NeuroConText: Contrastive Learning for Neuroscience Meta-Analysis with Rich Text Representation

Fateme Ghayem*, Raphaël Meudec, Jérôme Dockès, Bertrand Thirion, Demian Wassermann

Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France

* fatemeh.ghayyem@inria.fr

Abstract

Coordinate-based meta-analysis (CBMA) is the common way to gather information about human brain function across the existing literature in order to formulate hypotheses and contextualize new findings. However, automated CBMA tools face challenges such as inconsistent terminology and difficulties in analyzing long texts and capturing semantic meaning because they still rely on bag-of-words approaches; furthermore, sparse coordinate reporting in CBMA distorts the activation distribution due to incomplete data. This paper introduces NeuroConText, an automated CBMA tool that bridges neuroscience text, brain location coordinates and brain images by creating a shared latent space for encoding text and brain maps. This relies on a multi-objective loss combining contrastive and reconstruction terms. It leverages large language models (LLMs) to capture neuroscientific information from full-text articles, plus an LLM-based text augmentation strategy to handle short-text inputs. Quantitative and qualitative analyses demonstrate NeuroConText's ability to enhance text-to-brain retrieval performance and reconstruct brain maps from neuroscience texts. We also show that CBMA tools can infer brain activations in regions discussed in articles but absent in reported coordinates, potentially addressing the challenge of sparse coordinate reporting.

Author summary (should be 150-200 words)

Neuroscience studies often suffer from limited sample sizes, resulting in findings with low statistical power and uncertain reproducibility. Coordinate-based meta-analysis (CBMA) addresses this problem by pooling results across studies to identify consistent associations between cognitive processes and brain regions. This article introduces NeuroConText, an automated CBMA tool designed to integrate heterogeneous data sources, including brain images, full-text articles, and reported coordinate tables. NeuroConText uses large language models (LLMs) to capture rich semantic information from full-text of neuroscience articles to link neuroscience text to brain images and coordinates. The goal of NeuroConText is to retrieve brain locations or brain maps associated with text-based descriptions and to reconstruct brain activation patterns from text. NeuroConText is also designed to support short input queries, which makes it applicable to a wider range of use cases. In addition, we show that meta-analysis tools can highlight regions not explicitly reported in coordinate tables, thus overcoming the limitations posed by sparsely reported coordinates in neuroscience articles. In summary, by improving associations between neuroscience texts and brain locations, NeuroConText enhances the capabilities of coordinate-based meta-analysis.

1 Introduction

Hundreds of articles are published annually in the field of human neuroscience, highlighting the continuous accumulation of results in cognitive and clinical neuroscience. Manually collecting and analyzing information from such a volume of research is inefficient and thus requires automated methods [1]. Each study may examine different aspects of behavior or use specific methods or experimental protocols, adding layers of complexity to integrating these diverse results into a coherent answer to a specific research question [2]. Individual neuroscience studies are often challenged by small sample sizes and limited statistical power. This affects the reliability of the reported findings and reduces the likelihood that a statistically significant result reflects a true effect [3]. In addition, different framing within the literature leads to terminological inconsistencies; for example, different studies may use terms such as *cognitive decline* and *cognitive impairment* interchangeably [4].

Brain imaging meta-analysis is often used to produce a synthesis of the results reported in the literature to highlight the neural correlates of cognitive or neurological concepts [1]. It combines results from multiple studies, which improves the reliability of the associations between brain locations and cognitive or clinical concepts [5]. It serves three key purposes: first, it synthesizes information from the literature to summarize the state of knowledge in any subfield of human neuroscience. This allows researchers to see where the field stands collectively and highlight consensus or inconsistencies [6]. Second, meta-analysis provides context to interpret new results by comparing novel experimental data with existing patterns in the literature, which clarifies how new data align with or deviate from established findings [7]. Third, meta-analysis generates hypotheses on candidate brain regions or relevant cognitive domains [5].

In human neuroscience studies, experimental results are typically available in three modalities: the text of articles, brain activation coordinates reported in the tables of articles, and brain images available in public repositories at various levels of processing. Integrating these diverse data types is important, because they complement each other: texts give detail on cognitive hypotheses and conclusions, experimental procedures and provide cognitive concepts jointly with brain locations; brain coordinates tables provide a synthetic account of the main results of each publications, directly comparable across studies; brain images provide a more complete view of the correlates of cognition and diseases. Recent advances in large language models (LLMs) have demonstrated their ability to predict neuroscience research outcomes by synthesizing vast amounts of literature and making forward-looking predictions. BrainGPT is an LLM fine-tuned on neuroscientific publications to specialize in neuroscience research, and it has shown that LLMs can outperform human experts in predicting study results [8].

Bridging activation coordinates with brain images allows researchers to identify consistent activation patterns across studies [5, 9, 10]. Furthermore, connecting text from neuroscientific articles to brain activation coordinates can help to integrate textual data with imaging results for hypothesis generation and validating new results [1, 11].

Coordinate-based meta-analysis (CBMA) is a meta-analysis tool that evaluates the stereotactic brain coordinates reported in neuroscience articles to link neuroscience terms and descriptions to specific brain regions [5]. It leverages the fact that virtually all recent literature uses the Montreal Neuroimaging Institute (MNI) reference for group-level brain mapping, which gives an absolute meaning to the ensuing coordinates [12, 13]. Several tools for CBMA have been developed to link text and brain activation coordinates. Initial works on meta-analysis relied on manual curation and annotated activation coordinates with a systematic taxonomy [14], but this process struggles to keep pace with the rapidly growing literature. Automated meta-analyses [1] relies on articles' abstracts and multilevel kernel density analysis (MKDA) [15] to generate brain maps, but this approach captures few terms and suffers from limited

semantic accuracy. Recent improvements have come from the formulation of the problem as a predictive approach, by relying on full texts and the use of semantic relatedness [16]. However, approaches based on bag-of-word miss important information from neuroscientific discussions. Large language models (LLMs) trained on larger corpora and text augmentation techniques instead capture rich semantic details from neuroscience text [17]. What remains unclear is how to leverage long-text inputs.

CBMA tools can help researchers navigate the rapidly expanding neuroimaging literature efficiently and allow them to extract relevant information for hypothesis testing and knowledge discovery. Retrieval and reconstruction are two key processes that support these goals, both involving the translation of textual descriptions into spatial brain data. This facilitates the synthesis of new findings and generates new insights into the neural activations that underlie cognition and behavior. The **retrieval** task focuses on associating relevant brain locations or images with specific textual queries or hypotheses, a process central to hypothesis testing and literature review. For example, when researchers search for studies related to a particular cognitive function, *e.g. working memory*, they retrieve the brain coordinates associated with that function. This enables the integration of findings across diverse studies and supports the development of new hypotheses. On the other hand, the **reconstruction task** involves mapping textual descriptions of brain activity (e.g., from published studies) onto spatial brain coordinates or activation patterns. This process is essential for interpreting and aggregating findings across studies, allowing the creation of meta-analytic maps that summarize brain-behavior relationships. Reconstruction generates brain maps representing the spatial distribution of activations associated with a given textual description, such as a cognitive function or experimental condition.

Existing CBMA tools are optimized for reconstruction and fall short when it comes to retrieval. A core challenge in the retrieval task lies in accurately aligning textual descriptions with spatial brain representations. **Contrastive learning** offers a promising solution for associating (i.e. retrieving) an input text with its corresponding image. This family of methods effectively bridges text and image by establishing a shared latent space between the two modalities [18–20]. **CLIP** is a key example that exploits contrastive learning [20]. Contrastive modeling learns to identify correct (image, text) pairings from a set of potential matches by optimizing cosine similarity for true pairs and minimizing it for mismatched pairs. This suggests that CLIP has the potential to effectively improve retrieval between texts and their corresponding MNI coordinates.

In this paper, we introduce **NeuroConText**, a meta-analysis model designed to address the retrieval task in CBMA [21]. NeuroConText captures the articles semantics with an LLM and associates them with the reported brain locations. A preliminary short publication of the same work in [21] demonstrated the promising retrieval performance of NeuroConText. Here, we expand NeuroConText and introduce a *joint text-to-brain retrieval and reconstruction* framework with dual training objectives which leverage contrastive and mean squared error (MSE) losses to address retrieval and reconstruction tasks simultaneously. We also extend our experiments and provide extensive quantitative and qualitative analyses for retrieval and reconstruction tasks on left-out articles, and demonstrate NeuroConText's ability to bridge text and brain activation. We further enhance NeuroConText's performance and generalization to short-text inputs by introducing *LLM-based text augmentation*.

The remainder of the paper is organized as follows: We introduce data preparation and preprocessing of the articles' activation coordinates and text in Section 3.1. NeuroConText framework combining contrastive and MSE losses is introduced in Section 3.2. The inference task for reconstruction and the downstream task for retrieval are explained in Sections 3.3. We evaluate NeuroConText's ability to retrieve brain maps from text through quantitative and qualitative assessments in Section 4.2. We

demonstrate NeuroConText's capability to reconstruct brain maps from text using quantitative and qualitative methods in Section 4.3. To enhance NeuroConText's performance with short text inputs, we leverage LLMs for the augmentation of short-text inputs detailed in Section 3.4, with the results detailed in Section 4.4. We present ablation studies in Section 4.6, exploring different elements of NeuroConText, such as the impact of the LLM type, comparing contrastive method with pure regression-based methods, and the impact of chunk aggregation strategies for long-text representation. Finally, Section 5 discusses the outcomes of this paper.

2 Prior Work

Early efforts for brain meta-analysis, such as **BrainMap** [14], provide structured information about published neuroimaging studies by curating reported brain coordinates and annotating them according to a specialized taxonomy. However, as the neuroimaging literature grows rapidly, manual curation of databases is not scalable enough to cover all publications. To address scalability, **NeuroSynth** [1] automates the meta-analysis of neuroscientific studies by identifying relevant articles through the terms in their *abstracts* and extracting brain activation coordinates to create brain statistical maps. In NeuroSynth, texts are represented by the frequency with which they appear in the abstract of an article. It uses multilevel kernel density analysis (MKDA) [15] to convert reported coordinates into statistical maps. This automated process allows efficient and automated analysis of the entire corpus. However, NeuroSynth has some key limitations: it relies on statistical testing, which means that it is not designed to make accurate predictions on novel queries. It also uses a bag-of-words approach, based only on the abstracts and not the full text of publications, so it may miss out on complex topics, new research terms and syntactic information (negations, conditionals etc.). Finally, it struggles with rare terms, which limits how well it can cover less common research topics. Building on this, **NeuroQuery** [16] addresses several of these limitations by using a multivariate model trained on the *full texts* of publications rather than just abstracts to predict relevant brain locations for neuroscience queries. It uses kernel density estimation (KDE) [22, 23] to represent lists of coordinates as brain maps. NeuroQuery employs the term frequency-inverse document frequency (TF-IDF) method [24] to analyze texts, measuring the importance of words within a document relative to their frequency across the entire corpus. It applies semantic smoothing techniques that leverage term co-occurrences to improve the identification and mapping of infrequent terms. Although NeuroQuery improves upon NeuroSynth by using full texts and a broader vocabulary, it remains constrained by its reliance on a bag-of-words model that does not fully take into account the semantics of the text, treating word occurrences independently without considering their context or relationships. More recently, **Text2Brain** [17] has introduced a large language model (LLM)-based approach to link text with reported brain locations. It uses a transformer-based text encoder built on the SciBERT architecture [25] to represent the semantics of the text. During training, Text2Brain uses an augmentation strategy where it selects various types of text inputs, including the first sentence of table captions, articles' titles, keywords, abstracts, and randomly selected subsets of sentences from the discussion sections. The encoded text is then fed into a 3D convolutional neural network (CNN) to estimate spatial brain activation maps.

While prior tools have advanced automated meta-analysis by linking text to brain, they remain limited by weak text representations, a lack of retrieval capabilities, or restricted input formats. **NeuroConText** addresses these challenges by using contrastive learning on full-text data to jointly enable retrieval and reconstruction of brain activation patterns. In the following, we detail its architecture and learning strategy.

3 Method

In this section, we present **NeuroConText**, a joint text-brain retrieval and reconstruction model that uses contrastive learning grounded in the full semantic context of neuroscience text. NeuroConText links neuroscience text and distribution maps of brain activation peaks by learning latent representations of text and brain locations that support the retrieval of brain activation patterns from text and the estimation of relevant brain maps from a neuroscience text.
NeuroConText consists of four main modules: 1) data preparation and preprocessing, 2) training, 3) inference and downstream tasks, and 4) text augmentation at the inference phase. In the following, we detail each module.

3.1 Data Preparation and Preprocessing

In this section, we describe the procedures that we use for the preparation of text and coordinates data in the neuroscientific articles. We focus on transforming full-body texts and reported brain coordinates into representations suitable for NeuroConText's computational analysis.

- **Text data representation using LLM embeddings:** To capture the semantics of neuroscientific articles, we use LLMs that leverage attention mechanisms. We aim to process the **entire** text of each article. However, due to the token size limitations of LLMs, we segment the full-body text into fixed-size chunks. Specifically, we use Mistral-7B [26] with a context window of 8,192 tokens, which ensures that each chunk does not exceed this limit. For texts shorter than the chunk size, we apply padding using a special token to maintain uniform input dimensions. After obtaining the embeddings for each chunk, we aggregate them through mean pooling to create a unified text representation for each article. This approach ensures that the semantic information from the entire article is preserved while respecting the token constraints of the LLM.

- **Representing coordinates as brain images using KDE:** We bridge between sets of coordinates and brain images using a KDE approach [22, 23, 27] which proceeds as follows: Let $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_v] \in \mathbb{R}^{3\times v}$ represent a fixed sampling grid of voxel positions in a 3D space with the total number of v voxels, where each $\mathbf{g}_j \in \mathbb{R}^3$ denotes the position of the j-th voxel. Let $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ be the set of peak coordinates extracted from the brain image, where each $\mathbf{p}_i \in \mathbb{R}^3$ is the position of the i-th peak, and l is the number of peaks. The RBF kernel κ_σ centered at each peak is defined by:

$$\kappa_\sigma(\mathbf{g}_j, \mathbf{p}_i) = \exp\left(-\frac{\|\mathbf{g}_j - \mathbf{p}_i\|_2^2}{2\sigma^2}\right) \tag{1}$$

where σ is the kernel parameter. Then, the KDE representation of the estimated image $\mathbf{m} = [m_1, \ldots, m_v]^T \in \mathbb{R}^v$ over the grid $\mathbf{G}$ is calculated by summing the contributions of the Gaussian kernels centered at each peak across all grid points:

$$m_j = \sum_{i=1}^{l} \kappa_\sigma(\mathbf{g}_j, \mathbf{p}_i) \quad \forall j \in [\![1, v]\!] \tag{2}$$

This equation defines how each voxel $\mathbf{g}_j$ in the grid is influenced by the Gaussian kernels placed at each peak $\mathbf{p}_i$, which results in a continuous map $\mathbf{m}$ proportional to the spatial density of activity peaks.

- **Brain image dimension reduction with DiFuMo atlas:** To address the high dimensionality of the brain images reconstructed by KDE, we employ the Dictionary of

Functional Modes (DiFuMo), a probabilistic atlas that effectively reduces data dimensionality [28]. This method provides a linear decomposition of the brain images, where the data matrix $\mathbf{M} \in \mathbb{R}^{v \times n}$, containing n brain images, is represented as $\mathbf{M} = \mathbf{D}\mathbf{X}$. Here, $\mathbf{D} \in \mathbb{R}^{v \times k}$ is the dictionary matrix containing basis signals called atoms as its columns, under non-negativity and sparsity constraints. The coefficients matrix $\mathbf{X} \in \mathbb{R}^{k \times n}$ contains the linear combination coefficients for this decomposition. The number of atoms $k = 512$ is significantly smaller than the number of voxels v, which ensures a substantial dimensionality reduction.

In this study, we use the set of k DiFuMo coefficients as image features. The process of deriving DiFuMo coefficients from KDE images is formulated as a linear regression problem.

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^k} \|\mathbf{m} - \mathbf{D}\mathbf{x}\|_2^2 = \mathbf{D}^\dagger \mathbf{m}, \tag{3}$$

where $\mathbf{D}^\dagger = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T$ is the pseudo-inverse of $\mathbf{D}$.

3.2 Training module

To jointly model the relationship between neuroscience text and brain activation patterns, NeuroConText model is trained using a contrastive learning objective that aligns representations of text and brain maps in a shared latent space. The model learns to associate semantically related pairs while distinguishing them from unrelated ones. This training process enables both retrieval (matching text to brain data) and reconstruction (generating brain maps from text) within a unified framework. Fig.1 shows NeuroConText's training process, which consists of the following steps:

- **Encoding and Decoding:** To perform the retrieval task, we define a shared latent space by mapping text embeddings through a text encoder and DiFuMo representations of brain maps through an image encoder. We use **Mistral-7B** [26] as the LLM to generate text feature representations with an embedding dimension $d = 4096$. The latent space dimension is set to k, and a bottleneck in the text encoder reduces the dimension from d to k. Next, we implement a decoder to reconstruct DiFuMo coefficients of brain maps from the text's latent representation. The architecture of the text encoder, image encoder, and decoder are depicted at the bottom of Fig.1. The encoders and decoder consist of blocks, each containing a fully connected (FC) layer with k units, followed by GELU activation, dropout (50%), and layer normalization. Specifically, the text encoder comprises three blocks, with the first block including a residual connection. The image encoder contains three blocks, and the decoder has two blocks.

- **Multi-objective loss and alternating optimization:** We train **NeuroConText** to perform joint text-brain retrieval and brain map reconstruction. Consider $\Theta = [\theta_{\text{text}}, \theta_{\text{img}}, \theta_{\text{dec}}]$ to be NeuroConText's trainable model parameters corresponding to the weights of the text encoder denoted by the function $f_{\text{text}} : \mathbb{R}^d \to \mathbb{R}^k$, image encoder $f_{\text{img}} : \mathbb{R}^k \to \mathbb{R}^k$, and decoder $f_{\text{dec}} : \mathbb{R}^k \to \mathbb{R}^k$ with the architecture shown at the bottom of Fig.1 detailed in the previous part. $k = 512$ is the DiFuMo dimension, and $d = 4096$ is the Mistral-7B embedding size. We show the input text embeddings of N samples by the matrix $\mathbf{Y} = [\mathbf{y}^{[1]}, \ldots, \mathbf{y}^{[N]}]$ where columns $\mathbf{y}^{[i]} \in \mathbb{R}^d$ represent the ith sample. We denote the DiFuMo coefficients of the input data by the matrix $\mathbf{X} = [\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[N]}]$ where each column $\mathbf{x}^{[i]} \in \mathbb{R}^k$ corresponds to the sample i. NeuroConText's model optimizes a multi-objective loss function that minimizes the sum of the InfoNCE contrastive loss for retrieval and the mean squared error (MSE) loss for reconstruction:

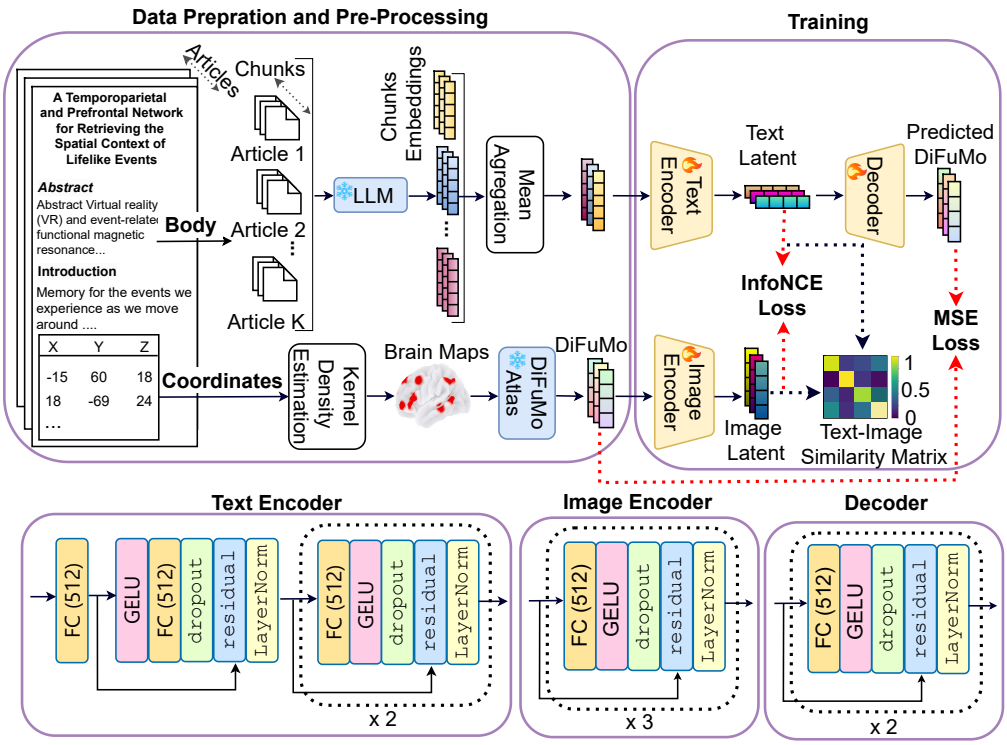


Fig 1. NeuroConText training : The training pipeline consists of three main steps: (1) Data preparation and preprocessing: Neuroscience articles are split into chunks, and a pretrained large language model (LLM) is used to encode each chunk and generate LLM-based embeddings. These embeddings are aggregated using a mean aggregation step. In parallel, brain coordinates from the articles are transformed into brain maps using kernel density estimation (KDE), which are then encoded in the DiFuMo atlas. (2) Encoding into a shared latent space and decoding: Text embeddings are passed through a text encoder, while DiFuMo coefficients are passed through an image encoder, both of which are trained to project inputs into a shared latent space. The architecture of the text encoder, image encoder, and decoder is shown below. (3) Learning objectives: A contrastive loss using the InfoNCE objective ensures that text and brain map embeddings from the same article are brought close in the latent space. A mean squared error (MSE) loss guides the decoder to reconstruct brain maps in the DiFuMo space from the text embeddings. In this figure, the blue modules with the snowflake icons and the yellow modules with the fire icons denote the frozen (pretrained) and trainable models, respectively.

$$\hat{\Theta} = \arg\min_{\Theta} \left[\mathcal{L}_{\text{InfoNCE}}(\mathbf{Y}, \mathbf{X}; \theta_{\text{text}}, \theta_{\text{img}}) + \mathcal{L}_{\text{MSE}}(\mathbf{Y}; \theta_{\text{text}}, \theta_{\text{dec}}) \right]. \tag{4}$$

Here, $\mathcal{L}_{\text{InfoNCE}}$ is the InfoNCE loss, defined as:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{Y}, \mathbf{X}; \theta_{\text{text}}, \theta_{\text{img}}) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(\text{sim}(\mathbf{z}_{\text{text}}^{[i]}, \mathbf{z}_{\text{img}}^{[i]})/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}(\mathbf{z}_{\text{text}}^{[i]}, \mathbf{z}_{\text{img}}^{[j]})/\tau\right)}, \tag{5}$$

where $\mathbf{z}_{\text{text}}^{[i]} = f_{\text{text}}(\mathbf{y}^{[i]}; \theta_{\text{text}})$ and $\mathbf{z}_{\text{img}}^{[i]} = f_{\text{img}}(\mathbf{x}^{[i]}; \theta_{\text{img}})$ are the latent representations of the text and image, respectively. The function $\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \|\mathbf{b}\|}$ denotes the cosine

similarity, with $\langle \cdot, \cdot \rangle$ as the inner product and $\| \cdot \|$ as the Euclidean norm, and τ is a temperature parameter. This loss encourages alignment of the text and image representations in the shared latent space.

The second objective function, $\mathcal{L}_{\text{MSE}}$, is the MSE loss, defined as:

$$\mathcal{L}_{\text{MSE}}(\mathbf{Y}; \theta_{\text{text}}, \theta_{\text{dec}}) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^{[i]} - f_{\text{dec}}(\mathbf{z}_{\text{text}}^{[i]})\|^2, \tag{6}$$

Learnable parameters $\Theta = [\theta_{\text{text}}, \theta_{\text{img}}, \theta_{\text{dec}}]$ for the text encoder, image encoder, and the decoder respectively, are structured as matrices of dimensions $k \times k$ for each FC layer. This configuration involves approximately 4.2 million parameters.

For training NeuroConText's model parameters Θ, we first optimize the parameters θ_{text} and θ_{img} of the encoders to minimize the InfoNCE loss to ensure alignment between text and image representations in the shared latent space. We optimize θ_{text} and θ_{dec} to minimize the MSE loss by focusing on the reconstruction of DiFuMo coefficients from text embeddings. We alternate between these two optimization steps, wherein during each iteration, θ_{text} is adjusted based on the contrastive loss initially and later fine-tuned using the MSE loss. See Appendix A for the detailed algorithm and the theoretical proof of convergence.

3.3 Inference Module and Downstream Task

Once NeuroConText's model has been trained, we leverage the model to perform inference and consider a downstream task to evaluate the efficiency of our model. This is illustrated in Fig.2.

- **Text-brain retrieval (downstream task):** We perform a retrieval task to identify the relevant brain map for a given textual query. As depicted in Fig. 2-bottom, we compare the input text embedding with brain map embeddings from our dataset in the shared latent space. We calculate a text-image similarity vector and rank the brain map embeddings to determine if the ground-truth map is among the top K retrieved items.

- **Brain map reconstruction (inference):** The inference task focuses on reconstructing a brain map from a given text. As shown in Fig.2-top, we input text into the LLM to generate chunk-level embeddings. These embeddings are aggregated to form the input text representation. This representation is passed through the trained text encoder and decoder to produce a brain map in the DiFuMo space. We then apply the DiFuMo inverse transform to recover the brain map in the original image domain.

3.4 LLM-based text augmentation for short input text at inference

While NeuroConText is trained on full-text articles, we adapt it for use with small texts, such as functional task descriptions or contrast specifications, similar to those available in the Cognitive Atlas. This can result in suboptimal performance due to a distribution shift in the embeddings generated by the language model for short and long texts, that is, the statistical properties of the embeddings differ depending on text length (see Appendix B).
To enhance NeuroConText's ability to reconstruct brain maps from short-text descriptions, we employ LLM-based text augmentation using GPT-4o. The augmentation process expands short descriptions into long article-like formats by including sections such as introduction, methodology, experiments, and discussion. To

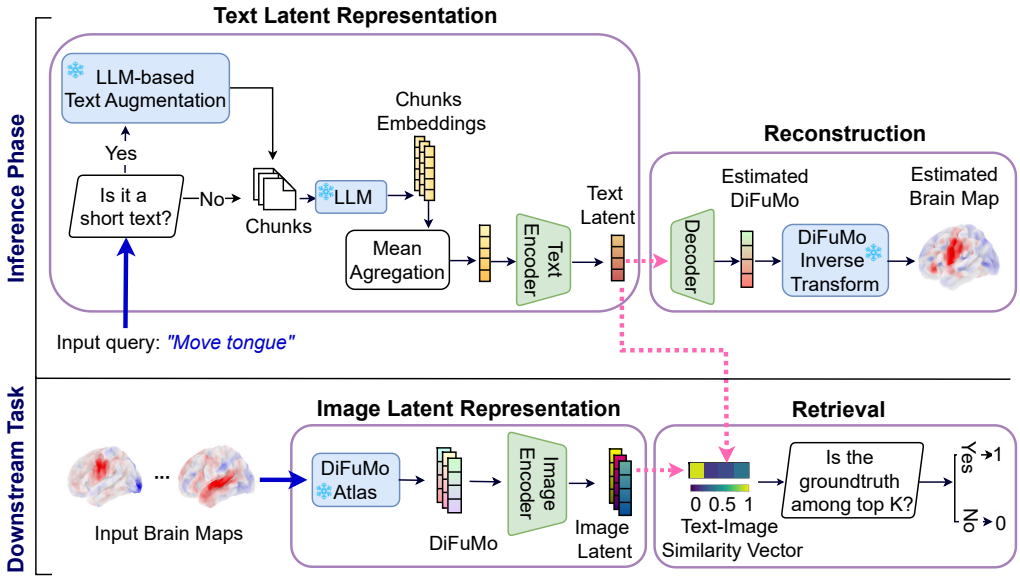


Fig 2. Use of NeuroConText- Inference phase (Top): The inference pipeline includes two tasks: brain reconstruction from an input text and article retrieval. Short input queries (e.g., "Move tongue") are processed through LLM-based text augmentation, then the augmented text is split into chunks. Next, the chunks are encoded and aggregated to form the input text embedding. This aggregated embedding is passed through NeuroConText's trained text encoder and decoder (green modules) to reconstruct the DiFuMo coefficients corresponding to the KDE representation of the article's reported coordinates. Then, we map estimated DiFuMo coefficients into the voxel space using the DiFuMo inverse transform. **Downstream Task (Bottom):** In the downstream task, we retrieve NeuroVault's brain contrast maps corresponding to the input queries, which are NeuroVault's short-text contrast definitions. To do so, the aggregated input text embedding is compared to the brain map embeddings in the database in the latent space through the text-image cosine similarity vector, where we expect to retrieve the corresponding map among the top-k retrieved items. In this figure, the blue modules with the snowflake icons and the green modules denote the frozen (pretrained) and trained NeuroConText models, respectively.

ensure the validity of the augmented texts and minimize hallucination risks, we ask domain experts to review the results, though occasional discrepancies may still exist. In the Results Section, we evaluate the impact of this distribution shift by comparing NeuroConText's performance on NeuroVault [29] short-text inputs before and after text augmentation. See Appendix C for the prompt used to augment NeuroVault contrast descriptions.

3.5 Evaluation metrics

We consider different evaluation metrics for text-brain retrieval and reconstruction tasks.

- **Retrieval evaluation metrics:** To evaluate how well the model can retrieve the correct brain map given a textual description, we use similarity scores (as illustrated in Fig. 2) between the input text and all candidate brain maps in the test set. The evaluation assesses whether the model ranks the correct brain map higher than the incorrect ones. In this regard, we use the following two metrics:

Recall@K — This metric measures how often the correct brain map is found among the top-K most similar maps retrieved by the model for each input text. A higher Recall@K indicates that the model better aligns text inputs with their corresponding brain maps.

Mix&Match — This metric, which is adapted from Mitchell et al. [30], tests whether the model can correctly identify the brain map that matches the input text. For each text, it compares the similarity of the retrieved map to the ground truth map versus a randomly selected incorrect map. The score reflects how often the retrieved map is more similar to the ground truth map than the random map, with higher values indicating better retrieval accuracy.

- **Reconstruction evaluation metric:** We use **Dice Score**, as in the Text2Brain paper, to asses NeuroConText's performance to predict the density of activation peaks from input text descriptions. The Dice Score measures spatial overlap between the predicted maps and actual activation maps. For a given threshold t, the Dice score D_t quantifies how well activations co-occur between the ground truth activation map $\mathbf{y}$ and the predicted map $\hat{\mathbf{y}}$:

$$D_t = \frac{2\,|\mathbb{1}_{\mathbf{y}>t} \cap \hat{\mathbf{y}}>t|}{|\mathbb{1}_{\mathbf{y}>t}| + |\mathbb{1}_{\hat{\mathbf{y}}>t}|},$$

where $\mathbb{1}_{\mathbf{y}>t}$ is an indicator function that identifies voxels with activation values above the threshold t, and $|\cdot|$ represents the volume of activated regions in the image space.

Combining the above metrics, our evaluations assess NeuroConText's ability to retrieve and generate brain maps from textual descriptions. This dual evaluation strategy reflects the model's ability to bridge neuroscience text and brain maps.

3.6 Implementation Details and Code Availability

For the KDE representation of the coordinates, we set FWHM $= 9mm$. NeuroConText model was trained using a batch size of 125, with the Adam optimizer [31]. The learning rate for the encoder and decoder was set to 1×10^{-4}, with a weight decay of 0.1. Dropout was applied at a rate of 0.6 for both the encoder and decoder to mitigate overfitting. The temperature parameter τ for the InfoNCE loss is set to 10. The training was performed over 50 epochs.
We used Python 3.10.9, PyTorch 2.0.1+cu117, and the Hugging Face Transformers library (version 4.39.3) for model implementation and inference. For text representation, we used the Mistral-7B-v0.1 model from the Hugging Face Model Hub, along with its associated tokenizer. To handle short input texts during inference, we employed LLM-based text augmentation using GPT-4o. For extracting Mistral embeddings of the text, we used a computing system consisting of a CPU node with 2 × Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz, 256 GB RAM, and a GPU node with 3 × Nvidia V100 32G. For model training, inference, and other experiments, we conducted our work on a system equipped with a Tesla V100-DGXS GPU (Driver Version: 450.51.05, CUDA Version: 11.0) and an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz with 40 cores and 251 GB of RAM. The code for reproducing our results is publicly available at `https://github.com/ghayem/NeuroConText`. This repository includes preprocessed text and coordinates data and scripts for NeuorConText model training and inference.

4 Experiments and Results

4.1 Data Preparation and Experiment Setup

We downloaded and prepared neuroscience articles from PubMed Central (PMC) using Pubget. Pubget is an open access tool that extracts text, metadata, stereotactic coordinates, and Term Frequency-Inverse Document Frequency (TF-IDF) data from the articles [24]. We extended the NeuroQuery dataset [16] by increasing its size from 14K to 20K articles including more recent publications. After downloading the articles and obtaining their text and coordinates, we obtained the DiFuMo representation of the coordinates, and the LLM embeddings of the full-body text. In our experiments, we split the neuroscience articles into 19K training and 1K test samples. We then trained the NeuroConText model with the training set and evaluated its performance in the retrieval and reconstruction tasks on the test set. We used a 19-fold cross-validation for model selection and a separate 15-fold cross-validation to evaluate the model's performance on unseen test data. We compared NeuroConText to two baselines: NeuroQuery and Text2Brain. For a fair comparison with the baselines and to avoid leakage, we ensure that articles used in NeuroQuery's training set are excluded from the test set.

4.2 Brain Map Retrieval from Textual Input

In the first experiment, we evaluated the ability of NeuroConText to associate the text of neuroscientific articles with their corresponding coordinates reported in the articles. To assess NeuroConText's performance on the retrieval task, we followed the procedure outlined in Section 3.3 and shown in Fig.2. We aligned the input articles' text (title, abstract, full-body) with their corresponding coordinates set based on texts-coordinates cosine similarities in NeuroConText's latent space. We then computed the retrieval metrics—Recall@K for $K \in \{10, 100\}$ and Mix&Match—on the left-out articles according to their definitions provided in Section 3.5. To compare with other meta-analysis tools, we also calculated these metrics for NeuroQuery and Text2Brain using their pre-trained models to estimate brain maps from the article's text in the test set and calculate their similarities with the KDE representation of the coordinates.

NeuroConText leads in retrieving brain activation from long text inputs

Our experiments on brain map retrieval from text show that NeuroConText outperforms NeuroQuery and Text2Brain. When evaluating recall@10 for full-body text, NeuroConText achieves a score of 22.6%, far from NeuroQuery (7%) and Text2Brain (1.4%), as shown in Fig.3. This trend holds when the input text is the article's abstract or title. Furthermore, in NeuroConText, the results for full-body text are higher compared to the abstract, with Recall@10 improving from 12.8% (abstract) to 22.6% (body text). This shows the impact and importance of leveraging the richer information in the full-body text compared to the relatively concise abstract.

4.3 Reconstructing Brain Maps from Article's Text

We assessed NeuroConText's ability to generate brain activation maps from text latent representations to obtain insights into its internal latent representations. In this experiment, we used the full text of the articles as input. We compared NeuroConText's reconstructed maps with the KDE representation of coordinates using the Dice Score metric explained in Section 3.5. For comparison with the baselines, we applied NeuroQuery and Text2Brain to the same test articles and computed Dice Scores between their reconstructed maps and the articles' KDE representations.

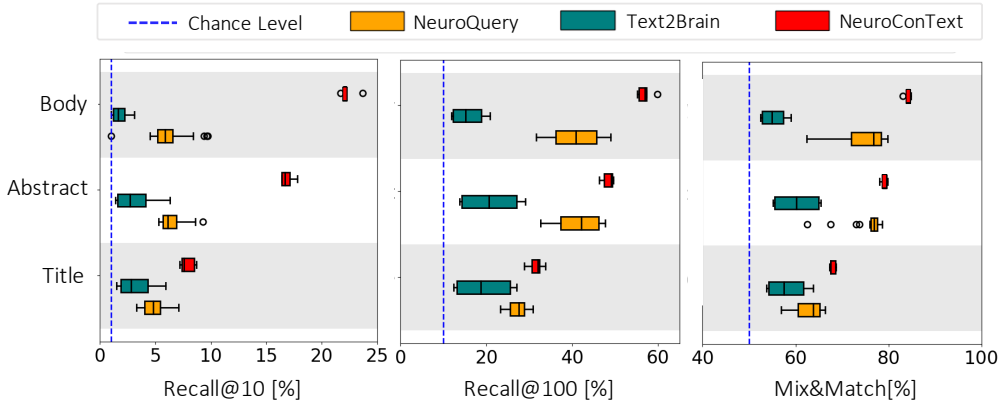


Fig 3. Retrieval of brain coordinates from articles' text input: NeuroConText achieves higher retrieval scores compared to NeuroQuery and Text2Brain, particularly for full-body text input. This figure presents results on left-out articles: Given a text query—either the title, abstract, or body of a neuroscientific publication—we report Recall@10, Recall@100, and Mix&Match, computed based on cosine similarities between text and coordinate representations. The dashed blue line is the chance level used as a baseline. These metrics assess the ability to retrieve the coordinates reported in the articles that best represent the input text.

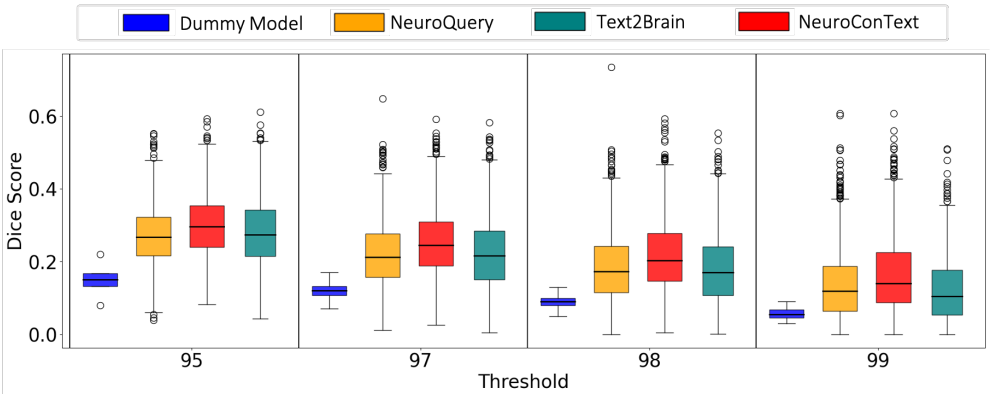
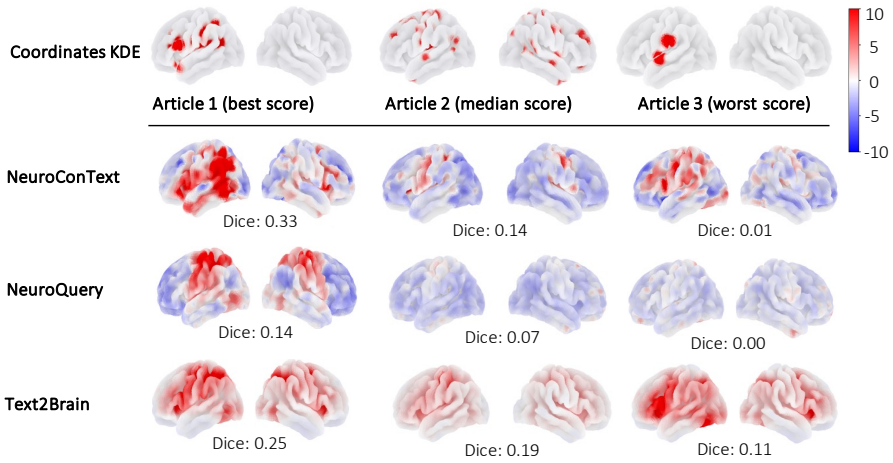


Fig 4. Reconstructing brain maps from articles' text input: NeuroConText outperforms on par with NeuroQuery and Text2Brain in reconstructing brain activation maps from textual input. NeuroConText achieves comparable Dice scores across different thresholds. The thresholds correspond to percentiles of the brain maps histograms. This figure presents the Dice score evaluation of brain maps reconstructed from full-body text input using NeuroConText, NeuroQuery, and Text2Brain. The Dice Score measures the spatial overlap between the reconstructed maps and the KDE representation of the articles' coordinates. The Dummy Model in blue serves as a baseline to highlight the added value of CBMA tools. Threshold values correspond to quantiles of the images.

NeuroConText is on par with the state of the art in the reconstruction task for left-out articles

NeuroConText consistently achieves Dice scores on par with NeuroQuery and Text2Brain across all thresholds for brain map reconstruction from text, as shown quantitatively in Fig. 4. We also provide a qualitative comparison of the brain maps reconstructed by NeuroConText and the baseline models in Fig.5 for the best, median,



(a) Target articles where NeuroConText achieved the highest, median, and lowest Dice scores

Summary of article 1 (best Dice Score)	Summary of article 2 (median Dice Score)	Summary of article 3 (worst Dice Score)
Patients with apraxia exhibit specific distortions in visuo-motor temporal integration, which are associated with damage to the left inferior frontal gyrus (IFG) and inferior parietal lobule (IPL).	DTI (diffusion tensor imaging) revealed widespread white matter abnormalities in ALS patients, particularly in the corticospinal tract (CST), corpus callosum, and frontal regions, which correlated with clinical measures. The involvement of the cerebellum suggests a compensatory mechanism for motor function loss, highlighting DTI's potential as a stable biomarker for ALS.	The cerebellum VI's intrinsic functional connectivity with cerebral regions is associated with rapid automatized naming, a key predictor of reading performance. This study provides new evidence for the cerebellar deficit hypothesis of dyslexia and the functional segregation of the cerebellum.

(b) Summary of the articles' full-body text.

Brain regions of article 1 (best Dice Score)	Brain regions of article 2 (median Dice Score)	Brain regions of article 3 (worst Dice Score)
visuo-motor integration and apraxia include, Left Inferior Frontal Gyrus (IFG), Left Inferior Parietal Lobule (IPL), Left Precentral Gyrus, Left Rolandic Operculum, Left Insula, Left Supramarginal Gyrus.	precentral gyrus, supplementary motor cortex, corticospinal tract (CST), corpus callosum, parahippocampal gyrus, frontal areas, insula, limbic system, thalamus, cerebellum, temporal and parietal lobes, internal and external capsules.	Cerebellum VI, Left Supramarginal Gyrus, Left Insula, Left Inferior Frontal Gyrus, Left Lateral Temporal Cortex, Left Fusiform Gyrus Cerebellum V

(c) Brain regions mentioned in the articles' text.

Fig 5. Qualitative comparison of reconstructed brain maps from articles' text input: In the best-performing cases, NeuroConText performs on par with Text2Brain and NeuroQuery. In the median and worst cases, although the overlap with the KDE representation is smaller, we still observe that NeuroConText and Text2Brain reconstructions match with the brain regions listed for the articles. This suggests that these tools can infer activations beyond the sparse coordinates reported in the articles and highlights the limitations of using Dice scores based on KDE maps as the sole evaluation metric. (a) Comparison of brain activation maps reconstructed by different models for the articles with the best, median and worst achieved Dice scores. The colorbar represents the normalized intensity values of brain activation. (b) Summary of the articles. (c) Brain regions discussed in the articles.

and worst reconstruction performances of NeuroConText. Using the Mistral language model, we provide a summary of the articles and the brain regions discussed in the articles to compare the reconstructed maps with the content of the articles' text. Comparing the NeuroConText reconstruction with the KDE map in Article 1, we observe activation in the superior temporal gyrus in both maps, which also appears among the regions extracted from the article text. In Addition, activations are present in the angular gyrus, supramarginal gyrus, and insula—regions listed in the article, athough they are not observed in the KDE representation. In Article 2, we observe activations in the precentral gyrus, corpus callosum, and external capsules for NeuroConText, while Text2Brain shows additional activations, such as in the supplementary motor cortex. In the worst case for NeuroConText, Article 3, we observe overlaps with the KDE in the left insula and left supramarginal gyrus. In addition, for both NeuroConText and Text2Brain, activations appear in the left precentral gyrus and cerebellum VI, which are not explicitly present in the KDE map but are listed among the brain regions of the article.

4.4 LLM-Based Text Augmentation Aligns NeuroConText to Short Text Descriptions for Brain Map Reconstruction

To assess NeuroConText's generalization to short-text inputs, we evaluate it on 196 contrast descriptions from the collection 2138 of the NeuroVault dataset [32], along with some associated contrast. The ID of the chosen samples is provided in NeuroConText GitHub repository. NeuroVault is a large open-access repository of fMRI statistical maps from group or single-subject studies. Each NeuroVault sample also includes associated cognitive terms or labels. A key challenge in this context is the **distribution shift** between LLM embeddings of long and short texts. By applying Principal Component Analysis (PCA) to text embeddings, we observe a prominent separation between short NeuroVault contrast definitions and longer publication texts (abstracts and full-body) in the embedding space, particularly along two principal components (see Appendix B for details). This suggests that the disparity in text length impacts how text information is represented and influences NeuroConText's performance.
In this subsection, to quantitatively assess the impact of text augmentation, we apply NeuroConText to reconstruct brain maps from NeuroVault descriptions in two forms: (1) the original short descriptions and (2) augmented descriptions transformed into longer, article-like formats using GPT-4o. We evaluate the reconstruction accuracy using Dice scores over thresholds corresponding to the 95%, 97%, 98%, and 99% quantiles of the image to compare the performance of NeuroConText, NeuroQuery, and Text2Brain and to show the ability of each model to recover activation patterns from textual input. Next, we perform a qualitative analysis of brain map reconstructions to further evaluate the impact of text augmentation. We also study the impact of LLM-based text augmentation on retrieval performance. In this experiment, we test how augmentation affects the ability of NeuroConText to associate NeuroVault contrast descriptions with corresponding brain activation maps. This retrieval task was performed using the original short descriptions and the augmented long descriptions. In the following, we describe the results.

LLM-based text augmentation enhances NeuroConText reconstruction performance on short-text input:

Table 1 shows the impact of text augmentation on reconstruction performance by comparing the results before and after the augmentation explained in Section 3.4 for the NeuroVault short-text descriptions. We evaluated the performance across various brain map thresholds. We can see that augmented text significantly improves NeuroConText's

Table 1. Impact of LLM-based text augmentation on reconstructing brain map from short-text input: LLM-based text augmentation improves the reconstruction performance of NeuroConText and NeuroQuery when applied to NeuroVault short labels. This confirms that these methods benefit from longer, semantically richer input text. In contrast, Text2Brain, which is optimized for short text input, performs better with the original (non-augmented) labels. This table shows the Dice scores for NeuroConText, NeuroQuery, and Text2Brain over different brain map thresholds for reconstructing NeuroVault maps from their labels. For each method, we compare the results obtained with the original short-text descriptions (org.) and the augmented descriptions (aug.).

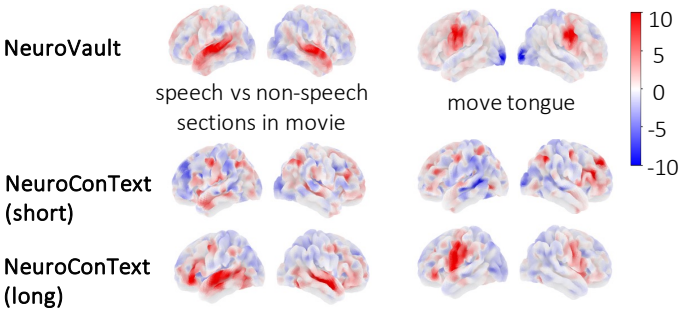
Method	**Threshold** [%] / **Text Type**	**Dice Across Thresholds** [%] ↑			
		95	**97**	**98**	**99**
Text2Brain	Augmented description	22	17	13	8
	Original description	25	20	17	13
	Difference (aug. - org.)	-3	-3	-4	-5
NeuroQuery	Augmented description	24	20	18	14
	Original description	23	19	17	13
	Difference (aug. - org.)	1	1	1	1
NeuroConText	Augmented description	21	15	12	7
	Original description	15	10	7	4
	Difference (aug. - org.)	**6**	**5**	**5**	**3**

performance. We can also see a slight improvement in NeuroQuery performance as well. This is due to the fact that NeuroConText and NeuroQuery models are optimized for processing long texts. In contrast, Text2Brain performs better with short descriptions, as it is not designed to handle long text effectively. The extended NeuroVault descriptions are open access in NeuroConText GitHub repository.

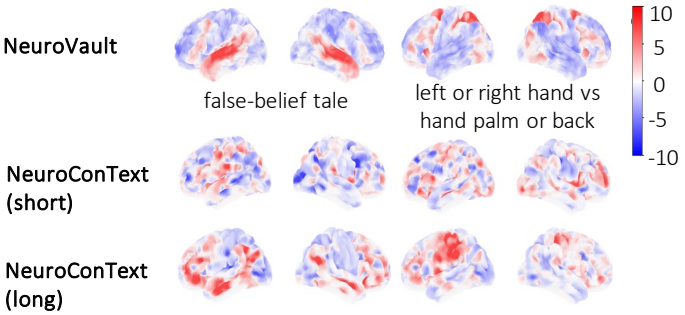
In addition to the quantitative results, we performed a qualitative analysis of the brain maps reconstructed by NeuroConText for NeuroVault descriptions before and after augmentation, as shown in Fig.6. In this figure, we have considered three cases: samples where NeuroConText achieved the highest Dice scores, samples with median scores, and samples with the lowest scores. The comparison between long and short inputs in the best and median cases shows that reconstructions from augmented descriptions provide more focal activations, with notable overlap between maps generated by NeuroConText and the corresponding NeuroVault statistical maps. We also note that NeuroConText performs poorly on contrasts such as *0-back vs 2-back* and *evaluation of food*. This poor performance is mainly due to limitations in the labeling and missing information. For instance, in the case of *evaluation of food*, the label lacks the term "image", e.g. *evaluation of food image*, which would imply the involvement of the visual cortex. Similarly, for *0-back vs 2-back*, is the opposite of a working memory highlighting contrast, and as such may like salient structures. As a result, NeuroConText fails to capture the upper peaks of this image.

LLM-based text augmentation enhances NeuroConText retrieval performance on short-text input:

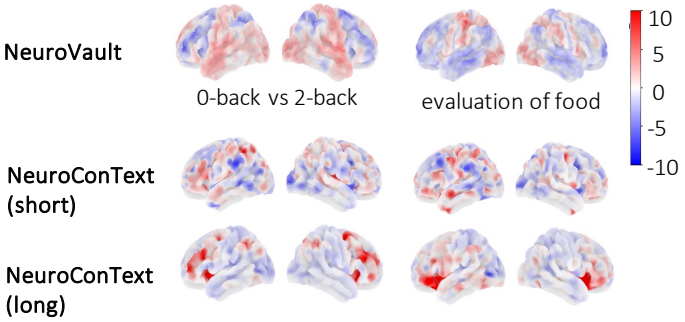
We studied the generalization of NeuroConText on retrieval tasks for short input text. The results detailed in Table 5 in Appendix D show that NeuroConText initially does not generalize well for the short text inputs of NeuroVault descriptions. To enhance NeuroConText's generalization on short texts, we applied our proposed LLM-based text augmentation module presented in Section 3.4. Augmented descriptions consistently improve the performance of NeuroConText both in NeuroConText's latent space and



(a) NeuroVault contrast where NeuroConText achieved the highest Dice Score



(b) NeuroVault contrast with the median Dice Score in NeuroConText



(c) NeuroVault contrast where NeuroConText achieved the lowest Dice Score

Fig 6. Qualitative comparison of brain maps reconstructed from NeuroVault descriptions: This figure presents a qualitative comparison of reconstructed brain maps for three cases where NeuroConText achieved the (a) highest, (b) median, and (c) lowest Dice scores. Each row shows the original NeuroVault statistical map, followed by the reconstructed maps using NeuroConText with short and augmented long text descriptions. In the best and median cases, reconstructions from augmented descriptions show improved localization and overlap with the NeuroVault maps. In the worst-performing cases, low-quality labeling and missing key terms contribute to poor alignment. Specifically, the label *evaluation of food* omits the word "image" leading to missing visual cortex activations, and the *0-back vs 2-back* is not expected to outline clear positive effects. NeuroConText produces more focal reconstructions when using augmented (long) descriptions, particularly in high and median performing cases.

the decoded image space across all metrics. Specifically, in the latent space, augmented descriptions lead to substantial improvements across all three metrics, with Recall@10

increasing by 9%, Recall@100 by 16%, and Mix&Match by 11% compared to the original short descriptions. In contrast, NeuroQuery and Text2Brain do not show significant improvements. In particular, Text2Brain struggles in retrieval performance on the augmented descriptions due to its limited compatibility with analyzing long text.

4.5 CBMA Tools Infer Regions Beyond Reported Coordinates, Mitigating the Limitation of Sparse Coordinate Reporting

In CBMA, brain activation coordinates reported in research articles are often sparse and do not capture all activated regions. This limitation affects the accuracy of the KDE maps, which rely on these coordinates to estimate full brain activation maps as it may fail to reconstruct the complete activation patterns. As CBMA tools leverage text information from research articles, aggregating both experimental descriptions and reported coordinates, we hypothesize that they have the potential to mitigate the challenge of sparse coordinate reporting in CBMA. To evaluate this hypothesis, we extracted all brain regions discussed in each article using Mistral LLM, then mapped these regions to standard brain atlases available using Nilearn [33]. We employed an *atlas label matching* procedure, in which brain region names extracted from the articles' texts were matched to standardized labels from brain atlases. We used six atlases available in Nilearn: AAL [34], Harvard-Oxford [35], DiFuMo [28], Juelich [36], MSDL [37], and Pauli [38]. These atlases are matched with 30.2% of the articles' extracted regions. Using the matched atlas components, we constructed an **atlas-based reference map** by normalizing and averaging the relevant components to each article. We then compared this map to both KDE-based reconstructions and NeuroConText-generated maps to assess whether NeuroConText captures active regions absent in KDE-based maps due to sparse coordinate reporting. The complete pipeline for the atlas matching process is shown in Fig.9 of Appendix G.

To evaluate our approach, we performed both qualitative and quantitative analyses. For quantitative analysis, we computed Dice scores to measure spatial overlap between the atlas-based reference and both KDE- and CBMA-derived maps. Specifically, we calculated Dice scores for the (atlas, KDE) and (atlas, CBMA) pairs across different brain map thresholds and averaged the results across all articles. For qualitative analysis, we visually compared meta-analysis tools' reconstructions with KDE-based maps and the atlas-based reference to determine whether meta-analysis tools identified regions missing from the KDE representation but present in the atlas-based reconstruction.

Our quantitative evaluation presented in Fig. 12 of Appendix G indicates that CBMA tools outperform KDE representation, with higher Dice scores for the (atlas, CBMA) pair than the (atlas, KDE) pair. This supports our hypothesis that CBMA tools have the potential to mitigate the limitations of sparse coordinate reporting in CBMA. In addition, our qualitative analysis for the best, median and worst performing samples of NeuroConText is shown in Appendix G-Fig.13. We also provided the list of articles' regions and the matched atlas labels in Tables 10, 11, and 12 for the best, median, and worst scores. The full lists of atlas-matched regions for all articles are available in the NeuroConText GitHub repository. From these results, we observe that NeuroConText's reconstructions include several active regions in the atlas-based reference but were missing from KDE-based maps. This is particularly observed in the worst-case example, where several valid activations align with the regions discussed in the article, despite the Dice score being very low. This highlights that a low Dice score, when using the KDE map as a reference, does not necessarily indicate poor reconstruction by a meta-analysis tool. Instead, it may result from the limited number of reported coordinates in the article, which underscores the limitations of KDE-based evaluation.

4.6 Validating NeuroConText Elements

4.6.1 NeuroConText vs regression-based strategies

We compared NeuroConText with regression-based methods to associate texts with brain maps. For NeuroConText, we retrieve the text and coordinates latents of the test set through NeuroConText's trained encoders. Then, we calculate retrieval metrics, Recall@K for $K \in \{10, 100\}$, and Mix&Match, on the test data. As a regression-based approach, we consider two models: RidgeCV (linear model) and a non-linear model. For the non-linear model, we implemented a feedforward neural network comprising three fully connected layers of 512 units. Non-linear transformations are introduced through hyperbolic tangent (tanh) activation functions following the first two layers. To enhance training stability, layer normalization is applied after each activation. In addition, a dropout layer with a 50% rate is incorporated after the first activation to mitigate overfitting. This model was trained over 50 epochs using the Adam optimizer [31] with a learning rate of $5e^{-4}$. We train these two models on the average Mistral-7B embeddings of the full-body chunks to estimate the DiFuMo coefficients. We evaluated the association performance of the trained models on test data and compared them with those obtained from NeuroConText and the baseline models. We can see that regression-based methods fail to associate text with brain locations effectively.

Table 2. Comparison of NeuroConText contrastive model with regression-based models for retrieving task: NeuroConText outperforms regression-based models (RidgeCV and non-linear models) in associating texts with brain maps. This table presents the retrieval performance of NeuroConText, RidgeCV (linear regression), and non-linear regression on the test set articles using full-body text. The evaluation metrics include Recall@10, Recall@100, and Mix&Match. NeuroConText achieves the highest scores across all metrics [21].

Method \ Metric [%]	recall@10	recall@100	mix&match
NeuroConText (ours)	$\mathbf{22.6 \pm 1.4}$	$\mathbf{57.8 \pm 1.6}$	$\mathbf{84.2 \pm 0.9}$
RidgeCV (linear model)	14.9 ± 0.8	43.7 ± 1.2	76.5 ± 0.4
Non-linear model)	11.4 ± 0.4	39.4 ± 1.2	74.1 ± 0.8

4.6.2 Long-Text Chunks Aggregation: Mean Aggregation Outperforms Other Strategies

Here, we assess the chunk aggregation strategy for long texts by comparing alternative aggregation methods. We compare the following methods:

- **Random chunk selection:** We selected one random chunk from each article. By choosing a single chunk at random, we avoid introducing any bias from specific parts of the article. We repeated this random selection multiple times to ensure the robustness and consistency of the results.

- **Train-set augmentation over chunks:** To expand the dataset, we construct all possible (chunk, DiFuMo) pairs for each article, where each article has multiple text chunks but is associated with a single DiFuMo. During training, we ensure that no more than one chunk from the same article appears in a batch, maintaining a diagonal structure in the similarity matrix between text and coordinates representation in NeuorConText latent space. For the test set, we explored two approaches while ensuring that the similarity matrix maintains a diagonal structure: 1) randomly selecting one chunk per article and 2) using the average of all chunks per article.

- **Voting strategy via Borda Count:** We use the Borda Count [39] approach for the voting strategy, where each chunk from an article produces a ranked list of brain maps. These rankings reflect how likely each brain map is to match the content of that chunk. To make a final prediction for the article, we apply the Borda Count by combining the ranks from all chunks: brain maps that appear near the top of many chunk-level rankings receive higher overall scores. The brain map with the best overall rank is selected as the final prediction. We evaluate this strategy using recall metrics to assess how well it identifies the correct brain map.

Table 3 shows a comparison the four different aggregation strategies based on evaluation metrics Recall@10, Recall@100, and Mix&Match. From this table, we conclude that averaging chunks performs the best. This is possibly due to the fact that averaging creates embeddings that are more tightly grouped within chunks of an article while maintaining a clear separation between different articles' chunks embeddings in the latent space. To explore this, we used **UMAP projections** to visualize the clustering of embeddings (See Fig.8 in Appendix F). The UMAP visualizations show that the embeddings of any given text form compact groups, that are clearly separated between articles. The small distances within these average embeddings highlight strong grouping, which implies that averaging leads to a good representative of each article's multiple embeddings.

Table 3. Comparison of chunk aggregation strategies for long texts input in NeuroConText: Averaging text chunks' embeddings results in the best performance in associating long neuroscientific articles with brain maps. The averaging outperforms other aggregation strategies by producing more compact and representative embeddings. This table presents the retrieval performance of different chunk aggregation methods in NeuroConText, evaluated using Recall@10, Recall@100, and Mix&Match. The methods compared include random chunk selection, train-set augmentation with different test-time strategies (random chunk vs. averaging chunks), and a voting strategy using Borda Count.

Chunks Aggregation Method	Recall@10	Recall@100	Mix&Match
train: augment - test: average chunks	20.5 ± 1.7	53.5 ± 1.6	81.9 ± 0.5
train: augment - test: random chunk	17.6 ± 1.2	49.9 ± 1.4	79.4 ± 0.6
Voting (Borda Count)	18.5 ± 0.8	51.9 ± 1.9	81.4 ± 0.5
Averaging chunks	**22.6 ± 1.4**	**57.8 ± 1.6**	**84.2 ± 0.9**

4.6.3 Impact of large language model and DiFuMo size

In our final experiment, we compare different LLMs and DiFuMo sizes on the performance of our proposed method. In NeuroConText model, we change the LLM to GPT-Neo-1.3B, GPT-Neo-125M, or SciBERT, alongside two DiFuMo sizes: $k = 256$ and $k = 512$. Then we compare the impact of each on the retrieval performance through the metrics Recall@K for $K \in \{10, 100\}$, and Mix&Match. The findings are summarized in Table 4. We see that the larger language model with Mistral-7B significantly enhances performance, achieving a score more than 4% higher than SciBERT. Moreover, the results indicate that the DiFuMo with size 512 outperforms 256, improving scores by approximately 3%.

Table 4. Impact of LLM and DiFuMo Size on retrieval performance: Using a larger language model (Mistral-7B) and a higher-resolution DiFuMo (512 components) improves retrieval performance in NeuroConText. This table presents the retrieval performance of NeuroConText when using different LLMs (Mistral-7B, GPT-Neo-1.3B, GPT-Neo-125M, and SciBERT) and DiFuMo sizes (256 and 512). The evaluation is conducted using Recall@10, Recall@100, and Mix&Match metrics. The results indicate that Mistral-7B consistently achieves the highest scores, outperforming SciBERT by over 4%. Additionally, increasing the DiFuMo resolution from 256 to 512 improves retrieval metrics by approximately 3% [21].

Setup \ Metric [%]		recall@10	recall@100	mix&match
DiFuMo 256	Mistral-7B	$\mathbf{19.8 \pm 0.9}$	$\mathbf{51.6 \pm 1.3}$	$\mathbf{81 \pm 0.6}$
	GPT-Neo-1.3B	18.1 ± 0.6	48.2 ± 1.3	79.4 ± 0.3
	GPT-Neo-125M	15.1 ± 0.6	42.3 ± 1.3	76.4 ± 0.3
	SciBERT	15.1 ± 0.6	42.8 ± 0.9	76.9 ± 0.3
DiFuMo 512	Mistral-7B	$\mathbf{22.6 \pm 1.4}$	$\mathbf{57.8 \pm 1.6}$	$\mathbf{84.2 \pm 0.9}$
	GPT-Neo-1.3B	21.5 ± 1.1	54.8 ± 1.1	82.7 ± 0.5
	GPT-Neo-125M	17.5 ± 1.1	48.2 ± 1.5	79.7 ± 0.7
	SciBERT	17.9 ± 0.8	50.3 ± 1.5	81 ± 0.8

5 Discussion

In this paper, we have presented NeuroConText, a coordinate-based brain meta-analysis tool for text-brain retrieval and reconstruction of brain activation maps from text. It leverages a multi-objective loss using contrastive and MSE losses for retrieval and reconstruction, along with an ensemble architecture that simultaneously encodes text and brain image data in a shared latent space and decodes brain images from text latent representation. By using an advanced Large Language Model (LLM), NeuroConText captures the semantic content of entire neuroscientific articles for contextual brain map generation. NeuroConText employs LLM-based text augmentation to enhance its performance for short text lengths during inference. Our experimental results show that NeuroConText is capable of inferring additional semantic information from the text to improve the reconstructed brain map, which establishes that it effectively targets the issue that reported coordinates are often sparse and do not entirely cover all regions reported in the articles' tables.

Retrieval Analysis of NeuroConText

We highlight several insights derived from the quantitative and qualitative analyses of NeuroConText's performance in the retrieval task. The quantitative evaluation presented in Fig.3 demonstrates that NeuroConText consistently outperforms existing baselines across all text lengths—from titles and abstracts to full articles—and retrieval metrics. This superior performance comes from the use of contrastive learning in NeuroConText, a novelty with respect to previous meta-analysis tools. Contrastive learning enables NeuroConText to align text and the location of brain activation peaks in a shared latent space, ensuring that semantically similar texts are positioned close to their corresponding brain patterns while unrelated pairs are separated. This allows the model to identify the correct match between a textual description and its associated brain activity while filtering out incorrect associations. By leveraging LLMs, NeuroConText further enhances this process, as their rich, context-aware embeddings effectively handle long-text inputs and capture detailed semantics, which leads to more accurate mappings between text and brain patterns. The positive impact of the

contrastive technique on retrieval performance is demonstrated in Table 2, where we compare it with RidgeCV and MLP models that rely solely on MSE loss. Furthermore, our results indicate that utilizing the full-body text of articles significantly enhances retrieval outcomes, which underscores the importance of leveraging complete textual information within articles.

Table 4 illustrates the status quo regarding advanced semantic extraction. It may change in the future with the advent of novel LLMs. By using larger and more sophisticated language models, NeuroConText achieves improved semantic analysis, which in turn boosts its retrieval capabilities. This finding demonstrates the critical role of deep semantic analysis in improving the accuracy of retrieved brain activation maps.

Reconstruction analysis of NeuroConText

In this paper, we provided both quantitative and qualitative evaluations to assess the performance of NeuroConText in reconstructing brain maps from input queries, resulting in several findings. Fig.4 quantitatively demonstrates that NeuroConText's performance in reconstructing full-body text from left-out articles is on par with the baselines. Qualitative analyses in Fig.5 provide typical illustrations of NeuroContext behavior in view of the brain coordinates and regions reported in each article.

LLM-based text augmentation impact

To accommodate short input queries typical in practical applications, we adapted NeuroConText using an LLM-based text augmentation technique. Our experiments assessed the impact of text augmentation on NeuroConText's performance with short NeuroVault descriptions. Augmenting short texts to mimic full-length articles considerably improved the model's reconstruction (Table 1) and retrieval accuracy (Table 5). In figure 6, text augmentation enhances the reconstruction task in the best and median-performing cases, resulting in more focal activations with a noticeable overlap with the NeuroVault image. Additionally, Fig.6-c illustrates the least successful cases for short texts, emphasizing the challenge NeuroConText encounters when processing tasks with diffuse activations or incomplete descriptions.

CBMA tools can infer brain activation beyond reported coordinates

CBMA methods rely on reported activation coordinates, which are often sparse and may not fully capture the underlying brain regions discussed in the articles. We have shown that meta-analysis tools that leverage text information and aggregate reported coordinates across thousands of articles can infer meaningful brain activations beyond the reported coordinates. Our qualitative and quantitative analyses in Fig. 12 and Fig. 13 of Appendix G show that in some cases, compared with the KDE map, the meta-analysis tools' reconstructions align more closely with atlas-based maps, capturing additional relevant regions absent in KDE-based representations. We also showed that when using KDE as a reference, low Dice scores do not necessarily indicate poor reconstruction but rather reflect the limitations of sparse coordinate reporting. These findings highlight that Dice score using KDE representations of reported coordinates as reference maps provides a limited metric for evaluating CBMA tools, especially when low scores result from the inherent sparsity of reported coordinates rather than poor reconstruction. Meanwhile, comparisons with atlas-based maps could complement the analysis of reconstructed map quality for the evaluation of CBMA tools.

Averaged chunks, larger language models, and larger DiFuMo dictionaries boost NeuroConText

The ablation study in Table 3 demonstrated that for NeuroConText, averaging chunks of text from articles provides the best text representation, and outperforms other methods like random selection and Borda Count voting. Additionally, Table 2 showed that NeuroConText outperformed standard regression-based models (RidgeCV and non-linear models), which highlights the impact of contrastive learning to capture text-brain multimodal data relationships. Further tests in Table 4 showed that larger language models and dictionary sizes enhance performance, suggesting that increased model complexity and richer feature representations are beneficial for accurately mapping neuroscience texts to brain activity.

Limitations of NeuroConText

NeuroConText faces several limitations and challenges. 1) *Text augmentation risks:* While text augmentation using LLM can enhance short-text inputs, it may introduce inaccuracies or hallucinations that require domain expert verification. This reliance on expert oversight can slow the pipeline and increase the effort needed to maintain text data integrity. 2) *Atlas-dependent evaluation for new inferred activations:* Another limitation of our study is the reliance on atlas-based label matching with articles' extracted brain regions. We applied preprocessing steps to harmonize the atlas labels and the brain regions extracted from the articles. This approach allowed us to match 30.2% of the regions to the atlas components. However, we observed in Table 9 of Appendix G that many of the least frequently matched labels were inconsistently formatted or contained abbreviations. This highlights the importance of label quality and the need for further curation and standardization of atlas labels to improve matching coverage.

Acknowledgments

This work is supported by the KARAIB AI chair (ANR-20-CHIA-0025-01), the ANR-22-PESN-0012 France 2030 program, and the HORIZON-INFRA-2022-SERV-B-01 EBRAINS 2.0 infrastructure project.

References

1. Yarkoni T, Poldrack RA, Nichols TE, Van Essen DC, Wager TD. Large-scale automated synthesis of human functional neuroimaging data. Nature methods. 2011;8(8):665–670.

2. Poldrack RA, Gorgolewski KJ. Making big data open: data sharing in neuroimaging. Nature neuroscience. 2014;17(11):1510–1517.

3. Button KS, Ioannidis JP, Mokrysz C, Nosek BA, Flint J, Robinson ES, et al. Power failure: why small sample size undermines the reliability of neuroscience. Nature Reviews Neuroscience. 2013;14(5):365.

4. Poldrack RA, Yarkoni T. From brain maps to cognitive ontologies: informatics and the search for mental structure. Annual review of psychology. 2016;67:587.

5. Eickhoff SB, Bzdok D, Laird AR, Kurth F, Fox PT. Activation likelihood estimation meta-analysis revisited. Neuroimage. 2012;59(3):2349–2361.

6. Fox PT, Lancaster JL. Mapping context and content: the BrainMap model. Nature Reviews Neuroscience. 2002;3(4):319–321.

7. Wager TD, Smith EE. Neuroimaging studies of working memory. Cognitive, Affective, & Behavioral Neuroscience. 2003;3(4):255–274.

8. Luo X, Rechardt A, Sun G, Nejad KK, Yáñez F, Yilmaz B, et al. Large language models surpass human experts in predicting neuroscience results. Nature human behaviour. 2024; p. 1–11.

9. Laird AR, Fox PM, Price CJ, Glahn DC, Uecker AM, Lancaster JL, et al. ALE meta-analysis: Controlling the false discovery rate and performing statistical contrasts. Human brain mapping. 2005;25(1):155–164.

10. Fox MD, Raichle ME. Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. Nature reviews neuroscience. 2007;8(9):700–711.

11. Yarkoni T, Westfall J. Choosing prediction over explanation in psychology: Lessons from machine learning. Perspectives on Psychological Science. 2017;12(6):1100–1122.

12. Mazziotta JC, Toga AW, Evans A, Fox P, Lancaster J, et al. A probabilistic atlas of the human brain: theory and rationale for its development. Neuroimage. 1995;2(2):89–101.

13. Collins DL, Neelin P, Peters TM, Evans AC. Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space. Journal of computer assisted tomography. 1994;18(2):192–205.

14. Laird AR, Lancaster JJ, Fox PT. Brainmap. Neuroinformatics. 2005;3(1):65–77.

15. Wager TD, Lindquist MA, Nichols TE, Kober H, Van Snellenberg JX. Evaluating the consistency and specificity of neuroimaging data using meta-analysis. Neuroimage. 2009;45(1):S210–S221.

16. Dockès J, Poldrack RA, Primet R, Gözükan H, Yarkoni T, Suchanek F, et al. NeuroQuery, comprehensive meta-analysis of human brain mapping. Elife. 2020;9:e53385.

17. Ngo GH, Nguyen M, Chen NF, Sabuncu MR. Text2Brain: Synthesis of Brain Activation Maps from Free-Form Text Query. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2021. p. 605–614.

18. Tian Y, Krishnan D, Isola P. Contrastive multiview coding. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. Springer; 2020. p. 776–794.

19. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 9729–9738.

20. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PMLR; 2021. p. 8748–8763.

21. Meudec R, Ghayem F, Dockès J, Wassermann D, Thirion B. NeuroConText: Contrastive text-to-brain mapping for neuroscientific literature. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2024. p. 325–335.

22. Silverman BW. Density estimation for statistics and data analysis. Routledge; 2018.

23. Scott DW. Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons; 2015.

24. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information processing & management. 1988;24(5):513–523.

25. Beltagy I, Lo K, Cohan A. SciBERT: A pretrained language model for scientific text. arXiv preprint arXiv:190310676. 2019;.

26. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, et al. Mistral 7B. arXiv preprint arXiv:231006825. 2023;.

27. Rubin TN, Koyejo O, Gorgolewski KJ, Jones MN, Poldrack RA, Yarkoni T. Decoding brain activity using a large-scale probabilistic functional-anatomical atlas of human cognition. PLoS computational biology. 2017;13(10):e1005649.

28. Dadi K, Varoquaux G, Machlouzarides-Shalit A, Gorgolewski KJ, Wassermann D, Thirion B, et al. Fine-grain atlases of functional modes for fMRI analysis. NeuroImage. 2020;221:117126.

29. Gorgolewski KJ, Varoquaux G, Rivera G, Schwarz Y, Ghosh SS, Maumet C, et al. NeuroVault. org: a web-based repository for collecting and sharing unthresholded statistical maps of the human brain. Frontiers in neuroinformatics. 2015;9:8.

30. Mitchell TM, Shinkareva SV, Carlson A, Chang KM, Malave VL, Mason RA, et al. Predicting human brain activity associated with the meanings of nouns. science. 2008;320(5880):1191–1195.

31. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

32. Pinho AL, Amadon A, Ruest T, Fabre M, Dohmatob E, Denghien I, et al. Individual Brain Charting, a high-resolution fMRI dataset for cognitive mapping. Scientific Data. 2018;5:180105. doi:10.1038/sdata.2018.105.

33. Abraham A, Pedregosa F, Eickenberg M, Gervais P, Mueller A, Kossaifi J, et al. Machine learning for neuroimaging with scikit-learn. Frontiers in neuroinformatics. 2014;8:14.

34. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, et al. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage. 2002;15(1):273–289.

35. Harvard-Oxford Cortical and Subcortical Structural Atlases;. `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases`.

36. Eickhoff SB, Paus T, Caspers S, Grosbras MH, Evans AC, Zilles K, et al. Assignment of functional activations to probabilistic cytoarchitectonic areas revisited. Neuroimage. 2007;36(3):511–521.

37. Varoquaux G, Gramfort A, Pedregosa F, Michel V, Thirion B. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In: Biennial International Conference on information processing in medical imaging. Springer; 2011. p. 562–573.

38. Pauli WM, Nili AN, Tyszka JM. A high-resolution probabilistic in vivo atlas of human subcortical brain nuclei. Scientific data. 2018;5(1):1–13.

39. Dwork C, Kumar R, Naor M, Sivakumar D. Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web; 2001. p. 613–622.

40. Nesterov Y, et al. Lectures on convex optimization. vol. 137. Springer; 2018.

41. Bertsekas DP. Nonlinear programming. Journal of the Operational Research Society. 1997;48(3):334–334.

42. Rudin W, et al. Principles of mathematical analysis. vol. 3. McGraw-hill New York; 1964.

Supporting Materials

A NeuroConText algorithm and its convergence proof

Proposition 1 establishes the convergence of NeuroConText algorithm, followed with the proof. We note that to set the step size μ, we monitor the convergence of our approach and continuously track the decrease in InfoNCE and MSE losses, assess model performance on a validation set, and examine the norms of the gradients and the stability of parameter updates. This ensures that each optimization step contributes effectively to the overall training objective.

Proposition 1 (Monotonic decrease of the total loss)**.** *Consider the objective functions* $\mathcal{L}_{\text{InfoNCE}}(\theta_{text}, \theta_{img})$, $\mathcal{L}_{\text{MSE}}(\theta_{text}, \theta_{dec})$, *and define*

$$\mathcal{L}\big(\theta_{text}, \theta_{img}, \theta_{dec}\big) \;=\; \mathcal{L}_{\text{InfoNCE}}\big(\theta_{text}, \theta_{img}\big) \;+\; \mathcal{L}_{\text{MSE}}\big(\theta_{text}, \theta_{dec}\big).$$

Assume $\mathcal{L}_{\text{InfoNCE}}$ *is* L_1*-smooth in* $(\theta_{text}, \theta_{img})$*, and* $\mathcal{L}_{\text{MSE}}$ *is* L_2*-smooth in* $(\theta_{text}, \theta_{dec})$*. Then, for a sufficiently small step size* μ*, the update scheme in* ***Algorithm 1*** *produces a nonincreasing sequence* $\{\mathcal{L}(\theta_{text}^{(k)}, \theta_{img}^{(k)}, \theta_{dec}^{(k)})\}$*, which is therefore convergent to a local minimum since* $\mathcal{L}$ *is non-negative and thus lower-bounded.*

Algorithm 1 Alternating Minimization of $\mathcal{L}_{\text{InfoNCE}}(\theta_{\text{text}}, \theta_{\text{img}}) + \mathcal{L}_{\text{MSE}}(\theta_{\text{text}}, \theta_{\text{dec}})$

Inputs: Training data $(\mathbf{X}, \mathbf{Y})$, step size $\mu > 0$, initial parameters $\theta_{\text{text}}^{(0)}, \theta_{\text{img}}^{(0)}, \theta_{\text{dec}}^{(0)}$
while *stopping criterion* not met **do**
  Step 1: Intermediate update of θ_{text} ***w.r.t*** $\mathcal{L}_{\text{InfoNCE}}$***:***

$$\tilde{\theta}_{\text{text}}^{(k+1)} \;=\; \theta_{\text{text}}^{(k)} \;-\; \mu\, \nabla_{\theta_{\text{text}}} \mathcal{L}_{\text{InfoNCE}}\Big(\theta_{\text{text}}^{(k)},\; \theta_{\text{img}}^{(k)}\Big).$$

  Step 2: Update of θ_{img} ***w.r.t*** $\mathcal{L}_{\text{InfoNCE}}$***:***

$$\theta_{\text{img}}^{(k+1)} \;=\; \theta_{\text{img}}^{(k)} \;-\; \mu\, \nabla_{\theta_{\text{img}}} \mathcal{L}_{\text{InfoNCE}}\Big(\theta_{\text{text}}^{(k)},\; \theta_{\text{img}}^{(k)}\Big).$$

  Step 3: Final update of θ_{text} ***w.r.t*** $\mathcal{L}_{\text{MSE}}$***:***

$$\theta_{\text{text}}^{(k+1)} \;=\; \tilde{\theta}_{\text{text}}^{(k+1)} \;-\; \mu\, \nabla_{\theta_{\text{text}}} \mathcal{L}_{\text{MSE}}\Big(\tilde{\theta}_{\text{text}}^{(k+1)},\; \theta_{\text{dec}}^{(k)}\Big).$$

  Step 4: Update of θ_{dec} ***w.r.t*** $\mathcal{L}_{\text{MSE}}$***:***

$$\theta_{\text{dec}}^{(k+1)} \;=\; \theta_{\text{dec}}^{(k)} \;-\; \mu\, \nabla_{\theta_{\text{dec}}} \mathcal{L}_{\text{MSE}}\Big(\theta_{\text{text}}^{(k+1)},\; \theta_{\text{dec}}^{(k)}\Big).$$

  Increase iteration counter: $k \leftarrow k + 1$
  Check *stopping criterion*
end while
Output: $\theta_{\text{text}}^{(k)},\; \theta_{\text{img}}^{(k)},\; \theta_{\text{dec}}^{(k)}$

Proof. Let's define $\theta_1 := \theta_{\text{text}}$, $\theta_2 := \theta_{\text{img}}$, $\theta_3 := \theta_{\text{dec}}$ and two sub-losses $f(\theta_1, \theta_2) \;=\; \mathcal{L}_{\text{InfoNCE}}(\theta_1, \theta_2)$ and $g(\theta_1, \theta_3) \;=\; \mathcal{L}_{\text{MSE}}(\theta_1, \theta_3)$. Then, the total objective becomes:

$$\mathcal{L}(\theta_1, \theta_2, \theta_3) \;=\; f(\theta_1, \theta_2) \;+\; g(\theta_1, \theta_3).$$

We divide the proof into two parts: (1) showing that the θ_1, θ_2 updates (Steps 1–2 in Algorithm 1) reduce f; and (2) showing that the θ_1, θ_3 updates (Steps 3–4 in Algorithm 1) reduce g. We then argue that these reductions are not undone by the intervening updates, provided μ is sufficiently small.

1. Steps 1–2 decrease $f(\theta_1, \theta_2)$:

Step 1 (gradient descent on θ_1 wrt f):

$$\tilde{\theta}_1^{(k+1)} \;=\; \theta_1^{(k)} \;-\; \mu\, \nabla_{\theta_1} f\big(\theta_1^{(k)}, \theta_2^{(k)}\big).$$

Under L_f-smoothness of f and for $\mu \le 1/L_f$, standard gradient-descent theory [40, 41] guarantees

$$f\big(\tilde{\theta}_1^{(k+1)}, \theta_2^{(k)}\big) \;\le\; f\big(\theta_1^{(k)}, \theta_2^{(k)}\big) \;-\; \Delta_{f,1}^{(k)},$$

where $\Delta_{f,1}^{(k)} > 0$ depends on μ and $\|\nabla_{\theta_1} f(\cdot)\|^2$.

Step 2 (gradient descent on θ_2 wrt f):

$$\theta_2^{(k+1)} \;=\; \theta_2^{(k)} \;-\; \mu\, \nabla_{\theta_2} f\big(\theta_1^{(k)}, \theta_2^{(k)}\big).$$

By L_f-smoothness in θ_2, we get

$$f\big(\theta_1^{(k)}, \theta_2^{(k+1)}\big) \;\le\; f\big(\theta_1^{(k)}, \theta_2^{(k)}\big) \;-\; \Delta_{f,2}^{(k)},$$

for some positive $\Delta_{f,2}^{(k)}$. Combining these sub-steps yields

$$f\big(\tilde{\theta}_1^{(k+1)}, \theta_2^{(k+1)}\big) \;\le\; f\big(\theta_1^{(k)}, \theta_2^{(k)}\big) \;-\; \Delta_f^{(k)}, \quad \Delta_f^{(k)} = \Delta_{f,1}^{(k)} + \Delta_{f,2}^{(k)} > 0.$$

Hence, after Steps 1–2, f is strictly decreased from its value at the start of iteration k.

2. Steps 3–4 Decrease $\boldsymbol{g(\theta_1, \theta_3)}$.

Step 3 (gradient descent on θ_1 wrt g):

$$\theta_1^{(k+1)} \;=\; \tilde{\theta}_1^{(k+1)} \;-\; \mu\, \nabla_{\theta_1} g\big(\tilde{\theta}_1^{(k+1)}, \theta_3^{(k)}\big).$$

If g is L_g-smooth in θ_1 and $\mu \le 1/L_g$, then

$$g\big(\theta_1^{(k+1)}, \theta_3^{(k)}\big) \;\le\; g\big(\tilde{\theta}_1^{(k+1)}, \theta_3^{(k)}\big) \;-\; \Delta_{g,1}^{(k)}.$$

Step 4 (gradient descent on θ_3 wrt g):

$$\theta_3^{(k+1)} \;=\; \theta_3^{(k)} \;-\; \mu\, \nabla_{\theta_3} g\big(\theta_1^{(k+1)}, \theta_3^{(k)}\big).$$

By the same L_g-smoothness argument,

$$g\big(\theta_1^{(k+1)}, \theta_3^{(k+1)}\big) \;\le\; g\big(\theta_1^{(k+1)}, \theta_3^{(k)}\big) \;-\; \Delta_{g,2}^{(k)},$$

so combining both sub-steps gives

$$g\big(\theta_1^{(k+1)}, \theta_3^{(k+1)}\big) \;\le\; g\big(\tilde{\theta}_1^{(k+1)}, \theta_3^{(k)}\big) \;-\; \Delta_g^{(k)}, \quad \Delta_g^{(k)} = \Delta_{g,1}^{(k)} + \Delta_{g,2}^{(k)} > 0.$$

Thus, after Steps 3–4, g is strictly decreased from its value at the end of Step 2.

3. Ensuring the second update of θ_1 (Step 3) cannot undo the decrease in f. A critical concern is that θ_1 is changed *twice* in the same iteration: once w.r.t. f (Step 1), once w.r.t. g (Step 3). In the following, we prove that by choosing μ sufficiently small, the second update does not cause f to increase again. In particular, a common sufficient condition is

$$0 < \mu \;<\; \frac{1}{L_f + L_g} \quad \text{or} \quad 0 < \mu \;<\; \frac{1}{2 \times \max(L_f,\, L_g)}.$$

3.1 First-order vs. second-order effects.

1. *Direct update on f* (Step 1) is a *first-order* improvement. By a first-order Taylor expansion [42]:

$$f(\theta_1 - \mu\,\nabla f(\theta_1)) \;\approx\; f(\theta_1) \;-\; \mu\,\|\nabla f(\theta_1)\|^2,$$

   indicating a change proportional to $\mu\,\|\nabla f\|$.

2. *Change in f due to a step on g* (Step 3) is a *second-order* Lipschitz effect. If $\Delta\theta_1 = -\mu\,\nabla g$, then by L_f-smoothness:

$$\big|\Delta f\big| \;=\; \big|f(\theta_1 + \Delta\theta_1) - f(\theta_1)\big| \;\le\; \tfrac{L_f}{2}\,\|\Delta\theta_1\|^2 \;=\; \mathcal{O}(\mu^2).$$

Hence, the first-order decrease from Step 1 (of order μ) dominates the second-order bump (of order μ^2) for sufficiently small μ. In practice, one often takes $\mu < 1/(L_f + L_g)$ so that $\mu^2 \ll \mu$. Similarly, Steps 1–2 cannot undo progress on g from Steps 3–4.

4. Conclusion: Non-increase of $\mathcal{L}$ in one full iteration. Therefore, if μ is small enough,

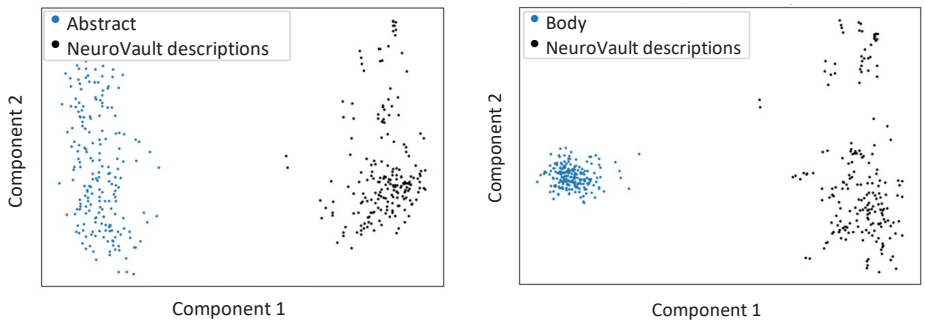
$$f\big(\theta_1^{(k+1)}, \theta_2^{(k+1)}\big) \;+\; g\big(\theta_1^{(k+1)}, \theta_3^{(k+1)}\big) \;\le\; f\big(\theta_1^{(k)}, \theta_2^{(k)}\big) \;+\; g\big(\theta_1^{(k)}, \theta_3^{(k)}\big).$$

As a result,

$$\mathcal{L}\big(\theta_1^{(k+1)}, \theta_2^{(k+1)}, \theta_3^{(k+1)}\big) \;\le\; \mathcal{L}\big(\theta_1^{(k)}, \theta_2^{(k)}, \theta_3^{(k)}\big).$$

Since $\mathcal{L}$ is lower-bounded (non-negative), this non-increasing sequence converges to a finite limit. □

B Visualizing the distribution shift between short and long text



(a) PCA: NeuroVault contrasts vs abstracts (b) PCA: NeuroVault contrasts vs body

Fig 7. PCA decomposition of LLM embeddings comparing NeuroVault contrast definitions (short text) with abstracts and full-body texts (long text). The left plot shows a clear distribution shift along the first component between NeuroVault descriptions and abstracts, while the right plot shows a more pronounced shift along both the first and second components between NeuroVault descriptions and full-body texts. This indicates a significant distribution shift between short NeuroVault descriptions and full-length articles.

C Prompt for augmenting NeuroVault contrast descriptions

Below is the prompt we used to extend NeuroVault contrast descriptions with GPT-4o:

```
I have 169 descriptions of the NeuroVault contrasts dataset for the
brain. On the other hand, I have trained an autoencoder to encode
the brain text and images from neuroscientific articles and to
decode images from the text's latent (encoded) representation. Now,
I want to use my trained autoencoder to encode the NeuroVault
descriptions and decode (estimate) the NeuroVault brain contrasts.
However, the NeuroVault descriptions are very short, while my
autoencoder networks have been trained on lengthy neuroscientific
articles. Therefore, I want to extend the NeuroVault descriptions to
 resemble the body of an article, including sections like
introduction, framework/method, experiments, discussion, conclusion,
 etc., similar to an 8-page or longer article. I will provide you
with the NeuroVault descriptions, and I would like you to extend
them for me. I want the text to be as long as possible and to
include all the relevant key terms related to the description.
```

D The impact of LLM-based text augmentation on the performance of the retrieval task

Table 5. Retrieval task on NeuroVault: the impact of NeuroConText contrastive approach and LLM-based text augmentation. Augmented descriptions consistently improve the performance of NeuroConText (both in latent and decoded forms) across all metrics—Recall@10, Recall@100, and Mix&Match—indicating that LLM-based text augmentation enhances the model's ability to retrieve relevant brain maps. For NeuroConText, augmented descriptions result in significant gains in Recall@10 (+9%), Recall@100 (+16%) and Mix&Match (+11%) compared to original descriptions. Similarly, NeuroConText (decoded) benefits from augmentation, showing improvements of +10%, +7%, and +8% across the three metrics, respectively. In contrast, NeuroQuery and Text2Brain do not show significant improvements, with Text2Brain showing slight declines across all metrics when using augmented descriptions.

Method	**Metric** [%] ↑ / **Description Type**	**Recall@10**	**Recall@100**	**Mix&Match**
NeuroConText (in latent)	Augmented description	13	71	64
	Original description	4	55	53
	Difference (aug. - org.)	9	16	11
NeuroConText (decoded)	Augmented description	20	70	66
	Original description	10	63	58
	Difference (aug. - org.)	10	7	8
NeuroQuery	Augmented description	20	70	65
	Original description	20	67	63
	Difference (aug. - org.)	0	3	2
Text2Brain	Augmented description	12	61	58
	Original description	13	67	62
	Difference (aug. - org.)	-1	-6	-4

E Prompt used for article summarization

We used a customized prompt to extract essential information about brain regions and associated cognitive processes to generate summaries of neuroscientific articles. The prompt instructed the latest version of the Mistral language model to act as a neuroscientific expert, asking relevant questions about the article and providing answers. Below is the exact prompt used:

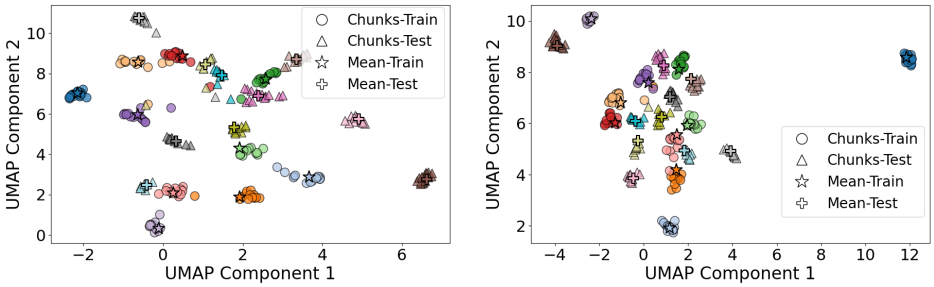
```
You are a neuroscientific expert. Your task is to write a summary of
 a neuroscientific article. When presented with the article, come up
 with interesting questions to ask and answer each question.
Afterward, combine all the information and write a summary.

article: {essay}
Instructions:
*Summarize: In clear and concise language, summarize the key points
and themes presented in the article.

*Write a short summary: Write a short summary in 2 sentences.

*The brain regions: Write the brain regions and any cognitive or
affective processes related to brain activation.
```

F UMAP visualizations of the chunks' LLM embeddings



(a) Original Mistral-7B embeddings space (b) NeuroConText text latent space

Fig 8. UMAP visualizations of the top 10 articles with the highest number of chunks. The left figure shows the original Mistral-7B embeddings and the right figure shows the encoded text embeddings in the shared latent space of NeuroConText model. Each color corresponds to the chunks of a given article. In both representations, chunks corresponding to the same article are closely clustered around their mean, while the means of different articles remain well-separated. This pattern supports the effectiveness of the averaging strategy, as it preserves proximity within an article's chunks to their mean while maintaining clear separation from the chunks of other articles in the latent space.

G Pipeline and results to assess meta-analysis tool's ability to mitigate sparse coordinate reporting challenges in CBMA

To evaluate how well meta-analysis tools reconstruct brain activation patterns discussed in the literature, we first construct a reference map for each article based on anatomical labels. Using Mistral-7B, we extract all brain regions mentioned in each article. We then match these extracted regions to labels from six atlases available in Nilearn (AAL, Harvard-Oxford, DiFuMo, Juelich, MSDL, and Pauli), applying some preprocessing and normalization to both sets of labels (atlases' labels and extracted articles' regions) to improve alignment (Table 6). Hemisphere sensitivity is also considered during matching, *e.g.* regions like "left Hippocampus" are only compared to atlas components from the corresponding hemisphere, according to the rules in Table 7. Note that we consider more directional restrictions when matching article region names to atlas labels; our pipeline considers that all terms (words) from the article region must appear in the atlas label after standardization. This includes directional terms like "superior", "inferior", "anterior", and "posterior". If such a term is present in the article region but missing in the atlas label, the match fails. Table 8 provides an example for matching with directional terms. With this preprocessing, we obtained 30.21% coverage of the articles' regions by the 6 atlas labels. Fig. 10 presents the distribution of matched brain region labels across different atlases, showing that the majority of matches (over 90%) originate from the DiFuMo atlas. Also, the list of 30 most frequently matched labels (all associated with DiFuMo atlas), and the 30 least frequent matched labels are shown in Figure 11 and Table 9, respectively.

After matching the atlas labels with the article regions, we average the matched atlas components to form a reference brain map that represents the reported regions for each article. We compute the Dice score between this atlas-based reference map and the reconstructed maps obtained from different meta-analysis tools, as well as the KDE representation of activation coordinates. Meta-analysis tools (NeuroConText,

NeuroQuery, and Text2Brain) consistently achieve higher Dice scores than KDE, suggesting their superior ability to capture activation patterns and mitigate limitations due to sparse coordinate reporting in CBMA.

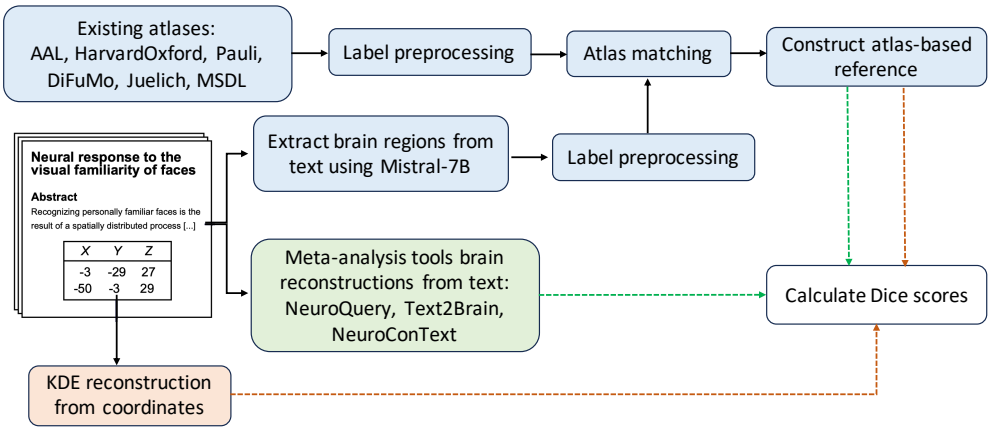


Fig 9. Atlas-based reference map framework to assess the reconstruction performance of meta-analysis tools: This framework illustrates the pipeline for evaluating brain map reconstructions from meta-analysis tools using atlas-based reference maps. Brain regions are first extracted from article text using Mistral-7B. After label preprocessing, these extracted regions are matched to components from six brain atlases (AAL, Harvard-Oxford, DiFuMo, Juelich, MSDL, Pauli). The matched components are averaged to construct a reference map for each article. Separately, reconstructed maps are obtained from meta-analysis tools (NeuroQuery, Text2Brain, NeuroConText) as well as KDE. Dice scores are then computed between each reconstructed map and the corresponding atlas-based reference. The blue blocks highlight the steps involved in constructing the atlas-based reference maps.

Table 6. Text preprocessing rules for atlas labels and article regions.

Before	After
LH, lh, _lh, _l	left
RH, rh, _rh, _r	right
hemi, h, hemisphere	hemisphere
gyri	gyrus
cortex	(removed)
lobe	(removed)
hemisphere	(removed)
Extra spaces	Removed
All text	Lowercased

Table 7. Hemisphere matching between article regions and atlas components

Article region hemisphere	Assigned atlas component hemisphere.
right	right
left	left
left and right	left, right, right & left, non
non	left, right, right & left, non

Table 8. Examples of region matching with directional terms.

Article Region	Atlas Label	Match?
superior frontal gyrus	superior frontal gyrus	Yes
superior frontal gyrus	frontal gyrus	No
frontal gyrus	superior frontal gyrus	Yes
posterior cingulate	posterior cingulate cortex	Yes
posterior cingulate	cingulate cortex	No
inferior parietal lobule	inferior parietal lobule	Yes
inferior parietal lobule	parietal lobule	No

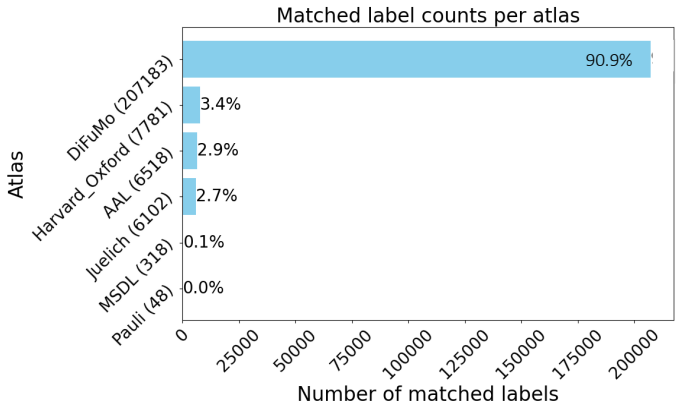


Fig 10. Number of matched brain regions per atlas across all articles.

In Fig. 13, we show three qualitative examples to compare the atlas-based reference maps with the KDE map and the reconstructed map generated by NeuroConText. These examples correspond to the articles for which NeuroConText achieved the best, median, and worst Dice scores. The first row displays KDE-based reconstructions, the second row shows the atlas-based reference maps averaged over six atlases, the third row shows the reference using only the DiFuMo atlas, and the final row presents NeuroConText-generated maps. Tables 10, 11, and 12 report the brain regions extracted from the articles using a large language model (LLM), followed by the matched atlas labels for Articles 1, 2, and 3, respectively. The matching was performed across six different atlases. The full lists of atlas-matched regions are available in the NeuroConText GitHub repository.

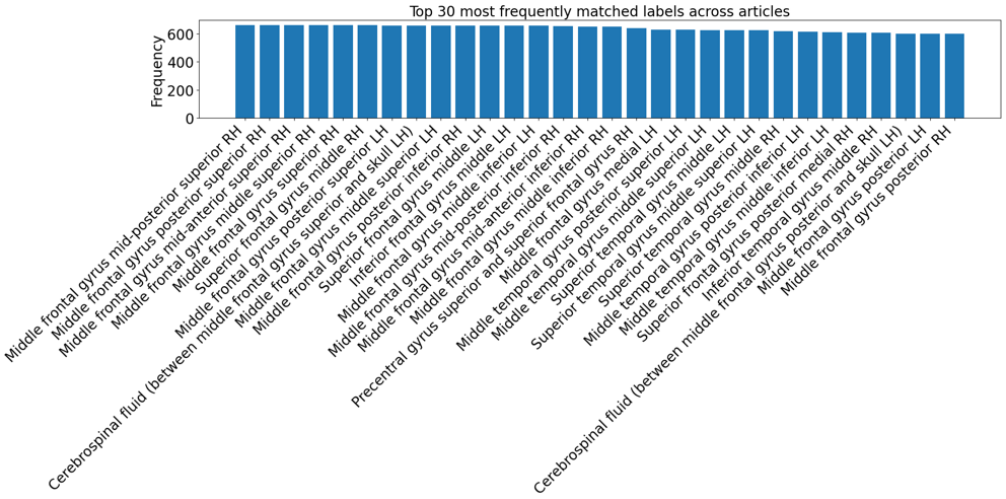


Fig 11. List of 30 most frequently matched labels.

Table 9. 30 least frequent matched labels with their corresponding atlases

Label	Count	Atlas
Vermis_10	2	AAL
Vermis_1_2	2	AAL
Vermis_3	2	AAL
Vermis_4_5	2	AAL
Vermis_6	2	AAL
Vermis_7	2	AAL
Vermis_8	2	AAL
Vermis_9	2	AAL
Suborbital sulcus	2	DiFuMo
WM Acoustic radiation	2	Juelich
WM Uncinate fascicle	2	Juelich
Broca	2	MSDL
Med DMN	2	MSDL
STH	2	Pauli
GM Amygdala_centromedial group	3	Juelich
GM Amygdala_laterobasal group	3	Juelich
GM Amygdala_superficial group	3	Juelich
Basal	3	MSDL
Cereb	3	MSDL
Cing	3	MSDL
Front DMN	3	MSDL
Occ post	3	MSDL
Striate	3	MSDL
Sup Front S	3	MSDL
Vis	3	MSDL
Ca	3	Pauli
EXA	3	Pauli
GPe	3	Pauli
GPi	3	Pauli
HN	3	Pauli

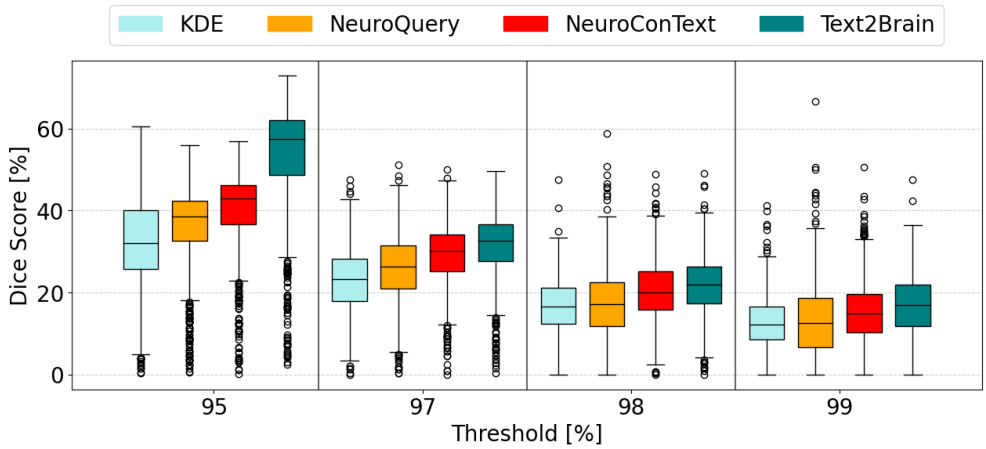


Fig 12. Dice similarity between atlas-based reference maps and reconstructed brain maps: Dice scores are computed between atlas-based reference maps (constructed by matching article-extracted region names to atlas components) and reconstructed brain maps generated by KDE, NeuroQuery, Text2Brain, and NeuroConText. Scores are shown across different threshold levels (95–99%) applied to the reconstructed maps. Meta-analysis tools consistently yield higher Dice scores than KDE, especially at lower thresholds. This indicates meta-analysis tools provide better overlap with the region-based reference.

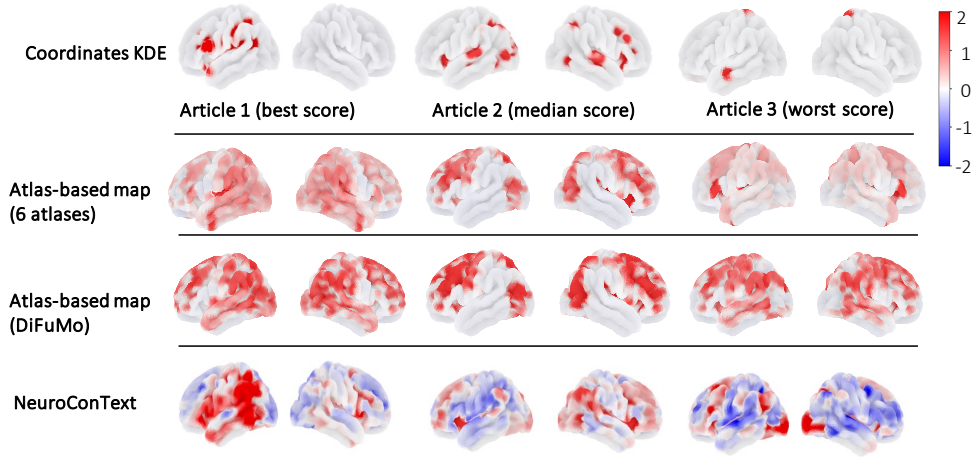


Fig 13. Qualitative analysis to asses NeuorConText's ability to mitigate the sparse coordinate limitation of CBMA:(a) Visualization of reconstructed brain maps for three representative articles with the best, median, and worst Dice scores. The first row shows KDE-based reconstructions, the second row presents the atlas-based reference maps averaged over 6 different atlases, the third row presents the atlas-based reference maps based on only DiFuMo atlas, and the last row displays NeuroConText-generated maps. The comparison highlights that NeuroConText captures broader activation patterns compared to KDE, aligning more closely with the atlas-based reference.

Table 10. Brain regions and atlas label matching for **Article 1 - (best Dice score)**. The full list is available in the NeuroConText GitHub repository for *PMID 30210434*.

Source	Sample Regions
Article regions extracted via Mistral LLM	**Left Ventro-dorsal Stream**: Inferior parietal lobule (incl. Supramarginal gyrus), Inferior frontal gyrus (ventral premotor cortex, opercular, triangular), Precentral gyrus, Rolandic operculum, Insula. **Left Frontal Cortex**: Precentral gyrus, Superior/Middle/Inferior frontal gyri (incl. opercular, triangular, orbital), Rolandic operculum. **Left Parietal Cortex**: Postcentral gyrus, Superior and Inferior parietal lobules (incl. Supramarginal and Angular gyri). **Left Temporal Cortex**: Superior and Middle temporal gyri, Superior temporal pole. **Subcortical Regions**: Putamen, Pallidum, Thalamus. **Other**: Left Insula, Left Angular Gyrus.
AAL	Pallidum_L, Pallidum_R, Putamen_L, Putamen_R, Thalamus_L, Thalamus_R
Harvard-Oxford	Angular Gyrus, Frontal Medial Cortex, Intracalcarine Cortex, Middle Temporal Gyrus (anterior division), Planum Temporale, Superior Frontal Gyrus
DiFuMo (1024)	Angular gyrus anterior LH, Frontal pole medial, Fusiform gyrus posterior RH, Lingual gyrus posterior LH, Middle frontal gyrus mid-anterior RH, Thalamus superior. *Full region mapping available in NeuroConText GitHub repository.*
Juelich	GM Parietal operculum OP1–OP4, GM Superior parietal lobule 7A, 7PC

Table 11. Brain regions and atlas label matching for **Article 2 - (median Dice score)**. The full list is available in the NeuroConText GitHub repository for *PMID 36166644*.

Source	Sample Regions
Article regions extracted via Mistral LLM	**Frontal Regions**: Frontal to central areas, Medial frontal area, Middle frontal gyrus (incl. right), Inferior frontal gyrus, Anterior insula (incl. dorsal anterior and right dorsal anterior insula), Precentral gyrus, Rolandic operculum. **Parietal Regions**: Inferior and Superior parietal lobules, General parietal areas. **Temporal Regions**: Superior temporal gyrus (bilateral and posterior), Superior temporal plane (STP), Planum temporale (PT), Temporo-parieto-occipital junction (TPO). **Occipital Regions**: Occipital areas, Inferior occipital gyrus, Visual cortex. **Cingulate Regions**: Anterior cingulate cortex (ACC), Dorsal ACC (dACC), Midcingulate cortex (MCC). **Other**: Fusiform gyrus, Optic nerve, Optic chiasm, Salience network, Superior olivary nucleus, Right/Left/Bilateral hemispheres, Contralateral and Ipsilateral attention systems.
AAL	Insula_L, Insula_R
DiFuMo (1024)	Anterior cingulate cortex anterior LH, Anterior insula (multiple subdivisions), Middle frontal gyrus (multiple subdivisions), Superior parietal lobule (multiple subdivisions), Dorsolateral prefrontal cortex LH, Fusiform gyrus RH. *Full region mapping available in the repository.*

Table 12. Brain regions and atlas label matching for **Article 3 - (worst Dice score)**. The full list is available in the NeuroConText GitHub repository for *PMID 30885230*.

Source	Sample Regions
Article regions extracted via Mistral LLM	**Limbic and Subcortical**: Amygdala, Hippocampus (bilateral), Anterior cingulate cortex (ACC), Middle cingulate cortex. **Frontal Regions**: Dorsolateral prefrontal cortex (DLPFC), General prefrontal cortex, Frontal areas. **Parietal Regions**: Intraparietal sulcus (IPS), Paracentral gyrus (bilateral). **Temporal and Occipital Regions**: Visual number form (posterior inferior temporal gyrus), Left temporal gyrus. **Other**: Insula.
AAL	Amygdala_L, Amygdala_R, Hippocampus_L, Hippocampus_R, Insula_L, Insula_R, Caudate_L, Caudate_R, Cuneus_L, Cuneus_R, Precuneus_L, Precuneus_R, Putamen_L, Putamen_R, Thalamus_L, Thalamus_R
Harvard-Oxford	Angular Gyrus, Insular Cortex, Lingual Gyrus, Middle Frontal Gyrus, Occipital Fusiform Gyrus, Paracingulate Gyrus, Postcentral Gyrus, Precentral Gyrus, Superior Frontal Gyrus
DiFuMo (1024)	Amygdala (anterior/posterior), Anterior insula (multiple subdivisions), Caudate (multiple subdivisions), Hippocampus posterior RH, Cuneus (anterior/posterior/superior), Middle and Superior frontal gyri (multiple subdivisions), Angular gyrus, Fusiform gyrus posterior RH. *Full region mapping available in the repository.*
Juelich	GM Hippocampus (cornu ammonis, dentate gyrus, subiculum), GM Insula (Id1, Ig1, Ig2), GM Hippocampus entorhinal cortex, GM Hippocampus-amygdaloid transition area