domains. In neuroscience, applying MLEM to human brain data (fMRI, MEG, ECoG) could help bridge the gap between findings from computational models and the principles of neural computation in biological systems. In artificial intelligence, MLEM provides a powerful tool for interpreting the internal representations of deep learning models across various modalities.

The method's ability to uncover hierarchical clustering based on feature importance (Figure 6) demonstrates its potential for discovering interpretable structure in high-dimensional representations. This capability could prove valuable for understanding how abstract concepts are organized in both artificial and biological neural systems.

4.7 Limitations

Large and Heterogeneous Datasets As explained in Section 2.2.1, the implementation of MLEM uses a subsampling strategy to avoid the quadratic complexity of pairwise comparisons. Optimization often converges long before the entire set of stimulus pairs has been seen. This efficiency, however, poses a risk for very large and heterogeneous datasets. If convergence is reached too early, some informative stimulus pairs, especially those related to sparse features, may not be sampled, potentially biasing the learned metric. While the adaptive batch sizing procedure (Section 2.2.1) is designed to mitigate this by ensuring feature correlations are stable within batches, it may lead to computationally expensive large batches for highly diverse datasets.

Weighting Units in the Neural Space One could consider extending MLEM to learn weights for individual units (e.g., neurons or voxels) in the neural space, similar to some variants of RSA (Oswal et al., 2016). For instance one could learn a diagonal matrix λ to compute a weighted neural distance: $D^N_{ij} = \|Y_i - Y_j\|_\lambda$. While this could improve the alignment between feature and neural spaces (i.e., achieve a higher Spearman correlation), it would come at the cost of interpretability. The resulting feature importances would explain a distorted or degenerated neural space, making it difficult to draw conclusions about the original representations. For spatial data like fMRI, a searchlight analysis remains a more advisable approach for localizing feature information. If such a feature were to be implemented, regularization on λ would be necessary to prevent it from becoming overly sparse, which would amount to selecting only a few neural units.

5 Conclusion

This work presented MLEM, a novel framework for interpreting high-dimensional representations by modeling their geometry as a learned metric over theoretical features. By explicitly accounting for feature interactions through a principled metric learning approach and leveraging a robust optimization objective, MLEM provides more accurate, stable, and efficient interpretations than previous methods. Its effectiveness has been demonstrated in both controlled simulations and in analyzing the representations of a large language model. MLEM represents a significant advance over existing approaches by combining the strengths of encoding models with the ability to capture distributed neural representations and modeling feature interactions. The framework's versatility makes it a powerful tool for researchers seeking to uncover the principles of neural information processing in both artificial and biological systems.

References

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, volume 2 of *ASPLOS '24*, pp. 929–947, New York, NY, USA, April 2024. Association for Computing Machinery. ISBN 979-8-4007-0385-0. doi: 10.1145/3620665.3640366. URL https://dl.acm.org/doi/10.1145/3620665.3640366.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. Probing for Constituency Structure in Neural Language Models. arXiv, 2022. doi: 10.48550/ARXIV.2204.06201. URL https://arxiv.org/abs/2204.06201.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, April 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Yonatan Belinkov and James Glass. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, April 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00254. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00254/43503/Analysis-Methods-in-Neural-Language-Processing-A.

Mathieu Blondel, O. Teboul, Quentin Berthet, and Josip Djolonga. Fast Differentiable Sorting and Ranking. *ArXiv*, February 2020. URL https://www.semanticscholar.org/paper/Fast-Differentiable-Sorting-and-Ranking-Blondel-Teboul/5eef6a00d9eab08f3071ef19ea3e4b545421e8cb.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Shael Brown and Reza Farivar. The Topology of Representational Geometry, February 2024. URL https://www.biorxiv.org/content/10.1101/2024.02.16.579506v1.

C. Caucheteux, Alexandre Gramfort, and J. King. Long-range and hierarchical language predictions in brains and algorithms. *ArXiv*, November 2021. URL https://www.semanticscholar.org/paper/Long-range-and-hierarchical-language-predictions-in-Caucheteux-Gramfort/fc915ec260e3511963c2d17092db16f0a3742882.

Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):134, February 2022. ISSN 2399-3642. doi: 10.1038/s42003-022-03036-1. URL https://www.nature.com/articles/s42003-022-03036-1.

Ahmad Chamma, Bertrand Thirion, and Denis Engemann. Variable Importance in High-Dimensional Settings Requires Grouping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11195–11203, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i10.28997. URL https://ojs.aaai.org/index.php/AAAI/article/view/28997.

Davide Crepaldi, Marco Marelli, and Simona Amenta. For a probabilistic and multidisciplinary approach to the investigation of morphological processing. *Cortex*, 116:1–3, July 2019. ISSN 00109452. doi: 10.1016/j.cortex.2019.05.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0010945219302072.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL http://aclweb.org/anthology/N19-1423.

Pablo Diego-Simón, Stéphane D'Ascoli, Emmanuel Chemla, Yair Lakretz, and Jean-Rémi King. A polar coordinate system represents syntax in large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pp. 105375–105396, Red Hook, NY, USA, December 2024. Curran Associates Inc. ISBN 979-8-3313-1438-5.

Apostolos P. Georgopoulos, Andrew B. Schwartz, and Ronald E. Kettner. Neuronal Population Coding of Movement Direction. *Science*, 233(4771):1416–1419, September 1986. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.3749885. URL https://www.science.org/doi/10.1126/science.3749885.

Ariel Goldstein, Avigail Grinstein-Dabush, Mariano Schain, Haocheng Wang, Zhuoqiao Hong, Bobbi Aubrey, Samuel A. Nastase, Zaid Zada, Eric Ham, Amir Feder, Harshvardhan Gazula, Eliav Buchnik, Werner Doyle, Sasha Devore, Patricia Dugan, Roi Reichart, Daniel Friedman, Michael Brenner, Avinatan Hassidim, Orrin Devinsky, Adeen Flinker, and Uri Hasson. Alignment of brain embeddings and artificial contextual embeddings in natural language points to common geometric patterns. *Nature Communications*, 15(1):2768, March 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-46631-y. URL https://www.nature.com/articles/s41467-024-46631-y.

John Hewitt and Percy Liang. Designing and Interpreting Probes with Control Tasks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275. URL https://www.aclweb.org/anthology/D19-1275.

John Hewitt and Christopher D. Manning. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North*, pp. 4129–4138, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL http://aclweb.org/anthology/N19-1419.

Eghbal A. Hosseini, Martin Schrimpf, Yian Zhang, Samuel Bowman, Noga Zaslavsky, and Evelina Fedorenko. Artificial Neural Network Language Models Predict Human Brain Responses to Language Even After a Developmentally Realistic Amount of Training. *Neurobiology of Language*, 5(1):43–63, April 2024. ISSN 2641-4368. doi: 10.1162/nol_a_00137. URL https://direct.mit.edu/nol/article/5/1/43/119156/Artificial-Neural-Network-Language-Models-Predict.

James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. Disentangling Semantics and Syntax in Sentence Embeddings with Pre-trained Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1372–1379, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.108. URL https://aclanthology.org/2021.naacl-main.108.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research*, 61:907–926, April 2018. ISSN 1076-9757. doi: 10.1613/jair.1.11196. URL http://jair.org/index.php/jair/article/view/11196.

Alexander G. Huth, Wendy A. De Heer, Thomas L. Griffiths, Frédéric E. Theunissen, and Jack L. Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458, April 2016. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature17637. URL https://www.nature.com/articles/nature17637.

Kamila M. Jozwik, Nikolaus Kriegeskorte, Katherine R. Storrs, and Marieke Mur. Deep Convolutional Neural Networks Outperform Feature-Based But Not Categorical Models in Explaining Object Similarity Judgments. *Frontiers in Psychology*, 8:1726, October 2017. ISSN 1664-1078. doi: 10.3389/fpsyg.2017.01726. URL http://journal.frontiersin.org/article/10.3389/fpsyg.2017.01726/full.

Philipp Kaniuth and Martin N. Hebart. Feature-reweighted representational similarity analysis: A method for improving the fit between computational models, brains, and behavior. *NeuroImage*, 257:119294, August 2022. ISSN 10538119. doi: 10.1016/j.neuroimage.2022.119294. URL https://linkinghub.elsevier.com/retrieve/pii/S105381192200413X.

Seyed-Mahdi Khaligh-Razavi, Linda Henriksson, Kendrick Kay, and Nikolaus Kriegeskorte. Fixed versus mixed RSA: Explaining visual representations by fixed and mixed feature sets from shallow and deep computational models. *Journal of Mathematical Psychology*, 76:184–197, February 2017. ISSN 0022-2496. doi: 10.1016/j.jmp.2016.10.007. URL https://www.sciencedirect.com/science/article/pii/S0022249616301134.

Jean-Rémi King, François Charton, David Lopez-Paz, and Maxime Oquab. Back-to-back regression: Disentangling the influence of correlated factors from multivariate observations. *NeuroImage*, 220:117028, October 2020. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2020.117028. URL https://www.sciencedirect.com/science/article/pii/S1053811920305140.

Nikolaus Kriegeskorte and Pamela K Douglas. Interpreting encoding and decoding models. *Current Opinion in Neurobiology*, 55:167–179, April 2019. ISSN 0959-4388. doi: 10.1016/j.conb.2019.04.002. URL https://www.sciencedirect.com/science/article/pii/S0959438818301004.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, November 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008. URL https://www.frontiersin.org/journals/systems-neuroscience/articles/10.3389/neuro.06.004.2008/full.

J. B. Kruskal. Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika*, 29(1):1–27, March 1964. ISSN 0033-3123, 1860-0980. doi: 10.1007/BF02289565. URL https://www.cambridge.org/core/product/identifier/S003331230004206X/type/journal_article.

Brian Kulis. Metric Learning: A Survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000019. URL http://www.nowpublishers.com/articles/foundations-and-trends-in-machine-learning/MAL-019.

Abhinav Kumar, Chenhao Tan, and Amit Sharma. Probing classifiers are unreliable for concept removal and detection. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 17994–18008, Red Hook, NY, USA, November 2022a. Curran Associates Inc. ISBN 978-1-7138-7108-8.

Sreejan Kumar, Theodore R. Sumers, Takateru Yamakoshi, Ariel Goldstein, Uri Hasson, Kenneth A. Norman, Thomas L. Griffiths, Robert D. Hawkins, and Samuel A. Nastase. Shared functional specialization in transformer-based language models and the human brain. June 2022b. doi: 10.1101/2022.06.08.495348. URL http://biorxiv.org/lookup/doi/10.1101/2022.06.08.495348.

Yair Lakretz, Gal Chechik, Evan-Gary Cohen, A. Treves, and N. Friedmann. Metric Learning for Phoneme Perception. *ArXiv*, September 2018. URL https://www.semanticscholar.org/paper/Metric-Learning-for-Phoneme-Perception-Lakretz-Chechik/9022a60866a999cf1882834fa0df209e5bf6d2c1.

Jean-Benoist Leger. Parametrization Cookbook: A set of Bijective Parametrizations for using Machine Learning methods in Statistical Inference, January 2023. URL http://arxiv.org/abs/2301.08297.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113(523):1094–1111, July 2018. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1307116. URL https://www.tandfonline.com/doi/full/10.1080/01621459.2017.1307116.

Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The Geometry of Concepts: Sparse Autoencoder Feature Structure. October 2024. URL https://www.

semanticscholar.org/paper/The-Geometry-of-Concepts%3A-Sparse-Autoencoder-Li-Michaud/f64f0b8d2b0886196aa38edff165f3797c76108a.

I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, November 2017. URL https://www.semanticscholar.org/paper/Decoupled-Weight-Decay-Regularization-Loshchilov-Hutter/d07284a6811f1b2745d91bdb06b040b57f226882.

Ashin Mukherjee and Ji Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(6):612–622, December 2011. ISSN 1932-1864, 1932-1872. doi: 10.1002/sam.10138. URL https://onlinelibrary.wiley.com/doi/10.1002/sam.10138.

Subba Reddy Oota, Manish Gupta, and Mariya Toneva. Joint processing of linguistic properties in brains and language models. 2022. doi: 10.48550/ARXIV.2212.08094. URL https://arxiv.org/abs/2212.08094.

Urvashi Oswal, Christopher R. Cox, M. L. Ralph, T. Rogers, and R. Nowak. Representational Similarity Learning with Application to Brain Networks. In *International Conference on Machine Learning*, June 2016. URL https://www.semanticscholar.org/paper/Representational-Similarity-Learning-with-to-Brain-Oswal-Cox/111666d0d30d160f0d5c842bee9253fc043e93c1.

Christophe Pallier, Anne-Dominique Devauchelle, and Stanislas Dehaene. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108(6):2522–2527, February 2011. doi: 10.1073/pnas.1018711108. URL https://www.pnas.org/doi/abs/10.1073/pnas.1018711108.

Alexandre Pasquiou, Yair Lakretz, John Hale, Bertrand Thirion, and Christophe Pallier. Neural Language Models are not Born Equal to Fit Brain Data, but Training Helps, July 2022. URL http://arxiv.org/abs/2207.03380.

Alexandre Pasquiou, Yair Lakretz, Bertrand Thirion, and Christophe Pallier. Information-Restricted Neural Language Models Reveal Different Brain Regions' Sensitivity to Semantics, Syntax, and Context. *Neurobiology of Language*, 4(4):611–636, December 2023. ISSN 2641-4368. doi: 10.1162/nol_a_00125. URL https://direct.mit.edu/nol/article/4/4/611/117823/Information-Restricted-Neural-Language-Models.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Stephan Pohl, Edgar Y. Walker, David L. Barack, Jennifer Lee, Rachel N. Denison, Ned Block, Florent Meyniel, and Wei Ji Ma. Clarifying the conceptual dimensions of representation in neuroscience. March 2024. URL https://www.semanticscholar.org/paper/Clarifying-the-conceptual-dimensions-of-in-Pohl-Walker/10b609f2efad9f17292cdd485a0b3a589c6ab0c1.

David E. Rumelhart, James L. McClelland, and AU. Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. The MIT Press, 1986. ISBN 978-0-262-29140-8. doi: 10.7551/mitpress/5236.001.0001. URL https://direct.mit.edu/books/book/4424/Parallel-Distributed-ProcessingExplorations-in-the.

Martin Schrimpf, Idan Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua Tenenbaum, and Evelina Fedorenko. The neural architecture of language: Integrative modeling converges on predictive processing. June 2020. doi: 10.1101/2020.06.26.174482. URL http://biorxiv.org/lookup/doi/10.1101/2020.06.26.174482.

Katherine R. Storrs, Tim C. Kietzmann, Alexander Walther, Johannes Mehrer, and Nikolaus Kriegeskorte. Diverse Deep Neural Networks All Predict Human Inferior Temporal Cortex Well, After Training and Fitting. *Journal of Cognitive Neuroscience*, pp. 1–21, August 2021. ISSN 1530-8898. doi: 10.1162/jocn_a_01755. URL https://direct.mit.edu/jocn/article/doi/10.1162/jocn_a_01755/103001/Diverse-Deep-Neural-Networks-All-Predict-Human.

Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307, December 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-307. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-307.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? Probing for sentence structure in contextualized word representations. *ArXiv*, May 2019. URL https://www.semanticscholar.org/paper/What-do-you-learn-from-context-Probing-for-sentence-Tenney-Xia/e2587eddd57bc4ba286d91b27c185083f16f40ee.

Sebastiano Vigna. A Weighted Correlation Index for Rankings with Ties. *Proceedings of the 24th International Conference on World Wide Web*, pp. 1166–1176, May 2015. doi: 10.1145/2736277.2741088. URL https://dl.acm.org/doi/10.1145/2736277.2741088.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. Van Der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul Van Mulbregt, SciPy 1.0 Contributors, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius De Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2.

Leila Wehbe, Brian Murphy, Partha Talukdar, Alona Fyshe, Aaditya Ramdas, and Tom Mitchell. Simultaneously Uncovering the Patterns of Brain Regions Involved in Different Story Reading Subprocesses. *PLoS ONE*, 9(11):e112575, November 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0112575. URL https://dx.plos.org/10.1371/journal.pone.0112575.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing: System Demonstrations, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Junhai Xu, Haibin Dong, Na Li, Zeyu Wang, Fei Guo, Jianguo Wei, and Jianwu Dang. Weighted RSA: An Improved Framework on the Perception of Audio-visual Affective Speech in Left Insula and Superior Temporal Gyrus. *Neuroscience*, 469:46–58, August 2021. ISSN 0306-4522. doi: 10.1016/j.neuroscience.2021.06.002. URL https://www.sciencedirect.com/science/article/pii/S0306452221002876.

Supplementary Information

A Relative Clause dataset

A.1 Dataset construction and balancing

The Relative Clause dataset was generated to emphasize syntactic variation in a 2×2 design that manipulates embedding site (center-embedded vs peripheral) and relative-clause role (subject vs object). Sentences were produced using controlled grammars to fill a combinatorial space of linguistic features; the final dataset contains approximately 7.7k sentences and 12 features (listed in Table S1). Features include binary syntactic properties (e.g., relative-clause type, attachment site), morphological attributes (subject/object/embedded number and gender), lexical frequency bins (Zipf scores), and the verb lemma. The dataset was balanced with respect to these features to avoid spurious correlations between feature RDMs; Figure S1 shows the (near-zero) correlations across feature RDMs, indicating a well-balanced design where permutation-based importances remain interpretable.

Table S1: Linguistic features and their values in the Relative Clause dataset.

Feature	**Values**	**Distance**	**Description**
Relative Clause type	Subject, Object	Indicator	Whether the relative clause modifies the subject or the object of the main clause.
Attachment site	Peripheral, Center-embedded	Indicator	Position of the relative clause relative to the main clause.
Subject number	Singular, Plural	Indicator	Grammatical number of the main clause subject.
Subject gender	Feminine, Masculine	Indicator	Grammatical gender of the main clause subject.
Subject frequency	4.0, 4.5, 5.0, 5.5	Euclidean	Zipf frequency of the main clause subject.
Object number	Singular, Plural	Indicator	Grammatical number of the main clause object.
Object gender	Feminine, Masculine	Indicator	Grammatical gender of the main clause object.
Object frequency	4.0, 4.5, 5.0, 5.5	Euclidean	Zipf frequency of the main clause object.
Embedded number	Singular, Plural	Indicator	Grammatical number of the embedded clause subject.
Embedded gender	Feminine, Masculine	Indicator	Grammatical gender of the embedded clause subject.
Embedded frequency	4.0, 4.5, 5.0, 5.5	Euclidean	Zipf frequency of the embedded clause subject.
Verb lemma	admire, see	Indicator	Lemma of the verb.

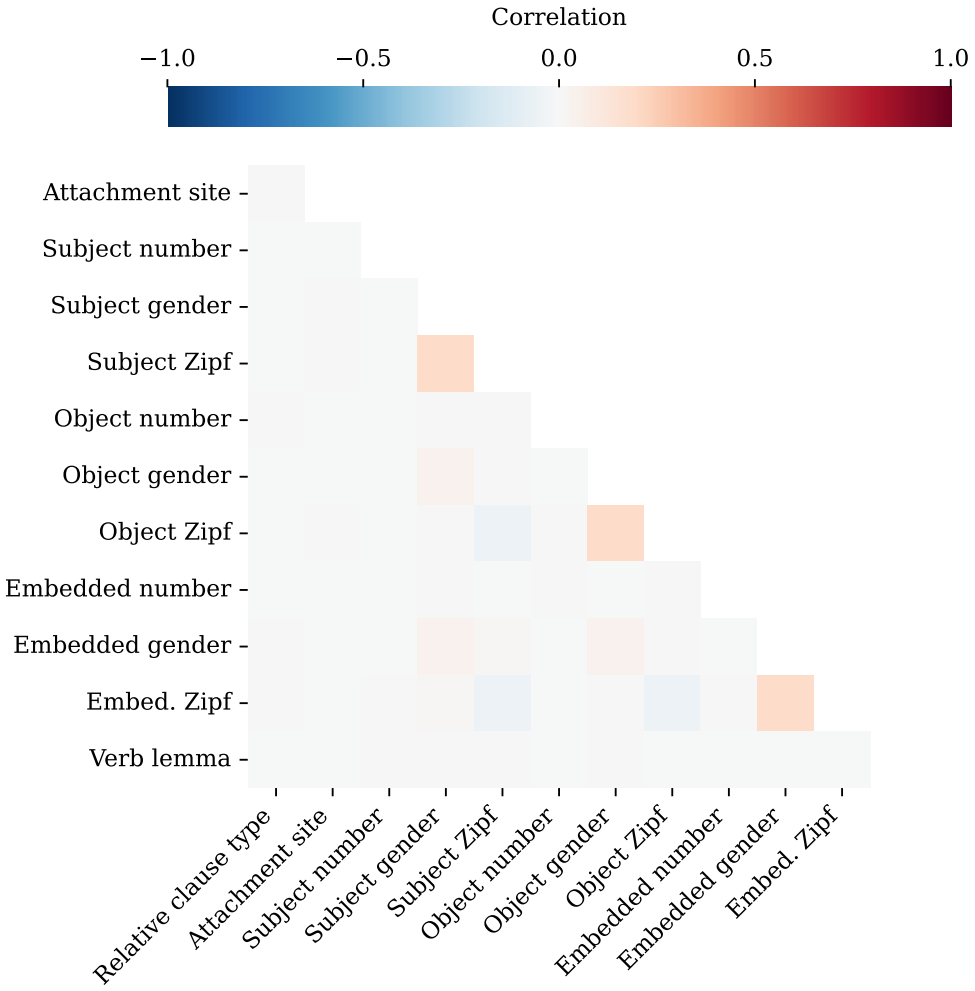


Figure S1: **Feature RDMs in the Relative Clause dataset are uncorrelated.** This heatmap shows the Pearson correlation between the Representational Dissimilarity Matrices (RDMs) of the 12 linguistic features in the dataset. The correlations are computed over all pairs of stimuli. The low values across the matrix (the largest being 0.125) indicate that the dataset is well-balanced and that the features are largely independent. This is crucial for the interpretability of methods like Permutation Feature Importance, as it minimizes the risk of attributing importance to confounded features.

B Optimization Details

Optimizing the Spearman correlation while constraining the weight matrix W to be symmetric positive definite (SPD) is a non-trivial problem. This optimization is performed through stochastic gradient descent on a differentiable relaxation of the Spearman correlation introduced by Blondel et al. (2020) and we used the optimized PyTorch implementation from torchsort.

To enforce the SPD constraint on the weight matrix W, a parametrization based on the Cholesky decomposition is used, as described in the Parametrization Cookbook (Leger, 2023). Any SPD matrix W can be uniquely decomposed as $W = LL^T$, where L is a lower-triangular matrix with strictly positive diagonal entries. Instead of optimizing W directly under constraints, an unconstrained matrix A is optimized. L is then constructed from A by taking the lower-triangular part of A and ensuring its diagonal entries are positive by applying the softplus function. Specifically, $L_{ij} = A_{ij}$ for $i > j$, $L_{ii} = \text{softplus}(A_{ii}) = \log(1 + e^{A_{ii}})$, and $L_{ij} = 0$ for $i < j$. The resulting matrix $W = LL^T$ is guaranteed to be SPD. This construction is differentiable so A can be optimized even if W is used in practice. A is implemented as an `nn.Linear` module from the PyTorch library (Ansel et al., 2024).

The `AdamW` optimizer (Loshchilov & Hutter, 2017) is used with a learning rate of 0.1 and no weight decay. Training stops when the optimized Spearman has not increased for 50 steps or after a maximum of 1000 steps.

For numerical stability the weight matrix W is normalized to have Frobenius norm 1 at each step, and online min–max scaling is applied to neural distances D^N so they remain in [0, 1]. These normalizations do not change the optimization problem as Spearman correlation is invariant to scaling.

C Batch Size Selection

To select an appropriate batch size for stochastic gradient descent, a preprocessing procedure is run on the feature values to estimate *correlation variability*. The goal is to find a batch size that is small enough for computational efficiency but large enough to provide a stable estimate of the correlations between feature distances. Formally, $p = 64$ batches are sampled: $B_1, ..., B_p$. Starting with a small batch size $b = 4096$, each batch B_r is a set of b pairs of indices:

$$B_r = \{(i_1, j_1), ..., (i_b, j_b)\}$$

The corresponding feature (and interaction) distances are computed and correlations between them are evaluated within the different batches. Then, if the individual variabilities of those correlations across the p batches are not too high (e.g., standard deviation below 0.01), this batch size is kept; otherwise it is increased by multiplying by a factor $\gamma = 1.2$ ($b \leftarrow b \times \gamma$) and the estimation is restarted. In practice, this preprocessing step can be unbounded for imbalanced datasets with very sparse features and relaxing the threshold may be necessary.

D Permutation Feature Importance Details

A trained MLEM model is defined by a matrix $W \in \mathbb{S}_m^{++}$ that can be used to predict neural distances D_{ij}^N from feature distances D_{ij}^F as follows:

$$D_{ij}^N \sim \widehat{D_{ij}^N} = \|D_{ij}^F\|_W = \sqrt{\sum_{1 \leq k,l \leq m} W_{kl} D_{ij}^k D_{ij}^l}$$

To obtain feature importance for the interactions between features (corresponding to off-diagonal W_{kl}), the input space of the MLEM is reshaped to fit the Permutation Feature Importance framework. Instead of considering the m-dimensional vector D_{ij}^F, it is transformed to an m_p-dimensional vector P_{ij} where $m_p = \frac{m(m+1)}{2}$ is the number of pairs of features (including pairs (f, f) corresponding to a single feature). For an

index $1 \leq k' \leq m_p$ corresponding to a pair of features (k, l) with $k \leq l$ define:

$$P_{ij}^{k'} = \begin{cases} D_{ij}^k D_{ij}^l & \text{if } k = l \\ 2D_{ij}^k D_{ij}^l & \text{if } k \neq l \end{cases}$$

The factor 2 accounts for symmetry and avoids duplication. Then the MLEM model can be rewritten as a mapping $h_W : \mathbb{R}^{m_p} \to \mathbb{R}$:

$$h_W(P_{ij}) = \sqrt{\sum_{1 \leq k' \leq m_p} W_{kl} \, P_{ij}^{k'}}$$

With the same correspondence between k' and (k, l) as above. Considering different pairs of stimuli as samples, Permutation Feature Importance can be applied to h_W with Spearman correlation as the score to obtain importance for each feature and for each interaction between them.

E Weights

In addition to permutation importances, the absolute value of learned weights is inspected to understand the model's parameterization. Figure S2 reports absolute weights for the top features across BERT layers for MLEM and FR-RSA-I. While weight profiles are broadly similar across methods and layers, raw weights can be misleading. A large absolute weight does not necessarily imply that a feature explains the geometry; it may reflect parameter scaling or compensate for other learned interactions. For instance, the interaction between object frequency and embedded frequency has a large absolute weight for both methods but negligible importance.

The MDS visualization in Figure S3 provides an example where weight magnitudes suggested a strong interaction, but the geometry did not clearly reflect the purported clustering. This discrepancy suggests that feature importance is a more reliable measure than weights for understanding the geometry of neural representations.

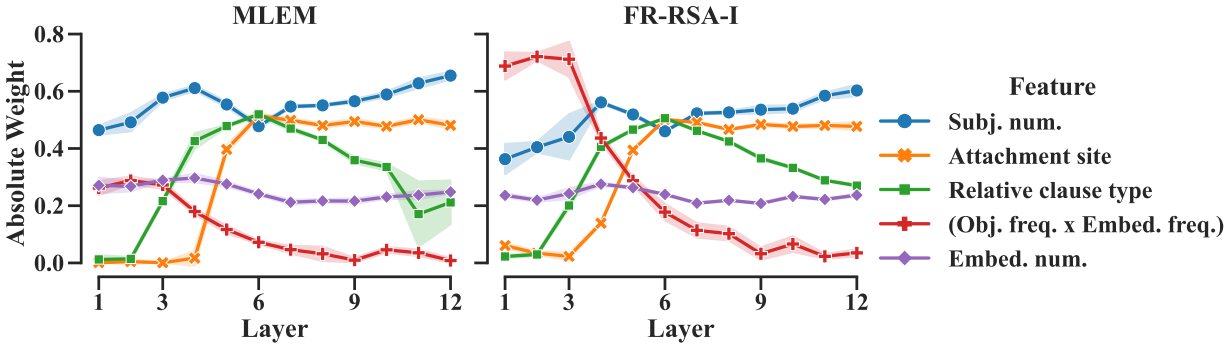


Figure S2: **Weight profiles are similar across methods but can be misleading.** This figure shows the absolute values of the learned weights for top features in FR-RSA-I (left) and MLEM (right) across BERT's 12 layers. While the profiles are broadly similar, raw weights are not always a reliable indicator of feature importance. For example, the interaction between "object frequency" and "embedded frequency" receives a large weight from both methods, but its permutation importance is negligible. This suggests that large weights can arise from model-internal parameter adjustments rather than true explanatory power.

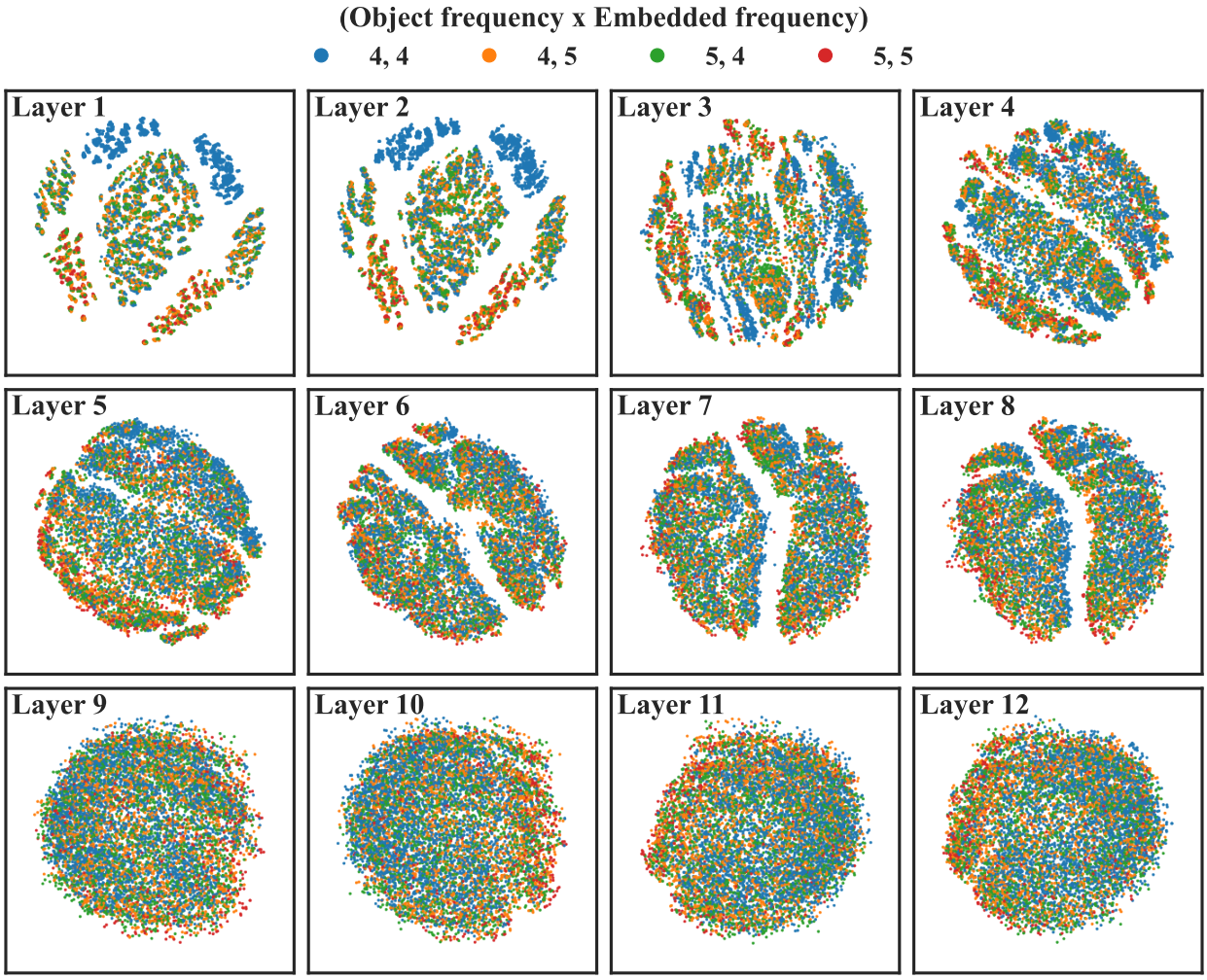


Figure S3: **Large weights do not always correspond to clear geometric structure.** This figure shows 2D MDS visualizations of BERT's sentence representations, colored by the interaction of "object frequency" and "embedded frequency". The clustering with respect to this interaction in early layers is not as clear as its weight seemed to suggest in Figure S2, in particular by FR-RSA-I. This provides a concrete example of why permutation feature importance is a more reliable measure than raw weights for interpreting the geometry of neural representations.

F Quantitative comparison of methods

To provide a more rigorous comparison between methods, we introduce quantitative measures of similarity for both weight and feature importance profiles. These measures can be used to compare results from different methods (MLEM vs. FR-RSA-I), different models, or under varying conditions like noise level.

For weights, we use the Frobenius distance between two weight matrices. This is used to compare estimated weights to ground truth in simulations (Figure 2a) and to assess robustness to noise (Figure 2b).

For feature importance profiles, standard correlation measures like Spearman's ρ are suboptimal as they treat all features equally. The agreement on top-ranking features is of particular interest, as these have the most explanatory power. Misranking features with low importance is less critical than misranking features with high importance.

Kendall's weighted τ (Vigna, 2015) is a rank correlation coefficient that addresses this by assigning more weight to disagreements involving higher-ranked items. Given n items (features in our case) and two lists of scores for those items (feature importances in our case), R and S, it is defined as:

$$\tau_w(R, S) = \frac{\sum_{i<j} w(i,j)\text{sgn}(R_i - R_j)\text{sgn}(S_i - S_j)}{\sum_{i<j} w(i,j)}$$

where the weight $w(i, j)$ is a function of the ranks of items i and j. A classical choice is hyperbolic weighting (Vigna, 2015) which we describe next. Let $\rho_R(i)$ be the rank of item i in list R (starting from 0). The weight for a pair of items (i, j) can be based on their ranks in list R:

$$w_R(i,j) = \frac{1}{\rho_R(i)+1} + \frac{1}{\rho_R(j)+1}$$

Similarly, a weight function w_S can be defined based on the ranks in S. To ensure symmetry, the final correlation τ_h is the average of the correlations computed with each weighting scheme:

$$\tau_h(R, S) = \frac{\tau_{w_R}(R,S) + \tau_{w_S}(R,S)}{2}$$

This symmetric approach ensures the measure does not depend on an arbitrary choice of which ranking to use for weights. In practice, `weightedtau` from the `scipy` library (Virtanen et al., 2020) is used, which implements this hyperbolic weighting scheme by default.

F.1 Robustness to noise

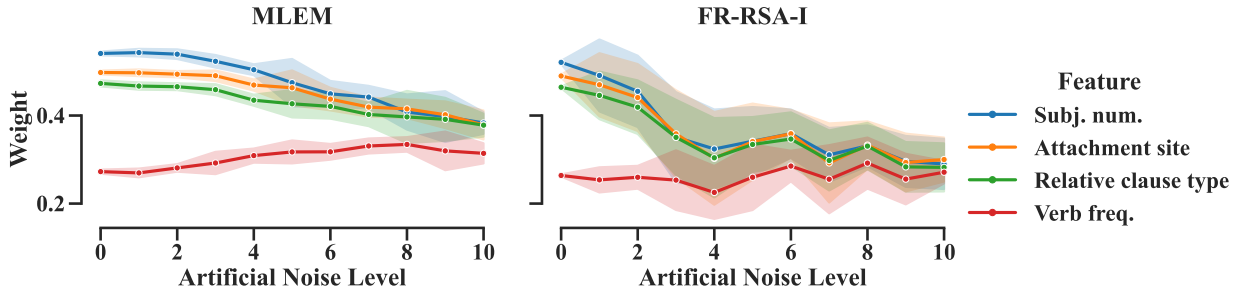


Figure S4: **MLEM's weight profile is more stable under noise than FR-RSA-I's.** The plots show the learned weights for top features in BERT's 7th layer for MLEM (left) and FR-RSA-I (right) under increasing levels of artificial noise. While both methods start with similar profiles at zero noise, the weight profile of FR-RSA-I degrades more rapidly as noise increases, showing greater instability. MLEM's profile remains more consistent, highlighting its superior robustness.

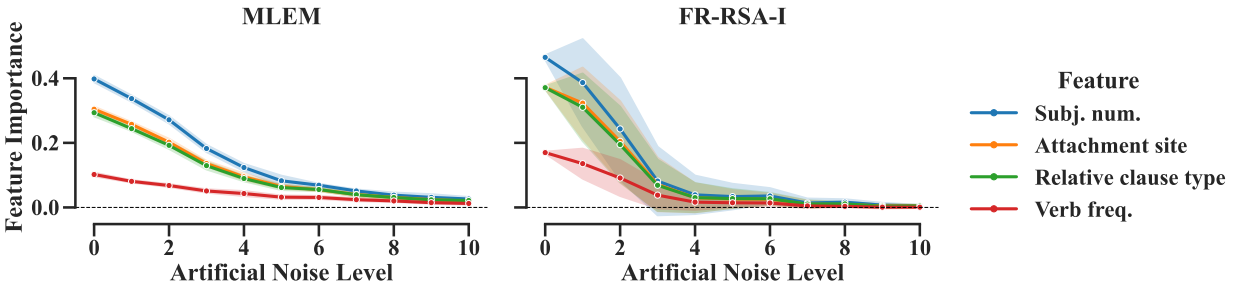


Figure S5: **MLEM's feature importance profile is more robust to noise.** This figure compares the feature importance profiles for FR-RSA-I (left) and MLEM (right) on BERT's 7th layer as artificial noise is added. Similar to the weight profiles, the feature importance profile for FR-RSA-I shows degradation with increasing noise. In contrast, MLEM's feature importance profile remains relatively stable, indicating that its interpretations are more reliable in noisy conditions.

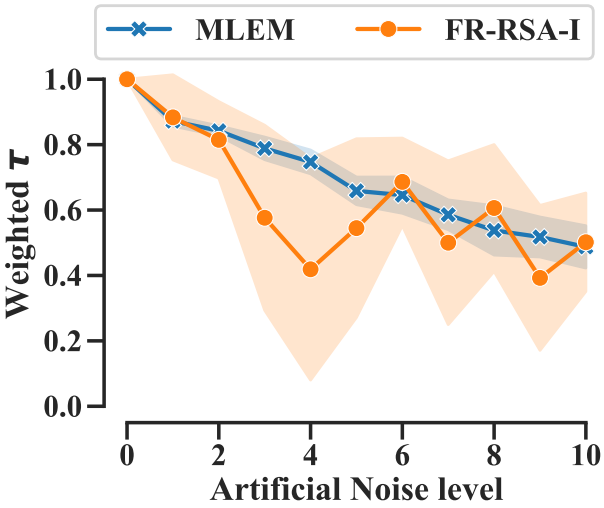


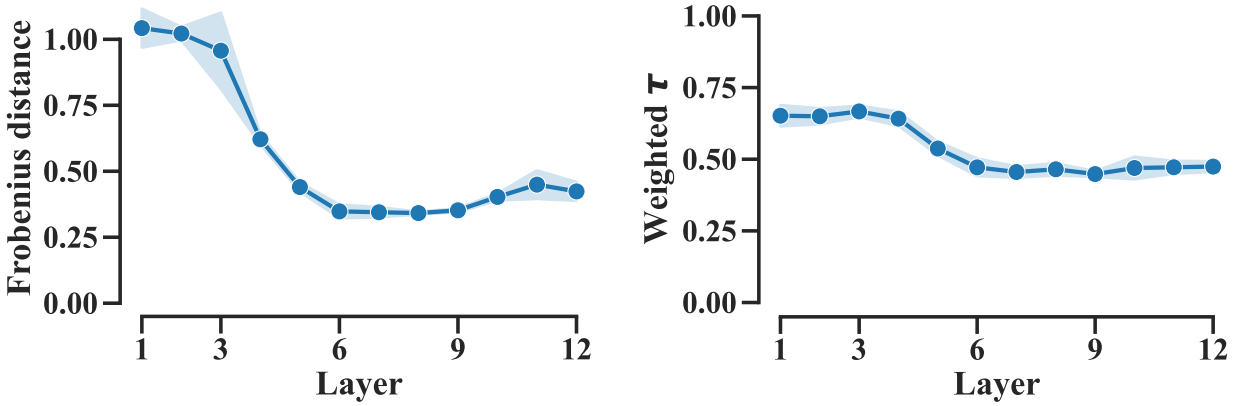
Figure S6: **MLEM's feature importance rankings are more stable against noise.** This plot shows Kendall's weighted τ correlation between the feature importance profile at each noise level and the noiseless feature importance profile, for MLEM (blue) and FR-RSA-I (orange). A higher correlation indicates greater stability. MLEM maintains a higher and more stable correlation as noise increases, whereas FR-RSA-I's correlation drops and shows high variability. This quantifies MLEM's superior robustness in preserving feature importance rankings under noise.

F.2 Comparing MLEM and FR-RSA-I

To further compare MLEM and FR-RSA-I, the Frobenius distance between their weight profiles and the Kendall's weighted τ between their feature importance profiles are computed across all BERT layers and noise levels.

Figure S7a shows that the Frobenius distance between the weight matrices of the two methods decreases in the middle and higher layers of BERT, suggesting they converge to more similar weight solutions in these layers. Conversely, Figure S7b shows that the correlation between their feature importance profiles is highest in the early layers and decreases in later layers. This indicates that while the raw weights may become more similar, the resulting interpretations in terms of feature importance diverge. Given the earlier finding that feature importance is a more reliable measure, this divergence is significant.

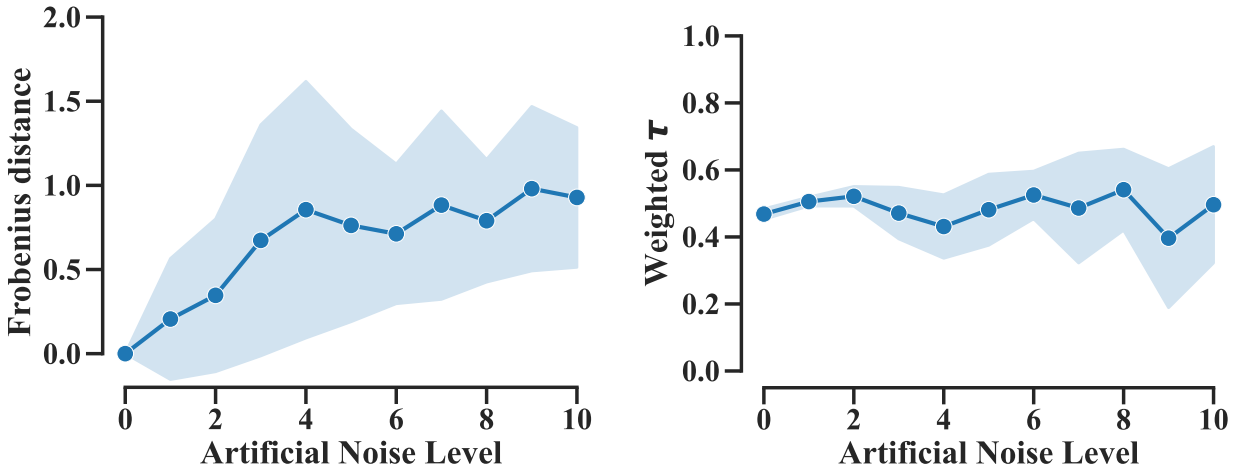
When examining the effect of noise (Figure S8a and Figure S8b), the weight matrices of the two methods diverge as noise increases, as expected. However, the agreement on feature importances remains relatively stable, albeit with increased variability.



(a) Dissimilarity of weight matrices.

(b) Agreement on feature importance.

Figure S7: **Comparison of MLEM and FR-RSA-I across BERT layers. (a)** The Frobenius distance between the weight matrices learned by MLEM and FR-RSA-I decreases in later BERT layers, suggesting the models converge to more similar weight solutions. **(b)** In contrast, the agreement on feature importance (Kendall's weighted τ) is highest in early layers and decreases thereafter. This suggests that even when raw parameters become more similar, their interpretations in terms of feature importance can diverge.



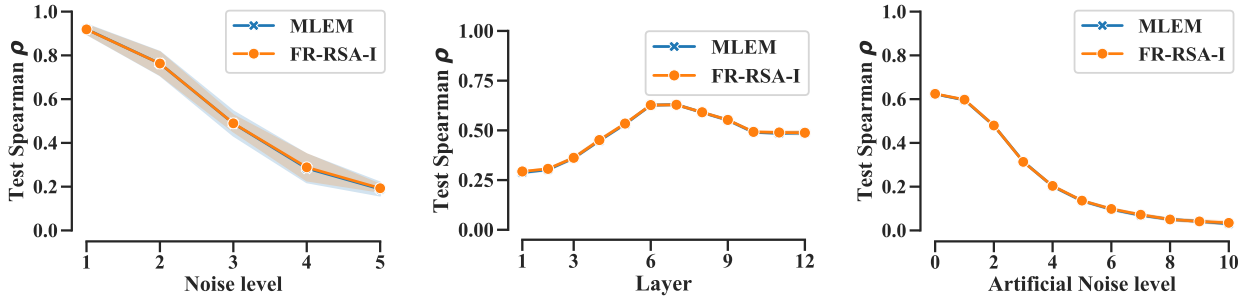
(a) Dissimilarity of weight matrices.

(b) Agreement on feature importance.

Figure S8: **Comparison of MLEM and FR-RSA-I under noise. (a)** The Frobenius distance between the weight matrices of MLEM and FR-RSA-I for BERT's 7th layer increases with noise, indicating they diverge more in noisier conditions. **(b)** Despite the growing dissimilarity in their weight matrices, the agreement on the ranking of important features (Kendall's weighted τ) remains relatively stable, though with increased variability.

F.3 MLEM and FR-RSA-I obtain the same encoding performance

Although MLEM yields improved recovery of weight structure, improved robustness, and faster convergence, the two methods obtain similar final encoding performance measured by Spearman correlation on pairs of held-out stimuli (see Figure S9). That is, both methods achieve comparable alignment to observed neural distances in terms of standard predictive scores. The additional contributions of MLEM are therefore interpretability and stability: MLEM recovers more faithful feature interactions (by Frobenius to ground truth), yields more reliable feature importance profiles, and converges faster, while preserving encoding power.



(a) Simulated data. (b) BERT layers. (c) Artificial noise.

Figure S9: **MLEM and FR-RSA-I achieve similar encoding performance.** The plots show the test Spearman correlation for MLEM (blue) and FR-RSA-I (orange). **(a)** On simulated data, both methods achieve comparable performance, which degrades as noise increases. **(b)** Across BERT layers, both models show nearly identical predictive performance. Performance peaks in middle layers, consistent with prior work (Hewitt & Manning, 2019). **(c)** On BERT's 7th layer with artificial noise, performance degrades similarly for both methods. This confirms that MLEM's advantages do not come at the cost of encoding performance.

G A decoding baseline is not informative

To contrast the encoding-based approach with a standard decoding baseline, linear classifiers are trained to decode linguistic features from BERT's layer-wise representations. As shown in Figure S10, decoding accuracy for key features is high across all layers. While this confirms that feature information is present, the flat accuracy profiles provide little insight into how the representational geometry changes across layers. This highlights a key advantage of encoding models like MLEM, which are designed to reveal the structure of representations, not just the presence of information.

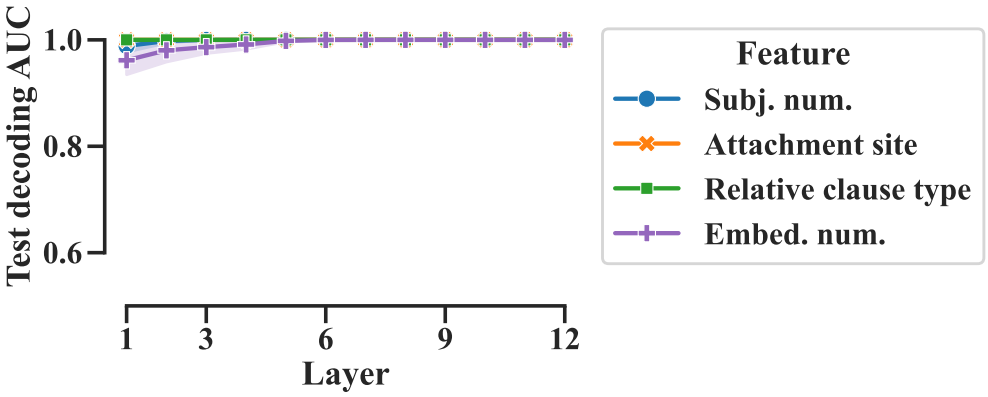


Figure S10: **Decoding accuracy provides limited insight into representational geometry.** This figure displays the decoding accuracy of selected linguistic features from BERT's layers using a linear classifier. Unlike the profiles of weights or feature importance, decoding accuracy remains consistently high across all layers, offering little differentiation of the underlying representational structure. The mean test AUC (full line) and standard deviation (shaded area) from 5-fold cross-validation are reported.

H A multivariate encoding baseline

To further contextualize MLEM's performance, we compare it to a multivariate encoding baseline: a random forest regressor. A random forest regressor was trained to predict the multi-dimensional BERT representations at each layer from the theoretical features. Random forests can naturally handle multi-dimensional targets and categorical inputs. We used the `RandomForestRegressor` implementation from `scikit-learn` and the built-in feature importance. As shown in Figure S11, this baseline surprisingly recovers a feature importance profile very similar to that of MLEM and FR-RSA-I, with syntactic features peaking in the middle layers. However, similarly to FR-RSA, standard random forests cannot model interactions between input features, hence the absence of an importance value for the interaction term. Furthermore, such an approach does not benefit from the noise and overfitting robustness conferred by the distance-based and Spearman-optimized framework of MLEM.

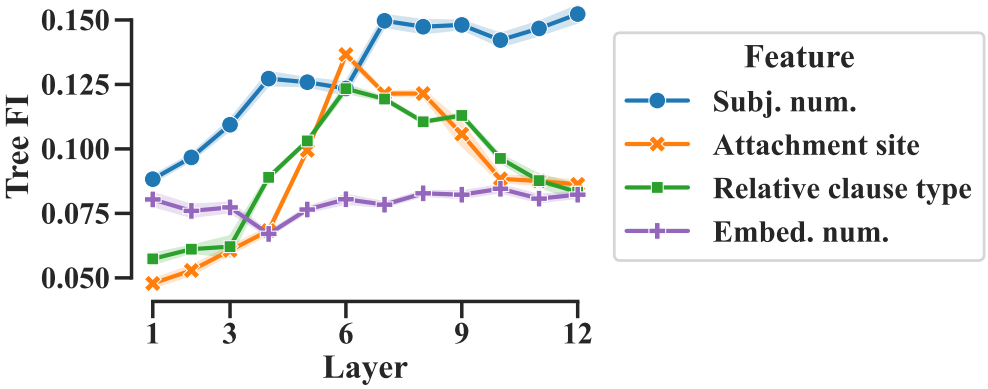


Figure S11: **A random forest encoding baseline recovers similar feature importance profiles but cannot model interactions.** This figure displays the feature importance from a random forest model trained to predict BERT's layer-wise representations from the theoretical features. The importance profile is similar to that found by MLEM, with syntactic features gaining importance in the middle layers. However, the model cannot capture feature interactions by design. The full line represents the average feature importance across the 100 trees of the forest, and the shaded area represents the standard deviation.

I Univariate vs. Multivariate MLEM

To assess whether MLEM captures distributed patterns of neural activity, the performance of the standard multivariate MLEM is compared against a univariate baseline. The multivariate MLEM is trained on all the units of a given layer simultaneously, allowing it to learn weights that reflect interactions across units. In contrast, a separate univariate MLEM is trained for each individual unit within a layer. This univariate model can only capture information encoded in the activation of a single unit. Figure S12 compares the encoding performance (test Spearman correlation) of the multivariate MLEM with the distribution of performances from all univariate models for each layer of BERT.

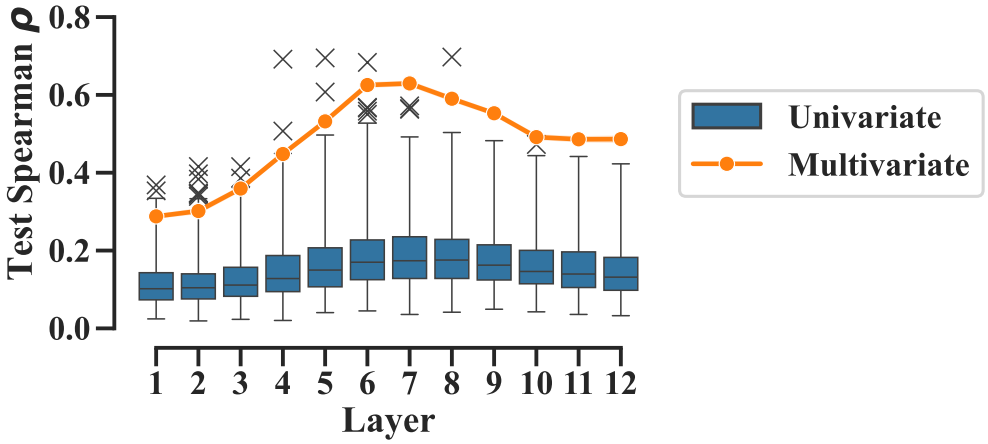


Figure S12: **Multivariate MLEM captures distributed representations more effectively than univariate models.** This figure compares the encoding performance (test Spearman correlation) of a single multivariate MLEM (orange) against the distribution of performances from univariate MLEMs (blue) for each layer of BERT. Each univariate model is trained on a single unit, while the multivariate model is trained on all the units in a layer. The superior performance of the multivariate MLEM indicates that it successfully captures geometric structure encoded across multiple units, which individual univariate models cannot. Units with a performance 3 times the interquartile range above the third quartile are considered outliers and are shown as crosses.

J Other clusters of the BERT representations

Figure 6 in the main text provides a walkthrough of the hierarchical clustering for a subset of the data. For completeness, Figure S13 shows the full partitioning of the representation space for layer 6, illustrating how the top three features organize all sentence representations into a nested structure.

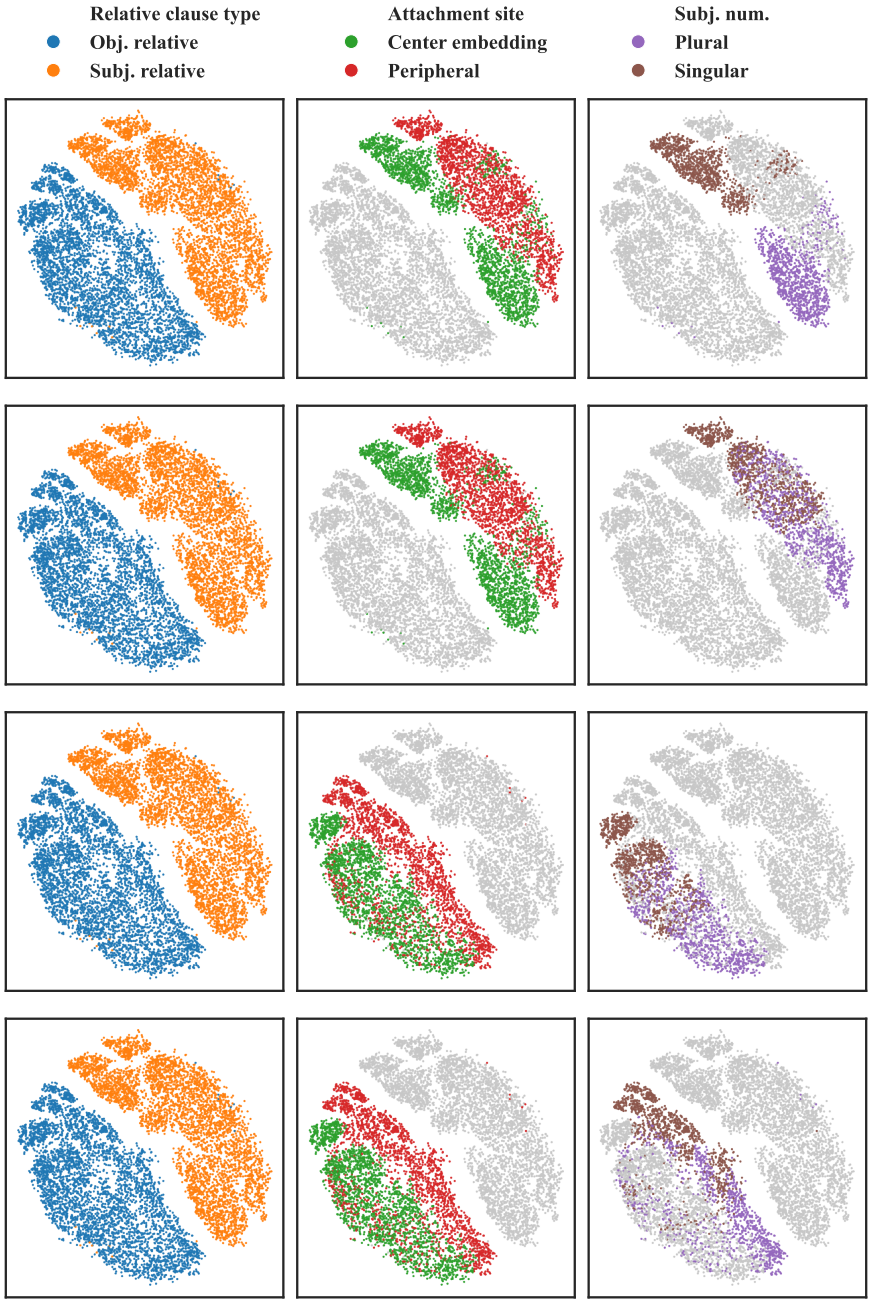


Figure S13: **Complete hierarchical clustering of BERT layer 6 representations.** This figure extends Figure 6 to show the full hierarchical clustering defined by the top three features from MLEM. Each row corresponds to a primary cluster based on "Relative Clause type" and "Attachment site". Within each of these, sub-clusters are formed based on "Subject number". This complete view illustrates how the combination of these three features partitions the entire representation space into a fine-grained, interpretable geometric structure.

K Convergence speed

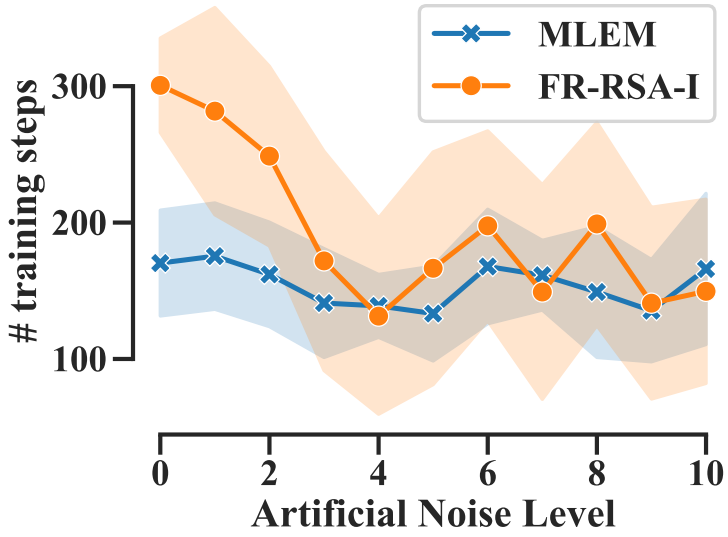


Figure S14: **MLEM's speed advantage diminishes at high artificial noise levels.** This plot shows the number of training steps for MLEM (blue) and FR-RSA-I (orange) on BERT's 7th layer with increasing artificial noise. While MLEM is faster at low noise levels, its speed advantage over FR-RSA-I decreases as noise increases, with both methods showing similar convergence times at high noise. However, MLEM's number of training steps exhibits less variability across runs.