FlatClip: Reusing Image Foundation Models for fMRI Representation Learning via Cortical Flatmaps

Mo Wang[1,2] **Wenhao Ye**[1,3] **Zihan Ning**[1] **Jiayu Zuo**[1]
Junfeng Xia[1] **Hongkai Wen**[4,2,*] **Quanying Liu**[1,*]

[1]Southern University of Science and Technology, Shenzhen, China
[2]University of Warwick, Coventry, United Kingdom
[3]Shenzhen University, Shenzhen, China
[4]Fudan University, Shanghai, China

[*]Co-corresponding authors:
hongkai.wen@gmail.com, liuqy@sustech.edu.cn

Abstract

Can the spatial organization of cortical activity help transfer frozen image representations to fMRI prediction? We introduce FlatClip, which renders cortical activity as anatomy-linked flatmaps and extracts features with a frozen SigLIP2 encoder. Only a lightweight downstream probe is trained, without fMRI-specific encoder pretraining or fine-tuning. Across four resting-state benchmarks, FlatClip outperforms the evaluated ROI-level baselines on HCP and ADNI, but is weaker on PPMI and below the strongest voxel-level models overall. On NSD visual-fMRI decoding, restricting inputs to visual or NSD-provided task-active cortex improves recognition over whole-cortex inputs. Spatial perturbations reduce performance with both retrained and fixed readouts, and anatomy-linked arrangements outperform cortical-pixel permutations across three colormaps. Together, these results demonstrate a practical route for reusing image-pretrained features in fMRI and highlight cortical arrangement and input-region selection as important design choices. Code is available here.

1 Introduction

Foundation models have recently become an important direction for fMRI representation learning. Instead of training a separate model for each dataset, task, or cognitive paradigm, fMRI foundation models aim to learn reusable brain representations from large-scale unlabeled neuroimaging data. Such models can reduce task-specific engineering and may improve transfer across heterogeneous downstream analyses [1, 2, 3]. However, fMRI representation learning depends strongly on how brain activity is represented. Unlike natural images, fMRI is a high-dimensional 4D spatiotemporal signal with both spatial organization and temporal dynamics. Choosing the right spatial scale is therefore a central modeling decision (Fig. 1).

Most existing fMRI foundation models operate at either ROI level [1, 2, 4, 5, 6] or voxel level [3, 7, 8, 9, 10, 11]. ROI- and connectome-based models summarize voxel signals within atlas-defined regions and then model regional time series or inter-regional relationships. These representations are efficient and scalable, but they are coarse: they depend on parcellation choices and may discard fine-grained cortical geometry [12, 13]. Voxel-based or volumetric models preserve more spatial detail and can capture local structure more directly, but they require specialized 3D/4D architectures, substantial computation, and storage of high-dimensional fMRI volumes. This trade-off raises a practical question: how effectively can a frozen image encoder use the spatial organization of cortical activity? Pang et al. showed that macroscopic fMRI activity can be parsimoniously explained by geometric

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

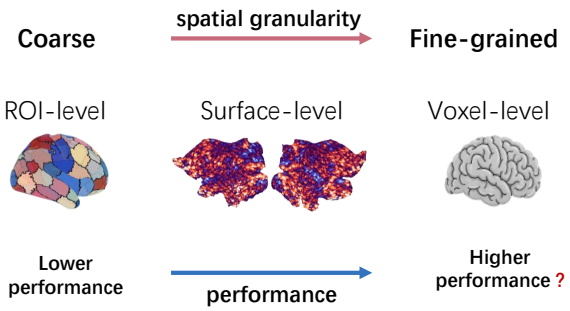


Figure 1: **Motivation.** fMRI representations vary in spatial granularity. FlatClip studies surface-level cortical flatmaps as a geometry-aware middle ground.

eigenmodes derived from brain shape, and that these modes explain spontaneous and task-evoked activity more effectively than connectome-derived modes [14]. This motivates representations that retain the spatial organization of cortical activity rather than reducing all signals to coarse regional averages.

We introduce **FlatClip**, a surface-level baseline that renders cortical activity as flatmap sequences, extracts features with a frozen SigLIP2 image encoder, and trains a lightweight downstream probe. The flatmaps provide a consistent anatomy-linked 2D+time layout between coarse ROI summaries and native volumes. Because the encoder requires no fMRI-specific pretraining or fine-tuning, FlatClip tests the predictive utility of cortical layouts using features learned outside neuroimaging.

We first evaluate FlatClip on four resting-state fMRI benchmarks, where subject-level prediction has no explicit stimulus region. Under a frozen-backbone, lightweight-probe protocol, FlatClip exceeds most ROI-level baselines while requiring no fMRI-specific encoder pretraining, although the strongest voxel-level models remain higher overall. We then evaluate visual-fMRI decoding on NSD COCO80, a stimulus-evoked setting in which the relevant cortical region is more explicit; NSD is a large-scale 7T fMRI dataset collected while subjects viewed natural images [15]. Performance improves from whole-cortex input to HCP-MMP visual cortex and NSD-provided subject-level task-active regions, highlighting the importance of cortical input-region selection for visual decoding.

Motivated by the resting-state and visual-fMRI observations, we conduct control experiments to test whether the gains are associated with spatial organization and spatial granularity. Geometry-disruption controls show that block-shuffling, cortical-pixel permutation, and FC heatmaps reduce performance relative to real flatmaps, indicating that the benefit is not merely due to image-like formatting. In a separate voxel-based comparison, adaptive patching improves HCP prediction, while its effect on AD diagnosis varies across metrics. Together, these results support cortical spatial organization and task-relevant input selection as useful design choices for frozen-feature fMRI prediction. Our main findings are summarized as follows:

- We introduce FlatClip, a surface-level approach that reuses a frozen SigLIP2 image encoder for fMRI prediction through anatomy-linked cortical flatmaps, training only a lightweight downstream probe.
- Across four resting-state benchmarks, FlatClip outperforms the evaluated ROI-level baselines on HCP and ADNI, with task-dependent performance relative to voxel- and surface-based models.
- We show that cortical spatial arrangement and input-region selection affect the predictive utility of frozen image features, using spatial-perturbation controls, three colormaps, and comparisons across cortical masks.

2 Related Work

fMRI foundation models across representation scales. ROI- or atlas-based methods reduce voxel-level signals into region-wise time series or functional-connectivity graphs, and models such as BrainLM, Brain-JEPA, BrainMASS, LCM, and related graph-based approaches learn from ROI

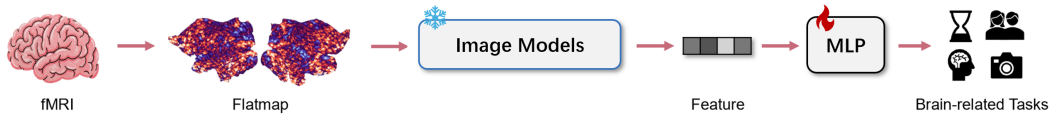


Figure 2: **FlatClip pipeline.** Preprocessed cortical signals are rendered in a fixed surface layout and encoded by frozen SigLIP2. Resting-state prediction averages 40 frame features into one subject feature; NSD prediction uses a stimulus-level GLM response map. Only the downstream MLP is trained.

tokens, temporal dynamics, or graph-structured connectivity [1, 2, 4, 5, 6, 16]. At the other end of the scale, voxel- or volume-based models operate directly on 3D or 4D fMRI and preserve more fine-grained spatial structure [17, 16, 7, 3, 8, 9, 10, 11, 18]. Our evaluations compare these pretrained representations with surface-level features under a frozen-backbone readout protocol.

Geometry-aware cortical surface and flatmap representations. Geometric and spectral analyses motivate retaining the spatial layout of cortical activity [14, 19]. Pycortex is a standard toolbox for projecting volumetric neuroimaging data onto cortical surfaces and generating flattened cortical visualizations [20]. Recent work has explored image-like or flatmap-based interfaces for fMRI, including visual decoding from fMRI signals and spatiotemporal masked-autoencoding on fMRI flatmap videos [21, 22]. FlatClip adopts this existing surface-rendering interface for frozen image-feature transfer.

Mesh- and sphere-based cortical models. Cortical learning also has a substantial literature beyond ROI and volumetric inputs. MoNet learns local filters on graphs and manifolds [23], while Spherical U-Net defines convolution and pooling on spherical cortical meshes [24]. SiT represents cortical measurements as triangular spherical patches processed by a Transformer [25]. X-SiT and super-vertex vision Transformers further investigate interpretable and multiscale surface representations for dementia analysis [26, 27]. CortexMAE-F instead pretrains a spatiotemporal encoder on fMRI flatmaps [22]. These approaches motivate our surface-model comparisons. FlatClip uses an existing image-pretrained encoder without fMRI-specific backbone training, offering a complementary reference for evaluating cortical representations.

Image-pretrained encoders as out-of-domain frozen baselines. Large-scale image-pretrained models have become reusable feature extractors for downstream vision tasks. Representative approaches include masked image modeling, contrastive image-text pretraining, self-distillation, and sigmoid-loss image–text pretraining, including MAE, CLIP, DINOv2, and SigLIP2 [28, 29, 30, 31]. SigLIP2 extends the SigLIP family with captioning-based pretraining, self-supervised losses, and online data curation, improving visual representation transfer and dense-feature performance [31].

3 Method

3.1 Overview

Figure 2 summarizes the three stages: cortical flatmap construction, frozen feature extraction, and downstream prediction. Resting-state inputs are sequences of time-point maps; NSD inputs are stimulus-level GLM response maps. The following sections specify their construction, pooling, and prediction losses.

3.2 Geometry-aware Flatmap Sequence Adapter

For each subject i, we construct a sequence of cortical flatmap frames

$$\mathcal{F}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,T}\},$$

where each frame

$$x_{i,t} \in \mathbb{R}^{H \times W \times C}$$

is an RGB rendering ($C = 3$) of the cortical activity at one time point. Starting from preprocessed volumetric fMRI, activity is projected onto cortical vertices; surface inputs already in fsLR space enter at this step. Pycortex rasterizes the vertex values onto a deterministic flatmap layout. We normalize and color-map the scalar values, apply the cortex or task-region mask, crop the rendered map, and composite the background onto RGB. Native cropped dimensions are reported in Appendix D.1; the SigLIP2-NaFlex processor then prepares the encoder input. By stacking frames along the temporal dimension, the sequence can be represented as a flatmap tensor

$$\mathbf{X}_i = \mathrm{Stack}_{t=1}^{T}(x_{i,t}) \in \mathbb{R}^{T\times H\times W\times C}.$$

For resting-state fMRI, t indexes time points and the sequence length T is fixed within each dataset. For visual-task fMRI, each sample corresponds to a stimulus-level response map estimated by a general linear model (GLM), where each response map is associated with one viewed image.

For NSD visual-fMRI decoding, we evaluate three cortical input regions: whole cortex, HCP-MMP visual cortex, and the NSD-provided subject-level task-active region, denoted as NSDgeneral. The HCP-MMP visual cortex mask is defined from visual ROIs in the HCP-MMP atlas in 32k fsLR space [32]. The NSDgeneral mask is provided by NSD and represents subject-level task-active cortex [15]. These regions are rendered as flatmaps and processed using the SigLIP2-NaFlex input processor.

3.3 Frozen SigLIP2 Encoder and Feature Construction

Let f_θ denote the frozen SigLIP2 image encoder. For each flatmap frame, we extract one global image representation:

$$z_{i,t} = f_\theta^{\mathrm{global}}(x_{i,t}) \in \mathbb{R}^d,$$

where $f_\theta^{\mathrm{global}}(\cdot)$ denotes the global representation returned by the encoder. The encoder parameters θ are fixed throughout all experiments, and no gradient is propagated into the image encoder.

For resting-state fMRI, each subject has T flatmap frames. We aggregate the frame-level global representations by average temporal pooling:

$$h_i = \frac{1}{T}\sum_{t=1}^{T} z_{i,t} \in \mathbb{R}^d.$$

In our main resting-state experiments, we use $T = 40$ frames for each subject and $d = 768$.

For NSD, each sample is a stimulus-level GLM response map and uses patch-feature pooling rather than the encoder's global output. Let $P_{s,j}$ denote the valid patch features returned by the frozen vision encoder, arranged according to the processor's spatial shape. We remove trailing padding tokens and adaptively average-pool this feature grid to 16×16:

$$Q_{s,j} = \mathrm{AdaptiveAvgPool}_{16\times16}(P_{s,j}), \qquad h_{s,j} = \frac{1}{256}\sum_{k=1}^{256} Q_{s,j,k} \in \mathbb{R}^{768}.$$

The NaFlex processor allows up to 1,024 input patches in this extraction pipeline. The 256 pooled tokens are a post-encoder representation; their mean forms the main NSD feature, while token-rich variants retain all 256. Both tasks therefore provide a 768-dimensional main feature, with different pooling operations.

3.4 Downstream Prediction

For each downstream task, we train a lightweight probe g_ϕ on top of the frozen SigLIP2 feature. Only the probe parameters ϕ are optimized.

The main resting-state probe is an MLP with two hidden layers, $768 \to 256 \to 256 \to C$, using BatchNorm, GELU, and dropout of 0.2. Features are standardized using training-set statistics. We use class-weighted cross-entropy, AdamW, a cosine learning-rate schedule, and seeds 42, 43, and 44; checkpoints are selected by validation weighted F1. Linear-probe sensitivity results are reported in Appendix D.5. NSD uses a task-specific multilabel MLP and class-weighted binary cross-entropy.

For single-label resting-state classification, let

$$\mathcal{D}_{\text{rest}} = \{(h_i, y_i)\}_{i=1}^{N_{\text{sub}}}$$

denote the subject-level training set. The probe predicts

$$\hat{y}_i = g_\phi(h_i),$$

and is trained with class-weighted cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N_{\text{sub}}} \sum_{i=1}^{N_{\text{sub}}} \sum_{c=1}^{C} \alpha_c \mathbf{1}[y_i = c] \log p_\phi(y_i = c \mid h_i),$$

where α_c is the class weight estimated from training labels, C is the number of classes, and N_{sub} is the number of training subjects.

For visual-fMRI recognition, let $\mathcal{D}_{\text{vis}}$ denote the set of subject–stimulus pairs (s, j), and let

$$N_{\text{vis}} = |\mathcal{D}_{\text{vis}}|.$$

Each pair has a feature $h_{s,j}$ and a COCO80 multi-hot target

$$y_j \in \{0, 1\}^C, \quad C = 80,$$

where y_j depends only on the stimulus image and is shared across subjects. The lightweight probe outputs

$$o_{s,j} = g_\phi(h_{s,j}) \in \mathbb{R}^C,$$

and is trained with class-weighted binary cross-entropy:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N_{\text{vis}}} \sum_{(s,j)\in\mathcal{D}_{\text{vis}}} \sum_{c=1}^{C} \left[w_c y_{j,c} \log \sigma(o_{s,j,c}) + (1 - y_{j,c}) \log(1 - \sigma(o_{s,j,c})) \right],$$

where $\sigma(\cdot)$ is the sigmoid function and w_c is the positive-class weight computed from the training split.

4 Evaluating Geometry-Aware Surface Representations

4.1 Resting-state fMRI Benchmarks

We first evaluate FlatClip on four subject-level resting-state fMRI benchmarks: HCP sex classification, PPMI diagnosis prediction, ADNI MCI vs. CN classification, and ADNI AD vs. CN classification [33, 34, 35]. These benchmarks cover demographic prediction in healthy adults and diagnostic-label prediction across neurodegenerative cohorts. All methods are evaluated on the same subject-level splits with frozen representations and lightweight downstream heads. Because the models have different native input requirements, the input construction follows each model's interface, while the readout protocol and splits are matched. This provides a matched readout protocol for comparing frozen representations. Voxel-level models were evaluated using 40 input frames, following their model-specific requirements; FlatClip used the same temporal window for consistency. In contrast, ROI/FC-based models used approximately 200 frames for feature computation, according to the requirements of their respective pipelines. We report Accuracy and weighted F1 against general-purpose fMRI foundation-model baselines, including ROI-level models (Brain-LM [2], Brain-MASS [4], LCM [6], Brain-Harmony [36]) and voxel-level models (SwiFT [7], NeuroSTORM [9], Omni-fMRI [3]). Details about datasets, task design, ablation of time frame, and model-specific settings are provided in Appendix A.

As shown in Table 1, FlatClip achieves competitive performance on HCP and ADNI without fMRI-specific encoder pretraining or end-to-end fine-tuning. It outperforms most ROI-level baselines while remaining below the strongest voxel-level models, consistent with its role as a surface-level middle-ground representation. This pattern establishes cortical flatmaps as a competitive surface-level representation under the evaluated frozen-backbone protocols. FlatClip is relatively weaker on PPMI, which may reflect a limitation of the current cortical-only input. Parkinson's disease involves subcortical and brainstem systems that are not fully captured by cortical surface flatmaps [37, 38].

Table 1: Performance on HCP, ADNI (MCI), ADNI (AD), and PPMI with Accuracy/weighted F1. Darker cell color indicates stronger performance after column-wise min–max normalization.

Model \ Dataset	HCP *Sex Classif.*		ADNI (MCI) *Diagnosis*		ADNI (AD) *Diagnosis*		PPMI *PD Diagnosis*	
	ACC ↑	**wF1 ↑**	**ACC ↑**	**wF1 ↑**	**ACC ↑**	**wF1 ↑**	**ACC ↑**	**wF1 ↑**
Brain-LM	64.12±2.89	64.14±2.84	45.86±5.18	45.78±5.12	66.09±2.22	66.19±2.11	62.50±4.89	57.15±6.62
BrainMASS	68.24±1.29	68.18±1.36	53.79±5.27	53.89±5.18	60.87±5.32	60.91±5.44	52.50±3.22	52.95±2.89
LCM	67.94±3.23	67.93±3.26	53.10±3.99	50.89±4.39	64.35±3.53	64.05±3.82	58.89±3.75	49.65±4.92
Brain Harmony	67.84±2.56	67.73±2.57	58.33±3.86	58.28±4.01	70.90±5.28	71.04±5.35	60.55±5.17	57.51±4.98
FlatClip	82.06±2.05	82.04±2.05	61.11±1.48	60.75±1.16	75.46±2.12	75.41±2.11	54.86±1.96	55.72±1.63
SwiFT	86.03±1.39	84.91±1.76	69.78±4.63	69.31±5.23	74.69±2.86	74.70±2.63	61.55±2.52	58.23±4.20
NeuroSTORM	85.72±1.44	84.36±2.00	57.25±3.18	56.76±3.21	71.64±0.66	65.35±2.06	66.78±0.51	57.37±0.86
Omni-fMRI	92.86±0.63	92.08±0.70	63.40±2.55	56.67±9.60	84.26±0.87	83.65±1.37	68.13±0.60	63.65±2.04

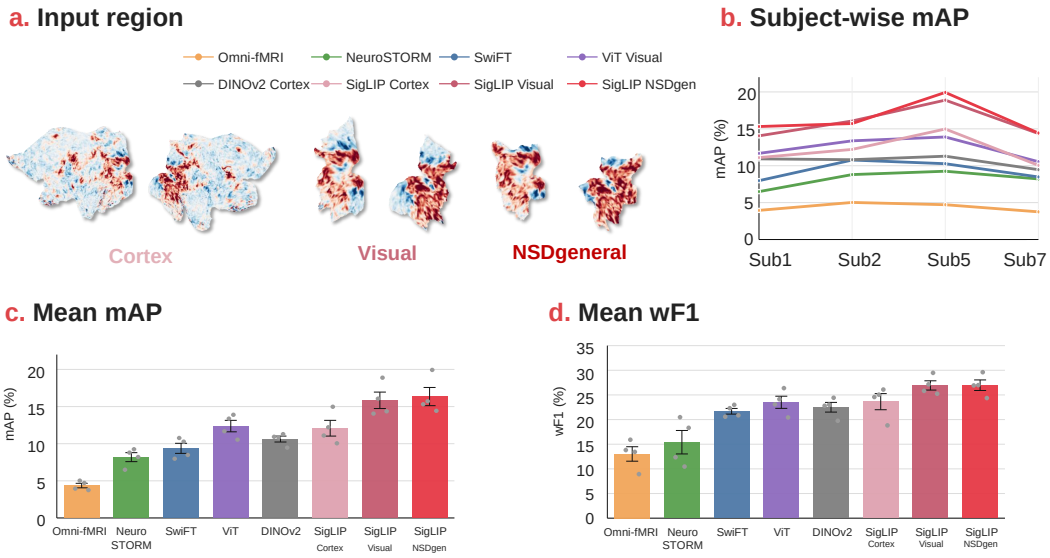


Figure 3: Visual-fMRI COCO80 multi-label recognition on NSD. (a) Illustration of the cortical input regions used by FlatClip: whole cortex, HCP-MMP visual cortex, and the NSD-provided subject-level task-active region (NSDgeneral). (b) Subject-wise mAP across four NSD subjects for Omni-fMRI, NeuroSTORM, SwiFT and FlatClip variants. (c,d) Mean mAP and weighted F1 across subjects. All FlatClip variants use one global representation per GLM response map and the same downstream MLP classifier. Dots denote individual subjects, and error bars denote SEM across subjects. Stimulus-level alignment statistics are reported in Appendix B.2.

4.2 Visual-fMRI COCO80 Multi-label Recognition

We next evaluate FlatClip on stimulus-level visual-fMRI decoding using NSD COCO80 multi-label recognition. Each sample corresponds to a GLM-estimated cortical response map evoked by one NSD stimulus image, and the target is an 80-dimensional COCO multi-hot label vector. Unlike resting-state benchmarks, this setting is stimulus-evoked and has a more explicit task-relevant cortical substrate. It therefore allows us to test whether preserving cortical geometry is most useful when the retained geometry is relevant to the prediction target. We report within-subject decoding using mean average precision (mAP), averaged across the 80 COCO labels, and support-weighted F1, and compare FlatClip with general-purpose fMRI foundation-model baselines [7, 3, 9].

For FlatClip variants, each GLM response map is encoded by the frozen SigLIP2 encoder into one global representation, followed by the same lightweight probe for COCO80 prediction. We vary the cortical input region across whole cortex, HCP-MMP visual cortex, and the NSD-provided subject-level task-active region, denoted NSDgeneral. The NSDgeneral setting is treated as a task-informed

input-region setting. For voxel-level baselines on NSD, we follow each model's closest available visual-stimulus evaluation protocol. NeuroSTORM includes an NSD visual-stimulus evaluation, and Omni-fMRI provides an NSD-style protocol using repeated GLM response maps. For SwiFT, which supports variable-length inputs, each GLM response map is treated as a single-volume sequence.

As shown in Fig. 3, FlatClip variants achieve stronger COCO80 recognition performance than the evaluated general-purpose fMRI foundation-model baselines. More importantly, performance improves as the input region becomes more specific to the visual task: whole-cortex input is weaker, HCP-MMP visual cortex performs better, and the NSDgeneral subject-level task-active region achieves the best mean mAP. For weighted F1, both visual-cortex and NSDgeneral inputs improve over whole-cortex input, while their difference is smaller. This pattern suggests that visual-fMRI decoding benefits not only from preserving cortical geometry, but also from selecting geometry that is relevant to the evoked visual task. Additional image–fMRI feature-alignment analyses in Appendix B.2 further show that SigLIP2-NSDgeneral features are more closely aligned with the corresponding image-feature geometry than both fMRI foundation-model baselines and broader FlatClip region choices. More details are provided in Appendix B.

4.3 Geometry Controls and Representation Scale

We next test how cortical spatial arrangement affects the predictive utility of flatmap features. We evaluate spatial perturbations of flatmap inputs and a separate comparison of ROI time-series and voxel-based models. Detailed control construction is provided in Appendix C.

Geometry reduction. We perturb cortical flatmaps while keeping the SigLIP2 encoder frozen and the MLP architecture and training protocol fixed. An independent probe is trained for each input condition, with checkpoint selection on the corresponding validation set. These controls measure the task information recoverable from each perturbed representation. Spatial block-shuffling permutes image blocks, while cortical-pixel permutation reassigns RGB values within the rendered cortical mask. Left-right hemisphere swap exchanges hemispheric image positions; within-hemisphere shuffling disrupts layout within each hemisphere. A further control randomizes spatial phase and rank-matches the foreground intensity histogram, with approximate frequency matching. We also include FC heatmaps, random images, and a randomly initialized encoder as additional controls.

As shown in Table 2, real flatmaps achieve the best HCP sex-classification performance. The performance drops under spatial perturbations, especially within-hemisphere shuffling and phase-randomized, histogram-matched inputs, showing that anatomically arranged flatmaps yield more predictive features under the matched probe protocol. FC heatmaps and random images perform much worse, and the randomly initialized encoder also degrades performance, supporting the utility of both spatially organized flatmap inputs and pretrained visual representations. DINOv2 controls show the same trend (Appendix C).

ROI and voxel reference models. Table 3 compares a 450-ROI time-series model with native voxel models using standard or adaptive patching [3]. The voxel models obtain higher accuracy and weighted F1 than the ROI time-series model on both tasks. Adaptive patching improves HCP accuracy and weighted F1; on AD diagnosis, accuracy increases slightly while weighted F1 decreases. This comparison provides additional context for the evaluated representations and patching strategies, whose input formats and model architectures differ.

4.4 Implementation Ablations

Table 4 evaluates the image backbone, feature representation, and flatmap-value normalization. Backbone comparisons fix the input and readout; SigLIP2 is used as the default encoder. Retaining additional frame or patch information improves some tasks, while the main setting uses one compact subject-level feature. Detailed backbone and visual-fMRI ablations are provided in Appendices A and B.

4.5 Surface-Model Comparisons

Table 5 compares FlatClip with frozen CortexMAE-F and an end-to-end SiT-style parcel-token surface Transformer under the corresponding downstream splits. CortexMAE-F uses the released

Table 2: **Geometry-control ablation on HCP sex classification.** All input-control results use the frozen pretrained SigLIP2 encoder with one global representation per frame, except the explicitly labeled random-initialization control. The MLP architecture and training protocol are fixed, and a separate probe is trained for each condition. Results are reported as mean±std over three seeds. Significance markers indicate paired t-tests over the three probe-seed results, comparing Real flatmap against each control after Holm correction. **Red** indicates the best performance.

Control input	ACC ↑	wF1 ↑	Balanced ACC ↑	Macro F1 ↑
Real flatmap	**82.06 ± 2.05**	**82.04 ± 2.05**	**81.81 ± 2.03**	**81.88 ± 2.06**
Spatial block-shuffle	76.14 ± 1.83*	76.18 ± 1.83*	76.21 ± 1.85*	76.06 ± 1.84*
Left-right hemisphere swap	74.96 ± 1.17*	74.89 ± 1.23*	74.56 ± 1.31*	74.64 ± 1.28*
Within-hemisphere region shuffle	70.90 ± 1.63*	70.87 ± 1.62*	70.62 ± 1.61*	70.64 ± 1.62*
Cortical-pixel permutation	72.93 ± 1.92*	72.87 ± 1.97*	72.58 ± 2.05*	72.62 ± 2.02*
Real flatmap with random initialized	70.56 ± 1.34*	70.56 ± 1.36*	70.37 ± 1.41*	70.35 ± 1.38*
Phase-randomized, histogram-matched	64.64 ± 2.29*	64.49 ± 2.14*	64.12 ± 2.03*	64.14 ± 2.07*
FC heatmap	57.87 ± 2.69*	57.93 ± 2.68*	57.98 ± 2.52*	57.80 ± 2.62*
Random image	48.90 ± 3.26*	48.89 ± 3.42*	48.69 ± 3.69*	48.59 ± 3.65*

Markers denote Holm-corrected paired tests against Real flatmap within each metric: $^*p < 0.05$.

Table 3: **Comparison of ROI time-series and voxel-based models.** The ROI input contains 450 regional time series; voxel models use native volumetric signals with standard or adaptive patching. Results are Accuracy and weighted F1 (%).

Representation	Resolution	HCP Sex		AD Diagnosis	
		ACC ↑	wF1 ↑	ACC ↑	wF1 ↑
ROI time series	450 ROIs	65.68±4.68	65.51±4.47	60.65±5.61	60.75±3.90
Voxel model	Native 4D volume	80.74±2.41	80.32±3.21	72.22±10.2	**71.66±9.54**
Voxel model	Adaptive patches	**82.81±0.57**	**82.88±0.55**	**72.99±1.03**	70.76±3.16

frozen checkpoint with a short-window adaptation of the first 40 source frames. CortexMAE-F is slightly stronger on HCP, while FlatClip achieves higher accuracy and weighted F1 on both ADNI tasks. The SiT-style model is competitive on ADNI AD/CN. Appendix D.2 specifies the checkpoints and surface-input construction.

4.6 Spatial Arrangement Across Renderers and Readouts

We extend the spatial-perturbation experiments with a 2×3 geometry-by-rendering design. All conditions use a fixed 224×224 canvas, the same HCP split and 40-frame inputs, frozen SigLIP2, and the same MLP protocol. Each renderer is evaluated with real flatmaps and the same fixed cortical-pixel permutation, with a freshly trained probe for every condition. Table 6 shows a consistent weighted-F1 advantage for the real arrangement across all three colormaps.

Fixed readouts trained on real flatmaps degrade substantially under block shuffle, cortical-pixel permutation, and phase randomization. Retraining the linear probe recovers part of the performance but remains below the real-flatmap result. Appendix D.6 details these complementary tests of recoverable information and decision-boundary transfer.

4.7 Measured Covariates, Temporal Sampling, and Probe Choice

We assess the sensitivity of frozen FlatClip features to available demographic, motion, site, acquisition, and cohort variables. Covariate models are fitted on training subjects and applied unchanged to validation and test subjects before probe training. HCP weighted F1 is 80.88 after age adjustment, compared with 82.04 for the raw features. On ADNI AD/CN, motion adjustment yields 75.34 weighted F1, compared with 75.41 before adjustment. Joint adjustments produce task-dependent changes, reported in Appendix D.3.

Alternative non-overlapping 40-TR windows retain predictive performance on the ADNI tasks: MCI/CN weighted F1 is 62.69 and 63.20 in the paired window evaluation, and AD/CN weighted F1 is 75.41 and 73.35 (Appendix D.4). Linear probing reaches 84.77 accuracy and weighted F1 on

Table 4: Implementation ablations for frozen image backbone, feature representation, and flatmap normalization. Results are reported as Accuracy / weighted F1. (%)

Setting \ Dataset	HCP *Sex Classif.*		ADNI (MCI) *Diagnosis*		ADNI (AD) *Diagnosis*		PPMI *PD Diagnosis*	
	ACC ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑
DINOv2	80.37±2.55	80.33±2.53	60.68±2.67	60.44±1.98	75.46±4.01	75.20±3.69	51.39±3.18	51.58±2.25
ViT-B	80.10±1.71	80.12±1.68	54.48±2.80	54.82±3.18	76.23±4.06	**76.05±3.88**	56.67±4.51	54.03±5.19
SigLIP2	**82.06±2.05**	**82.04±2.05**	**61.11±1.48**	**60.75±1.16**	**75.46±2.12**	75.41±2.11	**54.86±1.96**	**55.72±1.63**
40 Patch-Mean	**84.94±1.06**	**84.91±1.06**	58.12±1.96	57.82±1.87	**76.39±4.17**	**76.36±3.93**	**56.94±4.21**	**56.48±4.27**
40 Global tokens	83.08±0.59	83.02±0.66	60.68±1.96	60.02±2.07	74.07±4.88	74.13±4.98	53.47±7.54	53.93±6.14
One Global token	82.06±2.05	82.04±2.05	**61.11±1.48**	**60.75±1.16**	75.46±2.12	75.41±2.11	54.86±1.96	55.72±1.63
Frame Z-Score	**82.23±1.34**	**82.23±1.28**	58.12±3.23	58.28±3.27	73.15±0.80	73.52±0.91	49.65±3.35	50.68±3.10
Voxel Z-Score	74.49±2.20	74.52±2.22	58.55±0.60	56.22±1.28	**80.56±1.96**	**79.68±2.33**	50.35±2.60	51.17±0.76
Global Z-Score	82.06±2.05	82.04±2.05	**61.11±1.48**	**60.75±1.16**	75.46±2.12	75.41±2.11	**54.86±1.96**	**55.72±1.63**

Table 5: **Comparison with surface models.** Accuracy / weighted F1 (%), mean±std over three runs. CortexMAE-F pretraining includes HCP. The SiT-style column reports the parcel-token configuration; implementation details are given in Appendix D.2.

Task	FlatClip	CortexMAE-F	SiT-style parcel-token surface
HCP sex	82.06 ± 2.05 / 82.04 ± 2.05	$\mathbf{82.40 \pm 1.27}$ / $\mathbf{82.37 \pm 1.26}$	55.84 ± 5.19 / 55.74 ± 5.15
ADNI AD/CN	$\mathbf{75.46 \pm 2.12}$ / $\mathbf{75.41 \pm 2.11}$	69.91 ± 6.26 / 70.43 ± 6.31	74.54 ± 2.85 / 73.07 ± 2.90
ADNI MCI/CN	$\mathbf{61.11 \pm 1.48}$ / $\mathbf{60.75 \pm 1.16}$	58.12 ± 0.74 / 57.46 ± 1.46	55.98 ± 3.96 / 55.59 ± 2.79

HCP, while the MLP is stronger on the ADNI and PPMI tasks. Increasing the number of labeled HCP training subjects improves both readouts, with MLP weighted F1 rising from 69.92 at 10% of the training set to 82.04 at 100% (Appendix D.10).

4.8 Direct-Volume Rendering Comparisons

We additionally apply frozen SigLIP2 to 2D renderings derived directly from acquired volumes. On HCP, axial mean projection gives 72.81 weighted F1, whereas independently encoding the non-empty slices of each volume and averaging their embeddings across slices and time gives 89.32. FlatClip gives 82.04 with one rendered cortical image per time point. These results illustrate the performance and image-count trade-off between the input representations. On NSD subj01, direct-volume slice encoding gives 4.342 mAP and 15.922 weighted F1, compared with the reported FlatClip-NSDgeneral result of 15.311 and 26.922. Full rendering and pooling details are provided in Appendix D.8.

Native temporal inputs to Omni-fMRI. We also evaluate the released Omni-fMRI encoder on NSD subj01 raw BOLD using continuous 40-TR windows and event-centered stacks. These inputs yield 3.804/16.531 and 3.840/16.188 mAP/weighted F1, respectively. The two temporal constructions perform similarly, with continuous windows giving slightly higher weighted F1. Appendix D.9 provides the segment construction, data split, and full results.

5 Discussion

The experiments highlight two useful design choices: retaining an anatomy-linked spatial arrangement and selecting cortical inputs relevant to the task. Spatial perturbations assess sensitivity to cortical arrangement, while comparisons across NSD input masks examine input-region selection. Same-mask baselines compare representations and readouts under matched cortical coverage. Together, these analyses show how spatial arrangement and cortical coverage influence prediction with frozen image features.

FlatClip also separates reusable feature extraction from task-specific readout. Different cortical inputs can be evaluated with the same frozen encoder. This makes FlatClip a practical reference without fMRI-specific backbone training for comparisons with domain-specific models. More broadly,

Table 6: **Geometry effect across renderers.** Weighted-F1 gain (percentage points) of real flatmaps over the corresponding fixed cortical-pixel permutation.

Renderer	Real minus permuted wF1
Inferno	$+7.26$
Grayscale	$+8.24$
RdBu_r	$+8.74$

the results highlight cortical input design as a means of reusing pretrained visual representations, complementing efforts to develop fMRI-specific encoders.

Limitations. Flatmaps introduce cuts and metric distortion and omit subcortical and brainstem signals. Temporal mean pooling also discards frame order. Future work will examine hybrid cortical–subcortical inputs and explicit temporal modeling, including naturalistic movie paradigms.

Acknowledgment

This work was supported by National Natural Science Foundation of China (3254100307, 62472206), Brain Science and Brain-like Intelligence Technology-National Science and Technology Major Project (2021ZD0200500, 2026ZD0220402), National Key R&D Program of China (2025YFC3410000), Shenzhen Science and Technology Innovation Committee (RCYX20231211090405003), Guangdong Basic and Applied Basic Research Foundation (2026B1515020099), Guangdong S&T Program (2026B0101110003), Shanghai Municipal Special Program for Basic Research on General AI Foundation Models (2025SHZDZX026D05), Shenzhen Loop Area Institute (FPF10120250012), and the open research fund of the Guangdong Provincial Key Laboratory of Mathematical and Neural Dynamical Systems, the Center for Computational Science and Engineering at Southern University of Science and Technology, Shenzhen Key Laboratory of Smart Healthcare Engineering. The computations in this research were performed using the CFFF platform of Fudan University.

References

[1] Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Chong, Fang Ji, Nathanael Tong, Christopher Chen, and Juan Helen Zhou. Brain-jepa: Brain dynamics foundation model with gradient positioning and spatiotemporal masking. *Advances in Neural Information Processing Systems*, 37:86048–86073, 2024.

[2] 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, et al. Brainlm: A foundation model for brain activity recordings. In *International Conference on Learning Representations*, 2024.

[3] Mo Wang, Wenhao Ye, Junfeng Xia, Junxiang Zhang, Xuanye Pan, Minghao Xu, Haotian Deng, Hongkai Wen, and Quanying Liu. Omni-fmri: A universal atlas-free fmri foundation model. In *Forty-third International Conference on Machine Learning*, 2026.

[4] 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, 2024.

[5] Xiangmin Han, Rundong Xue, Jingxi Feng, Yifan Feng, Shaoyi Du, Jun Shi, and Yue Gao. Hypergraph foundation model for brain disease diagnosis. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[6] Ziquan Wei, Tingting Dan, and Guorong Wu. Large connectome model: An fmri foundation model of brain connectomes empowered by brain-environment interaction in multitask learning landscape. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 2155–2163, 2026.

[7] Peter Kim, Junbeom Kwon, Sunghwan Joo, Sangyoon Bae, Donggyu Lee, Yoonho Jung, Shinjae Yoo, Jiook Cha, and Taesup Moon. Swift: Swin 4d fmri transformer. *Advances in Neural Information Processing Systems*, 36:42015–42037, 2023.

[8] 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. *arXiv preprint arXiv:2512.21881*, 2025.

[9] Cheng Wang, Yu Jiang, Zhihao Peng, Chenxin Li, Chang-Bae 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 functional MRI analysis. *Nature Biomedical Engineering*, 2026.

[10] 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. *IEEE Journal of Biomedical and Health Informatics*, 29:8719–8726, 2025.

[11] Junbeom Kwon, Jungwoo Seo, Heehwan Wang, Taesup Moon, Shinjae Yoo, and Jiook Cha. Predicting task-related brain activity from resting-state brain dynamics with fmri transformer. *Imaging Neuroscience*, 3, 2025.

[12] Mo Wang, Kaining Peng, Jingsheng Tang, Hongkai Wen, and Quanying Liu. DCA: Graph-guided deep embedding clustering for brain atlases. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] Mehraveh Salehi, Abigail S Greene, Amin Karbasi, Xilin Shen, Dustin Scheinost, and R Todd Constable. There is no single functional atlas even for a single individual: Functional parcel definitions change with task. *NeuroImage*, 208:116366, 2020.

[14] James C Pang, Kevin M Aquino, Marianne Oldehinkel, Peter A Robinson, Ben D Fulcher, Michael Breakspear, and Alex Fornito. Geometric constraints on human brain function. *Nature*, 618(7965):566–574, 2023.

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

[16] Junfeng Xia, Wenhao Ye, Xuanye Pan, Xinke Shen, Mo Wang, and Quanying Liu. Brain-dit: A universal multi-state fmri foundation model with metadata-conditioned pretraining. *arXiv preprint arXiv:2604.12683*, 2026.

[17] Mo Wang, Wenhao Ye, Junfeng Xia, Minghao Xu, Hongkai Wen, and Quanying Liu. Flexi-Brain: Resolution-agnostic voxel-level encoding for native fMRI. In *European Conference on Computer Vision*, pages 432–449. Springer, 2026.

[18] Junfeng Xia, Wenhao Ye, Junxiang Zhang, Xuanye Pan, Mo Wang, and Quanying Liu. Brain-World: A structural-prior-conditioned generative model for whole-brain 4D fMRI dynamics. *arXiv preprint arXiv:2606.17742*, 2026.

[19] Hamid Behjat and Martin Larsson. Spectral characterization of functional mri data on voxel-resolution cortical graphs. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 558–562. IEEE, 2020.

[20] James S Gao, Alexander G Huth, Mark D Lescroart, and Jack L Gallant. Pycortex: an interactive surface visualizer for fmri. *Frontiers in neuroinformatics*, 9:23, 2015.

[21] Jianxiong Gao, Yanwei Fu, Yuqian Fu, Yun Wang, Xuelin Qian, and Jianfeng Feng. MinD-3D++: Advancing fMRI-based 3D reconstruction with high-quality textured mesh generation and a comprehensive dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11802–11816, 2025.

[22] Connor Lane, Daniel Z Kaplan, Tanishq Mathew Abraham, and Paul S Scotti. Scaling vision transformers for functional mri with flat maps. *arXiv preprint arXiv:2510.13768*, 2025.

[23] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[24] Fenqiang Zhao, Shunren Xia, Zhengwang Wu, Dingna Duan, Li Wang, Weili Lin, John H. Gilmore, Dinggang Shen, and Gang Li. Spherical u-net on cortical surfaces: Methods and applications. In *Information Processing in Medical Imaging*, pages 855–866, 2019.

[25] Simon Dahan, Abdulah Fawaz, Logan Z. J. Williams, Chunhui Yang, Timothy S. Coalson, Matthew F. Glasser, A. David Edwards, Daniel Rueckert, and Emma C. Robinson. Surface vision transformers: Attention-based modelling applied to cortical analysis. In *Proceedings of the 5th International Conference on Medical Imaging with Deep Learning*, volume 172 of *Proceedings of Machine Learning Research*, pages 282–303, 2022.

[26] Fabian Bongratz, Tom Nuno Wolf, Jaume Gual Ramon, and Christian Wachinger. X-sit: Inherently interpretable surface vision transformers for dementia diagnosis. In *Medical Image Computing and Computer Assisted Intervention*, 2025.

[27] Geonwoo Baek, David H. Salat, and Ikbeom Jang. Improved multiscale structural mapping with supervertex vision transformer for the detection of alzheimer's disease neurodegeneration. *Human Brain Mapping*, 47(8):e70548, 2026.

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[31] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[32] Matthew F Glasser, Timothy S Coalson, Emma C Robinson, Carl D Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F Beckmann, Mark Jenkinson, et al. A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178, 2016.

[33] Clifford R Jack Jr, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L. Whitwell, Chadwick Ward, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 27(4):685–691, 2008.

[34] Kenneth Marek, Danna Jennings, Shirley Lasch, Andrew Siderowf, Caroline Tanner, Tanya Simuni, Chris Coffey, Karl Kieburtz, Emily Flagg, Sohini Chowdhury, et al. The parkinson progression marker initiative (ppmi). *Progress in neurobiology*, 95(4):629–635, 2011.

[35] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[36] Zijian Dong, Ruilin Li, Joanna Su Xian Chong, Niousha Dehestani, Yinghui Teng, Yi Lin, Zhizhou Li, Yichi Zhang, Yapei Xie, Leon Qi Rong Ooi, et al. Brain harmony: A multimodal foundation model unifying morphology and function into 1d tokens. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[37] Heiko Braak, Kelly Del Tredici, Udo Rüb, Rob AI De Vos, Ernst NH Jansen Steur, and Eva Braak. Staging of brain pathology related to sporadic parkinson's disease. *Neurobiology of aging*, 24(2):197–211, 2003.

[38] Ronald B Postuma, Daniela Berg, Matthew Stern, Werner Poewe, C Warren Olanow, Wolfgang Oertel, José Obeso, Kenneth Marek, Irene Litvan, Anthony E Lang, et al. Mds clinical diagnostic criteria for parkinson's disease. *Movement disorders*, 30(12):1591–1601, 2015.

A Details of Rest fMRI experiments

A.1 Details of Datasets and Task

Human Connectome Project (HCP). The HCP FlatClip feature cohort contains 1,000 subjects with matched features and labels, partitioned into 602 training, 201 validation, and 197 test subjects. Table 16 gives the class counts for this partition, which is also used by the supplementary HCP feature evaluations.

Parkinson's Progression Markers Initiative (PPMI). PPMI is a longitudinal observational study of Parkinson's disease progression [34]. Our three-way classification cohort contains 474 participants: 29 controls, 142 with PD, and 303 prodromal participants. The training, validation, and test partitions contain 331, 47, and 96 participants, respectively.

Alzheimer's Disease Neuroimaging Initiative (ADNI). The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal study of Alzheimer's disease progression [33]. The MCI/CN task uses 381 subjects, split into 266/37/78 training/validation/test subjects; AD/CN uses 350 subjects, split into 244/34/72. The alternative-window analysis is described in Appendix D.4.

A.2 Evaluation Protocol

Evaluation uses subject-level partitions, with all frames from a subject assigned to one partition. Models use the frozen-feature protocol in Section 3; checkpoints are selected on validation data and evaluated on the held-out test set. Supplementary probe experiments keep these partitions fixed and report mean and standard deviation over seeds 42, 43, and 44.

A.3 Details of Baseline Models

In this section, we introduce our baseline models.

BrainLM is the first fMRI foundation model specifically designed to capture the spatiotemporal dynamics of brain activity through a Transformer-based masked autoencoder architecture. It employs the AAL-424 atlas for the parcellation of brain regions, transforming fMRI recordings into a 424-dimensional set of ROIs. The model is trained on an extensive dataset consisting of 6,700 hours of fMRI data derived from 77,298 recordings across the UK Biobank and the Human Connectome Project.

BrainMASS uses Schaefer 100-region brain networks and combines masked ROI modeling with latent representation alignment. Its pretraining collection comprises 70,781 samples from 46,686 participants across 30 datasets [4].

Brain-Harmony is a multimodal brain foundation model designed to jointly represent brain morphology and function within a unified one-dimensional token space. The model first encodes functional MRI ROI time-series and structural T1-weighted MRI into modality-specific token embeddings, and then uses a Transformer-based harmonizer to fuse these tokens into a shared subject-level representation. In our baseline setting, we use the official pretrained Brain-Harmony backbone with a ViT-Base architecture and 128 latent tokens.

LCM is an fMRI foundation model for connectome-based brain representation learning, which formulates pretraining as a multitask learning problem by leveraging brain–environment interaction variables, including demographic and behavioral targets, together with large-scale functional neuroimaging data. Its architecture represents brain connectomes as node-level embeddings and uses a decoder-style prediction head with learnable task queries to support multiple downstream objectives such as sex prediction, behavioral recognition, and disease classification.

SwiFT (Swin 4D fMRI Transformer) is an end-to-end framework for modeling high-dimensional spatiotemporal fMRI dynamics directly from raw 4D volumes, avoiding the information loss that can arise from hand-crafted features or atlas-level summaries. It adopts a computationally efficient

Table 7: Ablation of frozen image backbone and feature extraction strategy. Results are reported as Accuracy / weighted F1.

Backbone	Feature	HCP *Sex Classif.*		PPMI *PD Diagnosis*		ADNI (MCI) *Diagnosis*		ADNI (AD) *Diagnosis*	
		ACC ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑	**ACC** ↑	**wF1** ↑
DINOv2	One global token	80.37±2.55	80.33±2.53	51.39±3.18	51.58±2.25	60.68±2.67	60.44±1.98	75.46±4.01	75.20±3.69
DINOv2	40 Global tokens	79.53±1.17	79.50±1.15	47.57±5.14	48.82±2.67	55.98±2.67	55.58±2.09	75.00±2.41	74.61±2.34
ViT-B	40 Global tokens	70.00±1.13	70.02±1.15	60.00±1.36	55.45±1.72	52.76±4.57	52.19±4.71	78.11±2.56	76.84±2.71
SigLIP2	One Global token	82.06±2.05	82.04±2.05	54.86±1.96	55.72±1.63	61.11±1.48	60.75±1.16	75.46±2.12	75.41±2.11
SigLIP2	40 Global tokens	83.08±0.59	83.02±0.66	53.47±7.54	53.93±6.14	60.68±1.96	60.02±2.07	74.07±4.88	74.13±4.98
SigLIP2	40 patch_mean	84.94±1.06	84.91±1.06	56.94±4.21	56.48±4.27	58.12±1.96	57.82±1.87	76.39±4.17	76.36±3.93

4D shifted-window attention design to capture local and long-range spatiotemporal dependencies in brain activity. SwiFT also supports contrastive self-supervised pretraining, which further improves downstream transfer performance on large-scale fMRI benchmarks.

NeuroSTORM is a general-purpose neuroimaging foundation model designed to learn representations directly from raw 4D fMRI volumes. Its backbone is based on Shifted-Window Mamba (SWM), which is used to model long-range spatiotemporal dependencies while maintaining computational efficiency. NeuroSTORM is pre-trained on large-scale fMRI data from more than 50,000 subjects across UK Biobank, ABCD, and HCP, and introduces a Spatiotemporal Redundancy Dropout (STRD) module to reduce redundant temporal information during representation learning. This design enables NeuroSTORM to serve as a transferable foundation model for downstream fMRI prediction tasks.

Omni-fMRI is an atlas-free fMRI foundation model designed to learn directly from voxel-level signals, avoiding the information loss and atlas-dependent biases introduced by predefined region-level parcellations. To scale pretraining to 49,497 fMRI sessions from nine datasets, Omni-fMRI introduces a dynamic patching mechanism that reduces computational cost while preserving informative spatial structure. The model is evaluated through a comprehensive benchmark suite covering 11 datasets and diverse resting-state and task-based fMRI tasks, where it consistently outperforms existing fMRI foundation models. In our experiments, we use Omni-fMRI as a strong voxel-level foundation-model baseline for assessing whether FlatClip can achieve competitive transfer performance without fMRI-specific encoder pretraining.

A.4 Ablation of Architectures

We further compare different frozen image backbones and feature extraction choices for resting-state prediction (Table 7). For the main comparisons, all methods use one global representation and the same downstream MLP probe to ensure a fair evaluation of the frozen backbone. This setting follows the standard use of global or `[CLS]` representations in ViT-style encoders for image-level prediction, where the global token summarizes the input image into a single feature vector. The ViT baseline uses a standard ImageNet-pretrained ViT-B/16 visual encoder from torchvision. Across backbones, SigLIP2 achieves the strongest performance on HCP and is used as the default backbone in subsequent geometry-control analyses. The frame-token and patch-mean variants assess the effect of retaining additional feature information.

A.5 Configuration

All FlatClip encoders are frozen during downstream evaluation. We first render scalar cortical fMRI activity into RGB cortical flatmaps and then extract image features with a frozen SigLIP2-NaFlex encoder. Only the downstream MLP probe is trained for task prediction. The main settings are summarized in Table 8. We choose $T = 40$, following the setting with prior works [9, 3].

Normalization and inference cost. Following the rendering steps in Section 3.2, resting-state inputs are globally z-scored over the selected vertex-by-time matrix. NSD responses are z-scored within each GLM map and mapped to RGB using a diverging colormap. Native crop dimensions

Table 8: Hyperparameter settings of FlatClip.

Configuration	*Value*
Flatmap rendering	
surface space	fsLR cortical surface
renderer	PyCortex / cached flatmap projection
Normalization	global z-score with bounded color scale
colormap	Inferno (HCP); RdBu_r (NSD)
background pixels	transparent outside cortex/ROI, composited on white RGB
flatmap crop	cropped to non-background cortical pixels
Resting-state configs	
image encoder	frozen SigLIP2-base-patch16-NaFlex
input format	sequence of cortical flatmap frames
input region	whole cortex
frames per subject	first 40 resting-state frames
encoder input	RGB flatmap, padded/processed by SigLIP2-NaFlex processor
frame feature	global image embedding
frame feature dimension	768
subject feature	mean pooling over 40 frame features
probe architecture	two hidden layers: $768 \to 256 \to 256 \to C$, dropout 0.2
optimizer	AdamW
learning rate	1×10^{-3}
weight decay	1×10^{-4}
batch size	64
training epochs	up to 250, early stopping patience 40
validation metric	weighted-F1
loss function	class-weighted cross-entropy
reported metrics	accuracy, weighted-F1
NSD visual-fMRI configs	
image encoder	frozen SigLIP2-base-patch16-NaFlex
input format	stimulus-level GLM cortical flatmap
input regions	whole cortex / HCP-MMP visual / NSDgeneral
encoder output	mean over 256 adaptively pooled feature tokens
main feature dimension	768
probe architecture	MLP with LayerNorm, hidden dim 1024, dropout 0.1
optimizer	AdamW
learning rate	3×10^{-4}
weight decay	1×10^{-4}
batch size	128
training epochs	10
validation metric	mAP
loss function	class-weighted BCE with logits
reported metrics	mAP, weighted-F1

are listed in Table 13; SigLIP2-NaFlex converts these variable-size images into a fixed-budget patch sequence.

Feature extraction is performed offline with the frozen SigLIP2-NaFlex encoder. Image counts below refer to extraction jobs; downstream cohort sizes refer to matched feature and label sets. From the logged extraction jobs, HCP required 3.96 hours on four A100 GPUs for 40,440 flatmap images, corresponding to 2.84 images/s in wall-clock throughput. PPMI required 39.5 minutes on four A100 GPUs for 18,960 images, and ADNI required 61.6 minutes on two A100 GPUs for 19,840 images. After features are cached, downstream MLP training is lightweight and does not require backpropagation through the image encoder.

A.6 The influence of frame number

We further evaluate the effect of temporal sampling length in resting-state FlatClip. For each subject, we uniformly sample 10, 20, or 40 frames from the available 40-frame sequence. Each

Table 9: Within-subject COCO80 multi-label classification on NSD. **Red** indicates the best performance. wF1 denotes weighted-F1. Unmarked FlatClip rows use the main setting: one 768-dimensional feature obtained by averaging a 16×16 feature grid adaptively pooled from valid NaFlex output tokens. This grid is formed after encoding and is distinct from the processor's variable-size patch grid. Rows marked with * retain all 256 pooled feature tokens.

Model \ Subject	Sub1		Sub2		Sub5		Sub7		Mean	
	mAP ↑	wF1 ↑	mAP ↑	wF1 ↑	mAP ↑	wF1 ↑	mAP ↑	wF1 ↑	mAP ↑	wF1 ↑
SigLIP2-NSDgeneral	**15.311**	**26.922**	15.714	26.986	**19.933**	**29.611**	**14.428**	24.366	**16.347**	**26.971**
SigLIP2-Visual	14.050	25.829	**16.065**	**27.151**	18.889	29.476	14.361	**25.238**	15.841	26.924
DINOv2-Visual	14.765	26.338	14.780	25.105	17.213	27.304	12.605	22.547	14.841	25.323
ViT-Cortex	11.674	23.092	13.373	24.128	13.895	26.376	10.544	20.432	12.371	23.507
SigLIP2-Cortex	11.100	24.576	12.211	25.048	14.974	26.090	10.053	18.822	12.085	23.634
DINOv2-Cortex	10.930	22.800	10.815	23.022	11.290	24.427	9.467	19.751	10.625	22.500
SwiFT	7.974	20.582	10.769	22.303	10.274	22.989	8.478	20.858	9.374	21.683
NeuroSTORM	6.491	12.358	8.798	20.484	9.253	10.471	8.209	18.326	8.188	15.410
Omni-fMRI	3.949	13.818	5.024	15.910	4.715	13.441	3.751	8.934	4.360	13.026
SigLIP2-Cortex*	16.604	29.776	15.771	28.262	19.218	31.379	14.129	27.836	16.430	29.313
SigLIP2-Visual*	20.192	**33.255**	19.623	29.727	23.323	**33.801**	19.490	30.457	20.657	31.810
SigLIP2-NSDgeneral*	**21.565**	32.330	**22.218**	**32.084**	**25.300**	32.744	**20.282**	**31.123**	**22.341**	**32.070**

Table 10: Ablation on the number of resting-state frames used by FlatClip. All results use frozen SigLIP2 with one global representation per frame. Selected frame features are mean-pooled into a 768-dimensional subject-level token. Results are reported as Accuracy / weighted F1 over three seeds.

Frames \ Dataset	HCP *Sex Classif.*		ADNI (MCI) *Diagnosis*		ADNI (AD) *Diagnosis*		PPMI *PD Diagnosis*	
	ACC ↑	wF1 ↑	ACC ↑	wF1 ↑	ACC ↑	wF1 ↑	ACC ↑	wF1 ↑
10 frames	76.31±4.61	76.27±4.64	59.40±3.23	58.52±3.13	76.39±1.39	76.39±1.43	54.17±2.08	54.12±2.11
20 frames	77.83±1.92	77.79±1.86	60.68±1.48	59.90±1.77	**78.24±0.80**	**78.29±1.26**	52.43±1.59	53.39±2.27
40 frames	**82.06±2.05**	**82.04±2.05**	**61.11±1.48**	**60.75±1.16**	75.46±2.12	75.41±2.11	**54.86±1.96**	**55.72±1.63**

frame is encoded independently by the frozen SigLIP2 encoder, and the selected frame-level global representations are mean-pooled into a single 768-dimensional subject-level feature. As shown in Table 10, increasing the number of frames generally improves or stabilizes performance, but the gain is modest under the one-token pooling strategy. HCP and ADNI show the clearest benefit from using more temporal samples, with 40 frames giving the best or tied-best performance in most metrics. ADNI (AD) already saturates around 20 frames, while PPMI does not show a consistent improvement with additional frames, suggesting that temporal coverage alone is not the main limiting factor for this task. Overall, these results indicate that FlatClip can extract useful subject-level information even from sparse temporal samples, while the 40-frame setting provides a more stable default for the main experiments.

B Details of NSD Visual-fMRI Experiments

Natural Scenes Dataset (NSD). We evaluate visual-fMRI decoding on the Natural Scenes Dataset (NSD), a large-scale 7T fMRI dataset in which subjects viewed natural images from COCO. We use stimulus-level GLM beta estimates, specifically the `betas_fithrf_GLMdenoise_RR` version, as neural response inputs. The decoding target is an 80-dimensional COCO multi-hot label vector. For each evaluated subject, 8,550 non-shared images are used for training, 450 non-shared images for validation, and the 1,000 shared images for testing. Image identities are disjoint across these partitions.

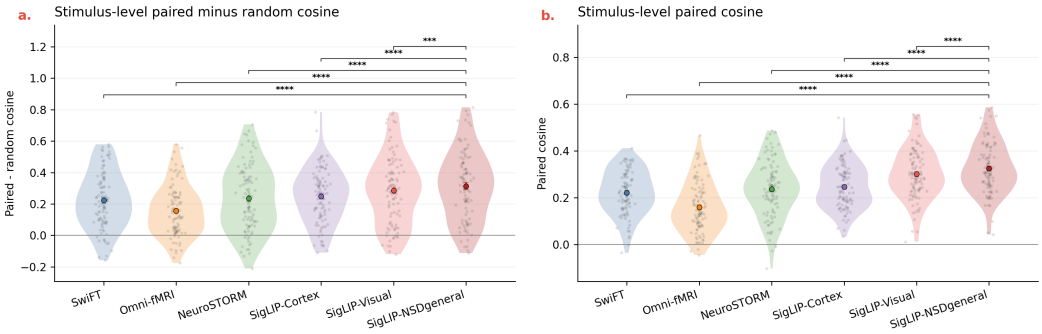


Figure 4: Stimulus-level image–fMRI feature alignment analysis on a randomly sampled subset of NSD shared1000 in SigLIP2 space. (a) Paired-random cosine margin, defined as the cosine similarity between the aligned fMRI-derived feature and the matched image feature minus a randomly mismatched image cosine. (b) Paired cosine similarity between the aligned fMRI-derived feature and its matched image feature. For each shared image, scores are first computed for each subject and then averaged across the four subjects. Dots denote individual shared stimuli, and large markers denote the mean. Significance brackets compare SigLIP2-NSDgeneral against each other method using paired Wilcoxon signed-rank tests over stimuli. Cosines are computed after separately fitted subject- and model-specific ridge alignment.

B.1 GLM for Baselines models

SwiFT extracted features in a volume-wise manner: each NSD beta volume is processed independently, and the resulting representation is used as the feature for that volume.

NeuroSTORM follows a similar strategy. Its NSD extraction pipeline also processes each beta volume independently with a 3D backbone and saves the resulting backbone feature for each volume.

Omni-fMRI expects 40 input channels. For the stimulus-level beta-map baseline, a single 3D beta volume is repeated along this dimension and encoded into a CLS feature for downstream prediction. The native-BOLD experiment in Appendix D.9 additionally evaluates temporally varying inputs with the same released encoder.

B.2 Supplementary Analysis of Image–fMRI Feature Alignment

We further quantified image–fMRI feature alignment in SigLIP2 space using a randomly sampled 100-image subset from NSD shared1000 (Fig. 4, Table 11). For each shared image, we computed the alignment score for each subject and then averaged across the four subjects, yielding one value per image. For paired cosine and margin, a separate ridge map is fitted for each subject and fMRI backbone to predict SigLIP2 image features. The evaluation set consists of 100 shared images sampled with seed 42; mapping uses the other available matched images, including shared images outside this held-out subset. Feature standardization and PCA are fitted on the mapping-training set only. The fMRI features are reduced to at most 256 PCA components, followed by ridge regression with $\alpha = 100$ and a fitted intercept. These hyperparameters are fixed rather than selected on the held-out images. Cosine similarity is computed between the ridge predictions and the matched image features standardized with the training-set statistics. Local-RDM is instead computed directly from distance profiles in each model's own feature space, without the learned mapping. For Local-RDM, each image was represented by its cosine-distance profile to the other evaluation images after row-wise standardization and L2 normalization, and we computed the Spearman correlation with the corresponding SigLIP2 image-feature distance profile. Paired cosine measured the similarity between the aligned fMRI-derived feature and its matched image feature, while the paired-random margin subtracted a randomly mismatched image cosine.

Across all three stimulus-level metrics, SigLIP2-NSDgeneral achieved the strongest alignment with the corresponding image features. Compared with the strongest non-FlatClip baseline, NeuroSTORM, SigLIP2-NSDgeneral improved Local-RDM by 0.083, paired cosine by 0.088, and paired-random margin by 0.079. The advantage also remained significant when compared with SigLIP2-Visual,

Table 11: Stimulus-level image–fMRI feature alignment analysis on a randomly sampled subset of NSD shared1000 in SigLIP2 space. Values are averaged over four subjects, leaving one value per image. Δ denotes SigLIP2-NSDgeneral minus the comparison method. p_t and p_w denote one-sided paired t-test and Wilcoxon signed-rank p-values, with Holm correction across the 15 comparisons within each test family. Confidence intervals are percentile bootstrap intervals over stimulus-level paired differences.

Metric	Comparison	Baseline	SigLIP2-NSDgeneral	Δ	95% CI	p_t	p_w	Improved
Local-RDM	vs SwiFT	0.026	0.134	0.108	[0.086, 0.129]	9.01×10^{-16}	3.18×10^{-12}	83.0%
Local-RDM	vs Omni-fMRI	0.009	0.134	0.125	[0.102, 0.147]	2.70×10^{-17}	5.27×10^{-13}	86.0%
Local-RDM	vs NeuroSTORM	0.052	0.134	0.083	[0.064, 0.101]	5.78×10^{-13}	1.34×10^{-10}	79.0%
Local-RDM	vs SigLIP2-Cortex	0.032	0.134	0.102	[0.084, 0.121]	2.09×10^{-17}	2.87×10^{-13}	89.0%
Local-RDM	vs SigLIP2-Visual	0.091	0.134	0.043	[0.026, 0.061]	5.41×10^{-6}	5.73×10^{-6}	74.0%
Paired cosine	vs SwiFT	0.222	0.325	0.103	[0.087, 0.119]	3.82×10^{-21}	7.31×10^{-15}	90.0%
Paired cosine	vs Omni-fMRI	0.159	0.325	0.166	[0.150, 0.182]	1.50×10^{-35}	3.61×10^{-17}	99.0%
Paired cosine	vs NeuroSTORM	0.237	0.325	0.088	[0.072, 0.104]	5.07×10^{-17}	4.31×10^{-14}	89.0%
Paired cosine	vs SigLIP2-Cortex	0.246	0.325	0.079	[0.064, 0.094]	3.70×10^{-16}	1.17×10^{-13}	89.0%
Paired cosine	vs SigLIP2-Visual	0.301	0.325	0.024	[0.015, 0.033]	3.83×10^{-6}	7.96×10^{-6}	69.0%
Margin	vs SwiFT	0.223	0.314	0.091	[0.066, 0.116]	7.19×10^{-10}	8.30×10^{-9}	76.0%
Margin	vs Omni-fMRI	0.156	0.314	0.157	[0.128, 0.186]	5.07×10^{-17}	2.87×10^{-13}	87.0%
Margin	vs NeuroSTORM	0.235	0.314	0.079	[0.053, 0.106]	1.81×10^{-7}	4.49×10^{-7}	72.0%
Margin	vs SigLIP2-Cortex	0.248	0.314	0.066	[0.042, 0.090]	1.38×10^{-6}	2.49×10^{-6}	69.0%
Margin	vs SigLIP2-Visual	0.286	0.314	0.028	[0.012, 0.044]	4.23×10^{-4}	4.06×10^{-4}	68.0%

indicating that restricting the flatmap input to NSD task-active cortex yields representations that are more tightly aligned with image-feature geometry than using a broader visual mask. This analysis complements COCO80 classification by quantifying stimulus-level correspondence between cortical and image representations.

B.3 Ablation and Detailed Results

Table 9 gives subject-wise results for the frozen image backbones and cortical regions. Unmarked rows use one pooled representation; starred rows retain all 256 adaptively pooled feature tokens to assess sensitivity to the feature representation.

C Details of Geometry perturbation Experiments

C.1 Geometry-control perturbations.

We constructed spatial-perturbation controls for HCP sex classification to assess the contribution of cortical image layout. All input controls used the same subjects, train/validation/test split, number of frames, frozen pretrained encoder, and MLP architecture and training protocol; the random-initialization condition replaces only the encoder weights. Each condition used an independently trained probe selected on validation data. For each subject, we used a fixed sequence of 40 frames. All rendered inputs were converted to the same image canvas before feature extraction, so that performance differences primarily reflect changes in spatial organization under matched downstream settings, while keeping image canvas handling and feature extraction fixed within each control family.

Real flatmap. The real-flatmap condition uses the original cortical flatmap rendering without spatial perturbation. Each frame preserves the native 2D cortical surface layout produced by the flatmap projection, including both local activity patterns and their global anatomical arrangement.

Left-right hemisphere swap. For the left-right swap control, we exchanged the left and right hemispheric halves of each rendered flatmap image. The same transformation was applied to all subjects and frames. This control preserves the pixel intensity distribution, local texture, and most low-level image statistics, but disrupts hemispheric laterality and the correspondence between cortical location and visual image position. It tests sensitivity to the placement of the two hemispheres within the image.

Within-hemisphere region shuffle. For the within-hemisphere shuffle control, we first separated the rendered flatmap into left- and right-hemisphere image regions. Within each hemisphere, the

valid cortical area was divided into local image blocks, and these blocks were randomly permuted only within the same hemisphere. The same block permutation was applied consistently across subjects and frames. This preserves hemispheric identity and local block-level activation patterns, while disrupting the finer global layout of cortical regions within each hemisphere. Compared with left-right swapping, this provides a stronger test of whether within-hemisphere cortical organization contributes to downstream prediction.

Phase-randomized, histogram-matched. To further control for rendering-induced low-level image statistics, we constructed a phase-randomized control for each flatmap frame. For each RGB channel, we fill the non-cortical background with the foreground mean, retain the resulting Fourier amplitudes, randomize phases, and transform back to image space. Foreground values are then rank-matched to the original foreground intensity histogram, and the white background is restored. This preserves the cortical silhouette and foreground intensity distribution while disrupting spatial organization; the subsequent histogram matching can change the Fourier amplitudes, so frequency matching is approximate.

Random initialized encoder. In addition to input perturbations, we included a randomly initialized SigLIP2 encoder control. This condition uses the real flatmap input and the same downstream MLP probe, but removes pretrained visual representations by replacing the frozen pretrained encoder with the same architecture initialized from random weights. This control tests whether the observed performance comes only from the model architecture and image-format input, or whether pretrained visual features contribute to the FlatClip representation.

Spatial block-shuffle. For the block-shuffle control, each flatmap image was divided into non-overlapping 16×16 image blocks. A single random block permutation was generated and applied consistently to all subjects and all frames. This preserves the local texture and activation pattern inside each block, but disrupts the global arrangement of cortical regions. Thus, this control tests whether local patch-level fMRI patterns alone are sufficient when large-scale cortical topology is scrambled.

Cortical-pixel permutation. For the cortical-pixel permutation control, we identified valid cortical pixels using the flatmap alpha mask and randomly permuted the values within this cortical mask. The same permutation was applied consistently across all subjects and frames. This preserves the rendered intensity distribution within each frame and leaves the background unchanged, while disrupting local image neighborhoods and the correspondence between cortical location and signal values. Therefore, this is a stronger geometry disruption than block-shuffling.

FC heatmap. For the FC heatmap control, we replaced frame-wise flatmaps with a subject-level functional connectivity image. For each subject, one 200-frame Schaefer100 ROI time series was used to compute a 100×100 Pearson correlation matrix after ROI-wise normalization. The resulting FC matrix was clipped to $[-1, 1]$, rendered as a red–blue heatmap, resized to the same image canvas, and repeated as 40 identical frames. This control keeps subject-level functional information but removes cortical surface geometry and frame-wise flatmap dynamics.

Random image. For the random-image control, each frame was replaced by a random RGB image with the same spatial size as the flatmap input. Random images were generated with deterministic subject- and frame-specific seeds. This control preserves only the image-format interface to the frozen visual encoder, while removing both fMRI signal and cortical geometry.

As an additional robustness check, we repeat the geometry-control ablation with a frozen DINOv2 encoder. Table 12 shows the same qualitative trend as the main SigLIP2 geometry-control experiment.

C.2 Details of ROI and voxel reference models

Table 3 compares a regional time-series model with native voxel models using standard or adaptive patching. The models use subject-level train/validation/test partitions and 40-frame inputs.

Native voxel-volume processing. For the voxel-volume setting, we use preprocessed 4D fMRI scans and a 40-frame sampling strategy, retaining voxel-wise signals. Voxels outside the selected brain or ROI mask are set to zero, and temporal z-score normalization is applied voxel-wise. This setting keeps fine-grained voxel-level spatial variation and serves as a stronger geometry-preserving volumetric reference.

Table 12: Geometry-control ablation on HCP sex classification. All results use the frozen DINOv2 encoder with one global token per frame, followed by an independently trained MLP with the same architecture and training protocol for every condition. Results are reported as mean±std over three seeds. Significance markers indicate paired t-tests over the three probe-seed results, comparing Real flatmap against each control after Holm correction. **Red** indicates the best performance.

Control input	ACC ↑	wF1 ↑	Balanced ACC ↑	Macro F1 ↑
Real flatmap	**80.37 ± 2.55**	**80.33 ± 2.53**	**80.08 ± 2.47**	**80.15 ± 2.53**
Spatial block-shuffle	$75.47 \pm 0.78^{*}$	$75.49 \pm 0.76^{*}$	$75.44 \pm 0.70^{*}$	$75.34 \pm 0.75^{*}$
Cortical-pixel permutation	$73.77 \pm 2.80^{*}$	$73.80 \pm 2.81^{*}$	$74.06 \pm 2.61^{*}$	$73.73 \pm 2.76^{*}$
FC heatmap	$54.99 \pm 5.08^{**}$	$55.04 \pm 5.07^{**}$	$54.80 \pm 5.08^{**}$	$54.77 \pm 5.07^{**}$
Random image	$49.24 \pm 1.52^{**}$	$49.33 \pm 1.52^{**}$	$49.24 \pm 1.54^{**}$	$49.14 \pm 1.53^{**}$

Markers denote Holm-corrected paired t-tests against Real flatmap: $^{*}p < 0.05$, $^{**}p < 0.01$.

Voxel models and patching. The voxel models use a ViT-Small backbone. We compare standard patching with adaptive patch allocation following Omni-fMRI [3].

40-frame ROI time-series processing. For the ROI time-series setting, we convert each 4D resting-state scan into a region-level temporal representation using the same ROI definition. At each time point, voxel signals within each ROI are averaged to obtain a regional activity vector. For subject i, this produces a time series

$$R_i = [r_{i,1}, r_{i,2}, \ldots, r_{i,T}], \quad r_{i,t} \in \mathbb{R}^K,$$

where K is the number of ROIs. The regional signals are temporally z-scored within each ROI, and the same 40-frame sampling strategy is applied to obtain

$$R_i^{40} \in \mathbb{R}^{40 \times K}.$$

This representation removes within-ROI spatial geometry and keeps only region-level temporal dynamics. We feed the 40-frame ROI sequence to a lightweight temporal model using the same subject-level train/validation/test partitions as the voxel reference models.

All experiments are performed at the subject level. Frames from the same subject are never split across train, validation, and test sets, preventing frame-level leakage.

D Additional Evaluation Protocols and Results

D.1 Native Image Dimensions

Table 13 lists the native width×height of the RGB flatmaps constructed in Section 3.2, before SigLIP2-NaFlex processing.

Table 13: Native cropped flatmap dimensions before encoder processing.

Input region	Width×height
Whole cortex	943×384
HCP-MMP visual	273×264
NSDgeneral, subject 1	293×208
NSDgeneral, subjects 2 and 5	293×209
NSDgeneral, subject 7	292×209

D.2 Surface-Model Implementations

CortexMAE-F. We use the released CortexMAE-F checkpoint and its standard surface preprocessing pipeline, pooling frozen backbone features into a subject-level representation for the downstream probe. Pretraining includes HCP-YA; the ADNI tasks evaluate transfer to a different cohort.

SiT-style parcel-token control. The SiT-style results in Table 5 use cortical time series read from fsLR CIFTI dtseries files. Schaefer1000 parcel averages are grouped into 320 connected surface patches, which are processed by a Transformer with width 192, depth 12, and three attention heads. The reported configuration uses learnable positional embeddings and is trained end to end with seeds 42, 43, and 44. It operates on surface signals without color mapping or image rendering.

Vertex-level spherical patching. For the additional vertex-level implementation, GIFTI spherical surfaces provide the geometry for resampling fsLR signals to ico6. We then use the official SiT ico2 indices, with 320 patches per hemisphere and 153 ico6 vertices per patch. In resting-state inputs, the 40-frame values within each patch are projected to a 192-dimensional token. The two hemispheres share the Transformer, and their logits are averaged at evaluation. For NSD, beta responses are used in place of temporal windows, and the visual or NSDgeneral mask selects the input support.

D.3 Measured-Confound Sensitivity

We fit covariate encodings and regression models on the training split only. Let H denote the frozen feature matrix and A the encoded covariates, including an intercept. We estimate a regression coefficient matrix B from $(A_{\text{train}}, H_{\text{train}})$ and construct adjusted features $H^{\text{adj}} = H - AB$ using the same fitted coefficients for validation and test subjects. We then standardize the features using training statistics and train the classification probe. The diagnostic labels remain the classification targets.

Table 14 reports each covariate adjustment alongside its task-specific raw reference. The PPMI raw row is the re-evaluation paired with its covariate analyses.

Table 14: **Measured-confound sensitivity.** Mean±std (%) over three probe seeds. ΔwF1 is the percentage-point change from the raw row for the same task.

Task	Adjustment	ACC	wF1	ΔwF1
HCP sex	Raw	82.06 ± 2.05	82.04 ± 2.05	Ref.
	Age	80.88 ± 1.06	80.88 ± 1.07	-1.16
ADNI MCI/CN	Raw	61.11 ± 1.48	60.75 ± 1.16	Ref.
	Age + sex	59.40 ± 0.74	59.03 ± 0.38	-1.72
	Age + sex + site	57.69 ± 0.00	56.47 ± 0.60	-4.28
	Age + sex + site + protocol	58.12 ± 1.96	57.69 ± 1.57	-3.06
ADNI AD/CN	Raw	75.46 ± 2.12	75.41 ± 2.11	Ref.
	Motion	75.00 ± 1.39	75.34 ± 1.56	-0.07
	Age + sex + motion	69.44 ± 4.81	69.96 ± 4.73	-5.45
PPMI three-way	Raw re-evaluation	54.51 ± 5.35	55.59 ± 4.20	Ref.
	Age + sex	57.29 ± 3.13	57.49 ± 3.62	$+1.90$
	Age + sex + acquisition/cohort	53.47 ± 3.35	52.52 ± 3.19	-3.07

For the MCI/CN age+sex+site condition, all three seeds yield the same number of correct test predictions, producing zero standard deviation in accuracy. Their error patterns differ, producing a nonzero standard deviation in weighted F1. Site identifiers and protocol descriptions are encoded as their respective metadata fields. Signal-quality proxies, when used in the volume controls, refer to volume-derived DVARS and global-signal summaries.

D.4 Alternative Forty-Frame Windows

Table 15 compares TR0–39 and TR98–137 using the same encoder and probe protocol. MCI/CN is re-evaluated on both windows. The alternative AD/CN window uses 243/34/72 training/validation/test subjects: one training acquisition contains only 97 TRs, while validation and test membership is unchanged.

Table 15: **Alternative 40-TR windows.** Accuracy and weighted F1 (%). The first-window MCI/CN row is the paired re-evaluation for this experiment.

Task	Window	ACC	wF1
ADNI MCI/CN	TR0–39	62.39	62.69
	TR98–137	62.82	63.20
ADNI AD/CN	TR0–39	75.46	75.41
	TR98–137	73.15 ± 2.89	73.35 ± 3.10

D.5 Class Distribution and Probe Sensitivity

The HCP feature cohort used for these supplementary controls contains 1,000 subjects, with the class distribution shown in Table 16. On this evaluation, the main MLP achieves 81.81 ± 2.03 balanced accuracy and 81.88 ± 2.06 macro F1, in addition to the reported accuracy and weighted F1.

Table 16: HCP class distribution for the supplementary frozen-feature evaluations. Class labels follow the saved split manifest.

Split	Class 0	Class 1	Total
Training	321	281	602
Validation	107	94	201
Test	107	90	197

For Table 17, we replace the MLP with a single trainable linear layer, retaining the frozen features, subject splits, training-set standardization, and validation-based checkpoint selection.

Table 17: **Probe sensitivity on resting-state tasks.** Accuracy / weighted F1 (%), mean±std over three runs. PPMI uses the raw re-evaluation paired with the linear experiment.

Task	Main MLP	Linear probe
HCP sex	82.06 ± 2.05 / 82.04 ± 2.05	84.77 ± 1.34 / 84.77 ± 1.34
ADNI MCI/CN	61.11 ± 1.48 / 60.75 ± 1.16	50.43 ± 6.45 / 50.61 ± 6.41
ADNI AD/CN	75.46 ± 2.12 / 75.41 ± 2.11	70.37 ± 10.79 / 71.10 ± 10.59
PPMI three-way	54.51 ± 5.35 / 55.59 ± 4.20	44.10 ± 0.60 / 47.75 ± 0.39

D.6 Retrained and Fixed Readouts

For the retrained controls, a new MLP or linear probe is fitted independently for every input condition, using the same architecture, optimizer, splits, and seeds. For the fixed-readout controls, we train the classifier only on real-flatmap features and then apply it unchanged to perturbed features, retaining the real-training-set standardizer. The real-feature endpoint reproduces the original saved predictions before cross-condition evaluation.

Table 18 reports both readout protocols for the same three perturbations.

D.7 Matched Cortical-Region Baselines on NSD

The same-mask comparison in Table 19 uses the four NSD subjects and image splits from Section 4.2. ROI-linear averages responses within masked HCP-MMP parcels and fits a linear classifier; the small MLP uses adaptively pooled pixels from the masked flatmaps. FlatClip achieves higher weighted F1 than both baselines under both masks and higher mAP on the Visual mask. On NSDgeneral, ROI-linear and FlatClip obtain comparable mAP.

Table 18: **Readout controls on HCP.** Weighted F1 (%), mean±std over seeds 42, 43, and 44. Retrained probes are fitted separately for each condition; fixed probes are fitted only on real flatmaps.

Input	Retrained MLP	Fixed MLP	Retrained linear	Fixed linear
Real flatmap	82.04 ± 2.05	82.04 ± 2.05	84.77 ± 1.34	84.77 ± 1.34
Spatial block shuffle	76.18 ± 1.83	28.65 ± 0.00	77.53 ± 1.05	32.18 ± 3.48
Cortical-pixel permutation	72.87 ± 1.97	28.65 ± 0.00	76.43 ± 1.41	28.65 ± 0.00
Phase randomization	64.49 ± 2.14	29.74 ± 1.89	65.87 ± 3.09	28.65 ± 0.00

Table 19: **Same-mask baselines on NSD.** Mean mAP and weighted F1 (%) across the four evaluated subjects.

	Visual		NSDgeneral	
Method	mAP	wF1	mAP	wF1
ROI-linear	14.28	22.91	**16.45**	23.80
Small MLP	14.06	24.66	15.18	26.04
FlatClip	**15.84**	**26.92**	16.35	**26.97**

D.8 Direct Acquired-Volume Renderings

HCP. The mean-projection control renders the nonzero-voxel axial mean of each $96 \times 96 \times 96$ volume as a 96×96 image. The slice-wise control retains each non-empty axial slice and encodes it independently. We mean-pool patch features within a slice, then average slice embeddings within each time point and the resulting embeddings across the first 40 TRs. Both produce a 768-dimensional subject feature for the same MLP protocol. The slice-wise method requires one image-encoder pass per slice rather than one per cortical frame; the retained HCP volumes contain a median of 77 non-empty slices per TR.

Table 20: **HCP direct-volume 40-frame results.** Accuracy and weighted F1 (%), mean±std over three probe seeds.

Representation	ACC	wF1
Axial mean projection	72.93 ± 1.63	72.81 ± 1.75
FlatClip cortical flatmap	82.06 ± 2.05	82.04 ± 2.05
Independent slices, then temporal mean	89.34 ± 1.34	89.32 ± 1.34

NSD subj01. We compare cortical flatmaps with a three-orthogonal-view mean-projection montage and independent axial slices. For the slice-wise condition, the three GLM-beta occurrences of each of the 10,000 images are first averaged voxel-wise. Each resulting volume has 80 non-empty axial slices. Each 96×96 slice is processed by frozen SigLIP2, and its patch features are pooled into one embedding; the 80 slice embeddings are then averaged into an image-level feature. The split contains 8,550 training, 450 validation, and 1,000 shared test images.

D.9 Omni-fMRI with Native Temporal Inputs

We evaluate the released Omni-fMRI checkpoint using official NSD subj01 1.8-mm preprocessed BOLD time series. Each target image has three presentations. We construct two 40-frame inputs:

1. **Continuous BOLD40.** A complete continuous 40-TR window is encoded for each presentation. The three presentation-level CLS features are averaged to obtain one feature per image.
2. **Event stack.** Response-centered 13-TR segments from the three presentations are concatenated and padded with one zero channel. The complete $3 \times 13 + 1$ stack is encoded once to produce an image-level CLS feature.

Table 21: **NSD subj01 direct-volume comparisons.** mAP and weighted F1 (%). Original FlatClip and mean-projection values are reported alongside the three-seed slice-wise experiment.

Input	mAP	wF1
FlatClip, whole cortex	11.100	24.576
FlatClip, Visual	14.050	25.829
FlatClip, NSDgeneral	15.311	26.922
Three-view mean projection	4.95	17.00
Independent axial slices	4.342 ± 0.171	15.922 ± 0.579

Both temporal constructions use 10,000 images with a 7,000/1,000/2,000 training/validation/test split. We train a class-weighted BCE MLP with seeds 42, 43, and 44, select checkpoints by validation weighted F1, and evaluate the test set with a fixed threshold of 0.5. NSD's rapid event-related design produces overlapping BOLD responses: continuous windows retain acquisition order, whereas event stacks concatenate segments from separate presentations.

Table 22 reports the two temporal experiments alongside the original beta-map results. The original FlatClip row uses the 8,550/450/1,000 image split and validation-mAP selection; the temporal rows use the shared protocol specified above.

Table 22: **Omni-fMRI temporal input adaptations on NSD subj01.** mAP and weighted F1 (%). Original beta-map results and new temporal-input results are grouped separately. Temporal rows report mean±std over three seeds.

Input	mAP	wF1
Original beta-map results		
FlatClip–NSDgeneral	15.311	26.922
Omni-fMRI, repeated beta $\times 40$	3.949	13.818
Native temporal inputs: 7,000/1,000/2,000 split		
Omni-fMRI, continuous BOLD40	3.804 ± 0.052	16.531 ± 0.357
Omni-fMRI, event stack $3 \times 13 + 1$	3.840 ± 0.035	16.188 ± 0.183

D.10 Labeled-Subject Scaling

We use cached HCP features and keep the validation and test subjects in Table 16 fixed. Class-stratified subsets contain 10%, 25%, 50%, or 100% of the training subjects and are nested within each seed. Standardization is fitted on each training subset, and checkpoints are selected by validation weighted F1. Table 23 reports both probes at each training fraction.

Table 23: **HCP labeled-subject scaling.** Accuracy / weighted F1 (%), mean±std over three seeds.

Training fraction	Subjects	Linear	MLP
10%	60	70.05 ± 4.51 / 70.01 ± 4.47	70.39 ± 2.55 / 69.92 ± 3.04
25%	150	77.83 ± 2.55 / 77.83 ± 2.59	75.97 ± 4.93 / 75.89 ± 4.96
50%	300	82.91 ± 0.29 / 82.89 ± 0.27	79.19 ± 1.76 / 79.11 ± 1.71
100%	602	84.77 ± 1.34 / 84.77 ± 1.34	82.06 ± 2.05 / 82.04 ± 2.05

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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: Limitations have been discussed in the paper.

   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 include theoretical results.

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 details of experiments are provided.

   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: Code is provided in supplementary.

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: Details are provided.

   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: Statistical significance is calculated and provided.

   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: Details of compute resources are provided.

   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: yes.

   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: [N/A]

    Justification: It's a general model research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets used in this project is licensed.

    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: [Yes]

    Justification: Code and documents are provided.

    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:

    Guidelines: The paper does not involve crowdsourcing.

    - 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 paper does not involve crowdsourcing nor research with human subjects.

    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: LLM is used only for writing, editing.

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.