Voxel as Token: A New Perspective for Zero-shot Cross-subject Vision Decoding

Yulong Liu[1] **Ziqiu Huang**[2] **Hua Xu**[1] **Guibo Zhu**[3,4,5] **Sirui Han**[1*] **Yike Guo**[1]
[1]The Hong Kong University of Science and Technology
[2]Northwest Polytechnical University
[3]Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences
[4]School of Artificial Intelligence, University of Chinese Academy of Sciences
[5]Wuhan AI Research
yliuom@connect.ust.hk huang_ziqiu@mail.nwpu.edu.cn
gbzhu@nlpr.ia.ac.cn {huaxu,siruihan,yikeguo}@ust.hk

Abstract

Recent advances in brain decoding have achieved impressive visual reconstruction under subject-specific settings, yet they fail to generalize to unseen subjects without retraining—hindering practical zero-shot cross-subject applications. The core challenges stem from inter-subject variability in voxel dimensionality and misaligned spatial response patterns. While existing methods rely on surface-based alignment or complex feature disentanglement, they often operate as black boxes and overlook the potential of volume-based fMRI data. To address this, we propose **Voxel as Token (VoxTok)**, a conceptual framework that treats each fMRI voxel as an independent token endowed with a learnable functional embedding. This formulation naturally handles variable-length inputs and subsumes existing adapter-based models as special cases where linear projections approximate voxel functionality. Guided by the hypothesis that large-scale spatial distributions of neural activity are consistent across subjects despite fine-grained variability, we introduce a KNN-based estimator to predict functional embeddings for unseen subjects. This approach outperforms existing volume-based zero-shot baselines by a large margin in retrieval and delivers competitive zero-shot reconstruction directly in native 3D voxel space, while enabling plug-and-play adaptation for existing models (e.g., MindEye2). Integrating multi-scale ROIs further boosts performance, with our best model achieving 40.4% zero-shot image retrieval accuracy, rivaling early supervised methods. Crucially, we find that decoding accuracy exhibits a positive trend with ROI size that is reproducible across datasets, indicating that global co-activation patterns drive cross-subject generalization.

1 Introduction

Decoding visual stimuli from human brain activity, particularly via functional MRI (fMRI), has advanced rapidly. State-of-the-art models like MindEye2 [1] achieve high-fidelity image reconstruction by training subject-specific adapters that map individual fMRI responses to a shared latent space, followed by a universal decoder. However, these approaches are inherently subject-specific: each new participant requires labeled data to fine-tune an adapter, precluding true zero-shot cross-subject generalization—a critical bottleneck for clinical and neuroscientific deployment.

The fundamental barriers to zero-shot transfer are twofold: (1) inter-subject variability in voxel count and location, and (2) inconsistent spatial response patterns even within anatomically aligned regions.

*Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Zero-shot decoding—requiring neither per-subject retraining nor natural-image pairs from the new subject—remains the key to practical deployment. Recent attempts to address these challenges, such as fixed-length interpolation [2], meta-learned cross-subject adapters [3], or surface-based pretraining with complex feature disentanglement [4], often operate as black boxes and obscure the mechanisms of cross-subject transfer. Moreover, leading zero-shot methods like ZEBRA [4] rely on surface-based representations that exclude subcortical structures and demand cumbersome preprocessing. In contrast, volume-based fMRI data preserves the full 3D brain anatomy yet remains largely unexplored for zero-shot decoding, leaving a critical gap in understanding how much generalization can be achieved directly from volumetric signals.

In this work, we introduce a conceptual shift: treating each fMRI voxel as a token. Inspired by the sequence modeling flexibility of Transformers, we reframe brain decoding as the learning and aggregation of functional vectors associated with individual voxel tokens. This formulation naturally accommodates variable input lengths, eliminates the need for rigid dimensional normalization, and reinterprets existing adapter-based models as special cases where linear projections approximate voxel functionality. Consequently, zero-shot cross-subject decoding simplifies to estimating functional embeddings for the voxels of unseen subjects.

To estimate these embeddings, we leverage prior evidence that cross-subject brain activity shares coarse-grained spatial similarities [5, 6]. We propose a KNN-based estimator that infers functional vectors for unseen subjects by interpolating from pretrained embeddings in a support set, conditioned on normalized 3D voxel coordinates. This strategy enables plug-and-play enhancement of existing adapter models without retraining, offering a practical solution for zero-shot cross-subject decoding. As illustrated in Fig. 1, our framework incorporates a multi-scale ROI fusion strategy and augments pretraining with KNN-estimated embeddings to improve robustness.

VoxTok significantly outperforms existing baselines in zero-shot retrieval, achieving 40.4% image retrieval accuracy and rivaling early supervised methods. While ZEBRA [4] attains higher reconstruction fidelity through large-scale surface-based masked autoencoding and an intricate feature-disentanglement pipeline, VoxTok reaches competitive zero-shot reconstruction directly in native 3D voxel space, without surface mapping or large-scale pretraining. Further analysis reveals that decoding accuracy exhibits a positive trend with ROI size that reproduces across datasets. Together with the performance gains from multi-scale ROI integration, these findings suggest that cross-subject generalization is driven by distributed, co-occurring activation patterns. Our work thus establishes a new, interpretable foundation for generalizable brain decoding.

This work's main contributions are summarized as follows:

- Voxel-as-Token Formulation: We reframe brain decoding by treating each fMRI voxel as a token with a learnable functional vector, enabling flexible input lengths and unifying existing adapter-based methods under a single theoretical framework.
- Plug-and-Play Zero-Shot Decoding: We propose a KNN-based estimator that infers functional embeddings for unseen subjects from pretrained 3D-coordinate-conditioned vectors, enabling training-free cross-subject generalization with multi-scale ROI fusion.
- Strong Empirical Results: VoxTok achieves 40.4% zero-shot image retrieval accuracy—surpassing prior volume-based zero-shot baselines by a large margin—and delivers competitive zero-shot reconstruction directly in volumetric space, trading some reconstruction fidelity for a simple pipeline and training-free transfer compared with surface-based methods.
- Interpretable Insights into Neural Generalization: Systematic analysis reveals that decoding accuracy exhibits a positive trend with ROI size that reproduces across datasets. Coupled with the gains from multi-scale ROI integration, these findings demonstrate that cross-subject generalization is driven by distributed, co-occurring activation patterns. Moreover, interpolating per-voxel functional vectors in the functional embedding space transfers across subjects substantially better than interpolating voxel responses in 3D voxel space.

2 Related Work

Visual brain decoding has evolved from reconstructing simple artificial stimuli with linear encoding models [7, 8, 9, 10, 11] to generating complex natural images via deep generative frameworks such as diffusion models [1, 12, 13, 14, 15, 16, 17]. Concurrently, the training paradigm has

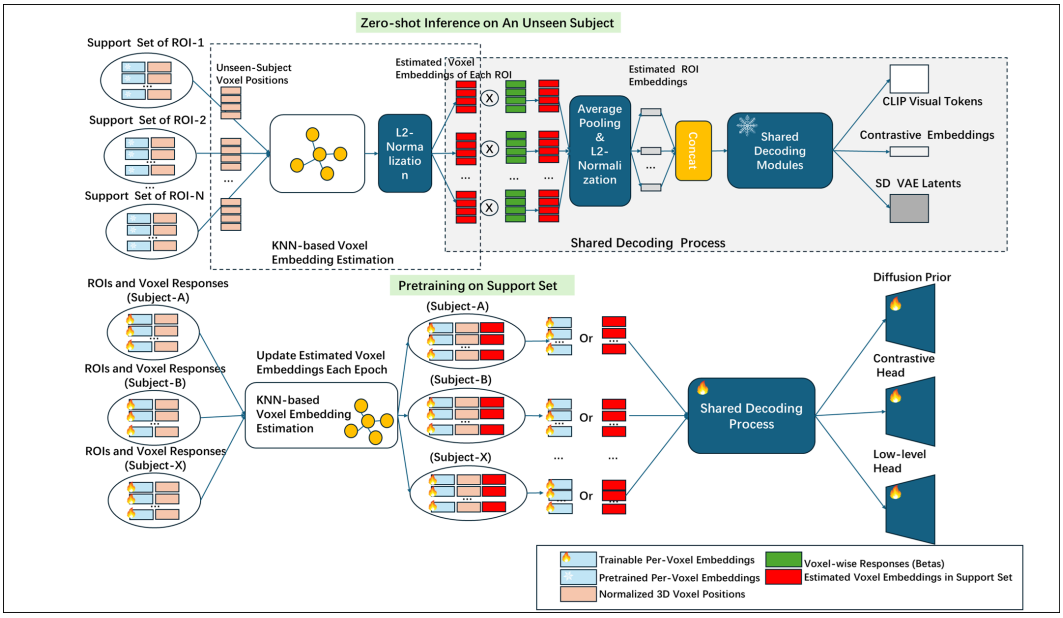


Figure 1: This figure illustrates the architecture of the proposed VoxTok Framework, which treats each voxel as a token. The voxels from the pretraining set are used as support sets. A KNN-based estimator is used to estimate voxel embeddings for an unseen subject. During pretraining, the KNN estimator is also used to conduct data augmentation(See Algorithm 1).

shifted from subject-specific modeling toward cross-subject foundation models that aim to capture subject-invariant neural representations [1, 12, 16, 18, 19, 20, 21]. While recent advances achieve impressive reconstruction quality under data-rich conditions, their performance degrades significantly in zero-shot scenarios for unseen subjects. Kong et al.[2] demonstrate above-chance retrieval on NSD via simple linear interpolation for dimension alignment. More recently, ZEBRA [4] pretrains a foundation model on surface-based fMRI using masked modeling and employs an adversarial disentanglement pipeline to isolate subject-invariant patterns, yielding competitive zero-shot transfer but at the cost of training complexity and interpretability. Alongside these decoding approaches, functional and anatomical alignment has long been studied for sharing information across subjects: hyperalignment and related techniques map individual responses into a common functional space to enable inter-subject decoding [22, 23]. In the encoding direction, the Universal Brain Encoder [24] jointly trains a voxel-centric encoder across diverse subjects and datasets, but it remains an *encoding* model (image $\rightarrow$ fMRI) and its transfer to a new subject still requires a small amount of that subject's data (few-shot adaptation), whereas VoxTok performs *decoding* (fMRI $\rightarrow$ image) with no natural-image pairs from the target subject at all. Notably, the best zero-shot works operate on surface-projected data, a representation derived from volumetric data. However, the extent to which native volume data can support cross-subject generalization remains underexplored. Therefore, we return to the native *volume-based* fMRI representation and propose a *voxel-as-token* perspective that treats individual voxels as discrete semantic units within their 3D spatial lattice. This framework enables direct analysis of cross-subject transferability from the original volumetric space—avoiding surface projection artifacts and complex multi-stage training—while providing interpretable insights into the spatial-semantic origins of generalization in brain decoding.

3 Method

3.1 Overview

The central hypothesis of this work is that fMRI decoding models operating under the multi-voxel pattern analysis (MVPA) framework primarily rely on large-scale spatial distributions of neural activity to extract stimulus-related information. Although individual subjects exhibit substantial variability in fine-grained response patterns to identical stimuli, recent studies have revealed that

coarse-grained response topographies are highly similar across subjects, suggesting that individual idiosyncrasies manifest in more fine-grained response patterns[5, 6]. This macroscopic consistency provides a mechanistic basis for cross-subject zero-shot decoding.

Our proposed framework enables zero-shot decoding of brain activity for unseen subjects by leveraging a shared representation space learned from a support set of subjects. The method consists of two distinct phases: (1) Pretraining on a Support Set, where a shared decoding module and voxel embeddings are learned from multiple subjects with known ROI definitions; and (2) Zero-shot Inference, where the pretrained model is applied to a new, unseen subject by estimating voxel embeddings via spatial proximity (KNN) to the support set, without requiring subject-specific retraining.

3.2 Problem Formulation

For a subject s, the brain activity signal is a vector of voxel-wise beta values, $\mathbf{b}^{(s)} = [b_1^{(s)}, b_2^{(s)}, \ldots, b_{N_s}^{(s)}]^\top \in \mathbb{R}^{N_s}$, which quantify the activation strength of each voxel in response to a viewed image. The number of voxels, N_s, varies across subjects due to anatomical differences and the ROI selection process. Each voxel i is associated with its 3D MNI coordinate $\mathbf{p}_i^{(s)} \in \mathbb{R}^3$ and a region-of-interest (ROI) label $r(i) \in \mathcal{R}$. The objective is to reconstruct or retrieve the visual stimulus I for a novel, unseen subject s_{test} without any subject-specific training data.

To address the fundamental challenge of individual variability, we introduce a novel conceptual framework: **Voxel as Token**. We treat each fMRI voxel as an independent token, whose representation is a composition of two distinct factors: a functional embedding that encodes its role in the visual decoding process, and a modulation based on its instantaneous activation strength (beta value). Formally, the token for voxel i is defined as:

$$\mathbf{t}_i = \gamma(b_i) \odot \mathbf{f}_i + \beta(b_i), \tag{1}$$

where $\mathbf{f}_i \in \mathbb{R}^{d_f}$ is the functional embedding, and $\gamma(\cdot), \beta(\cdot)$ are affine parameters generated from the scalar beta value b_i. This formulation decouples the static, transferable "what" (the functional role $\mathbf{f}_i$) from the dynamic, stimulus-specific "how much" (the activation b_i). This decoupling is the cornerstone of our approach to enabling zero-shot transfer: if the functional embedding $\mathbf{f}_i$ can be estimated for a new subject's voxels, their tokens can be constructed and processed by a shared decoder. In this work, the main model and the MLP-based variants instantiate Eq. (1) with $\gamma(b_i) = b_i$ and $\beta(b_i) = \mathbf{0}$, i.e., $t_i = b_i \mathbf{f}_i$: the token is the functional embedding scaled by the voxel's activation. The VoxelFormer variant (Appendix C) instead learns $\gamma(\cdot)$ and $\beta(\cdot)$ as linear functions of the beta value, following the full modulation form. The general formulation in Eq. (1) also subsumes existing adapter-based models, as discussed in Sec. 3.6.

Geometric Normalization. Absolute MNI coordinates are not directly comparable across subjects due to anatomical variability, so we Z-score the coordinates within each ROI. For a subject s and an ROI r, let $\mathcal{V}_r^{(s)} = \{j \mid r(j) = r\}$ denote the set of all voxels within that ROI; the normalized coordinate of voxel i is:

$$\tilde{\mathbf{p}}_i^{(s)} = \frac{\mathbf{p}_i^{(s)} - \boldsymbol{\mu}_r^{(s)}}{\boldsymbol{\sigma}_r^{(s)} + \epsilon}, \qquad \boldsymbol{\mu}_r^{(s)} = \frac{1}{|\mathcal{V}_r^{(s)}|} \sum_{j \in \mathcal{V}_r^{(s)}} \mathbf{p}_j^{(s)}, \qquad \boldsymbol{\sigma}_r^{(s)} = \sqrt{\frac{1}{|\mathcal{V}_r^{(s)}|} \sum_{j \in \mathcal{V}_r^{(s)}} (\mathbf{p}_j^{(s)} - \boldsymbol{\mu}_r^{(s)})^2}, \tag{2}$$

where $\epsilon = 10^{-6}$ ensures numerical stability. Z-scoring centers the coordinate system at the ROI's centroid and scales it by the ROI's spatial extent, removing global brain size, position, and local ROI scale differences between subjects. The resulting $\tilde{\mathbf{p}}_i^{(s)}$ is a subject-invariant spatial descriptor: voxels that are geometrically analogous across subjects receive comparable descriptors, which is precisely the correspondence that the KNN-based functional embedding estimator (Sec. 3.6) uses to transfer functional knowledge from the support set to an unseen subject.

Support Set. To facilitate zero-shot generalization, we construct a **Support Set** $\mathcal{D}_{\text{support}}$ using data from a pool of M seen subjects, denoted as $\mathcal{S}_{\text{train}} = \{s_1, s_2, \ldots, s_M\}$. The support set serves as a reference database of voxel representations organized by anatomical regions. Specifically, for each ROI $k \in \mathcal{R}$, we define a subset $\mathcal{D}_{\text{support}}^{(k)}$ that aggregates the pretrained voxel embeddings and their normalized coordinates from all subjects in $\mathcal{S}_{\text{train}}$.

Let $\mathbf{f}_j^{(s)} \in \mathbb{R}^d$ represent the pretrained per-voxel embedding for the j-th voxel of subject $s \in \mathcal{S}_{\text{train}}$, obtained during the pretraining phase, paired with its normalized coordinate $\tilde{\mathbf{p}}_j^{(s)}$ from Eq. 2. The support set for ROI k is formally defined as the collection of these embedding-position pairs:

$$\mathcal{D}_{\text{support}}^{(k)} = \left\{ \left(\tilde{\mathbf{p}}_j^{(s)}, \mathbf{f}_j^{(s)} \right) \mid s \in \mathcal{S}_{\text{train}},\, r(j) = k \right\}. \tag{3}$$

The complete support set is the union of these ROI-specific subsets: $\mathcal{D}_{\text{support}} = \bigcup_{k \in \mathcal{R}} \mathcal{D}_{\text{support}}^{(k)}$. During inference for s_{test}, we utilize $\mathcal{D}_{\text{support}}$ to estimate the functional embeddings of the unseen subject's voxels via spatial nearest-neighbor retrieval, as detailed in Sec. 3.6.

3.3 Multi-scale ROI Organization and Fusion

The human visual system processes information hierarchically, with early areas (e.g., V1, V2) encoding simple features and higher areas (e.g., IT, LO) encoding complex semantics. To leverage this inherent structure, we use multi-scale ROI groups based on their known functional hierarchy(e.g., V1, V2, V3, hV4, early visual areas, mid-ventral stream, high-ventral stream, and NSDGeneral). And during training, we randomly mask some ROIs with a learnable [MASK] token with probabilty of 50% to fully leverage the inter-ROI redundancy. We hypothesize that fusing information across these multiple scales is crucial for robust and generalizable decoding. As shown in Fig.1, our framework incorporates a dedicated fusion stage where representations from each scale group are first aggregated internally and subsequently combined to form a comprehensive global visual representation.

To mitigate inter-subject variability, all voxel embeddings (whether estimated or learned) are L2-normalized onto the unit sphere before forming tokens with the beta responses. Within each ROI, the voxel tokens are then averaged and L2-normalized again to obtain the ROI embedding:

$$\mathbf{e}_r^{(s)} = \text{L2Norm}\left(\frac{1}{|\mathcal{V}_r^{(s)}|} \sum_{i:r(i)=r} \mathbf{t}_i \right) = \text{L2Norm}\left(\frac{1}{|\mathcal{V}_r^{(s)}|} \sum_{i:r(i)=r} b_i^{(s)} \mathbf{f}_i^{(s)} \right), \tag{4}$$

where the second equality uses the token instantiation $\mathbf{t}_i = b_i \mathbf{f}_i$ adopted in this work. The per-ROI embeddings are then concatenated into the global input of the shared decoder:

$$\mathbf{H}_{\text{global}} = \bigoplus_{r \in \mathcal{R}} \mathbf{e}_r^{(s)}, \tag{5}$$

where $\bigoplus$ denotes concatenation along the feature dimension. This two-level normalization constrains the model to prioritize the directional geometry of neural representations, decoupling functional topology from amplitude fluctuations and measurement noise in the beta estimates, and it makes cross-subject aggregation more consistent and interpretable.

3.4 Pretraining with KNN-Augmented data

Using the KNN estimator of Sec. 3.6, we compute estimated functional embeddings for every support subject during pretraining and update them at the end of each epoch as the learned embeddings evolve. At each training step, both the estimated embeddings and the learned (subject-specific adapter) embeddings are independently propagated through the shared network. This iterative refinement, combined with dual-path forward processing, encourages the alignment of subject-specific and shared neural representations. Consequently, the model learns to disentangle idiosyncratic response variability from common functional topology, thereby establishing a robust foundation for zero-shot generalization to unseen subjects and tasks.

3.5 Network Architecture and Training Objective

The shared decoding modules employ a residual MLP backbone analogous to the MindEye[25] architecture, configured with 8 layers, a hidden dimension of 3,200 (structured to accommodate eight 400-dimensional ROI representations), and a dropout rate of 0.2. Similar architectural specifications are applied to the diffusion prior, contrastive head, and low-level reconstruction head. And the same diffusion prior loss, contrastive loss, and Low-level reconstruction loss are used as MindEye [25]. The CLIP Model used in this work is the ViT-L/14 version. And the CLIP vision tokens predicted by the brain decoding model are used to guide a pretrained versatile diffusion model[26] under img2img setting as MindEye[25] does with the low-level blurry reconstruction as the initial state.

3.6 The Unified perspective and a Plug-and-Play Version for Training-free Zero-shot Decoding

Our formulation also provides a unifying perspective on existing adapter-based models, such as MindEye2 [1]. In these models, a subject-specific linear layer $W^{(s)} \in \mathbb{R}^{N_s \times D}$ projects the beta vector into a shared latent space. We reinterpret the i-th row of $W^{(s)}$ as the functional embedding $\mathbf{f}_i$ for voxel i, and the projection operation $W^{(s)\top}\mathbf{b}^{(s)}$ as a simple, unweighted sum of the modulated tokens, $\sum_i b_i^{(s)}\mathbf{f}_i$. Consequently, the zero-shot cross-subject problem is reduced to the estimation of these functional embeddings $\mathbf{f}_i$ for a novel subject.

Therefore, we provide a training-free, plug-and-play solution to enable zero-shot inference for any existing adapter-based model (e.g., MindEye2). The motivation is that retraining or fine-tuning a model for each new subject is often impractical in real-world scenarios. We aim to leverage the functional embeddings already learned by a model on a cohort of training subjects and transfer them to a new subject using only geometric correspondence, without any modification to the pre-trained model.

Functional Embedding Estimation via KNN For a novel subject s, we process each ROI r independently to construct its subject-specific adapter using the support sets defined in Sec. 3.2 and the normalized coordinates of Sec. 3.2:

- For every voxel i with $r(i) = r$, compute its coordinate normalized within ROI r: $\tilde{\mathbf{p}}_i^{(s,r)}$.
- Query $\mathcal{D}_{\text{support}}^{(r)}$ to retrieve the K nearest neighbors of $\tilde{\mathbf{p}}_i^{(s,r)}$ (Euclidean distance), obtaining functional embeddings $\{\mathbf{f}_{\text{nn}(k)}^{(r)}\}_{k=1}^{K}$.
- Estimate the ROI-specific functional embedding for voxel i: $\hat{\mathbf{f}}_i^{(r)} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{f}_{\text{nn}(k)}^{(r)}$.
- Stack all estimated embeddings within ROI r to form the adapter matrix

$$\mathbf{W}_r^{(s)} = \begin{bmatrix} (\hat{\mathbf{f}}_1^{(r)})^\top \\ (\hat{\mathbf{f}}_2^{(r)})^\top \\ \vdots \\ (\hat{\mathbf{f}}_{N_r}^{(r)})^\top \end{bmatrix} \in \mathbb{R}^{N_r \times d},$$

  where N_r is the number of voxels in ROI r and d is the embedding dimension.

During decoding, given the subject's voxel activity pattern $\mathbf{a}_r^{(s)} \in \mathbb{R}^{N_r}$ within ROI r, the adapter projects it into the shared latent space:

$$\mathbf{z}_r^{(s)} = (\mathbf{W}_r^{(s)})^\top \mathbf{a}_r^{(s)} \in \mathbb{R}^d.$$

In this plug-and-play setting, the forward computation follows the original model's convention without L2-normalization, since the pretrained decoder was not trained with it; VoxTok's main model instead applies the L2-normalization of Eq. 4 during its own training. For multi-scale fusion, the per-ROI representations are concatenated as in Eq. 5 to form the global input $\mathbf{H}_{\text{global}}$ of the pre-trained, fixed shared decoder. This method requires no gradient computation or parameter updates, making it highly efficient and easy to deploy.

4 Experiments

4.1 Data and Settings

NSD Our study leverages the Natural Scenes Dataset (NSD) [27], a large-scale fMRI benchmark containing approximately 73,000 image–response pairs collected from eight subjects as they viewed natural images. Following established protocols in prior works [11, 28, 1], we partition each subject's data into a training set of roughly 9,000 images and a held-out test set of 1,000 unseen images (The training set of the tested subject is not used for training in the zero-shot setting). Unless otherwise noted, decoding performance is reported as the average across four subjects (1, 2, 5, and 7) who

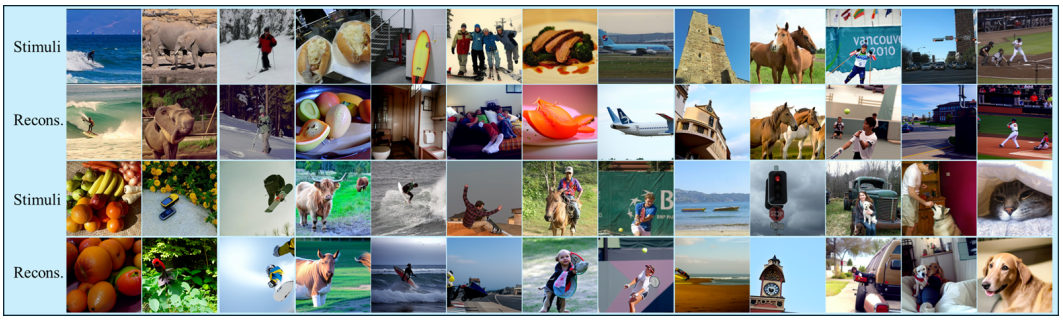


Figure 2: Zero-shot decoding Reconstruction samples from subject 1

Table 1: Quantitative results for image reconstruction and retrieval on the Natural Scenes Dataset (NSD) are reported, with performance averaged across participants. Baseline results are either directly cited from the original publications or reproduced using the authors' released model weights. Within each evaluation section, the best- and second best-performing methods are highlighted in red and blue, respectively.

Methods	Low- Level				High-Level				Retrieval	
	PixCorr↑	SSIM↑	Alex(2)↑	Alex(5)↑	Incep↑	CLIP↑	Eff↓	SwAV↓	Image↑	Brain↑
Fully fine-tuned										
Brain-diffuser[31]	.254	.356	94.2%	96.2%	87.2%	91.5%	.775	.423	21.1%	30.3%
BrainCLIP[32]	-	-	-	-	86.7%	94.8%	-	-	40.7%	-
DREAM[33]	.288	.338	93.9%	96.7%	93.7%	94.1%	.645	.418	-	-
MindBridge[34]	.151	.263	87.7%	95.5%	92.4%	94.7%	.712	.413	-	-
PSYCHOMETRY[35]	.297	.340	96.4%	98.6%	95.8%	96.8%	.628	.345	-	-
UMBRAE [36]	.283	.341	95.5%	97.0%	91.7%	93.5%	.700	.393		
MindEye[25]	.309	.323	94.7%	97.8%	93.8%	94.1%	.645	.367	93.6%	90.1%
NeuroPictor[12]	.229	.375	96.5%	98.4%	94.5%	93.3%	.639	.350	-	-
STTM[16]	.333	.334	95.7%	98.5%	95.8%	95.7%	.611	.338	92.8%	94.9%
MindEye2[1]	.285	.389	96.3%	98.7%	95.4%	93.0%	.619	.343	91.7%	97.4%
MindTuner[18]	.322	.421	95.8%	98.8%	95.6%	93.8%	.612	.340	98.9%	98.3%
NeuroSwift[37]	.335	.437	96.5%	98.1%	95.4%	97.1%	.615	.354	-	-
Few-shot										
MindEye2 [1] (1 hour)	.195	.419	84.2%	90.6%	81.2%	79.2%	.810	.468	79.0%	57.4%
MindTuner [18] (1 hour)	.224	.420	87.8%	93.6%	84.8%	83.5%	.780	.440	83.1%	76.0%
Zero-shot										
NeuroPictor*[12] [ECCV'24]	.057	.297	71.4%	74.7%	62.5%	66.0%	.939	.607	-	-
Kong et al.[2] [CLR'25]	-	-	-	-	-	-	-	-	1.0%	-
MindEye2[1] [ICML'24]	-	-	-	-	-	-	-	-	4.1%	-
BrainCoDec-200[3] [Arxiv'26]	-	-	-	-	-	-	-	-	22.7%	
ZEBRA-baseline[4] [NeurIPS'25]	.074	.316	70.8%	74.0%	63.5%	62.5%	.920	.602	-	-
ZEBRA[4] [NeurIPS'25]	.131	.375	74.6%	81.2%	72.2%	71.5%	.837	.506	-	-
VoxTok(MLP 3D-to-Function Embedder)	-	-	-	-	-	-	-	-	12.0 ±3.5%	11.8±3.6%
VoxTok(VoxelFormer)	.115	.325	67.5%	71.7%	64.3%	65.4%	.913	.578	23.2±6.3%	20.9±6.0%
VoxTok(MindEye2,training-free)	.113±0.022	.311±0.006	68.4±2.3 %	70.2±3.4%	60.4±4.1%	60.4±3.1%	.933±0.020	.591±0.014	17.5±2.3%	15.9±2.5%
VoxTok(Ours)	.164 ±0.016	.302 ±0.008	77.9 ±2.2%	80.4 ±3.0%	69.2 ±3.7%	70.4 ±3.8%	.871 ±0.020	.545 ±0.012	40.4 ±8.1%	37.2 ±7.1%
VoxTok(VoxelFormer,low-level)	.163	.464	57.8%	55.9%	51.9%	51.9%	1.01	.663	-	-
VoxTok(Ours,low-level)	.206 ±0.021	.471 ±0.003	59.8 ±2.3%	56.8 ±1.3%	51.7 ±0.6%	52.2 ±0.6%	1.01 ±0.001	.659 ±0.001	-	-

completed all 40 scanning sessions. We used eight vision-related ROI masks provided by NSD: V1, V2, V3, hV4, Early (early visual areas), mid-ventral stream, high-ventral stream, and NSDGeneral. Additionally, we extract the 3D spatial coordinates of each voxel for each subject in the same MNI template to incorporate geometric information into our modeling.

Pretraining settings for our main models Each VoxTok model is pretrained on data from seven subjects while holding out the test subject for zero-shot evaluation. The training process runs for 100,000 steps with a total batch size of 490 on a single NVIDIA H800 GPU (80 GB), taking approximately 30 hours to complete. We adopt the AdamW optimizer [29] with a weight decay of 0.01 and employ a OneCycle learning rate schedule [30] that peaks at 1×10^{-4}. Evaluation metrics for reconstruction and retrieval follow the protocol of MindEye2.

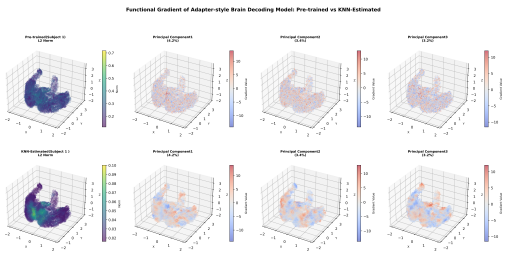
Other variants for VoxTok To get a comprehensive understanding about VoxTok, we tested with a series of variants of VoxTok, including "VoxTok(MLP 3D-to-function Embedder)", "VoxTok(VoxelFormer)", "VoxTok(MindEye2,training-free)", Among them, "VoxTok(MLP 3D-to-function embedder)" and "VoxTok(VoxelFormer)" use a trainable MLP network to learn the estimation of voxel functional embeddings based on ROI labels and the 3D positions. Only VoxelFormer is with a transformer-based backbone architecture while the others are MLP-based. More details and training settings for these VoxTok variants can be found in the Appendix C.

4.2 Zero-shot Decoding Results

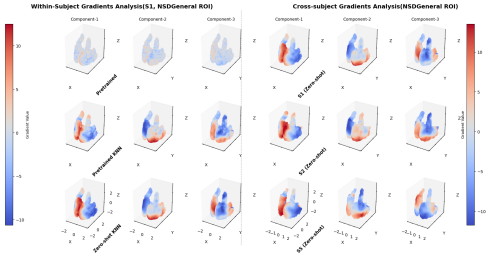
Fig. 2 displays some zero-shot reconstruction examples obtained by our VoxTok model. Even though not perfectly matched, these reconstructions still reflect some attributes highly-correlated with the original stimuli.

In Tab. 1, we quantitatively compare our VoxTok framework with recent works. As we can see, our full method achieves significantly better retrieval performance than all comparable zero-shot baselines, including BrainCoDec-200 [3] and Kong et al. [2]. On reconstruction, ZEBRA [4] remains stronger on most low-/high-level metrics; we note that ZEBRA relies on large-scale surface-based MAE pretraining and a complicated feature-disentanglement objective with seven loss terms, whereas VoxTok operates directly in native 3D voxel space with a much simpler pipeline. ZEBRA reports reconstruction only and does not release its weights, so a controlled retrieval comparison is not possible. The results also demonstrate that all VoxTok variants achieve significantly better accuracy than chance level, even under the training-free setting with the MindEye2 architecture. These results show that VoxTok can be a powerful but cheap method for cross-subject zero-shot decoding.

Cross-dataset adaption on GOD. Beyond NSD, we evaluate on the GOD dataset [7] with leave-one-subject-out fine-tuning from the NSD-pretrained model (the test subject is always excluded; KNN augmentation is applied during fine-tuning). The fine-tuned model reaches 72.0% image retrieval and 67.2% brain retrieval accuracy (50-way retrieval; chance 2%), approaching supervised single-subject GOD baselines on several reconstruction metrics (full details in Appendix Tab. 3).



(a) MindEye2 adapter-based model: L_2 norms and functional gradients of voxel embeddings. First row: learned embeddings; second row: KNN regressor estimates. The supervised decoder relies on large-scale co-activation patterns that are partly subject-specific, while the KNN estimates recover a subject-invariant component in which early visual areas and inferior temporal gyrus show consistent cross-subject importance.



(b) VoxTok model (full method): voxel functional gradients representing spatial distribution of embeddings' projection on principal components (See, Appendix A). VoxTok's KNN estimator leads to clear regional function clustering, and the KNN-estimated (zero-shot) embeddings closely match the pretraining-time layout, indicating a subject-invariant component.

Figure 3: Comparative visualization of functional gradient analyses for adapter-based and our VoxTok brain decoding models. Cross-subject comparsion for MindEye2 is provided in Appendix Fig.8

4.3 More Analysis

What transfers across subjects? Tab. 2 addresses the central question of *what* should be transferred across subjects. Tab. 2(a) compares transferring the learned functional embeddings (KNN) against transferring raw voxel responses via 3D spatial interpolation. A test-only interpolation also incurs a train-test mismatch, so we additionally train a decoder from scratch on interpolated responses; baseline that is trained with 3D interpolation of betas and with ROI Fusion strategy reaches only 12.21%, $\sim$28 points below VoxTok's KNN-based transfer. The gap tracks the transferred signal rather than the decoder family: within the identical VoxTok architecture, replacing the KNN-estimated embeddings with interpolated betas drops retrieval substantially, while providing the MindEye2 decoder with KNN-estimated embeddings raises it by a comparable margin. Tab. 2(b) further shows that spatially coarse alternatives—the ROI-level mean embedding or K randomly sampled voxels—collapse to chance (0.48–0.71%), confirming that fine-grained, spatially conditioned neighbor selection, not coarse ROI identity, drives the gain.

Table 2: Ablation studies on NSD (300-way retrieval; chance $\approx 0.33\%$; averaged over 4 subjects). (a) compares what is transferred across subjects—3D spatial interpolated voxel betas vs. KNN-estimated functional embeddings—under the same decoder family. In the training-free interpolation variant, the already-trained full model is applied to betas resampled onto a reference subject's input space (subjects 2, 5, 7 into subject 1, and subject 1 into subject 2). The trained 3D-interpolation baselines resample the raw betas of the 7 training subjects into the target subject's native input space; (b) replaces the KNN estimator by spatially coarse alternatives under the VoxTok full setup. (c) isolates L2-normalization and KNN-augmentation.

Method	Image Retrieval ↑	Brain Retrieval ↑
(a) What is transferred across subjects		
VoxTok (Ours, full method, KNN-estimated embeddings)	**40.4 ± 8.1**	**37.2 ± 7.1**
VoxTok (Ours, 3D interpolation of betas, training-free)	25.86 ± 10.74	23.60 ± 10.76
Train with 3D interpolation of betas (ROI Fusion, MindEye-style)	12.21 ± 2.33	11.03 ± 1.60
Train with 3D interpolation of betas (NSD-General, MindEye-style)	7.12 ± 1.46	6.99 ± 1.22
MindEye2 decoder (KNN-estimated embeddings, training-free)	17.5 ± 2.3	15.9 ± 2.5
MindEye2 decoder (3D interpolation of betas, training-free)	5.78 ± 2.67	5.69 ± 2.57
(b) Spatial-conditioning controls (VoxTok full setup)		
KNN estimator (ours)	**40.4 ± 8.1**	**37.2 ± 7.1**
ROI-level mean embedding	0.48 ± 0.03	0.52 ± 0.02
Random K voxels (K=450)	0.51 ± 0.01	0.71 ± 0.02
Random K voxels (K=200)	0.59 ± 0.06	0.68 ± 0.08
(c) L2-normalization And KNN augmentation		
VoxTok(Ours, w/o KNN-augmented data, w/o L2 Norm)	26.2 ± 5.6%	21.5 ± 4.5%
VoxTok(Ours, w/o KNN-augmented data)	28.3 ± 5.1%	23.6 ± 4.2%
VoxTok(Ours, w/o L2 norm)	38.2 ± 8.32	35.7 ± 7.36
VoxTok(Ours, full method)	**40.4 ± 8.1**	**37.2 ± 7.1**

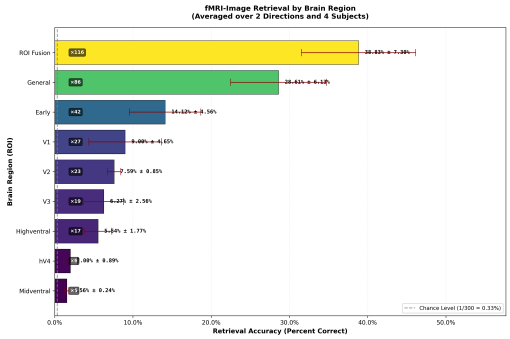
L2-Normalization and KNN-augmented Pretraining Tab. 2(c) isolates the two training components of the full model. L2-normalization removes cross-subject magnitude differences so that the embedding encodes functional direction rather than per-voxel SNR; KNN-augmented pretraining exposes the decoder to the estimated embeddings encountered at zero-shot inference, which is worth 12.1 points in image retrieval (28.3% → 40.4%). Both components adapt the decoder to the estimation-based inputs of unseen subjects.

How Do Adapter-based and VoxTok Models Use the Embeddings? Fig. 3a visualizes the L2 norm and functional gradients of the learned voxel embeddings of a MindEye2-style adapter model. High-norm voxels and functionally similar voxels are widely distributed across disparate anatomical regions, with no pronounced spatially contiguous clusters; under fully supervised training this large-scale co-activation pattern is partly *subject-specific*, which is why the trained adapter does not transfer to new subjects. Here, by *distributed co-occurring patterns*, we mean the broad-extent aggregation of signal and the redundancy it provides under measurement noise: even when a region dominates functionally, other regions carry complementary or redundant stimulus information, so importance is regionally concentrated while sufficiency is not (e.g., early visual voxels receive large embedding norms yet early areas alone cannot reproduce MindEye2's accuracy). In contrast, Fig. 3b shows that KNN-estimated voxel embeddings can recover the regional clustering, and its KNN-estimated (zero-shot) embeddings closely match the pretraining-time layout while showing consistent cross-subject similarity. Together, these results indicate that VoxTok's KNN step recovers the *subject-invariant* component of the co-activation pattern, the part that transfers across subjects, whereas the fully supervised MindEye-style decoder additionally relies on subject-specific components, which explains why it does not generalize directly.

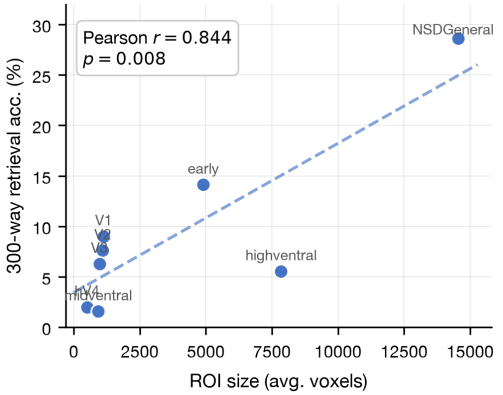
Multi-scale ROI Fusion and the ROI-Size Trend: Fig. 4a evaluates the contribution of individual ROIs by measuring zero-shot performance when only a single ROI remains active (all other ROIs are masked with a learnable [MASK] embedding). No single ROI matches the accuracy of multi-scale fusion—different ROIs capture complementary information—and the fusion gain does not depend on NSDGeneral: excluding it, the remaining ROIs still achieve 33.82%, substantially above the best single ROI (see Appendix Tab. 4). Fig. 4b further analyzes the relationship between ROI volume and zero-shot accuracy across the 8 ROIs, revealing a positive Pearson correlation ($r = 0.844$, $p = 0.008$).

This trend reproduces on the independent GOD dataset ($r = 0.710$; see Appendix Tab. 5) and in a pooled, size-normalized analysis across both datasets ($r = 0.79$, $p = 0.0004$; see Appendix D.2). Both observations are consistent with decoding relying on distributed response patterns(See Appendix D for more analysis).

The Effect of the Number of Neighbors for the KNN Estimator: In Fig. 5, we test VoxTok's zero-shot retrieval accuracy under different K values of the KNN estimator. The results suggest that the KNN estimator benefits from large neighborhood sizes without significant over-smoothing, reinforcing the importance of distributed, large-scale neural representations in zero-shot generalization. The final K used in this work is 450, which covers 0.44%–6.9% of each ROI's candidate pool (12.9% for the smallest ROI, hV4; see Appendix Tab. 6).



(a) Retrieval performance by Brain Region. The multiply factors are based on the chance-level.



(b) Correlation between ROI size and zero-shot accuracy

Figure 4: Analysis of ROI-based decoding performance. (a) Integrating distributed ROIs improves accuracy; (b) Strong positive correlation between ROI size and retrieval performance.

5 Conclusion and Limitation

We introduced *Voxel as Token*, a new perspective for zero-shot cross-subject visual decoding that treats each voxel as an independent token comprising a subject-invariant functional embedding and stimulus-specific activation. This perspective naturally resolves variable input dimensionality and enables direct modeling of transferability in native volumetric space. We evaluate this framework with a series of VoxTok variants. Experiments demonstrate that volume-based fMRI can achieve meaningful zero-shot generalization—approaching early supervised methods. Crucially, our analysis reveals a positive trend between ROI size and decoding accuracy that reproduces across datasets, suggesting that distributed co-activation patterns drive cross-subject transferability. By operating directly on volumetric data with minimal architectural complexity, our work establishes a simpler, more biologically complete foundation for generalizable brain decoding and offers a new lens for understanding the spatial-semantic structure of neural representations. While our analysis focuses on NSD, we additionally validate cross-dataset adaptation on GOD, where leave-one-out fine-tuning approaches supervised baselines (Appendix D.1). Our image reconstruction quality has not yet fully surpassed surface-based methods such as ZEBRA, which rely on large-scale surface MAE pretraining; in future work, we will further improve reconstruction fidelity and test the method on more datasets.

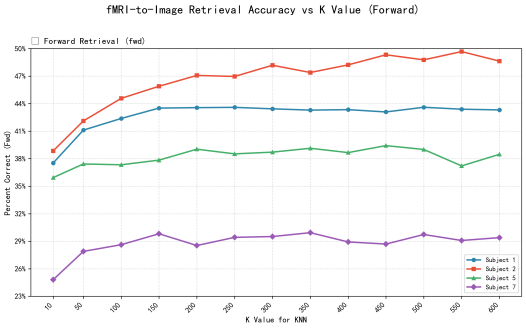


Figure 5: Retrieval accuracy trend of the KNN estimator as the number of neighbors varies.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. T2600237, the Early Career Scheme (ECS) of the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China, under Project No. 26601126, and the research funding under the HKUST-DXM AI for Finance Joint Laboratory (DXM25EG01), and the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123404).

References

[1] Paul S. Scotti, Mihir Tripathy, Cesar Kadir Torrico Villanueva, Reese Kneeland, Tong Chen, Ashutosh Narang, Charan Santhirasegaran, Jonathan Xu, Thomas Naselaris, Kenneth A. Norman, and Tanishq Mathew Abraham. Mindeye2: Shared-subject models enable fmri-to-image with 1 hour of data, 2024.

[2] Xiangtao Kong, Kexin Huang, Ping Li, and Lei Zhang. Toward generalizing visual brain decoding to unseen subjects. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 25484–25497, 2025.

[3] Mu Nan, Muquan Yu, Weijian Mai, Jacob S Prince, Hossein Adeli, Rui Zhang, Jiahang Cao, Benjamin Becker, John A Pyles, Margaret M Henderson, et al. Meta-learning in-context enables training-free cross subject brain decoding. *arXiv preprint arXiv:2604.08537*, 2026.

[4] Haonan Wang, Jingyu Lu, Hongrui Li, and Xiaomeng Li. ZEBRA: Towards zero-shot cross-subject generalization for universal brain visual decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[5] Po-Hsuan Cameron Chen, Janice Chen, Yaara Yeshurun, Uri Hasson, James Haxby, and Peter J Ramadge. A reduced-dimension fmri shared response model. *Advances in neural information processing systems*, 28, 2015.

[6] Umut Güçlü and Marcel AJ van Gerven. Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, 145:329–336, 2017.

[7] Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, 8(1):1–15, 2017.

[8] Bertrand Thirion, Edouard Duchesnay, Edward Hubbard, Jessica Dubois, Jean-Baptiste Poline, Denis Lebihan, and Stanislas Dehaene. Inverse retinotopy: inferring the visual content of images from brain activation patterns. *Neuroimage*, 33(4):1104–1116, 2006.

[9] David D Cox and Robert L Savoy. Functional magnetic resonance imaging (fmri)"brain reading": detecting and classifying distributed patterns of fmri activity in human visual cortex. *Neuroimage*, 19(2):261–270, 2003.

[10] Milad Mozafari, Leila Reddy, and Rufin VanRullen. Reconstructing natural scenes from fmri patterns using bigbigan. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[11] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. *bioRxiv*, pages 2022–11, 2022.

[12] Jingyang Huo, Yikai Wang, Yun Wang, Xuelin Qian, Chong Li, Yanwei Fu, and Jianfeng Feng. Neuropictor: Refining fmri-to-image reconstruction via multi-individual pretraining and multi-level modulation. In *European Conference on Computer Vision*, pages 56–73. Springer, 2025.

[13] Guohua Shen, Tomoyasu Horikawa, Kei Majima, and Yukiyasu Kamitani. Deep image reconstruction from human brain activity. *PLoS computational biology*, 15(1):e1006633, 2019.

[14] Roman Beliy, Guy Gaziv, Assaf Hoogi, Francesca Strappini, Tal Golan, and Michal Irani. From voxels to pixels and back: Self-supervision in natural-image reconstruction from fmri. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Guohua Shen, Kshitij Dwivedi, Kei Majima, Tomoyasu Horikawa, and Yukiyasu Kamitani. End-to-end deep image reconstruction from human brain activity. *Frontiers in Computational Neuroscience*, page 21, 2019.

[16] Yulong Liu, Yongqiang Ma, Guibo Zhu, Haodong Jing, and Nanning Zheng. See through their minds: Learning transferable brain decoding models from cross-subject fmri. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5730–5738, 2025.

[17] Ziqi Ren, Jie Li, Xuetong Xue, Xin Li, Fan Yang, Zhicheng Jiao, and Xinbo Gao. Reconstructing seen image from brain activity by visually-guided cognitive representation and adversarial learning. *NeuroImage*, 228:117602, 2021.

[18] Zixuan Gong, Qi Zhang, Guangyin Bao, Lei Zhu, Rongtao Xu, Ke Liu, Liang Hu, and Duoqian Miao. Mindtuner: Cross-subject visual decoding with visual fingerprint and semantic correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14247–14255, 2025.

[19] Xuelin Qian, Yun Wang, Jingyang Huo, Jianfeng Feng, and Yanwei Fu. fmri-pte: A large-scale fmri pretrained transformer encoder for multi-subject brain activity decoding, 2023.

[20] Zijiao Chen, Jiaxin Qing, Tiange Xiang, Wan Lin Yue, and Juan Helen Zhou. Seeing beyond the brain: Conditional diffusion model with sparse masked modeling for vision decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22710–22720, 2023.

[21] Yulong Liu, Hua Xu, Yiyang Cai, Chunyang Jiang, Sirui Han, and Yike Guo. Modeling the brain's grammar: Roi-guided fmri pretraining for transferable and interpretable vision decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6900–6909, 2026.

[22] Thomas Bazeille, Elizabeth DuPre, Hugo Richard, Jean-Baptiste Poline, and Bertrand Thirion. An empirical evaluation of functional alignment using inter-subject decoding. *NeuroImage*, 245:118683, 2021.

[23] Matteo Ferrante, Tommaso Boccato, Furkan Ozcelik, Rufin VanRullen, and Nicola Toschi. Through their eyes: multi-subject brain decoding with simple alignment techniques. *Imaging Neuroscience*, 2:imag–2, 2024.

[24] Roman Beliy, Navve Wasserman, Amit Zalcher, and Michal Irani. The wisdom of a crowd of brains: A universal brain encoder. *arXiv preprint arXiv:2406.12179*, 2024.

[25] Paul S. Scotti, Atmadeep Banerjee, Jimmie Goode, Stepan Shabalin, Alex Nguyen, Ethan Cohen, Aidan J. Dempster, Nathalie Verlinde, Elad Yundler, David Weisberg, Kenneth A. Norman, and Tanishq Mathew Abraham. Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors, 2023.

[26] Xingqian Xu, Zhangyang Wang, Gong Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7754–7765, 2023.

[27] Emily J Allen, Ghislain St-Yves, Yihan Wu, Jesse L Breedlove, Jacob S Prince, Logan T Dowdle, Matthias Nau, Brad Caron, Franco Pestilli, Ian Charest, et al. A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence. *Nature neuroscience*, 25(1):116–126, 2022.

[28] Furkan Ozcelik, Bhavin Choksi, Milad Mozafari, Leila Reddy, and Rufin VanRullen. Reconstruction of perceived images from fmri patterns and semantic brain exploration using instance-conditioned gans. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[30] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[31] Furkan Ozcelik and Rufin VanRullen. Natural scene reconstruction from fmri signals using generative latent diffusion, 2023.

[32] Yulong Liu, Yongqiang Ma, Wei Zhou, Guibo Zhu, and Nanning Zheng. Brainclip: Bridging brain and visual-linguistic representation via clip for generic natural visual stimulus decoding, 2023.

[33] Weihao Xia, Raoul de Charette, Cengiz Oztireli, and Jing-Hao Xue. Dream: Visual decoding from reversing human visual system. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8226–8235, 2024.

[34] Shizun Wang, Songhua Liu, Zhenxiong Tan, and Xinchao Wang. Mindbridge: A cross-subject brain decoding framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11333–11342, 2024.

[35] Ruijie Quan, Wenguan Wang, Zhibo Tian, Fan Ma, and Yi Yang. Psychometry: An omnifit model for image reconstruction from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 233–243, 2024.

[36] Weihao Xia, Raoul de Charette, Cengiz Oztireli, and Jing-Hao Xue. Umbrae: Unified multimodal brain decoding. In *European Conference on Computer Vision*, pages 242–259. Springer, 2024.

[37] Shiyi Zhang, Dong Liang, and Yihang Zhou. Neuroswift: A lightweight cross-subject framework for fmri visual reconstruction of complex scenes. *arXiv preprint arXiv:2510.02266*, 2025.

[38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[39] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[40] Bohan Zeng, Shanglin Li, Xuhui Liu, Sicheng Gao, Xiaolong Jiang, Xu Tang, Yao Hu, Jianzhuang Liu, and Baochang Zhang. Controllable mind visual diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6935–6943, 2024.

A An Explanation about Functional Gradients Used in This Work

To characterize the low-dimensional organization of voxel-wise functional representations, we performed a PCA-based functional gradient analysis on the learned embedding vectors. Specifically, Principal Component Analysis (PCA) was applied to the high-dimensional embedding matrix to extract the dominant axes of variance across voxels. The resulting principal component scores were interpreted as functional gradient values, representing each voxel's continuous coordinate along the primary organizational axes encoded in the embedding space. These gradient scores were subsequently mapped back to their native 3D anatomical coordinates for spatial visualization and topological interpretation. While conventional functional gradient analyses typically rely on nonlinear diffusion mapping applied to connectivity-derived similarity matrices, our approach serves as a linear approximation that efficiently captures the major variance directions within the learned representation space. The leading gradient components reflect the most salient continuous functional hierarchies, enabling the characterization of brain organization as smooth, large-scale transitions rather than discrete regional parcellations.

B Implementation and Evaluation Details

B.1 Model Pipeline and KNN-Augmented Pretraining

VoxTok's main model (VoxTok (Ours) in Tab. 1) follows the MindEye-style architecture [25, 1]. Each training subject s owns a subject-specific adapter matrix $\mathbf{W}^{(s)} \in \mathbb{R}^{N_s \times d}$ whose i-th row is the functional embedding $\mathbf{f}_i^{(s)}$ of voxel i; Sec. 3.6 reinterprets this matrix as the collection of voxel tokens. Per ROI, the voxel tokens are averaged and L2-normalized to form the ROI embedding $\mathbf{e}_r^{(s)}$ (Eq. 4), and the per-ROI embeddings are concatenated into $\mathbf{H}_{\text{global}}$ (Eq. 5) before being fed into the shared residual-MLP decoder (8 layers, hidden size 3,200, dropout 0.2), followed by a diffusion prior, a contrastive head, and a low-level reconstruction head trained with the same losses as MindEye [25]. During training, ROIs are randomly replaced by a learnable [MASK] embedding with probability 50%. Algorithm 1 summarizes the KNN-augmented pretraining procedure.

Algorithm 1 KNN-Augmented Pretraining of the VoxTok Main Model

Require: Support subjects $\mathcal{S}_{\text{train}}$ with fMRI–image pairs; ROI definitions; neighbor count K

Ensure: Shared decoder D; learned per-voxel functional embeddings $\{\mathbf{f}_j^{(s)}\}$

Initialize subject-specific adapter matrices $\{\mathbf{W}^{(s)}\}$ (rows are $\mathbf{f}_j^{(s)}$) and the shared decoder D
Compute the normalized coordinates $\tilde{\mathbf{p}}_j^{(s)}$ (Sec. 3.2) for all support voxels once
for each training epoch **do**
  Build the ROI-wise support sets $\mathcal{D}_{\text{support}}^{(k)}$ from the current learned embeddings
  For every support subject, compute KNN-estimated embeddings $\hat{\mathbf{f}}_j^{(s)}$ (Sec. 3.6) from the pooled embeddings of the other support subjects
  for each training step **do**
    Path A: construct tokens from the learned embeddings (Eq. 1) and forward through D, with voxel and ROI embeddings L2-normalized as in Eq. 4
    Path B: construct tokens from the KNN-estimated embeddings and forward through the same D, with the same L2-normalization
    Update $\{\mathbf{W}^{(s)}\}$ and D with the MindEye losses on both paths; randomly mask ROIs with a learnable [MASK] embedding (probability 50%)
  end for
end for

B.2 Evaluation Protocol

All metrics follow the MindEye protocol [25, 1]. *Image retrieval* (forward, fMRI-to-image) ranks the fMRI-evoked embedding against the CLIP embeddings of all candidate images in the test pool (300-way on NSD, chance $\approx 0.33\%$; 50-way on GOD, chance 2%). *Brain retrieval* (backward, image-to-fMRI) ranks the ground-truth image's fMRI embedding among the fMRI embeddings of the same test pool. Reconstruction is evaluated with PixCorr, SSIM, AlexNet(2)/(5) similarity, InceptionV3 similarity, CLIP similarity, and the learned perceptual metrics EffNet-B and SwAV; low-level reconstruction outputs are evaluated with the same metric set.

B.3 Details of the Ablation Experiments

3D spatial interpolation of betas (Tab. 2). The voxel responses are interpolated in 3D space onto the target subject's voxel locations using the same ROI-constrained KNN and normalized coordinates as our estimator (no fixed grid, no padding, no 3D CNN). In the test-only setting, the other subjects' betas are interpolated into the target subject's input space and fed to the already-trained decoder. In the train+test setting, the target subject is held out entirely: the betas of the remaining 7 subjects are resampled into the target subject's native input space and a residual-MLP decoder is trained from scratch on these interpolated responses; at test time the target subject's own betas are used in the same input space, so interpolation is applied consistently during training and testing.

Coarse controls (Tab. 2). *ROI-level mean embedding*: for each voxel, use the mean of all support embeddings of its ROI as the estimated embedding (a spatially constant estimate per ROI). *Random-*

K: average K voxels uniformly sampled at random from the ROI's candidate pool instead of the nearest neighbors; we report $K = 450$ and $K = 200$.

Progressive noise (Tab. 8). Additive noise is injected into the voxel beta responses (not the embeddings) with a linearly increasing schedule (max beta $= 1.0$, $T = 1000$ steps), and 1000-way retrieval (chance 0.1%) is measured at intermediate noise levels.

C VoxTok Variants

C.1 VoxelFormer — An End-to-End Architecture

With the "Voxel as Token" paradigm, we propose an end-to-end trainable model named VoxelFormer.

Relative 3D Positional Encoding of Voxels. To incorporate the geometric structure of voxels within each region of interest (ROI), we encode their *relative* 3D coordinates using high-frequency Fourier features, following the positional encoding scheme introduced in Neural Radiance Fields [38]. Let $\tilde{\mathbf{p}}_i^{(s)} = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i) \in \mathbb{R}^3$ denote the Z-score normalized coordinate of the i-th voxel, as defined in Section 3.2. This normalized coordinate is centered at the ROI's centroid and scaled by its spatial extent, making it invariant to global brain alignment across subjects. We map $\tilde{\mathbf{p}}_i^{(s)}$ to a $6L$-dimensional embedding via:

$$\text{FPE}(\tilde{\mathbf{p}}_i^{(s)}) = \big[\sin(2^0\pi\tilde{\mathbf{p}}_i^{(s)}), \cos(2^0\pi\tilde{\mathbf{p}}_i^{(s)}), \ldots, \sin(2^{L-1}\pi\tilde{\mathbf{p}}_i^{(s)}), \cos(2^{L-1}\pi\tilde{\mathbf{p}}_i^{(s)})\big]^\top, \quad (6)$$

where L is the number of frequency bands (set to $L = 10$ in our experiments). This encoding explicitly preserves fine-grained spatial relationships while being invariant to inter-subject anatomical misalignment. The resulting positional embeddings are concatenated with ROI-specific embeddings to form the input for the functional embedding generator.

Functional Embedding Generator. For each voxel i, its functional embedding $\mathbf{f}_i$ is generated by a multi-layer perceptron (MLP) that takes as input the concatenation of its ROI's learnable embedding $\mathbf{e}_{r(i)}$ and the high-frequency positional encoding $\text{FPE}(\tilde{\mathbf{p}}_i^{(s)})$. This allows the model to learn complex, non-linear mappings from a voxel's relative position and ROI context to its functional role in visual decoding.

Beta-aware Response Modulation. The voxel's beta value b_i is used to generate affine parameters γ_i and β_i via simple linear layers, which then modulate the functional embedding to produce the final token $\mathbf{t}_i = \gamma_i \odot \mathbf{f}_i + \beta_i$.

ROI Perceiver for Token Compression. Since an ROI can contain thousands of voxels, we employ a Perceiver-style module [39] to compress them. A set of M learnable latent queries attends to all voxel tokens within an ROI, producing a fixed-length, compressed representation $\mathbf{Z}_r$ that captures the ROI's essential information.

Multi-scale ROI Feature Fusion and Backbone. Following our general framework, the compressed representations $\{\mathbf{Z}_r\}$ are then concatenated to form the global input for the backbone. The global input is processed by a Transformer backbone with learnable query tokens that aggregate visual semantics.

Training objectives. The model is trained with a dual objective: a contrastive loss and an L2 loss to align with CLIP image features for high-level semantics, and an L1 loss to reconstruct VAE latents for low-level image details.

C.2 VoxTok(MLP 3D-to-Function Embedder)

We replace the KNN estimator in the main model with the same MLP voxel functional embedding estimator shown in the Fig. 6.

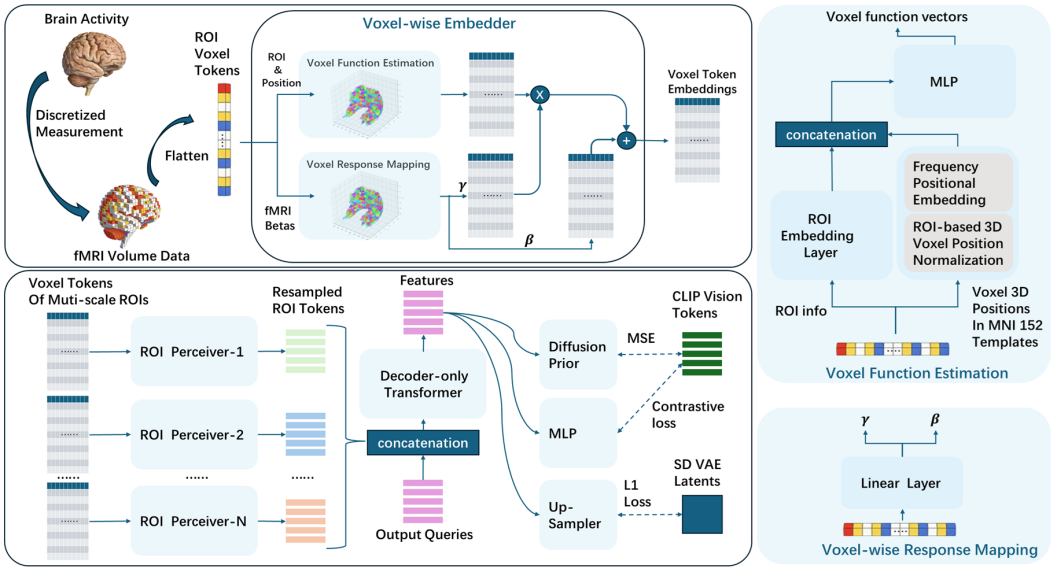


Figure 6: The architecture of VoxelFormer and the MLP-based voxel embedder used in VoxelFormer and VoxTok(MLP 3D-to-Function Embedder)

C.3 VoxTok(MindEye2, training-free)

We pretrain a decoding model in MindEye2's style with subject-specific adapters with a single ROI (NSDGeneral).

C.4 VoxTok(Ours, w/o KNN-augmented data, w/o L2 Norm)

We pretrain a decoding model in MindEye2's style with subject-specific adapters with multiple ROI(V1, V2, V3, hV4, Early (early visual areas), mid-ventral stream, high-ventral stream, and NSDGeneral). During training ROIs' embeddings are randomly replaced with a learnable [MASK] embedding with probability of 50%. There is no L2-normalization for voxel and aggregated ROI embeddings, and we do not use the KNN-estimated voxel embeddings.

C.5 VoxTok(Ours,w/o KNN-augmented data)

We pretrain a decoding model in MindEye2's style with subject-specific adapters with multiple ROI(V1, V2, V3, hV4, Early (early visual areas), mid-ventral stream, high-ventral stream, and NSDGeneral). During training ROIs' embeddings are randomly replaced with a learnable [MASK] embedding with probability of 50%. And L2-normalization is applied as the main model. The only difference between this variant and the main model is that the main model uses the KNN-estimated voxel embeddings.

C.6 Pretraining settings for VoxTok Variants

Pretraining of VoxelFormer VoxelFormer pretrained on data from seven subjects while holding out one subject for cross-subject evaluation. The training process runs for 100,000 optimization steps with a total batch size of 70 on a single NVIDIA H800 GPU (80 GB), taking approximately 60 hours to complete. We adopt the AdamW optimizer [29] with a weight decay of 0.01 and employ a OneCycle learning rate schedule [30] that peaks at 1×10^{-4}.

Pretraining of other MLP-based VoxTok Variants For these models we use the same settings as the main models.

D Additional Analyses

D.1 Cross-dataset Results on GOD

GOD (Generic Object Decoding) [7] records fMRI from 5 subjects viewing 1,250 ImageNet images (1,200 training and 50 test samples per subject). We use the publicly released preprocessed ROIs—V1–V4, early visual areas, high-level visual areas (high ventral stream), and the general visual cortex—together with the released 3D voxel coordinates; retrieval is 50-way over the 50 test images (chance 2%). Table 3 reports the full GOD results. The supervised baselines (IC-GAN [28], MinD-Vis [20], CMVDM [40]) are single-subject GOD methods included as a reference upper bound; they are trained on the target subject and report reconstruction only (no retrieval). The NSD-pretrained model is fine-tuned on the GOD training subjects with the same objective, optimizer, and learning-rate schedule as NSD pretraining; the test subject is always excluded, and KNN augmentation is applied during fine-tuning (each iteration uses both KNN-augmented and non-augmented training-subject data).

Table 3: Cross-dataset results on GOD (50-way retrieval, chance 2%; averaged over 5 subjects).

Method	Image Retr. ↑	Brain Retr. ↑	PixCorr ↑	SSIM ↑	AlexNet(2) ↑	AlexNet(5) ↑	IncepV3 ↑	CLIP ↑	EffNet-B ↓	SwAV ↓
Supervised (single-subject reference)										
IC-GAN [28]	–	–	.195	.386	88.1	95.3	85.8	84.0	.855	.486
MinD-Vis [20]	–	–	.119	.390	82.8	93.8	81.0	82.6	.833	.491
CMVDM [40]	–	–	.279	.454	88.4	93.9	81.6	82.0	.810	.485
Ours (fine-tuning, leave-one-out)										
VoxTok (GOD fine-tuned)	**72.0** ± **8.37**	**67.20** ± **4.32**	.2071 ± .0239	.3178 ± .0126	78.76 ± 5.09	80.67 ± 3.37	60.91 ± 4.12	71.23 ± 4.57	.9281 ± .0139	.5936 ± .0179

D.2 Single-ROI and Fusion Results on NSD

Table 4 reports the single-ROI zero-shot retrieval results (300-way, averaged over 4 subjects and 2 retrieval directions), together with the fusion results. Fusing all ROIs achieves 38.83%; excluding NSDGeneral still achieves 33.82%, substantially above any single ROI, showing that multi-ROI fusion is effective independently of the broad mask.

Table 4: Single-ROI and fusion zero-shot retrieval on NSD (300-way; chance $\approx 0.33\%$).

ROI (avg. voxels)	300-way Retrieval Acc. (%)
V1 (1129.9)	9.00 ± 4.65
V2 (1099.0)	7.59 ± 0.85
V3 (970.1)	6.27 ± 2.56
hV4 (497.8)	2.00 ± 0.89
midventral (921.6)	1.56 ± 0.24
highventral (7843.9)	5.54 ± 1.77
early (4904.9)	14.12 ± 4.56
NSDGeneral (14549.4)	28.61 ± 6.13
All ROI Fusion	38.83 ± 7.30
Fusion w/o NSDGeneral	33.82 ± 6.55

GOD ROI-size / accuracy. Table 5 reports the same ROI-size analysis on GOD (50-way retrieval, chance 2%; averaged over 5 subjects; fine-tuned model). The ROI-size trend also holds on GOD ($r = 0.710$, $p < 0.074$). Pooling NSD and GOD (ROI sizes normalized by each dataset's General ROI and accuracies by their maximum, with General excluded as the normalizer), the correlation is $r = 0.79$, $p = 0.0004$ over all ROIs, and $r = 0.8250$, $p = 0.0018$ when the outlier highventral is excluded.

D.3 The KNN Candidate Pool

Table 6 reports, for each ROI, the average size of the candidate pool (the union of ROI voxels from the 7 pretraining subjects) and the fraction covered by $K = 450$ neighbors. Except for the smallest

Table 5: GOD ROI-size / accuracy (50-way retrieval, chance 2%; averaged over 5 subjects).

ROI (avg. size)	Forward	Backward	Mean
V1 (802.2)	29.20 ± 6.10	27.20 ± 4.15	28.20 ± 5.31
V2 (947.8)	34.40 ± 7.95	32.00 ± 9.38	33.20 ± 8.78
V3 (853.2)	20.80 ± 4.60	19.60 ± 4.56	20.20 ± 4.62
hV4 (720.4)	6.00 ± 2.24	5.20 ± 3.35	5.60 ± 2.88
early (2140.4)	40.80 ± 5.22	40.80 ± 6.23	40.80 ± 5.75
highventral (2155.2)	11.60 ± 5.59	11.20 ± 5.55	11.40 ± 5.57
General (4403.2)	59.20 ± 8.90	54.40 ± 9.12	56.80 ± 9.32
ALL ROI Fusion	72.00 ± 8.37	67.20 ± 4.32	69.5 ± 7.08

ROI (hV4, 12.9%), the selected neighbors cover only a small portion of each pool, confirming that the estimator performs spatially conditioned neighbor selection rather than an ROI-wide average.

Table 6: Fraction of each ROI's candidate pool covered by $K = 450$.

ROI	Avg. candidate count (N)	K/N
V1	7909	$\approx$5.69%
V2	7693	$\approx$5.85%
V3	6791	$\approx$6.63%
hV4	3484	$\approx$12.92%
midventral	6451	$\approx$6.98%
highventral	54907	$\approx$0.82%
early	34334	$\approx$1.31%
NSDGeneral	101846	$\approx$0.44%

D.4 Extension to Adapters with a Bias Term

To show that the plug-and-play estimator extends beyond bias-free ridge adapters, we train a MindEye2-style decoder on the NSDGeneral ROI whose adapter is a linear layer *with* bias. For a training subject s with N_s voxels and hidden dimension d:

$$\mathbf{h}^{(s)} = (\mathbf{W}^{(s)})^\top \mathbf{b}^{(s)} + \mathbf{B}^{(s)}, \qquad \mathbf{W}^{(s)} \in \mathbb{R}^{N_s \times d}, \quad \mathbf{B}^{(s)} \in \mathbb{R}^d, \tag{7}$$

where $\mathbf{b}^{(s)}$ are the voxel responses, the i-th row of $\mathbf{W}^{(s)}$ is the functional embedding $\mathbf{f}_i^{(s)}$, and $\mathbf{B}^{(s)}$ is a subject-specific bias. Because $\mathbf{B}^{(s)}$ is an ROI-level parameter, we allocate it uniformly over voxels, $\mathbf{B}_i^{(s)} := \frac{1}{N_s}\mathbf{B}^{(s)}$, and estimate both $\hat{\mathbf{f}}_i^{(s)}$ and $\hat{\mathbf{B}}_i^{(s)}$ by the same anatomical KNN averaging used for $\mathbf{f}_i$; the ROI-level bias is recovered as $\hat{\mathbf{B}}^{(s)} = \sum_i \hat{\mathbf{B}}_i^{(s)}$, which scales automatically with N_s. The whole procedure remains training-free.

Table 7 compares the ridge adapter, the bias-adapter without estimating the bias ($\beta = \mathbf{0}$), and the bias-adapter with the KNN-estimated bias. Both settings fall within the unified form $t_i = \gamma(b_i) \odot \mathbf{f}_i + \beta(b_i)$

Table 7: Plug-and-play ablation on adapters with/without bias (300-way retrieval, chance $\approx 0.33\%$).

Adapter	$\gamma(b_i)$	β	FWD	BWD
Ridge regression	b_i	$\mathbf{0}$	17.5 ± 2.3	15.9 ± 2.5
FC with bias	b_i	$\mathbf{0}$ (not estimated)	14.69 ± 2.75	12.56 ± 1.76
FC with bias	b_i	$\hat{\mathbf{B}}^{(s)} = \sum_i \hat{\mathbf{B}}_i^{(s)}$	14.59 ± 2.90	12.14 ± 2.03

of Eq. (1), showing the framework extends to host decoders that expose a bias parameter.

D.5 Noise Robustness and Voxel Subsampling

Tables 8 and 9 provide two additional analyses supporting the large-scale distributed co-activation view. Table 8 progressively injects additive noise (linear scheduler, max beta $= 1.0$, $T = 1000$) and measures 1000-way retrieval (chance $= 0.1\%$; averaged over 4 subjects and both directions); larger ROIs retain a far larger fraction of their accuracy under noise (e.g., at step 400, NSDGeneral retains 85.1% vs. 67.8% for V1). Table 9 uniformly downsamples each ROI in 3D to a common voxel budget (one voxel per 3D cube, no beta smoothing, no retraining); keeping only 34% of NSDGeneral voxels retains 85.2% of its accuracy, and multi-ROI fusion consistently beats any single ROI at every budget.

Table 8: Retrieval accuracy under progressive additive noise (1000-way; ret@s_k = acc@s_k/acc@s_0; score@s_k = (acc@s_k − chance)/(min(acc@s_0) − chance)).

ROI	acc@s_0	acc@s_{400}	acc@s_{500}	acc@s_{600}	ret@400	ret@500	ret@600	score@400	score@500	score@600
NSDGeneral	17.06	14.53	10.49	4.70	85.1%	61.3%	27.1%	33.6×	24.2×	10.7×
early	7.46	5.45	3.95	1.83	72.7%	52.3%	23.5%	12.4×	9.0×	4.0×
V1	4.33	2.97	2.19	0.75	67.8%	49.4%	15.4%	6.7×	4.9×	1.5×
V2	3.55	2.64	1.62	0.80	73.6%	44.1%	20.3%	5.9×	3.5×	1.6×
V3	2.75	1.92	1.20	0.49	68.7%	41.5%	14.7%	4.2×	2.6×	0.9×
highventral	2.26	1.50	1.00	0.46	64.8%	41.7%	16.7%	3.3×	2.1×	0.8×
hV4	0.77	0.33	0.44	0.36	34.3%	50.7%	38.8%	0.5×	0.8×	0.6×
midventral	0.53	0.52	0.39	0.19	97.7%	67.4%	20.9%	1.0×	0.7×	0.2×

Table 9: Uniform 3D voxel subsampling of each ROI (300-way retrieval, forward/backward averaged, 4 subjects; support-set pool $= N_{\text{ROI}} \times 7$).

ROI (avg. size)	Original	5000	1500	1000	500	400	200
V1 (1129.9)	9.00	–	–	8.95	8.54	7.11	4.28
V2 (1099.0)	7.59	–	–	7.19	6.64	5.05	3.30
V3 (970.1)	6.27	–	–	–	4.62	5.46	4.03
hV4 (497.8)	2.00	–	–	2.09	–	1.58	1.57
midventral (921.6)	1.56	–	–	–	1.38	1.35	1.17
highventral (7843.9)	5.54	4.98	2.84	2.35	1.43	1.35	0.70
early (4904.9)	14.12	–	10.06	8.42	4.79	4.49	2.09
NSDGeneral (14549.4)	28.61	24.37	12.16	7.85	3.36	2.42	0.78
ALL ROI Fusion	38.83	37.10	31.08	26.37	18.29	15.38	7.47

E More results

E.1 Subject-specific results

In Tab. 10, we report the individual subject decoding performance on the Natural Scenes Dataset (NSD), corresponding to the final reconstruction results of the main models. These results demonstrate that different subjects' zero-shot decoding accuracy has an obvious gap.

E.2 More Results on Retrieval accuracy with Various K Values

In Fig. 7, we present both the forward(fMRI-to-image) and backward (image-to-fMRI) retrieval results.

E.3 Functional Gradients Cross-subject Comparison for Linear Adapter-based Brain Decoding Model

Fig. 8 gives two additional subjects' functional gradients visualization. As we can see, the pretrained adapters do not have an obvious regional clustering. But after cross-subject KNN estimation, the early visual areas show obvious larger L2 Norm, which indicates that these areas are consistently

Table 10: Individual decoding results on NSD.

Methods	Low- Level				High-Level				Retrieval	
	PixCorr↑	SSIM↑	Alex(2)↑	Alex(5)↑	Incep↑	CLIP↑	Eff↓	SwAV↓	Image↑	Brain↑
Ours(sub1)	.173	.310	78.4%	80.9%	68.4%	69.5%	.878	.549	43.1%	40.1%
Ours(sub2)	.178	.308	80.2%	83.4%	71.7%	73.1%	.854	.534	49.3%	45.1%
Ours(sub5)	.164	.296	78.2%	81.1%	72.5%	73.7%	.855	.538	39.4%	35.6%
Ours(sub7)	.141	.294	74.8%	76.1%	64.3%	65.4%	.897	.562	29.9%	28.1%
Ours(sub1, Low-level)	.220	.474	61.8%	57.8%	51.1%	52.0%	1.01	.660	-	-
Ours(sub2, Low-level)	.221	.473	62.6%	58.7%	52.4%	52.9%	1.01	.659	-	-
Ours(sub5, Low-level)	.207	.472	59.5%	56.9%	51.8%	52.7%	1.02	.658	-	-
Ours(sub7, Low-level)	.177	.468	57.3%	55.7%	51.4%	51.5%	1.02	.659	-	-

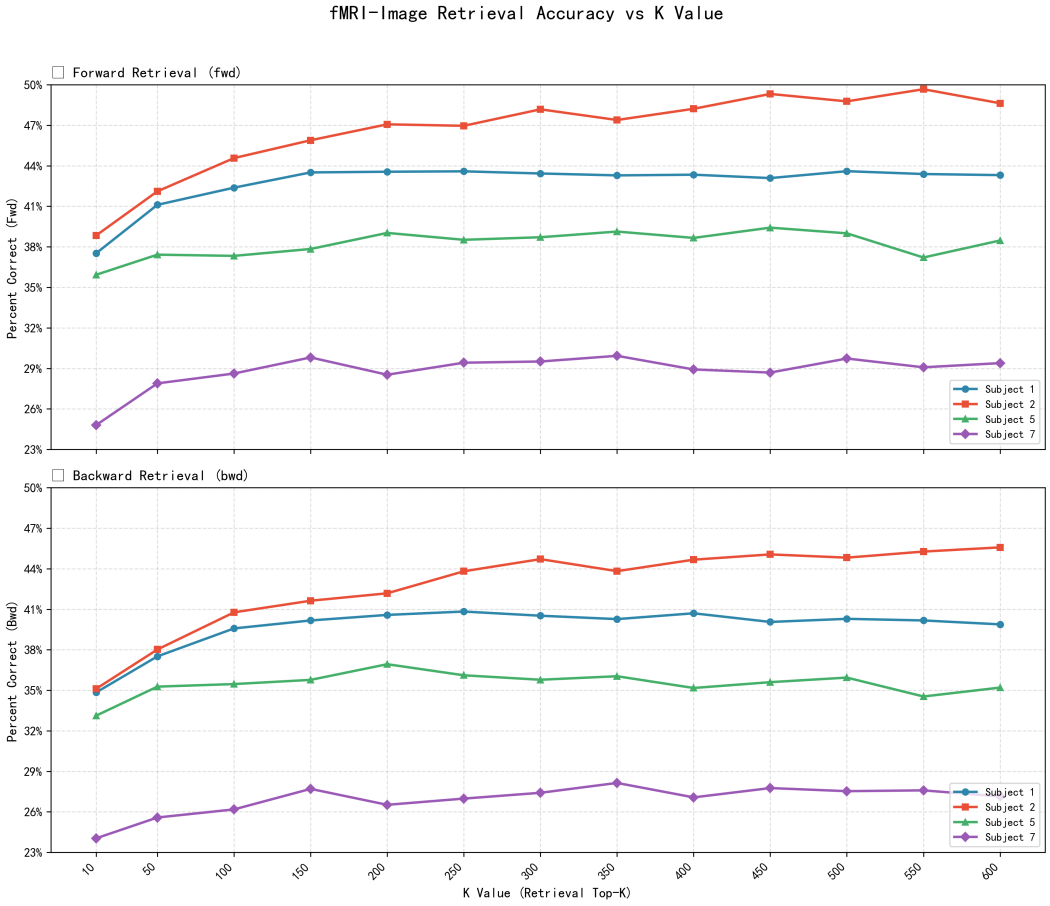


Figure 7: Forward(fMRI-to-image) and backward (image-to-fMRI) retrieval results trend as K value changes.

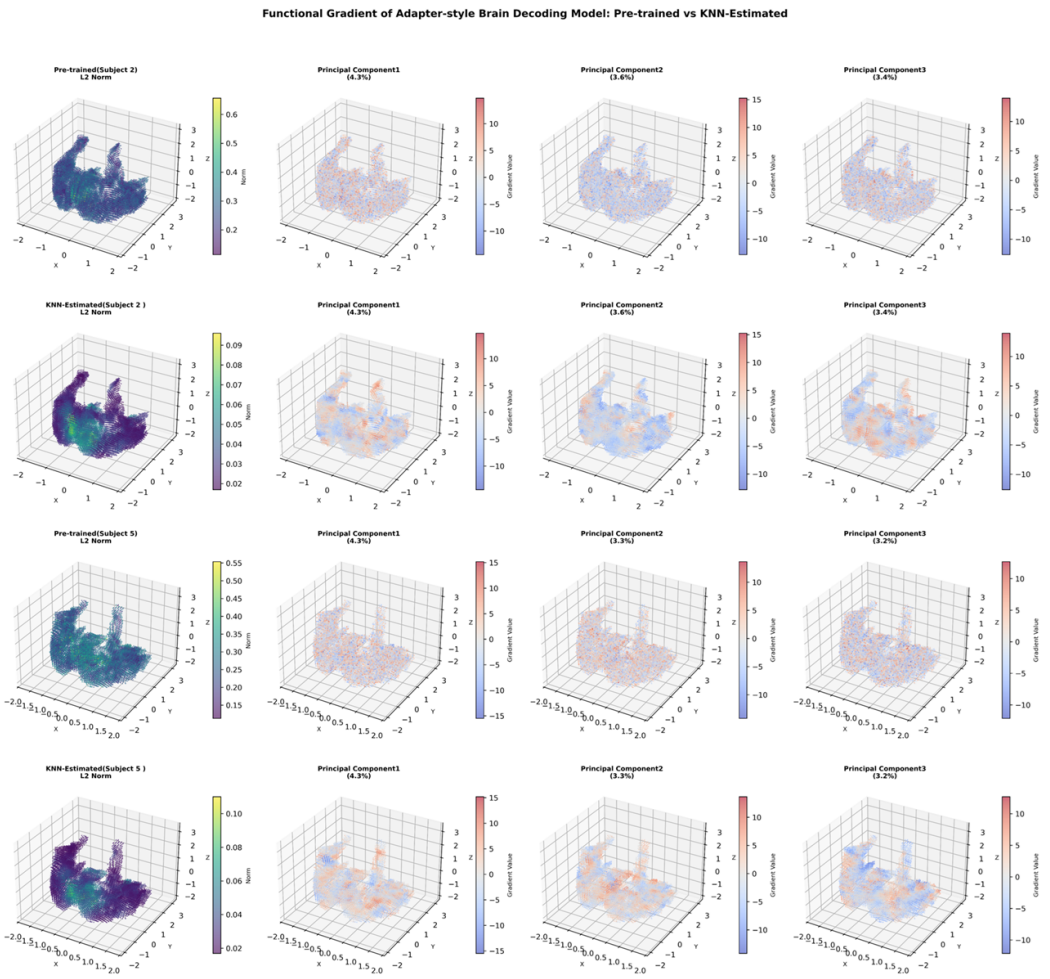


Figure 8: Functional Gradients Cross-subject Comparison for MindEye2 model.

important for cross-subject decoding. And we can also find that the first principal component of our main model's KNN estimation leads to the same regional clustering of voxel function gradients around these areas.

E.4 More Zero-shot Reconstructions

In Fig. 9, we sampled some reconstructions from the first 100 test samples. Our method is not perfect but can reflect some aspects of the original stimuli.

F Broader impacts

This work on zero-shot cross-subject brain decoding holds promise for accelerating brain-computer interface (BCI) development and cognitive neuroscience research by enabling generalizable neural decoding without subject-specific calibration. Such capabilities could lower deployment barriers for clinical assistive technologies, facilitate large-scale studies of human cognition, and support personalized neurofeedback systems. At the same time, the ability to infer mental states or neural representations across individuals without explicit training raises important ethical and privacy concerns, including the risk of unauthorized neural surveillance, potential misuse in behavioral prediction or targeted influence, and demographic biases in decoding accuracy. We emphasize that any real-world application must be grounded in explicit informed consent, rigorous data anonymization, transparent documentation of model limitations, and user-controlled opt-in frameworks. We advocate

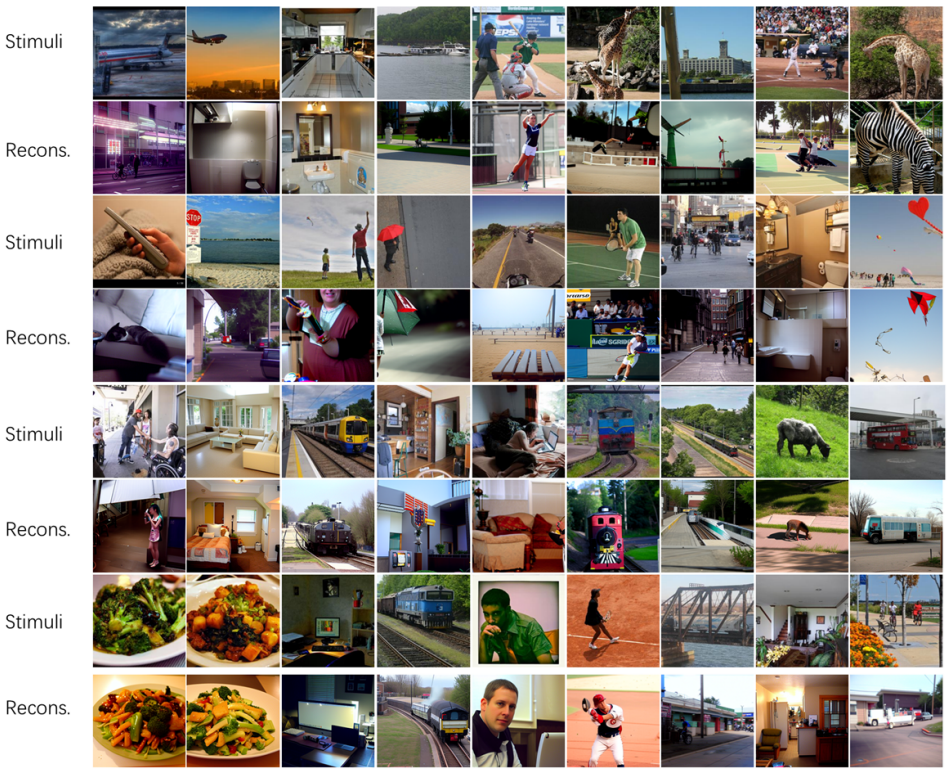


Figure 9: Reconstruction samples from the first 100 test samples

for interdisciplinary oversight and proactive policy development to ensure that zero-shot neural decoding is deployed equitably, responsibly, and in service of human well-being.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction include the claims made in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a paragraph describing the limitations of this work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our main assumption is that the subjects' brain activities are similar in large-scale spatial patterns, which makes the cross-subject zero-shot decoding tasks possible. And our results showed that this assumption may be true.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided most information that needed for reproduction. And we will release our code and pretrained weights later.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We used a opensource dataset, and provided basic information for reproduction. More details will be released in our code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen,type of optimizer) necessary to understand the results. We will opensource the code and pretrained weights upon being accepted.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, we provided error bars for our main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For each training session, we only need one H800 GPU with 80G memory, and pretraining process takes about 30h.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We respect the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide a short statement about Broader impacts in the Apendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[Yes]

    Justification: We cited their papers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new asset is introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer:[N/A]

    Justification: The paper does not involve crowdsourcing. Only Open-source dataset is used.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Only Open-source dataset is used.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology,

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.