A Hierarchical Tokenization Framework for Voxel-Level fMRI Representation Learning

Karanpartap Singh* **Yue Zhao** **Mohammad H. Abbasi** **Ehsan Adeli**
Stanford University

Abstract

We introduce **HBAR** (Hierarchical Brain Activity Representation), a hierarchical tokenization framework for learning compact, multi-scale representations of neuroimaging data. HBAR organizes voxel-level resting-state fMRI into a hierarchy spanning sub-parcel voxel groups, parcels, and large-scale functional networks, enabling up to $415\times$ compression while preserving fine-grained spatial information beyond conventional parcellation. The framework supports fixed atlas-based, fully learned, and softly prior-guided hierarchies, allowing us to systematically study the role of structured priors in brain representation learning. On HCP-YA, at approximately $415\times$ compression, HBAR achieves higher parcel and network correlations than rate-matched flat tokenizers, while at a lower $52\times$ compression rate, shuffling the anatomical grouping reduces parcel correlation from 0.920 to 0.886. Demographic readouts with autoregressive training also reveal potential gains from the hierarchical structure. By combining sub-parcel inputs, compact discrete tokens, and multiscale brain structure, HBAR offers a practical framework for representation learning in high-dimensional neuroimaging data.

1 Introduction

Many modern machine learning systems operate on high-dimensional structured data, where observations are defined over sets of interrelated entities rather than independent features. Examples include images defined over spatial grids (e.g. CNNs) [18], graphs defined over relational neighborhoods (e.g. Graph-CNNs) [16], and biological signals defined over anatomically organized domains such as brain networks [28]. A central challenge in these settings is to construct representations that are both compact enough to enable efficient learning and inference, yet expressive enough to preserve the dependencies and organization inherent to the data.

A common approach is *tokenization*, which maps high-dimensional inputs into compact representations for downstream modeling. *Discrete* tokenization offers additional benefits: fixed-rate codes provide storage bounds, enabling compression at biobank scale, while finite vocabularies support autoregressive sequence modeling. Existing strategies often rely on fixed partitions or uniform downsampling, which can be suboptimal for structured signals that lack regular spatial organization. An alternative is to view tokenization as learned aggregation, where elements are grouped according to their statistical dependencies [29]. This motivates *hierarchical tokenization*, which organizes representations from local to global scales. However, existing approaches often rely on fixed spatial hierarchies (e.g., image pyramids), limiting their applicability to more general structured domains.

Resting-state functional magnetic resonance imaging (rs-fMRI) provides a compelling testbed for hierarchical tokenization. Brain activity is inherently structured across multiple scales, from voxel-level signals to parcels and functional networks, and exhibits rich dependencies that are not strictly

*Correspondence to: karanps@stanford.edu
We make our code available at https://github.com/su-karanps/HBAR.

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

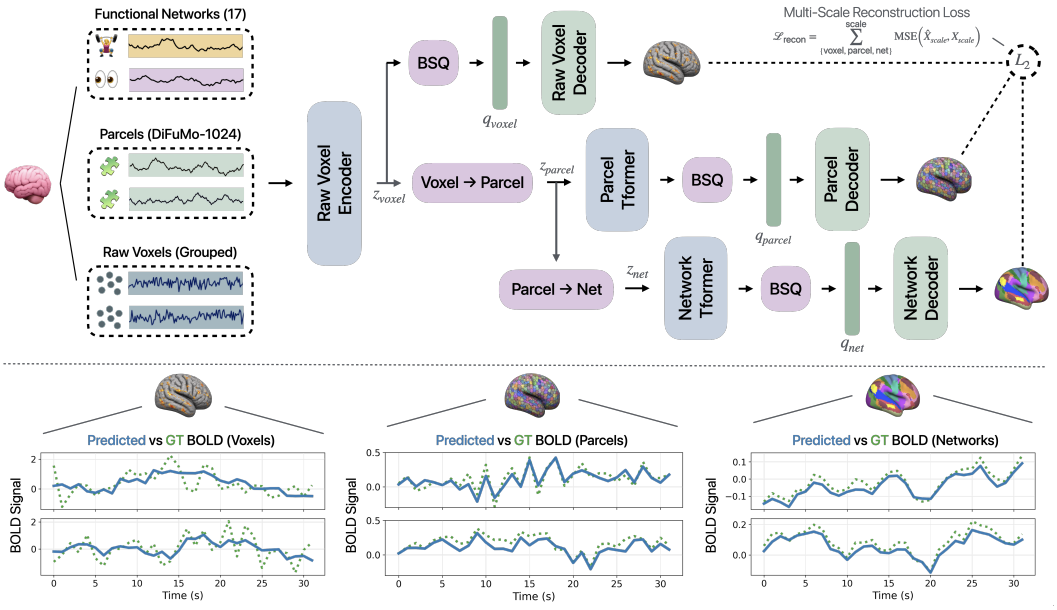

Figure 1: **Hierarchical tokenization and reconstruction for resting-state fMRI.** Voxel-level BOLD signals are organized into a three-level hierarchy spanning sub-parcel voxel groups, parcels (DiFuMo-1024), and large functional networks (Yeo-17). A shared encoder produces voxel-level representations, which are progressively aggregated to parcel- and network-level tokens via hierarchical mappings. At each scale, tokens are discretized using binary spherical quantization (BSQ) and decoded to reconstruct the corresponding signals. The model is trained using a multi-scale reconstruction objective, encouraging representations to preserve information across voxel, parcel, and network levels. Example reconstructions (bottom) illustrate recovery of BOLD dynamics at all scales.

localized in space. At the same time, existing representation learning approaches for rs-fMRI typically operate at one of two extremes: either directly modeling voxel-level data at high computational cost [31], or aggregating signals into predefined atlases or parcellations [6, 8, 36], which can discard fine-grained spatial information. This tension makes rs-fMRI an ideal setting in which to study learned, multiscale tokenization. To this end, we introduce **HBAR (Hierarchical Brain Activity Representation)**, a multiscale tokenization framework for structured neuroimaging data.

Our contributions are summarized as follows: (1) we introduce a general framework for hierarchical tokenization of structured data based on hierarchical aggregation rather than fixed spatial downsampling; (2) we instantiate HBAR for rs-fMRI using a voxel-to-parcel-to-network hierarchy, achieving up to $415\times$ compression and evaluating a higher-fidelity $52\times$ setting; (3) we systematically study the tradeoff between fully learned and prior-guided hierarchical organization, including a hybrid approach combining adaptive tokenization with structured neurobiological priors; and (4) we evaluate reconstruction and demographic prediction with repeated training seeds and matched readouts, finding that atlas-consistent grouping improves parcel reconstruction relative to shuffled grouping and may provide a strong inductive bias for hierarchical representation learning in fMRI.

2 Related Work

2.1 Tokenization and Hierarchical Tokenization

Modern tokenization emerged from variational autoencoders [15] and VQ-VAE [22], which introduced discrete latent representations compatible with autoregressive modeling. Single-scale discrete tokenizers such as VQGAN [9] and DALL-E [24] showed that these codes support large-scale generative modeling, and VQ-VAE-2 [25] introduced the multiscale formulation of coarse and fine latents later adopted by multiscale tokenizers like AudioLM [5]. Together, these works demonstrated

that strong tokenizers can compress high-dimensional signals into compact representations suitable for large-scale generative modeling.

Most hierarchical tokenizers define hierarchy through fixed spatial mechanisms such as multiresolution grids, residual quantization, or decoder pyramids. HR-VQVAE [2], RQ-VAE [19], and VAR [29] progressively refine representations across scales and show that multiscale tokenization improves efficiency and inductive bias. However, these hierarchies remain fundamentally tied to Euclidean image structure rather than structured scientific domains whose meaningful units may be spatially distributed. A second line of work instead learns aggregation directly. Perceiver [14] compresses large inputs into latent bottlenecks via cross-attention, while DiffPool [38] learns soft hierarchical graph coarsening. These works suggest that tokenization need not rely on fixed partitions, but can instead emerge through learned many-to-few aggregation.

2.2 Tokenization and Autoregressive Modeling for fMRI

Recent fMRI foundation models primarily focus on pretraining objectives rather than on learning the tokenizer itself. BrainLM [6] applies large-scale masked-prediction training to rs-fMRI and demonstrates transfer to clinical and demographic tasks, while Brain-JEPA [8] introduces latent-target prediction with spatiotemporal masking. SLIM-Brain [32] moves toward atlas-free voxel-level modeling using selective hierarchical encoding. These works show that both atlas-based and voxel-level pretraining can transfer effectively, but generally treat representational units as fixed.

Several recent works move closer to explicit tokenization and compression. BrainCodec [21] imports neural codec ideas into fMRI for denoising and reconstruction, while HST [35] combines hierarchical state-space modeling with clustered VQ-VAE quantization for brain-state analysis. Earlier voxelwise VAE work [12] similarly demonstrated that learned latent rs-fMRI representations preserve information useful for downstream prediction. Complementary autoregressive approaches include FARM [11], continuous-space autoregression [23], voxel-level transformer prediction [27], and multimodal next-token modeling [37]. Hierarchical functional organization has also been used as an inductive bias for supervised brain-network classification, e.g. DHGFormer [34] guides hierarchical graph modeling with Yeo-network structure and Zhu et al. [44] pool adaptively between ROI graphs. However, these operate on connectivity instead of voxel-level data. Together, these frameworks establish that compression and autoregressive modeling of fMRI signals are both feasible and useful.

2.3 Recent Tokenization and Quantization Techniques

Recent tokenizer research increasingly studies the geometry and scalability of the quantization bottleneck itself. LFQ/MAGVIT-v2 [39], FSQ [20], BSQ [41], and spherical Leech quantization [42] demonstrate that bottleneck geometry strongly influences tokenizer stability, compression efficiency, and scalability. Other approaches focus on compactness and codebook utilization. TiTok [40] compresses images into extremely short token sequences, VQGAN-LC [43] scales to very large vocabularies, IBQ [26] introduces differentiable code indices, and FQGAN [3] factorizes vocabularies into semantically enriched sub-codebooks. Recent work also increasingly co-designs tokenizers and autoregressive models. VAR [29], for example, replaces raster next-token prediction with next-scale prediction over multiscale latent maps generated by a hierarchical VQ-VAE. However, these approaches remain largely tied to predefined image pyramids and grid-structured data, whereas rs-fMRI exhibits distributed low-SNR structure spanning voxel, parcel, and network scales.

3 HBAR: Hierarchical Tokenization for fMRI

3.1 Problem Formulation

We consider data defined over a structured domain represented as a graph $\mathcal{G} = (V, E)$, where V denotes a set of nodes and E encodes relationships between nodes. Each observation is a signal:

$$X \in \mathbb{R}^{|V| \times d_{\text{in}}}$$

where nodes $i \in V$ are associated with feature vectors X_i. In rs-fMRI, nodes correspond to voxels, and edges reflect spatial adjacency or implicit functional relationships. Our goal is to construct a *hierarchical tokenization* of X, producing a sequence of progressively coarser representations:

$$Z^{(0)} \to Z^{(1)} \to \cdots \to Z^{(L)}$$

where $Z^{(0)} = X$, and $Z^{(l)} \in \mathbb{R}^{N_l \times d}$ denotes a set of N_l tokens at level $l > 0$, with $N_0 > N_1 > \cdots > N_L$. In our implementation for fMRI, the graph notation describes grouping structure, and attention within each level is global rather than restricted by an explicit adjacency matrix.

3.2 Hierarchical Tokenization as Aggregation

We define a general tokenization operator at level l as $Z^{(l+1)} = \mathcal{T}^{(l)}(Z^{(l)}, \mathcal{G})$ which maps fine-scale representations to coarser tokens. We consider two complementary regimes:

Fixed (prior-based) tokenization. Tokenization may be defined by a partition of nodes into groups $\{S_k \subset V\}_{k=1}^{N_{l+1}}$, yielding:

$$Z_k^{(l+1)} = \mathrm{Pool}\big(\{Z_i^{(l)} : i \in S_k\}\big)$$

where Pool is an aggregation operator (e.g., mean or attention pooling). In structured domains such as the brain, these partitions may be derived from large-scale datasets (e.g., functional atlases), providing a strong prior over meaningful organization. A large body of resting-state fMRI literature focuses on deriving large scale groupings of brain regions at various scales: as small as seven networks [28] and as large as 1024 [7] or even 3000 parcels [4]. Similarly, we instantiate our framework for resting-state fMRI using a three-level hierarchy spanning voxel, parcel, and network scales. For each subject, we assume a preprocessed 4D resting-state fMRI volume

$$X \in \mathbb{R}^{X \times Y \times Z \times T}$$

in standard space (e.g., 2 mm MNI), along with a gray-matter mask $M \in \{0,1\}^{X \times Y \times Z}$. Only voxels with $M = 1$ are retained, yielding $N_V = |V|$ voxels per subject. In our experiments, the mask retains $N_V = 190{,}468$ voxels and scans have $T \in [922, 1200]$ timepoints.

Learned tokenization via cross-attention. Alternatively, tokenization can be defined as a learned aggregation:

$$Z^{(l+1)} = \mathrm{Attn}(Q^{(l)}, Z^{(l)}),$$

where $Q^{(l)} \in \mathbb{R}^{N_{l+1} \times d}$ are learned query embeddings, and Attn denotes multi-head cross-attention. Each token aggregates information from all tokens at level l with learned, input-dependent weights, allowing the model to learn distributed and non-local structure directly from data.

In our architecture, at the bottom level, voxel-level sub-parcel tokens are first encoded using a temporal encoder operating on local voxel groups over short temporal windows. The resulting token representations are then aggregated hierarchically to produce parcel- and network-level representations. Finally, we also consider a soft hierarchy setting, in which the same learned cross-attention architecture is used, but representations are regularized toward atlas-derived groupings.

Overall, these two regimes represent complementary design choices: fixed tokenization leverages strong domain priors; learned tokenization can capture distributed, data-driven structure. Both are instances of one hierarchical aggregation operator, which generalizes the fixed local neighborhoods of CNNs and the predefined neighborhoods of graph networks to aggregation that may be fixed or learned at each scale. This flexibility is especially important in domains such as brain data, where meaningful structure is often distributed and not strictly localized in space.

3.3 Multiscale Representation

Each level of the hierarchy induces an implicit coarsening of the underlying graph. Tokens at level $l + 1$ can be interpreted as nodes in a coarsened graph $\mathcal{G}^{(l+1)}$, where each coarse node represents a subset or distribution over nodes at level l. Unlike classical graph coarsening methods, which rely on fixed clustering heuristics, our formulation allows tokenization to be defined either through explicit priors or learned aggregation, enabling flexible multiscale representations.

Voxel-level At the lowest level, we construct voxel-level tokens by partitioning each parcel into fixed-size voxel groups of size $K = 16$, preserving fine-grained spatial information that is typically lost in standard parcellations that average much larger groups of voxels. For each parcel, voxel coordinates are normalized and ordered deterministically, after which they are divided into contiguous groups, padding the final group if necessary. This results in $N_C = 12{,}358$ spatial chunks, with 7,260

padded slots. The number of temporal codes per chunk depends on the operating setting (Section 4.3). These ordered voxel lists need not form compact 3D cubes.

Parcel-level At the intermediate level, we use DiFuMo-1024 parcels [7].

Network-level At the top level, we use large-scale functional networks, specifically the Yeo-17 labels supplied with DiFuMo [28], plus an extra group for unassigned parcels, giving 18 groups in total. Importantly, this hierarchy is anchored in well-established functional atlases, which provide a strong prior over brain organization.

With C, P, N denoting chunks, parcels, and networks, the fixed encoder applies a temporal CNN/chunk projection E, factorized transformers F_ℓ, and feature means A_P, A_N:

$$z_C = F_C(E(x)), \qquad z_P = F_P(A_P z_C), \qquad z_N = F_N(A_N z_P). \tag{1}$$

The features have width 192, and transformer depths are 3/2/1 with each depth comprising a spatial and a temporal attention block. Learned pools replace the means with cross-attention. Appendix B specifies all blocks and positions.

Discretizations at multiple levels Vector [22], binary spherical [41], or other quantization [42] can be applied at intermediate levels to obtain discrete tokens; our hierarchical models use BSQ at all three levels. In practice, BSQ discretizes representations by projecting features onto a binary hypersphere using a straight-through sign operator, producing compact discrete tokens while maintaining stable optimization and efficient codebook utilization. Compared to conventional approaches, BSQ enables substantially higher compression without codebook collapse, making it especially attractive for extremely high-dimensional data such as fMRI.

3.4 Optimization and Regularization

Decoding proceeds hierarchically across scales: network-level predictions are first reconstructed from top-level tokens, parcel-level predictions are modeled as residuals relative to broadcast network predictions, and voxel-level reconstructions are modeled as residuals relative to parcel-level signals. This residual formulation encourages coarse representations to capture low-frequency global structure, while finer scales model higher-frequency deviations, and defines a coarse-to-fine residual decomposition. Every reconstruction path passes through discrete codes (Appendix B), with loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\Delta} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}} + \sigma\mathcal{L}_{\text{align}} + \lambda_{\text{mid}}\mathcal{L}_{\text{mid}} \tag{2}$$

Reconstruction terms Mean-squared- or mean-absolute-error (MSE/MAE) supervision individually at voxel, chunk, parcel, and network scales encourages the hierarchy to preserve both fine-grained and large-scale functional organization. The temporal consistency term $\mathcal{L}_\Delta$ encourages preserving local temporal dynamics in noisy voxel signals. Let V, C, P, N denote the voxel, chunk, parcel, and network reconstruction terms. Then, the terms, with weights in Appendix C, Table 3, are defined:

$$\mathcal{L}_{\text{recon}} = \sum_{\ell \in \{V,C,P,N\}} \lambda_\ell \mathcal{L}_\ell\Big(\hat{\mathbf{X}}_{(\ell)}, \mathbf{X}_{(\ell)}\Big) \tag{3}$$

$$\mathcal{L}_{\Delta} = \sum_{\ell \in \{V,C\}} \lambda_{\Delta,\ell} \mathcal{L}_\ell\Big(\Delta_t \hat{\mathbf{X}}_\ell, \Delta_t \mathbf{X}_\ell\Big) \tag{4}$$

Soft cross-scale alignment In the soft hierarchy regime, atlas-derived representations are treated as stopped-gradient targets and do not receive gradients. The alignment objective therefore acts as a soft structural prior, encouraging learned representations to remain close to known functional organization while still permitting flexible reassignment through attention-based pooling. In contrast, the fully learned hierarchy omits this term and learns cross-scale organization purely from reconstruction objectives, without the atlas alignment penalty.

$$\mathcal{L}_{\text{align}} = \left\|\mathbf{Z}^{\text{learned}}_{\text{parcel}} - \mathbf{Z}^{\text{atlas}}_{\text{parcel}}\right\|_2^2 + \left\|\mathbf{Z}^{\text{learned}}_{\text{network}} - \mathbf{Z}^{\text{atlas}}_{\text{network}}\right\|_2^2 \tag{5}$$

BSQ objective The quantization term is summed across all levels:

$$\mathcal{L}_{\text{vq}} = \sum_{\ell \in \{\text{voxel,parcel,network}\}} \mathcal{L}^{(\ell)}_{\text{BSQ}}. \tag{6}$$

For each level ℓ, with $\hat{\mathbf{z}}^{(\ell)} = \text{sign}\big(\mathbf{z}^{(\ell)}\big)$, namely $\mathcal{L}^{(\ell)}_{\text{BSQ}}$ is a weighted average of the commitment loss and the entropy penalty, $\mathcal{L}^{(\ell)}_{\text{BSQ}} = \mathcal{L}^{(\ell)}_{\text{commit}} + \frac{\zeta^{(\ell)}}{\tau^{(\ell)}_{\text{inv}}}\mathcal{L}^{(\ell)}_{\text{entropy}}$:

$$\mathcal{L}^{(\ell)}_{\text{commit}} = \beta^{(\ell)} \left\| \text{sg}\Big(\hat{\mathbf{z}}^{(\ell)}_{\text{scaled}}\Big) - \mathbf{z}^{(\ell)} \right\|_2^2, \quad \mathcal{L}^{(\ell)}_{\text{entropy}} = \gamma_0^{(\ell)} H^{(\ell)}_{\text{persample}} - \gamma^{(\ell)} H^{(\ell)}_{\text{codebook}} \tag{7}$$

where $\text{sg}(\cdot)$ is stop-gradient and β, γ_0, γ and ζ are hyperparameters. BSQ quantization is applied independently at the voxel-, parcel-, and network-token levels prior to hierarchical decoding (Figure 1).

4 Experiments

4.1 Data and Preprocessing

We use $N = 1096$ resting-state fMRI BOLD recordings from the Human Connectome Project Young-Adults (HCP-YA) [30], with one run per subject and TR 0.72 s. We use 877/110/109 subjects for training/validation/test, and train for 10 epochs with a 2×10^{-4} learning rate. The datasets are partitioned at the subject level to prevent any repeated or longitudinal recordings from the same individual from appearing in both training and testing sets. All datasets are preprocessed using standardized pipelines [1, 10], and all models and baselines are trained on identical data splits using approximately matched compression ratios to ensure fair comparison across tokenizer families (see Appendix D). Additional details are presented in Appendix A.

4.2 Baselines

We compare against flat tokenizer baselines at each storage budget and with four atlas-derived representations for linear probing.

Rate-matched flat tokenizers. Both operate directly on voxel chunks. Flat BSQ uses 19 sign bits per position; flat RVQ uses two residual codebooks with 1,024 and 512 entries (10+9 bits). Flat BSQ uses eight temporal positions per chunk at $52\times$ and one at $415\times$. The bit budgets are matched to within 0.2%; model size and training objectives can differ.

MS-VQVAE. A hierarchical VQ-VAE baseline inspired by multiscale autoregressive image models such as VAR [29]. This baseline progressively compresses representations across scales using learned hierarchical latent maps, but does not incorporate structured neurobiological priors or explicit atlas-based organization. To provide a bitrate-matched multiscale VQ baseline, we implement a two-level hierarchical VQ-VAE following the progressive latent compression strategy used in VAR [29]. The model has two discrete scales: a middle scale with one token per voxel chunk (12,358 per window), quantized with a two-stage residual VQ (2×512 codes, 18 bits), and a top scale with 1,024 tokens obtained by learned cross-attention pooling over the chunk tokens, quantized with a 4,096-code VQ (12 bits). Both quantizers use EMA codebook updates with dead-code restarts on a 32-dimensional projection, giving $\approx 415\times$ compression. Unlike our hierarchical BSQ framework, which uses three explicit scales (voxel, parcel, and network), the MS-VQVAE baseline uses only two discrete latent scales and does not incorporate structured priors or cross-scale alignment objectives.

DiFuMo Baselines. We use Yeo network means, DiFuMo parcel means, and two training-fitted PCA representations derived from parcel time series or parcel functional connectivity.

4.3 Temporal Windowing

Training is performed on overlapping temporal windows of length $W = 32$ with stride $s = 16$. For a sequence of length T, this yields $N_{\text{win}} = \lfloor \frac{T-W}{s} \rfloor + 1$ training windows per scan, corresponding to

approximately 56–74 windows in our setting. Each training sample consists of voxel-level windows, sub-parcel token groups, and corresponding parcel- and network-level summaries. Within the encoder, temporal windows are partitioned into smaller segments that are summarized into a fixed number of latent temporal tokens at each scale.

HBAR-52 stores eight temporal codes per spatial unit (one per four TRs), with 18, 12, and 6 bits at the chunk, parcel, and network levels. HBAR-415 stores one code per 32-TR window, with 18/12/24 bits. The compact setting also changes temporal pooling, the parcel decoder, and residual supervision, and is therefore a separately trained system rather than a pure rate ablation (Appendix B).

4.4 Evaluation and uncertainty

Reconstruction uses the same 16 windows for each of 109 test subjects. All models are evaluated by pooling their final voxel reconstructions with the true atlas: voxels are equally weighted within parcels and parcels are equally weighted within networks. Pearson correlation flattens time and units per window; scores are averaged over windows within each subject and then across subjects. We distinguish three-seed sample SD from per-model subject-bootstrap CIs and paired contrasts (Appendix E).

Linear probes use mean-pooled post-quantization features and logistic regression, with training-only standardization and five-fold inner-CV regularization selection. We report AUROC for sex and for three-bin age for seed 0. The AR readout applies the same type of probe to mean-pooled hidden states of a causal model trained on bottom-level tokens over four consecutive 32-TR windows. Probes and atlas baselines use matched stride-64 windows; full settings are in Appendix F.

Model	Ratio	Voxel Corr↑	Parcel Corr↑	Net Corr↑
Window-Mean Oracle	—	0.503	0.578	0.522
Random Baseline	—	0.000	0.000	−0.002
HBAR-415 (fixed)	415×	0.405 ± 0.004	$\mathbf{0.746 \pm 0.003}$	$\mathbf{0.895 \pm 0.002}$
Flat BSQ	415×	0.394 ± 0.002	0.627 ± 0.005	0.710 ± 0.010
Flat RVQ	415×	$\underline{0.413 \pm 0.009}$	$\underline{0.691 \pm 0.002}$	$\underline{0.803 \pm 0.010}$
MS-VQVAE	415×	$\mathbf{0.427 \pm 0.001}$	0.594 ± 0.000	0.641 ± 0.004
HBAR-52 (fixed)	52×	$\underline{0.526 \pm 0.004}$	$\mathbf{0.920 \pm 0.003}$	$\underline{0.965 \pm 0.005}$
Learned pooling	52×	0.519 ± 0.005	0.876 ± 0.005	0.938 ± 0.009
Soft pooling	52×	0.487 ± 0.063	0.796 ± 0.022	0.898 ± 0.030
Shuffled grouping	52×	0.523 ± 0.006	0.886 ± 0.004	$\underline{0.966 \pm 0.011}$
Flat BSQ	52×	$\mathbf{0.529 \pm 0.002}$	$\underline{0.890 \pm 0.002}$	$\mathbf{0.968 \pm 0.010}$

Table 1: Reconstruction performance across tokenizers and baselines on 109 held-out subjects with 16 windows per subject (1,744 windows). Higher correlation is better. Tokenizer results report mean ± sample standard deviation across three training seeds; ratios denote rounded per-window code-payload compression. The first two rows are target-informed, model-free references: the window-mean oracle predicts each unit's true within-window temporal mean, while the random reference samples Gaussian noise matched to the target window's global mean and standard deviation at each scope. Reference rows report point estimates. Bold and underlining indicate the highest and second-highest mean within each tokenizer rate block, respectively, not statistical significance.

5 Results

5.1 Reconstruction

At approximately $415\times$ compression, HBAR achieves parcel and network correlations of 0.746 and 0.895, compared with 0.691 and 0.803 for flat RVQ and 0.627 and 0.710 for flat BSQ (Table 1). Voxel correlation is slightly lower than for RVQ (0.405 versus 0.413) and higher than for flat BSQ (0.394), with the MS-VQVAE baseline achieving the highest correlation (0.427). At this bit budget, HBAR's gain over the baselines is concentrated at the coarser scales. At $52\times$, flat BSQ gives similar voxel and network correlations to HBAR, while fixed HBAR retains the highest parcel correlation (0.920).

Model	Linear Probing		AR Readout	
	Age AUROC↑	Sex AUROC↑	Age AUROC↑	Sex AUROC↑
Yeo-Networks	0.530 (0.44, 0.61)	0.551 (0.44, 0.66)	—	—
DiFuMo	0.561 (0.47, 0.65)	0.559 (0.45, 0.67)	—	—
DiFuMo → PCA	0.580 (0.49, 0.67)	0.574 (0.47, 0.68)	—	—
Parcel-FC + PCA	0.683 (0.61, 0.75)	**0.956** (0.92, 0.99)	—	—
HBAR-415 (fixed)	0.482 (0.41, 0.56)	<u>0.701</u> (0.60, 0.79)	**0.584** (0.50, 0.66)	**0.638** (0.53, 0.74)
Flat BSQ-415	**0.607** (0.53, 0.68)	0.648 (0.54, 0.75)	0.506 (0.43, 0.59)	0.556 (0.45, 0.66)
Flat RVQ-415	<u>0.557</u> (0.47, 0.64)	**0.722** (0.62, 0.81)	0.503 (0.42, 0.58)	<u>0.636</u> (0.53, 0.74)
MS-VQVAE-415	0.540 (0.45, 0.63)	0.605 (0.50, 0.71)	<u>0.515</u> (0.43, 0.60)	0.545 (0.43, 0.66)
HBAR-52 (fixed)	<u>0.606</u> (0.52, 0.68)	**0.795** (0.71, 0.87)	<u>0.557</u> (0.48, 0.63)	**0.792** (0.70, 0.87)
Learned pooling (52×)	0.593 (0.51, 0.68)	0.746 (0.65, 0.83)	**0.588** (0.50, 0.67)	0.742 (0.65, 0.83)
Soft pooling (52×)	**0.655** (0.56, 0.75)	0.744 (0.65, 0.83)	0.507 (0.43, 0.59)	0.711 (0.61, 0.81)
Shuffled grouping (52×)	0.588 (0.50, 0.67)	<u>0.769</u> (0.68, 0.85)	0.525 (0.43, 0.61)	<u>0.752</u> (0.66, 0.84)
Flat BSQ-52	0.580 (0.50, 0.66)	0.737 (0.64, 0.82)	<u>0.557</u> (0.47, 0.64)	0.640 (0.53, 0.74)

Table 2: Demographic prediction performance on 109 held-out test subjects using linear probing and autoregressive (AR) readout. Age is a three-bin classification task; values are AUROC with 95% subject-bootstrap confidence intervals. Linear probes use pooled tokenizer features or the indicated atlas-based representations. AR readouts use mean-pooled final-layer hidden states from causal sequence models trained on bottom-level token sequences. Tokenizer results use seed 0 in both panels. Bold and underlining indicate the highest and second-highest point estimates within each tokenizer rate block, respectively.

At 52×, shuffled grouping yields nearly the same voxel correlation as fixed atlas grouping (0.523 versus 0.526) and nearly the same network correlation (0.966 versus 0.965). The difference appears at the parcel scale: preserving the anatomical chunk-to-parcel assignments raises correlation to 0.920. A spectral analysis is included in Appendix G.

5.2 Downstream Performance

We next evaluate whether the learned representations support downstream modeling beyond reconstruction. We consider two evaluation settings: (*i*) linear probing using mean-pooled token embeddings, and (*ii*) autoregressive modeling using a causal sequence model trained directly on the tokenized representations. Table 2 reports demographic classification from both readouts.

The relative ordering of models differs between linear probing and autoregressive evaluation. At $52\times$, fixed HBAR achieves strong sex AUROC under both linear probing (0.795) and AR readout (0.792), leading the methods reported in Table 2. At $415\times$, HBAR achieves the highest observed AR AUROC for both age (0.584) and sex (0.638), despite not leading under linear probing. These results suggest that the utility of compressed representations depends on the downstream readout, and that linear decodability alone may not fully capture their suitability for sequence modeling. Parcel-FC with PCA remains the strongest overall linear baseline for sex prediction (AUROC 0.956), consistent with prior work demonstrating the predictive value of resting-state connectivity [33, 13].

Performance on age prediction is comparatively modest across all methods. At $415\times$, HBAR has the highest observed AR AUROC (0.584), while flat BSQ leads under linear probing (0.607). At $52\times$, soft pooling leads under linear probing (0.655) and learned pooling leads with the AR readout (0.588). HCP-YA's narrow, binned age range limits this task (see Limitations).

5.3 Learned vs. Fixed Hierarchies

To better understand the role of structured priors, we compare fully learned, fixed atlas-based, and softly aligned hierarchical tokenization strategies (Fig. 2). The soft hierarchy introduces a tunable alignment parameter σ controlling the tradeoff between learned flexibility and adherence to atlas structure. As shown in Fig. 2 (left), varying σ changes reconstruction only modestly and non-monotonically, and no setting closes the gap to the fixed hierarchy at parcel or network scale. The

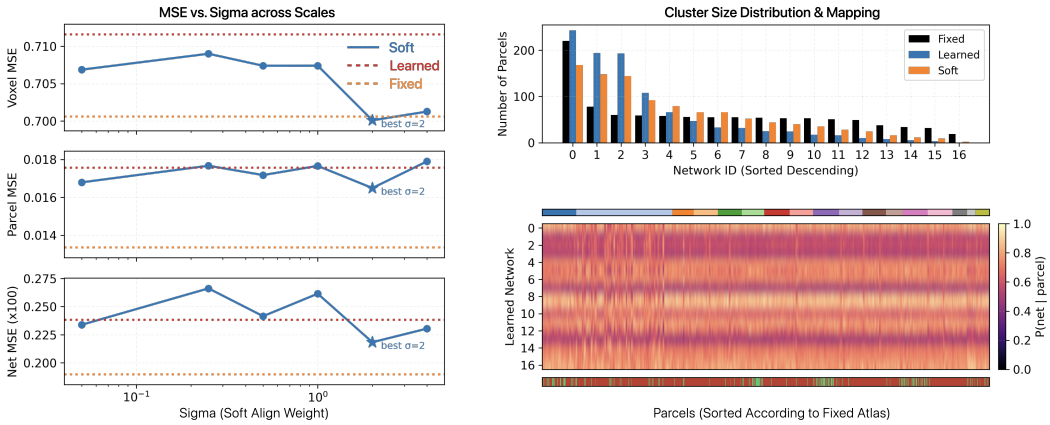

Figure 2: **Comparison of fixed, learned, and softly aligned hierarchical tokenization strategies.** (**Left**) Reconstruction error across voxel, parcel, and network scales as a function of the soft-alignment weight σ. Larger σ imposes stronger alignment to the atlas prior. Across the sweep, $\sigma = 2$ gives the lowest soft-hierarchy error at all three scales; the soft hierarchy matches the fixed hierarchy at voxel level but remains worse at parcel and network scales for every σ. (**Top right**) Distribution of parcel assignments across learned network slots for fixed, learned, and soft hierarchies. Because learned network indices are permutation-invariant, we compare distributions rather than direct one-to-one correspondence. Both learned and soft hierarchies exhibit skewed assignment patterns similar to the atlas-based hierarchy, suggesting convergence toward structured large-scale organization. (**Bottom right**) Soft parcel-to-network assignment matrix after Hungarian alignment between learned and atlas-derived network indices. Parcels are sorted according to fixed atlas labels, and colors indicate the probability of assignment to each aligned learned network. The diffuse structure and sparse hard matches indicate that fully learned hierarchies only partially recover atlas-like organization, despite exhibiting similar global assignment statistics.

best-performing configurations at coarser scales use predefined atlas organization, reinforcing the importance of structured priors in modeling large-scale brain activity.

Because learned network indices are permutation-invariant, direct comparison between learned and atlas-derived hierarchies is not meaningful. Instead, we analyze global assignment structure and align learned network slots to atlas networks using Hungarian matching [17]. Interestingly, the learned hierarchy exhibits assignment distributions broadly similar to the fixed atlas hierarchy, with several dominant network slots containing disproportionately large mass (Fig. 2, top right). This suggests that the model partially recovers large-scale organizational structure even without explicit supervision. However, the aligned soft assignment matrix reveals that learned parcel-to-network mappings remain substantially more diffuse than atlas-derived organization (Fig. 2, bottom right). Rather than recovering localized atlas-like partitions, learned hierarchies distribute mass, indicating that reconstruction objectives alone are insufficient to recover canonical functional organization.

5.4 Model Efficiency

HBAR contains 5.6M/7.3M parameters for the $52\times/415\times$ models, and processes a temporal window in 231 ms on 1x H100 GPU in our implementation. This corresponds to approximately 9 seconds to tokenize an entire 1200-frame resting-state scan, making the approach practical for patient-level analysis and large-scale neuroimaging datasets alike while operating directly on voxel-level signals.

6 Discussion

We introduced a hierarchical tokenization for structured data and instantiated it for resting-state fMRI through a voxel-to-parcel-to-network hierarchy. At $415\times$ compression, the hierarchy preserved parcel and network signals more accurately than baselines. At $52\times$, flat BSQ was competitive, although HBAR still maintained the best parcel reconstruction. At both scales, HBAR maintained strong downstream performance. The preferred tokenizer therefore depends not only on compression rate

and reconstruction scale, but also on how the resulting codes are used. In summary, we demonstrated that compact discrete representations can retain information from voxel-level fMRI while remaining practical for large-scale modeling.

A second key finding is that neurobiological priors can improve structural fidelity while retaining downstream predictive utility. At $52\times$ compression, the fixed anatomical hierarchy reconstructed parcel signals more accurately than learned or soft alternatives. Shuffling chunk-to-parcel assignments reduced parcel correlation across all three seeds, indicating that the benefit extends beyond the grouping sizes alone. Importantly, this structural advantage does not come at the expense of demographic information: fixed HBAR achieves strong sex prediction under both linear and autoregressive readouts, while at $415\times$ it leads the AR comparisons on both tasks. Together, these findings suggest that anatomical structure provides a useful inductive bias for compact representation learning.

More broadly, these findings highlight a useful design principle for representation learning in scientific domains. Unlike images or videos, where multiscale structure is often defined by spatial resolution, fMRI has meaningful organization across distributed functional systems. Our results indicate that hierarchical tokenization should not simply import grid-based multiscale designs, but should instead exploit domain structure. In this sense, atlas-based hierarchies are not merely preprocessing choices; they may provide powerful inductive biases that can guide interpretable and effective token formation.

Limitations This study focuses on resting-state fMRI from HCP-YA, which has a relatively narrow young-adult age range and discretized age labels. As a result, age prediction is a challenging and limited downstream task. Our experiments also use a fixed atlas hierarchy based on DiFuMo-1024 and Yeo-17, and although these are well-established priors, other parcellations or subject-specific functional networks may yield different tradeoffs. Additionally, our scale comparison contrasts one-scale flat tokenizers with the three-scale hierarchy and is not a sweep over hierarchy depth. Finally, while we evaluate autoregressive modeling and demographic prediction, we do not yet test broader clinical, cognitive, or cross-dataset transfer settings where the value of voxel-preserving tokenization may be even larger.

Future work Future work should evaluate hierarchical tokenization across more diverse neuroimaging cohorts, including lifespan, aging, and disease datasets. The framework also does not depend on volumetric input: because it assumes only a graph, the nodes could equally be CIFTI grayordinates with a surface-based parcellation at the intermediate level, which would likely improve cross-subject cortical correspondence at the cost of the subcortical and cerebellar coverage our gray-matter mask retains. A particularly important direction is subject-adaptive tokenization, where atlas priors provide initialization or regularization while allowing individualized deviations in functional organization. Another promising direction is to couple the tokenizer more tightly to downstream autoregressive objectives, so that the hierarchy directly shapes temporal brain dynamics modeling. Finally, while this paper focuses on fMRI, the framework is naturally applicable to other structured scientific signals, including electrophysiology, spatial transcriptomics, and sensor networks, where meaningful units are distributed and multiscale.

Conclusion Hierarchical tokenization provides a compact and expressive way to represent high-dimensional voxel-level fMRI. By preserving sub-parcel information while organizing representations across parcel and network scales, HBAR preserves meaningful brain signals under substantial compression while retaining information useful for downstream demographic prediction and autoregressive modeling. At the same time, our comparison of fixed, learned, and soft hierarchies shows that strong domain priors remain critical: established functional atlases provide structure that is difficult to recover from reconstruction alone. These results argue for a hybrid view of scientific representation learning, where learned tokenizers are most effective when paired with principled domain structure.

Acknowledgments. This work was supported by the Stanford Institute for Human-Centered AI (HAI) Hoffman-Yee Award, UST, and National Institutes of Health (NIH) Grant AG089169. KS was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE-2146755. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or NIH.

References

[1] Mohammad Hassan Abbasi and Ehsan Adeli. fMRI Processing Pipeline: A lightweight, end-to-end framework for BIDS-compatible fMRI preprocessing and quality control, 2025.

[2] Mohammad Adiban, Kalin Stefanov, Sabato Marco Siniscalchi, and Giampiero Salvi. Hierarchical Residual Learning Based Vector Quantized Variational Autoencoder for Image Reconstruction and Generation. 8 2022.

[3] Zechen Bai, Jianxiong Gao, Ziteng Gao, Pichao Wang, Zheng Zhang, Tong He, and Mike Zheng Shou. Factorized Visual Tokenization and Generation. 11 2024.

[4] Thomas Blumensath, Saad Jbabdi, Matthew F. Glasser, David C. Van Essen, Kamil Ugurbil, Timothy E.J. Behrens, and Stephen M. Smith. Spatially constrained hierarchical parcellation of the brain with resting-state fMRI. *NeuroImage*, 76:313–324, 8 2013. ISSN 10538119. doi: 10.1016/j.neuroimage.2013.03.024.

[5] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: a Language Modeling Approach to Audio Generation. 7 2023.

[6] Josue Ortega Caro, Antonio H. de O. Fonseca, Christopher Averill, Syed A. Rizvi, Matteo Rosati, James L. Cross, Prateek Mittal, Emanuele Zappala, Daniel Levine, Rahul M. Dhodapkar, Insu Han, Amin Karbasi, Chadi G. Abdallah, and David van Dijk. BrainLM: A foundation model for brain activity recordings, 9 2023.

[7] Kamalaker Dadi, Gaël Varoquaux, Antonia Machlouzarides-Shalit, Krzysztof J. Gorgolewski, Demian Wassermann, Bertrand Thirion, and Arthur Mensch. Fine-grain atlases of functional modes for fMRI analysis. *NeuroImage*, 221:117126, 11 2020. ISSN 10538119. doi: 10.1016/j.neuroimage.2020.117126.

[8] Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Su Xian Chong, Fang Ji, Nathanael Ren Jie Tong, Christopher Li Hsian Chen, and Juan Helen Zhou. Brain-JEPA: Brain Dynamics Foundation Model with Gradient Positioning and Spatiotemporal Masking. 9 2024.

[9] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. 6 2021.

[10] Oscar Esteban, Christopher J. Markiewicz, Ross W. Blair, Craig A. Moodie, A. Ilkay Isik, Asier Erramuzpe, James D. Kent, Mathias Goncalves, Elizabeth DuPre, Madeleine Snyder, Hiroyuki Oya, Satrajit S. Ghosh, Jessey Wright, Joke Durnez, Russell A. Poldrack, and Krzysztof J. Gorgolewski. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*, 16(1):111–116, 1 2019. ISSN 1548-7091. doi: 10.1038/s41592-018-0235-4.

[11] Rahul Garg, Guillermo A. Cecchi, and A. Ravishankar Rao. Full-brain auto-regressive modeling (FARM) using fMRI. *NeuroImage*, 58(2):416–441, 9 2011. ISSN 10538119. doi: 10.1016/j.neuroimage.2011.02.074.

[12] Eloy Geenjaar, Tonya White, and Vince Calhoun. Variational voxelwise rs-fMRI representation learning: Evaluation of sex, age, and neuropsychiatric signatures. 8 2021.

[13] Tong He, Ru Kong, Avram J. Holmes, Minh Nguyen, Mert R. Sabuncu, Simon B. Eickhoff, Danilo Bzdok, Jiashi Feng, and B.T. Thomas Yeo. Deep neural networks and kernel regression achieve comparable accuracies for functional connectivity prediction of behavior and demographics. *NeuroImage*, 206:116276, 2 2020. ISSN 10538119. doi: 10.1016/j.neuroimage.2019.116276.

[14] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General Perception with Iterative Attention. 6 2021.

[15] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. 12 2022.

[16] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2 2017.

[17] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 3 1955. ISSN 0028-1441. doi: 10.1002/nav.3800020109.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 00189219. doi: 10.1109/5.726791.

[19] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive Image Generation using Residual Quantization. 3 2022.

[20] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite Scalar Quantization: VQ-VAE Made Simple. 10 2023.

[21] Yuto Nishimura, Masataka Sawayama, Ayumu Yamashita, Hideki Nakayama, and Kaoru Amano. BrainCodec: Neural fMRI codec for the decoding of cognitive brain states. 10 2024.

[22] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. 5 2018.

[23] François Paugam, Basile Pinsard, Guillaume Lajoie, and Pierre Bellec. A benchmark of individual auto-regressive models in a massive fMRI dataset. *Imaging Neuroscience*, 2, 7 2024. ISSN 2837-6056. doi: 10.1162/imag{_}a{_}00228.

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 5 2021. URL `https://proceedings.mlr.press/v139/ramesh21a.html`.

[25] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. 6 2019.

[26] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable Image Tokenization with Index Backpropagation Quantization. 3 2025.

[27] Yifei Sun, Daniel Chahine, Qinghao Wen, Tianming Liu, Xiang Li, Yixuan Yuan, Fernando Calamante, and Jinglei Lv. Voxel-Level Brain States Prediction Using Swin Transformer. 6 2025.

[28] B. T. Thomas Yeo, Fenna M. Krienen, Jorge Sepulcre, Mert R. Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L. Roffman, Jordan W. Smoller, Lilla Zöllei, Jonathan R. Polimeni, Bruce Fischl, Hesheng Liu, and Randy L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3):1125–1165, 9 2011. ISSN 0022-3077. doi: 10.1152/jn.00338.2011.

[29] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction. 6 2024.

[30] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E.J. Behrens, Essa Yacoub, and Kamil Ugurbil. The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80:62–79, 10 2013. ISSN 10538119. doi: 10.1016/j.neuroimage.2013.05.041.

[31] Cheng Wang, Yu Jiang, Zhihao Peng, Chenxin Li, Changbae Bang, Lin Zhao, Wanyi Fu, Jinglei Lv, Jorge Sepulcre, Carl Yang, Lifang He, Tianming Liu, Xue-Jun Kong, Quanzheng Li, Daniel S. Barron, Anqi Qiu, Randy Hirschtick, Byung-Hoon Kim, Hongbin Han, Xiang Li, and Yixuan Yuan. Towards a general-purpose foundation model for fMRI analysis. 3 2026.

[32] Mo Wang, Junfeng Xia, Wenhao Ye, Enyu Liu, Kaining Peng, Jianfeng Feng, Quanying Liu, and Hongkai Wen. SLIM-Brain: A Data- and Training-Efficient Foundation Model for fMRI Data Analysis. 1 2026.

[33] Susanne Weis, Kaustubh R Patil, Felix Hoffstaedter, Alessandra Nostro, B T Thomas Yeo, and Simon B Eickhoff. Sex Classification by Resting State Brain Connectivity. *Cerebral Cortex*, 30 (2):824–835, 3 2020. ISSN 1047-3211. doi: 10.1093/cercor/bhz129.

[34] Rundong Xue, Hao Hu, Zeyu Zhang, Xiangmin Han, Juan Wang, Yue Gao, and Shaoyi Du. DHGFormer: Dynamic Hierarchical Graph Transformer for Disorder Brain Disease Diagnosis. pages 268–278. 2026. doi: 10.1007/978-3-032-05162-2{_}26.

[35] Yanwu Yang and Thomas Wolfers. Hierarchical Characterization of Brain Dynamics via State Space-based Vector Quantization. 6 2025.

[36] Yanwu Yang, Chenfei Ye, Guinan Su, Ziyao Zhang, Zhikai Chang, Hairui Chen, Piu Chan, Yue Yu, and Ting Ma. BrainMass: Advancing Brain Network Analysis for Diagnosis With Large-Scale Self-Supervised Learning. *IEEE Transactions on Medical Imaging*, 43(11):4004–4016, 11 2024. ISSN 0278-0062. doi: 10.1109/TMI.2024.3414476.

[37] Xuanhua Yin, Runkai Zhao, Lina Yao, and Weidong Cai. BrainVista: Modeling Naturalistic Brain Dynamics as Multimodal Next-Token Prediction. 2 2026.

[38] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling. 2 2019.

[39] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language Model Beats Diffusion – Tokenizer is Key to Visual Generation. 3 2024.

[40] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An Image is Worth 32 Tokens for Reconstruction and Generation. 6 2024.

[41] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and Video Tokenization with Binary Spherical Quantization. 6 2024.

[42] Yue Zhao, Hanwen Jiang, Zhenlin Xu, Chutong Yang, Ehsan Adeli, and Philipp Krähenbühl. Spherical Leech Quantization for Visual Tokenization and Generation. 12 2025.

[43] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the Codebook Size of VQGAN to 100,000 with a Utilization Rate of 99%. 6 2024.

[44] Qi Zhu, Shengrong Li, Xiangshui Meng, Qiang Xu, Zhiqiang Zhang, Wei Shao, and Daoqiang Zhang. Spatio-Temporal Graph Hubness Propagation Model for Dynamic Brain Network Classification. *IEEE Transactions on Medical Imaging*, 43(6):2381–2394, 6 2024. ISSN 0278-0062. doi: 10.1109/TMI.2024.3363014.

A Data and spatial layout

A.1 Cohort and preprocessing

We use one selected left-to-right resting-state run per HCP-YA subject (the longest eligible run in the data), yielding 1,096 subjects. Of these, 1,091 runs have 1,200 TRs and five have 922–1,124 TRs, for 1,314,326 TRs in total with a TR of 0.72 s. This produces 877 training, 110 validation, and 109 test subjects. With 32-TR windows, stride 16, and no incomplete trailing windows, the data comprises 64,855/8,125/8,066 train/validation/test windows (81,046 total). Splits are by subject, not by window. Family identifiers were not available to enforce family-disjoint splits.

Beyond fMRIPrep, HBAR preprocessing retains every voxel in the gray-matter mask, z-scores each voxel over time within its scan with $\epsilon = 10^{-6}$, and stores float16 arrays. HBAR adds no denoising, filtering, detrending, or censoring. Normalization uses the complete individual scan, including time

points outside a given window. The input grid is MNI-152 at 2 mm, $97 \times 115 \times 97$ (1,082,035 grid positions), of which the template mask retains 190,468 unique voxels. The mask derives from a template gray-matter mask resampled with nearest-neighbor interpolation, and the retained volume includes cortical, cerebellar, and subcortical regions; a small number of voxels in white-matter/CSF may also remain near mask boundaries.

A.2 Atlas construction and padding

The DiFuMo-1024 continuous maps are cubically resampled to the BOLD grid, then converted to a hard, nonoverlapping partition by an argmax across maps. Within each parcel, voxel coordinates are min–max normalized along each axis, ordered by $x + 10^{-2}y + 10^{-4}z$, and divided into consecutive groups of 16. Chunks are ordered voxel lists, not necessarily compact 3D cubes or a space-filling curve. Parcel order and slot order are deterministic, and each voxel belongs to exactly one chunk.

There are 12,358 chunks with 197,728 slots, comprising 190,468 valid voxels and 7,260 zero-filled padding slots. Of the chunks, 925 are partial and no chunk is entirely padding. Fourteen of the 1,024 parcels have no retained voxels, and their pooled signal is defined to be zero. They remain as token slots and are not attention-masked, so their codes still count toward storage. The model's CNN processes zero padding without an encoder mask; voxel reconstruction drops invalid slots, whereas chunk-level training losses include them.

The network mapping uses 17 Yeo networks plus an eighteenth group for unassigned parcels. That group contains 220 parcels, including all 14 empty parcels, and is not interpreted as a physiological network. For a nonempty parcel p with voxel set $\mathcal{V}_p$, the target signal is:

$$y_P[p,t] = \frac{1}{|\mathcal{V}_p|} \sum_{v \in \mathcal{V}_p} x[v,t], \qquad y_N[n,t] = \frac{1}{|\mathcal{P}_n|} \sum_{p \in \mathcal{P}_n} y_P[p,t] \tag{8}$$

The network mean is unweighted over parcels, not over voxels. The unassigned group's target includes its 14 zero parcel signals in its denominator of 220. Feature pooling in Eq. (1), by contrast, is an unweighted mean of child tokens; it does not weight partial chunks by voxel count.

B Architecture and decoding details

B.1 The 52× configuration

For each four-TR subwindow and voxel slot, the bottom encoder applies a shared 1D temporal CNN: $1 \to 64$ channels with kernel 5/padding 2, GELU, then $64 \to 64$ with kernel 3/padding 1 and GELU. It averages over the subwindow's four time positions, concatenates the 16 slot embeddings, and projects $16 \cdot 64 = 1,024$ values to width 192. The output has shape $[B, 12{,}358, 8, 192]$. Each of the three factorized encoders adds its own learned temporal-position table of shape $[1, 1, 8, 192]$.

Each transformer block is pre-layer-normalized attention with a residual connection, followed by a pre-layer-normalized feed-forward network $192 \to 768 \to 192$ with GELU and a residual connection. Attention has eight heads (head width 24) and dropout is zero. At a depth unit, spatial attention acts over all locations for each temporal token, followed by temporal attention over eight tokens at each location. Depths 3/2/1 therefore mean 6/4/2 attention blocks at chunk/parcel/network levels. These are full spatial attention operations, including 12,358 chunk positions; token compression does not imply a short encoder sequence or linear spatial-attention complexity.

The three quantized arrays, excluding batch, have shapes $[12{,}358, 8, 18]$, $[1{,}024, 8, 12]$, and $[18, 8, 6]$. Their last dimensions are sign bits. BSQ entropy groups have sizes 6/4/3, giving 3/3/2 groups at the respective levels. No lookup table of 2^{18} learned vectors is needed to define the bottom quantizer. Learned and soft cross-attention pools add query arrays of size $1,024 \times 192$ and 18×192, shared across all temporal tokens and subjects. Normalized queries and normalized key/value features enter eight-head attention; there is no query-residual bypass around the pool output.

For head h, the chunk-to-parcel pool uses weights:

$$\alpha_{pc}^{(h)}(w,\tau) = \mathrm{softmax}_c\left(\frac{(W_Q^{(h)}\,\mathrm{LN}(a_p))^\top W_K^{(h)}\,\mathrm{LN}(z_C[w,c,\tau])}{\sqrt{24}}\right) \tag{9}$$

where a_p is a learned query. The output is a projected concatenation of weighted value sums. Weights can change with subject, window, and time token through the input data, but no per-subject parameters are learned. Attention is dense and does not impose a one-parent assignment. There are no chunk/parcel/network spatial-position embeddings in this configuration. Within-chunk position enters through concatenation order, and temporal position enters through the position tables and CNN boundary effects.

Let $\pi(p)$ be parcel p's network, $\rho(c)$ chunk c's parcel, and $(c(v), k(v))$ voxel v's chunk and slot. For the temporal token covering time t, decoding is

$$\hat{y}_N[n,t] = H_N(r_N[n])_t, \tag{10}$$
$$\hat{y}_P[p,t] = \hat{y}_N[\pi(p),t] + H_P(r_P[p])_t, \tag{11}$$
$$\hat{y}_C[c,t,k] = \hat{y}_P[\rho(c),t] + H_C(r_C[c], \bar{y}_P[c], s_k)_t, \tag{12}$$
$$\hat{x}[v,t] = \hat{y}_C[c(v),t,k(v)] \tag{13}$$

Here $r_\ell = W_\ell^{\mathrm{post}} q_\ell$, $q_\ell = \mathrm{sign}(\bar{u}_\ell)/\sqrt{b_\ell}$, and $\bar{u}_\ell$ is the normalized linear projection of the encoder features. Temporal-token indices are suppressed.

After quantization, each level projects back to width 192. The network and parcel heads are linear maps $192 \to 4$. For chunk c and time token τ, define:

$$\bar{y}_P[c,\tau] = \frac{1}{4}\sum_{t=4\tau}^{4\tau+3} \hat{y}_P[\rho(c),t], \quad r'_C[c,\tau] = W_C^{\mathrm{post}} q_C[c,\tau] + W_{\mathrm{cond}}\bar{y}_P[c,\tau] \tag{14}$$

For each voxel slot k, a shared MLP $192 \to 192 \to 192 \to 4$ receives $r'_C[c,\tau] + s_k$ and produces the residual in Eq. (13); s_k is one of 16 learned 192-dimensional embeddings. Decoder heads have no attention layers. Parent broadcasts are the only cross-location coupling in decoding. No requirement forces voxel residuals to average to zero within parcels, or parcel residuals within networks. Direct coarse-head losses and coarse metrics on pooled voxel outputs are therefore distinct quantities.

B.2 The 415× configuration

HBAR-415 uses fixed pooling with one temporal token per 32-TR window at every level. Simply averaging the CNN features over all 32 TRs discarded much of their positional information in development. The final encoder instead learns eight weight vectors over the window positions, initialized to means of eight consecutive four-TR segments. The resulting $16 \cdot 64 \cdot 8 = 8,192$ features are projected to width 192 before the single bottom code. These eight continuous pooling slots are internal encoder features, not eight stored temporal codes.

The middle head is a three-layer MLP $192 \to 192 \to 192 \to 32$ with a learned 192-dimensional parcel-slot embedding added after quantization. The top head and bottom slot decoder also output 32 TRs per code. The code dimensions are 18/12/24, and the middle entropy group uses all 12 bits jointly. Its configuration value `bsq_mid_inv_temperature` is 0.3, versus 1 for the 52× configuration; this changes entropy regularization, not the hard sign code's bit count. The remaining encoder widths, attention heads, and level depths follow the 52× configuration. The final fixed models have 5,646,832 parameters at 52× and 7,313,174 at 415×; the compact code does not imply fewer model parameters.

The compact setting adds a direct middle-residual loss for stability:

$$10\ \mathrm{MSE}(H_P(r_P)[p,t],\ y_P[p,t] - y_N[\pi(p),t])\,. \tag{15}$$

The target uses the *true* network signal, not the predicted network output. The predicted parcel signal is detached when used as the bottom decoder's base and conditioning input; voxel-loss gradients

through these parent-prediction paths are stopped. This does not freeze the shared bottom encoder, which still feeds the higher levels. These changes in time representation, head capacity, and code allocation mean the 52×–415× comparison measures two complete trained systems rather than the isolated effect of temporal token count.

C Losses, optimization, and runtime

C.1 Loss reductions

For each tensor, MSE and MAE denote means over its batch, time, and spatial elements. Δ is a first difference along the TR axis and has $W - 1$ positions. The terms in Eq. (2) are:

$$\mathcal{L}_V = \mathrm{MSE}(\hat{x}, x), \qquad \mathcal{L}_{\Delta V} = \mathrm{MSE}(\Delta\hat{x}, \Delta x), \tag{16}$$

$$\mathcal{L}_C = \mathrm{MAE}(\hat{y}_C, y_C), \qquad \mathcal{L}_{\Delta C} = \mathrm{MAE}(\Delta\hat{y}_C, \Delta y_C), \tag{17}$$

$$\mathcal{L}_P = \mathrm{MSE}(\hat{y}_P, y_P), \qquad \mathcal{L}_N = \mathrm{MSE}(\hat{y}_N, y_N). \tag{18}$$

Voxel losses exclude padded slots because the voxel signal is assembled from valid indices. Both chunk L1 terms include padded slots, whose targets are zero (3.7% of chunk slots). They are not masked versions of the voxel losses. The parcel/network terms supervise the decoder's direct heads. Their targets are computed under the training grouping; the shuffled control consequently changes both aggregation and corresponding coarse supervision.

BSQ regularization sums a commitment penalty and entropy terms at each level:

$$\mathcal{L}_{\mathrm{BSQ},\ell} = \beta(s)\ \mathrm{MSE}(\bar{u}_\ell, \mathrm{sg}[q_\ell]) + \frac{\zeta(s)}{\eta_\ell}\big(H_{\mathrm{sample},\ell} - H_{\mathrm{code},\ell}\big). \tag{19}$$

Here s counts microbatches per rank and η_ℓ is the implementation's inverse-temperature setting. Both entropy coefficients inside the parentheses are 1. The per-sample entropy is computed analytically from the sign probabilities; code-distribution entropy uses the configured bit groups and aggregates token probabilities within the batch. It is a regularizer on soft assignments, not an entropy-coded payload estimate. Hard-code perplexity is a diagnostic of the empirical code distribution, not the number of bits needed to serialize the fixed-length representation.

Table 3: Final hierarchical training recipes. All listed weights are constant except the BSQ ramps described below. A dash means the term is inactive.

Setting	Fixed / learned / shuffled 52	Soft 52	Fixed 415
Epochs	10	10	10
Learning rate	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
Effective batch	32	32	32
$\lambda_V, \lambda_{\Delta V}$	$1, 0.5$	$1, 0.5$	$1, 0.5$
$\lambda_C, \lambda_{\Delta C}$	$10, 2$	$10, 2$	$10, 2$
λ_P, λ_N	$10, 10$	$10, 10$	$10, 10$
BSQ coefficient	0.03	0.03	0.03
Alignment coefficient (σ)	–	2	–
Direct middle-residual coefficient	–	–	10

C.2 Optimization and checkpoint selection

Hierarchical runs use AdamW, weight decay 0.01, gradient-norm clipping at 1, and bf16 automatic mixed precision on NVIDIA H100 80 GB GPUs. The learning rate is constant; there is no learning-rate warmup or scheduler. The final hierarchy runs use one GPU, batch size 4, and gradient accumulation of 8 (effective batch 32), with shuffled training batches. Commitment β ramps linearly from 0.01 to 0.1 and entropy weight ζ from 0.1 to 1 over the first 4,000 microbatches. On one GPU this is 500 optimizer updates. The final epoch, index 9, is used for every primary reconstruction and tokenizer readout. Seed labels 0, 1, and 2 identify independent training runs, not bootstrap draws.

C.3 Flat baseline settings

The flat BSQ baselines remove the parcel and network levels from HBAR and keep only its chunk-level encoder and decoder (same encoder architecture, 3-layer reconstruction head, and BSQ quantizer with L2 normalization and analytical entropy regularization). Each chunk token is quantized to 19 sign bits; because 19 is prime, the codebook-entropy term is computed bit-wise rather than over bit groups. Rates are matched to HBAR by the number of time tokens per chunk: flat BSQ-52 uses 8 tokens per chunk per 32-TR window (12,358 $\times$ 8 $\times$ 19 = 1,878,416 bits/window, vs. 1,878,720 for HBAR-52), and flat BSQ-415 pools the window to one token per chunk using the same temporal pooling as HBAR-415 (12,358 $\times$ 19 = 234,802 bits/window, vs. 235,164). The models have 2,965,719 (52$\times$) and 4,346,291 (415$\times$) parameters, vs. 5,646,832 and 7,313,174 for HBAR. The objective is HBAR's without the coarse-scale terms: voxel reconstruction MSE (weight 1), voxel temporal-difference MSE (0.5), chunk reconstruction L1 (10), chunk temporal-difference L1 (2), and the BSQ term (0.03); the parcel and network reconstruction losses do not exist at this depth. Training matches HBAR: 10 epochs, learning rate 2×10^{-4}, bf16 mixed precision, effective batch 32 (one GPU, batch 4, gradient accumulation 8), on the same 877 training subjects. The flat baselines are therefore rate- and recipe-matched to HBAR, but not parameter-matched, and differ from it by the absence of the parcel and network levels and their losses.

C.4 Cost and computational scope

Ten-epoch 52$\times$ hierarchical runs require approximately 36–38 H100 GPU-hours; representative peak PyTorch allocations are about 56–57 GB per GPU at batch 4, with roughly 70 GB reserved. Flat baselines use about 35 GPU-hours at $52\times$. These are indicative training costs, and several factors can affect elapsed time. $415\times$ runs completed in approximately 14 single-GPU hours. No matched final-model wall-clock-time or inference-speed comparison is claimed.

C.5 Loss-component ablations

To isolate the role of the three auxiliary terms in Eq. (2), we ran single-knob ablations during development: each changes one coefficient of the 52$\times$ recipe and holds everything else fixed, and each is compared with its own reference run at a matched epoch. These are single-seed development runs on validation subjects, not the final three-seed test-set models of Table 1, so the absolute values are not comparable across rows; the within-row comparison is the quantity of interest. Reconstruction is a voxel-level MSE, and atlas latent MSE is the squared distance between the model's pooled parcel/network features and the atlas-pooled targets, evaluated on the same windows. Code usage and perplexity are measured on the parcel-level (middle) quantizer.

Ablation	Recon. MSE↓	Atlas MSE↓	Usage↑	PPL↑
$\lambda_{\mathrm{vq}} : 0.03 \to 0.003$	0.668 / 0.670	–	0.100 / 0.208	24.7 / 243.5
$\lambda_{\Delta V}, \lambda_{\Delta C} : (0.5, 2) \to (0, 0)$	0.673 / 0.810	20.8 / 57.8	–	–
$\sigma : 2 \to 0.5$	0.779 / 0.733	40.7 / 66.0	–	–

Table 4: Single-knob ablations at $52\times$. Values are reference/ablated at matched epochs. The first two rows use fixed-hierarchy references; the third uses a soft-hierarchy reference. Rows are not directly comparable. Usage is the fraction of 2^{12} parcel codes observed; PPL denotes hard-code perplexity. Dashes indicate unmeasured quantities.

Reducing the quantization weight tenfold leaves reconstruction essentially unchanged but roughly doubles parcel-level code usage and raises perplexity by an order of magnitude, so this term mainly governs code utilization rather than fidelity. Removing the temporal-difference terms is clearly harmful: reconstruction MSE rises by 20% and the alignment distance to atlas-pooled features nearly triples, so $\mathcal{L}_\Delta$ shapes the latent structure as well as temporal fidelity. Weakening the alignment weight from $\sigma = 2$ to 0.5 exposes the intended trade-off: reconstruction MSE falls by 6% while atlas latent MSE rises by 62%, i.e., the soft hierarchy buys reconstruction by moving away from the atlas, which is the tunable behavior it was designed to provide.

D Complete bit accounting

The numerator is the preprocessed masked signal: 32 TRs, 190,468 unique voxels, 16 bits/value. It is not the full rectangular image, raw scanner data, or a float32 array. Padding is not counted as input data. Every stored latent code, including empty parcel slots and the unassigned network group, is counted in the denominator.

$$B_{52} = 12{,}358 \cdot 8 \cdot 18 + 1{,}024 \cdot 8 \cdot 12 + 18 \cdot 8 \cdot 6 \tag{20}$$
$$= 1{,}779{,}552 + 98{,}304 + 864 = 1{,}878{,}720,$$
$$B_{415} = 12{,}358 \cdot 18 + 1{,}024 \cdot 12 + 18 \cdot 24 \tag{21}$$
$$= 222{,}444 + 12{,}288 + 432 = 235{,}164.$$

Flat BSQ needs 19 bits per chunk, giving 234,802 bits/window. Flat models at $415\times/52\times$ emit 1/8 codes per chunk per 32-TR window.

Table 5: Compression accounting. "Window ratio" uses one 32-TR window. Scan ratios use a 1,200-TR scan: 38 nonoverlapping padded windows (stride 32) or 74 overlapping windows (stride 16).

Model	Bits/window	Window ratio	Bits/sample	Tiled scan	Stride-16 scan
HBAR-52	1,878,720	51.91	0.30824	51.22	26.30
Flat BSQ-52	1,878,416	51.92	0.30819	51.23	26.31
HBAR-415	235,164	414.69	0.03858	409.23	210.15
Flat BSQ-415	234,802	415.33	0.03852	409.86	210.47
Flat RVQ-415	234,802	415.33	0.03852	409.86	210.47
MS-VQVAE-415	234,732	415.45	0.03851	409.98	210.53

The scan calculation is $R_{\text{scan}} = 1,200 \cdot V \cdot 16/(n_w B_q)$. The rates assume ideal fixed-length bit packing with no entropy coding. For M windows, shared storage S bits, and per-window/container overhead H bits, a model-inclusive ratio is

$$R_{\text{total}}(M) = \frac{MB_x}{M(B_q + H) + S} \tag{22}$$

The recorded layout arrays occupy 1,041,946 bytes before container overhead; a full float32 weight array occupies $4P$ bytes for P parameters (about 22.6 MB for fixed $52\times$ and 29.3 MB for fixed $415\times$). Which encoder/decoder weights are distributed changes S. Finally, the 36- or 54-dimensional probe vector is a mean-pooled readout of many discrete tokens. It is not the complete encoded window and is not sufficient for the reconstruction decoder. Comparing 36 readout coordinates with 1,024 DiFuMo coordinates is a dimensionality comparison, not a 28-fold codec advantage. Quantization bounds the code alphabet; it does not guarantee predictive sufficiency or privacy.

E Reconstruction metrics and statistical analysis

For each level, a window's correlation is Pearson's r between vectors formed by flattening time and units. It is not the mean of independent per-voxel temporal correlations. Window scores are averaged within each subject, then equally across subjects. MSE is likewise computed per window then averaged by subject. There is no Fisher z transform. With $J = 16$ windows for subject i, the reported mean of metric m is

$$\bar{m} = \frac{1}{109} \sum_{i=1}^{109} \left(\frac{1}{16} \sum_{j=1}^{16} m_{ij} \right) \tag{23}$$

Windows from the same subject are not treated as independent observations for confidence intervals.

All coarse reconstructions are formed from valid voxel slots using Eq. (8), not the model's direct coarse decoder heads. In particular, each parcel prediction sums valid decoded voxels and divides by that parcel's valid voxel count. A reconstruction of the true chunk inputs must reproduce the dataset's parcel/network targets within float16 tolerance. A 300-window oracle check gave parcel MSE $3.1 \cdot 10^{-9}$ and network MSE $8.2 \cdot 10^{-10}$, with least-squares scales 1.000000 and 1.000002. The evaluator checks MSE below 10^{-7}, maximum absolute error below $4 \cdot 10^{-3}$, scale error below 10^{-4}, and zero targets for empty parcels. A correlation-only check would not detect a multiplicative scale error.

Parcel MSE averages over all 1,024 slots, including 14 zero-error empty parcels. Multiplying by $1,024/1,010 = 1.01386$ converts it to the mean over nonempty parcels. Correlations include the same zero slots for every model; they cannot be converted by that MSE factor. The final 18-group network metric includes the unassigned group. Coarse-scale correlation is consequently atlas- and convention-dependent.

Reconstruction CIs use 10,000 subject-bootstrap resamples; paired contrasts resample the same subjects for both models. The accompanying two-sided paired sign-flip test uses 10,000 draws and a plus-one correction, with minimum attainable $p = 1/10,001$. Many contrasts, including those between fixed-model seeds, reach this floor. They quantify consistency across these subjects for trained checkpoints. We emphasize effect sizes, paired CIs, and separately reported seed spread.

Paired tests show HBAR better at parcel scale than every comparator at every seed. At $415\times$, MS-VQVAE achieves the best voxel reconstruction across all seeds, while HBAR dominates for parcel and network reconstruction. At $52\times$, atlas-consistent grouping improved parcel correlation relative to shuffled grouping in all three seed pairs ($\Delta r = 0.029$–0.040). Network correlation, by contrast, was similar on average, and the direction of the difference varied across seeds ($+0.013$, -0.006, and -0.008).

F Downstream features, baselines, and probes

Linear probes use the same stride-64 windows for every method: 20,808 windows total, of which 16,651 belong to training subjects. This differs intentionally from the denser stride-16 training index and from the 16-window-per-subject reconstruction sample. A window's hierarchical feature concatenates the spatial/temporal means of its three post-quantization sign vectors; a subject's feature averages these window vectors. Dimensions are $18 + 12 + 6 = 36$ for $52\times$ and $18 + 12 + 24 = 54$ for $415\times$. Flat BSQ has 19 coordinates. The readout discards position and order and need not exploit all information present in the stored sequence.

Autoregressive readout. The AR model uses only finest-level codes, excluding MS-VQVAE's top-level tokens. HBAR's 18-bit codes are split into three 6-bit IDs (vocabulary 64); flat BSQ's 19-bit codes into 10- and 9-bit IDs (vocabulary 1,536); and RVQ/MS-VQVAE use their two codebook indices with offsets (vocabularies 1,536/1,024). IDs are ordered by temporal slot, chunk, and bit group. Non-overlapping 32-TR windows are concatenated chronologically within subjects. Each Transformer position embeds 64 consecutive IDs as a 256-dimensional vector (64×4), with learned absolute positional embeddings. A 4-layer, 4-head causal pre-norm Transformer (width 256, FFN 1,024, GELU, dropout 0.1) predicts all 64 IDs of the next patch using separate softmax heads and mean cross-entropy. Samples span four windows (128 TRs), truncated to complete patches. Sequence lengths and parameter counts (4.9M–31.7M) are given in Table 6.

AR models are trained on 877 training subjects for 10 epochs (seed 0) using AdamW (LR 5×10^{-5}, weight decay 0.01, batch size 8, gradient clipping 1.0, bf16), with 200 warmup steps followed by cosine decay to 20% of peak LR after the first epoch. Four-window samples are shuffled each epoch; the final checkpoint is used without validation-based selection. The frozen model processes each subject's sequence in segments of up to four windows; final-layer hidden states are averaged within and across segments to yield a 256-dimensional subject representation. These features are evaluated using the same standardized, balanced L2 logistic probe as above, with five-fold cross-validation for C and 10,000 subject-bootstrap resamples. Probes are fitted on 877 training subjects and evaluated on 109 held-out test subjects. Although training-subject features come from sequences seen during AR training, phenotype labels are never used by the AR model. Table 2 reports tokenizer and AR seed 0.

Table 6: AR input sequences per tokenizer. IDs/window counts finest-level IDs per 32-TR window; positions are 64-ID patches per four-window sample. Parameter counts include embeddings and the output head.

Tokenizer	Vocab.	IDs / window	Positions	AR params
HBAR-52 (fixed, learned, soft, shuffled)	64	296,592	18,537	9.02M
Flat BSQ-52	1,536	197,728	12,358	31.66M
HBAR-415 (fixed)	64	37,074	2,317	4.87M
Flat BSQ-415 / Flat RVQ-415	1,536	24,716	1,544	28.89M
MS-VQVAE-415	1,024	24,716	1,544	20.47M

The atlas baselines use the same preprocessed signal and the same 20,808 stride-64 windows. DiFuMo time means average each of the 1,024 parcel series over the window; Yeo time means do the same for 18 groups (the 17 Yeo networks plus one group of parcels without a Yeo assignment). The parcel-series PCA baseline flattens each $32 \times 1{,}024$ window and projects it to 256 coordinates. Parcel-FC computes Pearson correlations between parcel series within each window, vectorizes the upper triangle ($1{,}024 \cdot 1{,}023/2 = 523{,}776$ entries), and projects it to 256 coordinates. Both PCA bases are fit only to the 16,651 training-subject windows. Features are then averaged by subject. Neither validation nor test windows are used to fit PCA or feature standardization.

For classification, a training-set standardizer precedes an L2-regularized logistic regression with balanced class weights. Regularization C is selected by five-fold inner cross-validation on training subjects from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. We report AUROC for sex and macro-averaged one-vs-rest AUROC for the three age bins. There are 877 training and 109 test sex labels; class counts are 480/397 in training and 58/51 in test. Age has 865 training labels and all 109 test labels, with test bin counts 19/59/31. Numeric representatives 23, 28, and 33 encode the age categories; this is age-bin classification, not fine-grained continuous-age prediction.

Single-model CIs are 95% percentile intervals from 10,000 subject-bootstrap resamples of the test set, conditional on the trained tokenizer and probe; they reflect subject sampling, not training-seed variance. Paired differences use 10,000 bootstrap resamples with the same subject indices for both prediction sets, and a two-sided permutation test that randomly swaps the two models' predicted probabilities within each subject (10,000 permutations, add-one smoothed). We do not interpret nonsignificance as equivalence.

G Frequency-resolved reconstruction

For each 32-TR window, the evaluator uses an undetrended raw periodogram to compare Fourier power of the reconstruction residual with input power in broad frequency bands. For band $\mathcal{B}$, the error ratio is of the form

$$E_{\mathcal{B}} = \frac{\sum_v \sum_{f \in \mathcal{B}} |\mathcal{F}(\hat{x} - x)[v, f]|^2}{\sum_v \sum_{f \in \mathcal{B}} |\mathcal{F}(x)[v, f]|^2} \tag{24}$$

The exported window statistics are averaged within subjects and then across subjects, as for reconstruction. Zero indicates perfect reconstruction in the band; one indicates residual power equal to input power and need not imply statistical independence. Values can exceed one. It is not an explained-variance statistic and the bands are not validated partitions into neuronal, respiratory, and noise components.

With $W = 32$ and TR 0.72 s, the frequency spacing is $1/(32 \cdot 0.72) = 0.0434$ Hz and Nyquist is 0.6944 Hz. The 0–0.02 Hz band contains only DC; the 0.02–0.10 and 0.10–0.20 Hz bands contain two positive-frequency bins each, 0.20–0.35 Hz four and 0.35–0.70 Hz eight. At $52\times$, all models recover part of the within-window dynamics below 0.20 Hz, and HBAR-52 is comparable to the other $52\times$ variants. At $415\times$, every model has one temporal token per chunk and window, and its residual carries essentially all input power above 0.10 Hz (ratios ≥ 0.987). Differences are confined to the two 0.02–0.10 Hz bins, where HBAR-415 and flat RVQ-415 (0.94) retain slightly more than flat BSQ-415 (0.97) and MS-VQVAE (0.99), and to the DC bin, where MS-VQVAE is lowest. We therefore avoid

Table 7: Final test voxel spectral-error/input-power ratios (lower is better). Mean $\pm$ SD over three training seeds, rounded to two decimals. The DC column contains only the zero-frequency bin at this window length. SDs shown as 0.00 are below 0.005.

Model	DC	0.02–0.10	0.10–0.20	0.20–0.35	0.35–0.70 Hz
HBAR-52 (fixed)	0.38 ± 0.02	0.66 ± 0.02	0.85 ± 0.01	0.94 ± 0.01	0.90 ± 0.00
Learned pooling (52×)	0.42 ± 0.02	0.68 ± 0.02	0.85 ± 0.01	0.92 ± 0.00	0.88 ± 0.01
Soft pooling (52×)	0.45 ± 0.23	0.64 ± 0.08	0.81 ± 0.04	0.94 ± 0.01	0.99 ± 0.01
Shuffled grouping (52×)	0.40 ± 0.02	0.66 ± 0.02	0.84 ± 0.01	0.92 ± 0.01	0.89 ± 0.01
Flat BSQ-52	0.37 ± 0.03	0.68 ± 0.01	0.86 ± 0.00	0.93 ± 0.01	0.88 ± 0.02
HBAR-415 (fixed)	0.41 ± 0.01	0.94 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Flat BSQ-415	0.42 ± 0.01	0.97 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Flat RVQ-415	0.39 ± 0.03	0.94 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
MS-VQVAE-415	0.29 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

claims about resolving narrow slow bands or preserving complete resting-state dynamics. Strong flattened coarse correlations can coexist with substantial within-window temporal error. A long-scan spectral or connectivity-preservation study would answer a different question and is not established by this table.

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 describe the proposed hierarchical tokenization framework, its fMRI instantiation, compression/reconstruction goals, downstream evaluation, and key empirical findings.

   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: The discussion includes a limitations paragraph.

   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: [N/A]

Justification: The paper does not present formal theoretical results or proofs.

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: The paper describes the dataset, splits, preprocessing, model variants, baselines, losses, quantization details, and evaluation metrics needed to reproduce the main experiments.

   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: The paper uses Human Connectome Project (HCP) data that can be obtained by researchers, and the code repository is linked on the first page.

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: The Experiments section specifies HCP-YA data, subject-level splits, preprocessing, training epochs, learning rate, temporal window length/stride, baselines, metrics, and model settings.

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: Table 1 reports reconstruction correlations as mean ± sample standard deviation across three independent training seeds. Paired comparisons between HBAR and each comparator use subject-level bootstrap confidence intervals (10,000 resamples) and two-sided paired sign-flip permutation tests; the procedure is described in Appendix E and the outcomes are summarized in the Table 1 caption (HBAR exceeds every comparator at parcel scale at every seed; fixed vs. shuffled grouping $\Delta r = 0.029$–0.040). Windows from the same subject are not treated as independent observations. Table 2 reports 95% subject-bootstrap confidence intervals for every tokenizer result, conditional on the fitted tokenizer and probe (seed 0).

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: The paper reports model size, per-window runtime, scan-level runtime, and GPU type for the proposed model.

   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: The work uses established neuroimaging datasets and focuses on representation learning methodology. We do not identify individuals or make clinical deployment claims.

   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: [No]

Justification: We do not include a separate broader impacts discussion due to space constraints. Potential positive impacts include more efficient and expressive representation learning for large-scale neuroimaging data. Potential risks include privacy, fairness, and misuse concerns if demographic or clinical prediction systems are deployed without sufficient validation or oversight. These considerations are relevant to many machine learning systems operating on sensitive biomedical data.

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 does not release high-risk generative models, scraped datasets, or models with obvious misuse risk. The neuroimaging data used are from controlled-access/public scientific datasets subject to their original data-use terms.

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: The paper cites the HCP-YA dataset, preprocessing tools, atlases, and relevant model/tokenization methods. Dataset access and usage follow the terms of the original data providers.

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: The paper does not introduce a new dataset. Accompanying documentation for the code describes training and usage.

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 new crowdsourcing experiments or new human-subject data collection.

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: The study uses previously collected HCP-YA data rather than collecting new human-subject data. The original dataset was collected under its own ethical approvals and data-use procedures.

    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: LLMs are not used as part of the core methodology, experiments, or scientific contribution of this work.

    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.