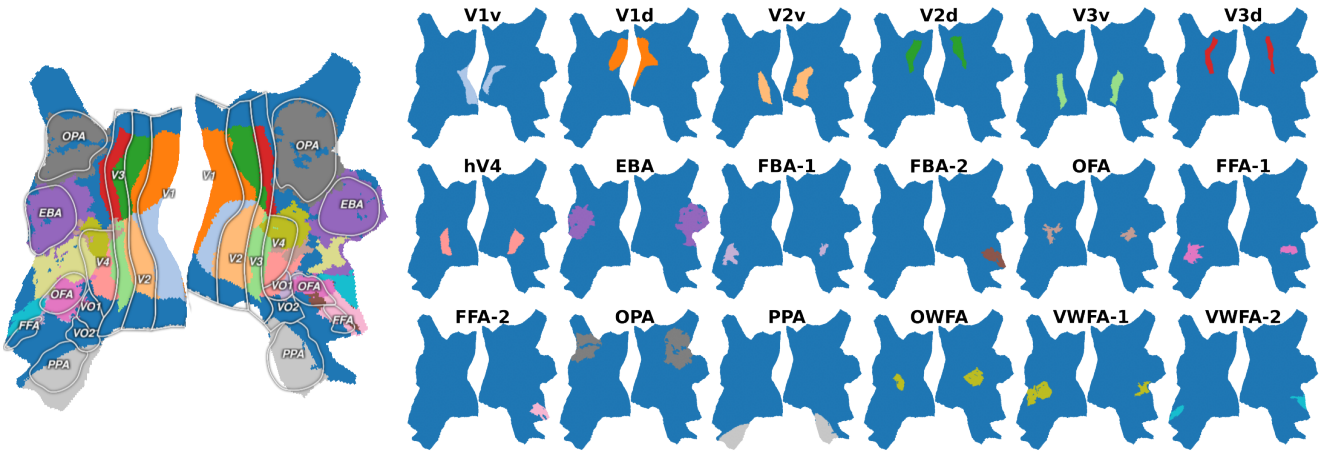


Figure 11. **Brain Region of Interests (ROIs)**. ***Left***: color is subject-specific ROI, and border overlay is subject-average common template ROIs. ***Right***: subject-specific ROIs. V1v: ventral stream, V1d: dorsal stream.

ROI name	V1 V2 V3	V4	EBA FBA	OFA FFA	OPA	PPA	OWFA VWFA
Known Function/Selectivity	primary visual	mid-level	body	face	navigation	scene	words

Table 4. Known function and selectivity of brain region of interests (ROIs).

A. Appendix Overview

1. Table 5 is an overview of key findings in this work.
2. Appendix B summarizes known function of brain ROIs.
3. Appendix C lists details of pre-trained models.
4. Appendix D is extended results with more ViT sizes, diffusion time steps, and more subjects.
5. Appendix E is implementation details of data processing and model training, pseudocode for visualization.
6. Appendix F summarizes state-of-the-art methods, ablation study of our methods.
7. Appendix G demonstrates the resulting brain-to-network mapping when trained with less data samples.

B. Brain Region Details

This section briefly summarizes the known functionality of brain regions of interest (ROI). In our primary result, we included numerical results for V1, V2, V3, OPA, PPA, EBA, and FFA. In this appendix, we further report numerical results on FBA, OFA, OWFA, and VWFA.

Figure 11 is an overview of brain ROIs. We used subject-specific ROIs provided by NSD [2], NSD defined subject-specific ROIs by population receptive field (prf) and functional localizer (floc) experiments. It's worth noting that common template ROIs are different from subject-specific ROIs.

Table 4 is known function and selectivity for each ROI. Briefly, V1 to V3 is the primary visual stream, they are further divided into ventral (lower) and dorsal (upper) streams. V4 is the mid-level visual area. EBA (extrastriate body area) and FBA (fusiform body area) are body-selective regions, FFA (fusiform face area) and OFA (occipital face area) are face-selective, OWFA (occipital word form area) and VWFA (visual word form area) are words selective. PPA (parahippocampal place area) is scene and place selective, and OPA (occipital place area) is related to navigation and spatial reasoning.

Key Observations	Sections	Figures & Tables
Brain Score		
MAE and SAM are relatively better for the early visual brain, CLIP and DiNOv2 are relatively better for high-level brain regions.	4.1, D.1	Fig. 4; Tab. 1, 13
SD late time-steps are uniformly good for all brain regions.	4.1, D.1	Fig. 4; Tab. 1, 13
SD late time-steps are better for early brain, mid-to-late time steps are better for late brain.	D.1	Tab. 13
Brain-Net Alignment		
Across all models included in this study, CLIP has the best brain-alignment.	4.2, 4.3	Fig. 5; Tab. 2
ImageNet and SAM last layer align to the mid-level visual brain, classification and segmentation are mid-level brain tasks.	4.2, D.2	Fig. 5, 15
DiNOv2 and MAE last layer does not align to any brain region, mask reconstruction deviates from the brain's task.	4.2, D.2	Fig. 5, 15
MoCov3 last layers align better with the late ventral stream ('what' part) than the dorsal stream ('what' part), self contrastive learning is more on semantics than spatial relationship.	4.2, D.2	Fig. 5, 15
CLIP and ImageNet early layers align with the early visual brain, SAM DiNOv2 MAE MoCov3 early layers deviate from the brain.	4.2, D.2	Fig. 5, 14
SD have less separation in layers but more in time steps. SD encoder layers have more separation than decoder layers. SD's brain-alignment is more 'soft' compared to ViT models. SD final time steps align to early brain regions, SD mid-late time steps align to the late brain.	4.2, D.2	Fig. 6, 16
Model Sizes		
CLIP's brain-net alignment improved as CLIP scaled up size and training data. In bigger CLIP models, both early and late layers become more aligned with the brain.	4.3, D.3	Fig. 5, 17
SAM, ImageNet, DiNOv2, MoCov3, and MAE's brain-net alignment decreased as they scaled up sizes. ImageNet and DiNOv2 bigger models' early layers deviate from the brain; SAM, MAE, and MoCov3 bigger models' late layers deviate from the brain.	4.3, D.3	Fig. 5, 18-22
Fine-tuning		
CLIP maintained brain-alignment after fine-tuning, DiNOv2 and MAE re-wired late layers.	4.4	Fig. 7, Tab. 3
Fine-tuning performance does not correlate to change of computation layout, CLIP had the best fine-tuning performance but DiNOv2 and MAE also had competitive performance.	D.4	Tab. 14
Channels and Brain ROIs		
Early visual brain uses similar channels but diverse spatial tokens. Late visual brain use diverse channels and global token.	4.5	Fig. 8, 3
The top selected channels reveal brain ROIs' function. Image space features also reveal differences in various pre-trained models.	D.5	Fig. 9, 29-34
Methods and Consistency		
Consistent subject difference exists in both brain prediction score and brain-net alignment.	D.1, 4.2	Fig. 12, 4; Tab. 13
Brain-Net mapping is consistent across random seeds within the same subject.	D.2	Fig. 13
Brain-Net mapping can be trained with limited training data samples. 3K data samples is a good trade-off for speed and quality.	G	Fig. 28, 27

Table 5. Overview of key observations in this work.

C. Pre-trained Model Details

This section briefly summarizes the models included in this study. All the models are ViT architecture except for U-Net Stable Diffusion. We primarily used models released by their original authors, we used models from third-party releases when size variants are unavailable from the official release. We did not run any pre-training ourselves.

Model	Layers	Width	Input Size	Patch Size	Training Data
CLIP XL	24	1024	224x224	14x14	DataComp-1B
CLIP L	12	768	224x224	16x16	DataComp-140M
CLIP M	12	768	224x224	32x32	DataComp-14M
CLIP S	12	768	224x224	32x32	DataComp-1.4M

Table 6. CLIP Models.

CLIP The objective of CLIP [18] (Contrastive Language-Image Pre-Training) is to match images with their corresponding text captions. The training objective is to minimize a contrastive loss that increases the similarity of paired images and text but decreases for unpaired ones. CLIP has two branches, one for vision and one for text, we only used the vision branch. We used a model released from the OpenCLIP [12] repository, models are pre-trained on data from DataComp [9]. Size variants of CLIP were trained on different sub-samples of data from 1B to 1.4M samples.

Model	Layers	Width	Input Size	Patch Size	Training Data
SAM H	32	1280	1024x1024	16x16	SA-1B
SAM L	24	1024	1024x1024	16x16	SA-1B
SAM B	12	768	1024x1024	16x16	SA-1B

Table 7. SAM Models.

SAM The objective of the Segment Anything Model (SAM) [13] is interactive segmentation with points, boxes, or text prompts as additional input. SAM was trained without the class label of the objects, but the text prompts (CLIP embeddings) enhanced SAM's understanding of the semantics. SAM was initialized from the MAE H model. Training was done on the SA-1B dataset, which was built by the SAM authors. SAM is an encoder-decoder design, we only took features from the encoder part. SAM's ViT architecture does not have a class token, we used global averaging pooling to replace the global token. We used the officially released model weights for SAM.

ImageNet This fully supervised model was trained to predict ImageNet [8] labels, the training was done on ImageNet-1K from scratch without any pre-training. We used model weights released by PyTorch Hub. We used a model from the improved training recipe that covers state-of-the-art training tricks and augmentations. Specifically, we used the IMAGENET1K_V1 weights, the base size model has 81.9 ImageNet accuracy, large size model has 79.7 accuracy.

Model	Layers	Width	Input Size	Patch Size	Training Data
ImageNet L	24	1024	224x224	16x16	IN-1K
ImageNet B	12	768	224x224	16x16	IN-1K

Table 8. ImageNet Models.

DiNOv2 The authors describe DiNOv2 [16] as DiNOv1 [4] plus iBOT [23] with the centering of SwAV [3]. DiNOv2 was trained with momentum self-distillation and mask reconstruction of latent tokens. The training was done on LVD-142M, which is a custom dataset made by the DiNOv2 authors. One notable difference to other models is that DiNOv2 smaller models were distilled from bigger models. We used the officially released model weights for DiNOv2.

Model	Layers	Width	Input Size	Patch Size	Training Data
DiNOv2 G	40	1536	224x224	14x14	LVD-142M
DiNOv2 L	24	1024	224x224	14x14	LVD-142M
DiNOv2 B	12	768	224x224	14x14	LVD-142M

Table 9. DiNOv2 Models.

MoCov3 The Momentum Contrastive (MoCo) [5] method trains contrastive loss with a momentum teacher encoder, which is an exponential moving average of the previous iteration models. The constrastive objective is to enforce the encoder to generate a similar representation to the momentum model. The training was done with the ImageNet-1K dataset. We used MoCov3 model weights released by MMPreTrain.

Model	Layers	Width	Input Size	Patch Size	Training Data
MoCov3 L	24	1024	224x224	16x16	IN-1K
MoCov3 B	12	768	224x224	16x16	IN-1K
MoCov3 S	12	384	224x224	16x16	IN-1K

Table 10. MoCov3 Models.

MAE The Mask Autoencoder (MAE) [11] objective is to reconstruct the masked patches of input images given the un-masked patches, reconstruction is in the image space. The training was done on the ImageNet-1K dataset. MAE used an encoder and decoder design, we only studied the encoder part. We used the official release from the original authors.

Model	Layers	Width	Input Size	Patch Size	Training Data
MAE H	32	1280	224x224	16x16	IN-1K
MAE L	24	1024	224x224	16x16	IN-1K
MAE B	12	768	224x224	16x16	IN-1K

Table 11. MAE Models.

SD The Stable Diffusion (SD) [19] model's objective is to generate photo-realistic images. Although SD was trained without supervision on the loss term, the content of the generated image is controlled by a text prompt (CLIP embeddings), and the text prompt enhanced the semantic understanding of the features. SD is a U-Net and ResNet design with cross-attention to CLIP embeddings. There are 8 layers in the U-Net encoder and 12 layers in the decoder, skip connection connects the encoder and decoder blocks. There's no class token in SD, we used global averaging pooling to replace it. In the feature extraction, we used an empty text prompt, we followed the 'inversion' time steps that chain the features of different time steps. SD was trained on LAION-5B [20] dataset. We used the Huggingface release of the SD version 1.5 model.

Encoder Layers	Decoder Layers	Width	Feature Size	Input Size	Training Data
1,2	10,11,12	320	64x64	512x512	LAION-5B
3,4	7,8,9	640	32x32	512x512	LAION-5B
5,6	4,5,6	1280	16x16	512x512	LAION-5B
7,8	1,2,3	1280	8x8	512x512	LAION-5B

Table 12. Stable Diffusion Layers.

D. Extended Results

In addition to the main results in Section 4 , this appendix presents extended results that cover more brain ROIs, more ViT model sizes, more diffusion time steps, and more subjects. The structure of this appendix section follows the main results:

1. **Brain Score**. Results on three subjects. Numerical results on more ROIs, all diffusion time steps.
2. **Training Objectives and Brain-Net Alignment**. Consistency check. Display of raw layer selector weights.
3. **Network Hierarchy and Model Sizes**. Layer selector results in more ViT model size variants.
4. **Fine-tuned Models**. Fine-tune performance score.
5. **Channels and Brain ROIs**. Top channel image feature display on more brain ROIs.

D.1. Brain Score

In addition to the brain score reported in main results Section 4.1 , we report 1) CLIP brain score on three subjects, 2) Numerical brain score results of ViT base size model on more ROIs, and 3) Stable Diffusion brain score results that cover the full time-step range. Also, in main results we reported the root summed square difference of brain score, in this appendix, we report the raw ROI-averaged brain score.

Three subjects In Figure 12 and Table 13, subject #2 has a more predictable V1 while subject #3 has a least predictable early visual cortex. Subject #1 has a most predictable FFA and FBA. The prediction score difference matches the brain-to-network mapping results that subject #3 has large uncertainty in the early visual cortex (Figure 13), and subject #2 and #3 are missing the FFA region that subject #1 has. Overall, individual difference is expected and consistent.

ViT models In Table 13, we report the raw ROI-average brain score. Among the ViT models, MAE has the best prediction power in early visual (V1 to V3) and navigation and spatial-relation region OPA. Interestingly, MAE has the best score in word and letter region OWFA but not for VWFA. CLIP has the best score in face, body, and scene-related regions (EBA, FBA, OFA, FFA, PPA) followed by DiNOv2.

Diffusion time steps In Table 13, we report brain score fixing each diffusion time step. $T < 25$ showed a suboptimal performance score in all regions. $T = 35$ showed the best performance on high-level regions (EBA, FBA, OPA, PPA), and $T = 45$ showed good performance for all regions from early visual to high-level. Surprisingly, $T = 0$ achieved relatively good brain score for the early visual ROIs.

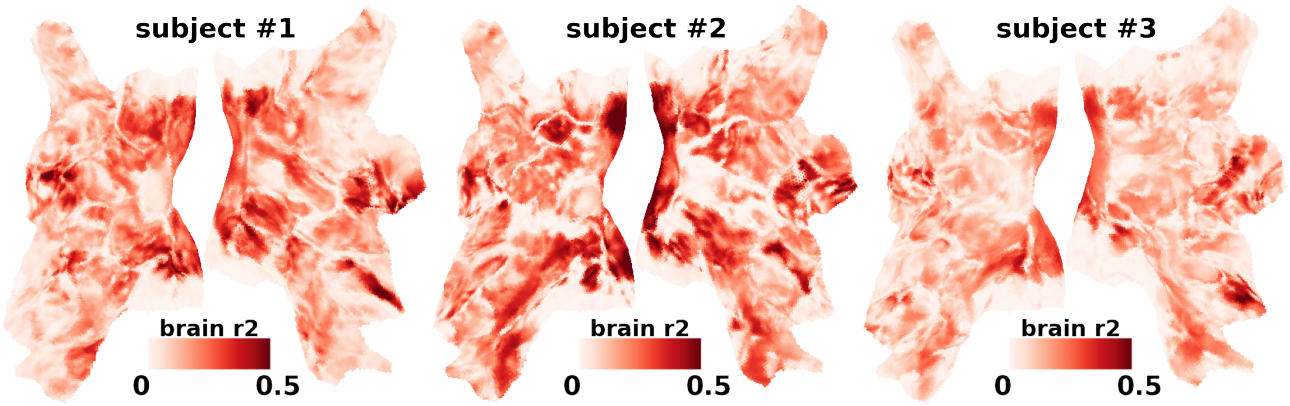


Figure 12. **Brain Score**. Three subjects, CLIP L (base size 12 layer) model.

	ROI Brain Score $R^2 \uparrow$ ($\pm$ 0.001)												
Model	**all**	**V1**	**V2**	**V3**	**V4**	**EBA**	**FBA**	**OFA**	**FFA**	**OPA**	**PPA**	**OWFA**	**VWFA**
CLIP model, three subjects													
CLIP (subject #1)	0.132	0.216	0.209	0.185	0.139	0.176	0.157	0.129	0.182	0.091	0.130	0.121	0.092
CLIP (subject #2)	0.154	0.247	0.183	0.192	0.188	0.182	0.134	0.098	0.136	0.126	0.199	0.083	0.140
CLIP (subject #3)	0.104	0.155	0.128	0.108	0.105	0.134	0.137	0.080	0.151	0.081	0.125	0.104	0.087
ViT models, subject #1													
CLIP	**0.132**	***0.216***	***0.209***	***0.185***	**0.139**	**0.176**	**0.157**	**0.129**	**0.182**	***0.091***	**0.130**	***0.121***	**0.092**
SAM	0.110	0.212	0.197	0.172	0.113	0.127	0.120	0.104	0.142	0.074	0.104	0.105	0.066
ImageNet	0.120	0.205	0.202	0.174	0.127	0.159	0.143	0.117	0.169	0.076	0.121	0.109	0.077
DiNOv2	0.126	0.208	0.202	0.175	0.127	***0.174***	***0.152***	0.122	***0.178***	0.083	***0.126***	0.111	***0.088***
MAE	***0.129***	**0.219**	**0.210**	**0.186**	***0.135***	0.165	0.148	***0.127***	0.173	**0.093**	***0.126***	**0.124**	0.086
MoCov3	0.126	0.214	0.208	0.181	0.134	0.163	0.150	0.120	0.176	0.086	0.124	0.115	0.081
Stable Diffusion time steps, subject #1													
T0	0.048	0.135	0.112	0.088	0.057	0.036	0.033	0.046	0.036	0.024	0.041	0.049	0.025
T5	0.062	0.151	0.130	0.103	0.070	0.053	0.050	0.055	0.056	0.039	0.055	0.058	0.033
T10	0.077	0.161	0.146	0.119	0.078	0.085	0.079	0.068	0.091	0.050	0.071	0.068	0.044
T15	0.095	0.187	0.169	0.141	0.097	0.111	0.105	0.085	0.123	0.063	0.090	0.083	0.055
T20	0.106	0.195	0.184	0.155	0.110	0.135	0.120	0.100	0.142	0.071	0.104	0.096	0.063
T25	0.112	0.199	0.191	0.161	0.109	0.149	0.127	0.109	0.151	0.076	0.112	0.103	0.068
T30	0.121	0.207	0.202	0.177	***0.129***	0.163	0.138	0.118	0.163	0.080	0.121	0.114	0.076
T35	**0.125**	0.212	0.205	0.178	0.128	**0.170**	**0.145**	***0.123***	***0.169***	**0.084**	**0.126**	0.118	**0.083**
T40	0.123	**0.215**	***0.207***	0.177	0.123	***0.169***	0.143	0.120	***0.169***	0.080	0.123	0.116	0.075
T45	**0.125**	**0.215**	**0.208**	**0.181**	**0.130**	**0.170**	**0.145**	**0.124**	**0.170**	***0.082***	***0.125***	***0.119***	0.078
T50	***0.124***	***0.213***	***0.207***	***0.179***	0.124	***0.169***	***0.144***	***0.123***	0.168	***0.082***	0.124	**0.120**	***0.081***

Table 13. **Brain Score**. ViT models are base size 12-layer. **Bold** marks best within each category, ***bold italic*** marks the second best. ***Top***: CLIP model on three subjects. ***Middle***: ViT models on subject #1. ***Bottom***: Stable Diffusion model time steps on subject #1. ***Insights***: 1) individual difference exists, subject #3's early visual cortex is significantly less predictable. 2) CLIP and DiNOv2 are better for late brain regions, and MAE is better for the early visual cortex. 3) Stable Diffusion T35 is better for late brain regions, and T45 is better for early visual cortex.

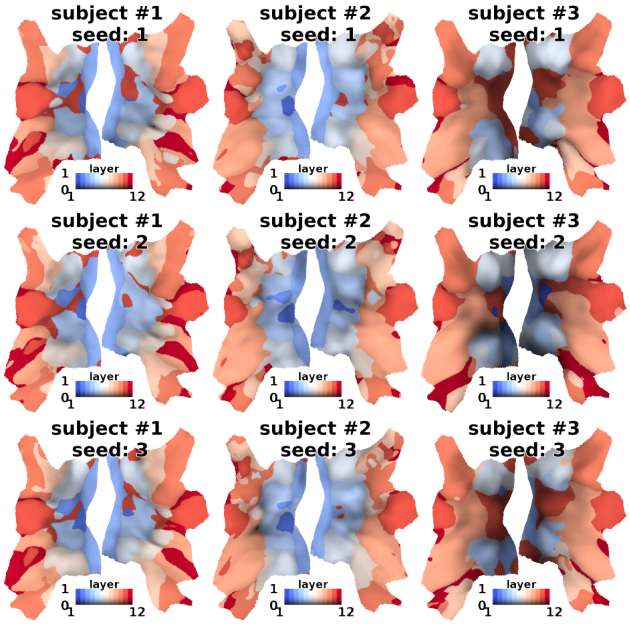
D.2. Training Objectives and Brain-Net Alignment

In this section, we show: 1) consistency of brain-net alignment across random seeds, and 2) expanded raw layer selector weights.

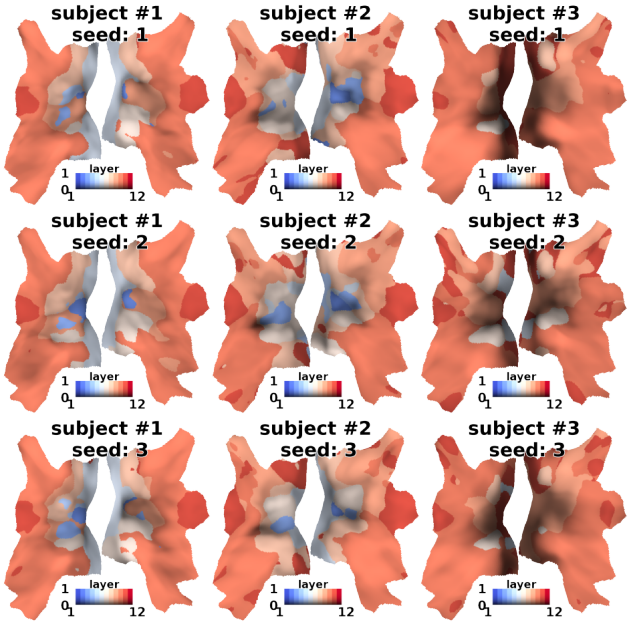
Random seed consistency In the main results Section 4.2 we found consistent differences across subjects. In this experiment, we repeated the same model and subject for 3 different random seeds. Results are in Figure 13, we found subjects #1 and #2 had consistent brain-to-layer mapping across random seeds. Subject #3 was less consistent across random seeds, note that subject #3 also had the lowest data quality (brain score, Table 13).

Raw layer selector weights In our main results Sections 3.2 and 4.2 , we displayed argmax and confidence of selected layers. In Figure 14-16, we display the raw output of layer selector weights for 1) 6 ViT base size 12-layer models, and 2) Stable Diffusion model fix time step T40 layer selection and fix decoder layer 6 time-step selection.

There are some interesting observations that are hard to conclude from the argmax plot but more visible in the raw weights: 1) CLIP layer 11 is strongly aligned to EBA but also weakly aligned to the mid-level dorsal stream. 2) ImageNet's last layer is weakly aligned to all regions expect EBA and FFA. 3) SAM's last layer is weakly aligned to the mid-to-high level dorsal stream and mid-level ventral stream. 4) DiNOv2's last two layers' alignment weakly follows layer 10. 5) MAE layer 10 strongly aligns to mid-to-high level dorsal and ventral stream, MAE last layer does not align to any brain regions. 6) MoCov3 layer 11 aligns with the late ventral stream but not the dorsal stream, and MoCov3's layer 12 aligns with EBA.



(a) CLIP



(b) MAE

Figure 13. **Random Seed Consistency**. CLIP (up) and MAE (down) model, 3 subjects (columns) and 3 random seeds (rows).

Figure 14. **Raw Layer Selector weights (Part 1)**. Layer 1 to 6 of ViT base size 12-layer models. The number tailing model name is the layer index.

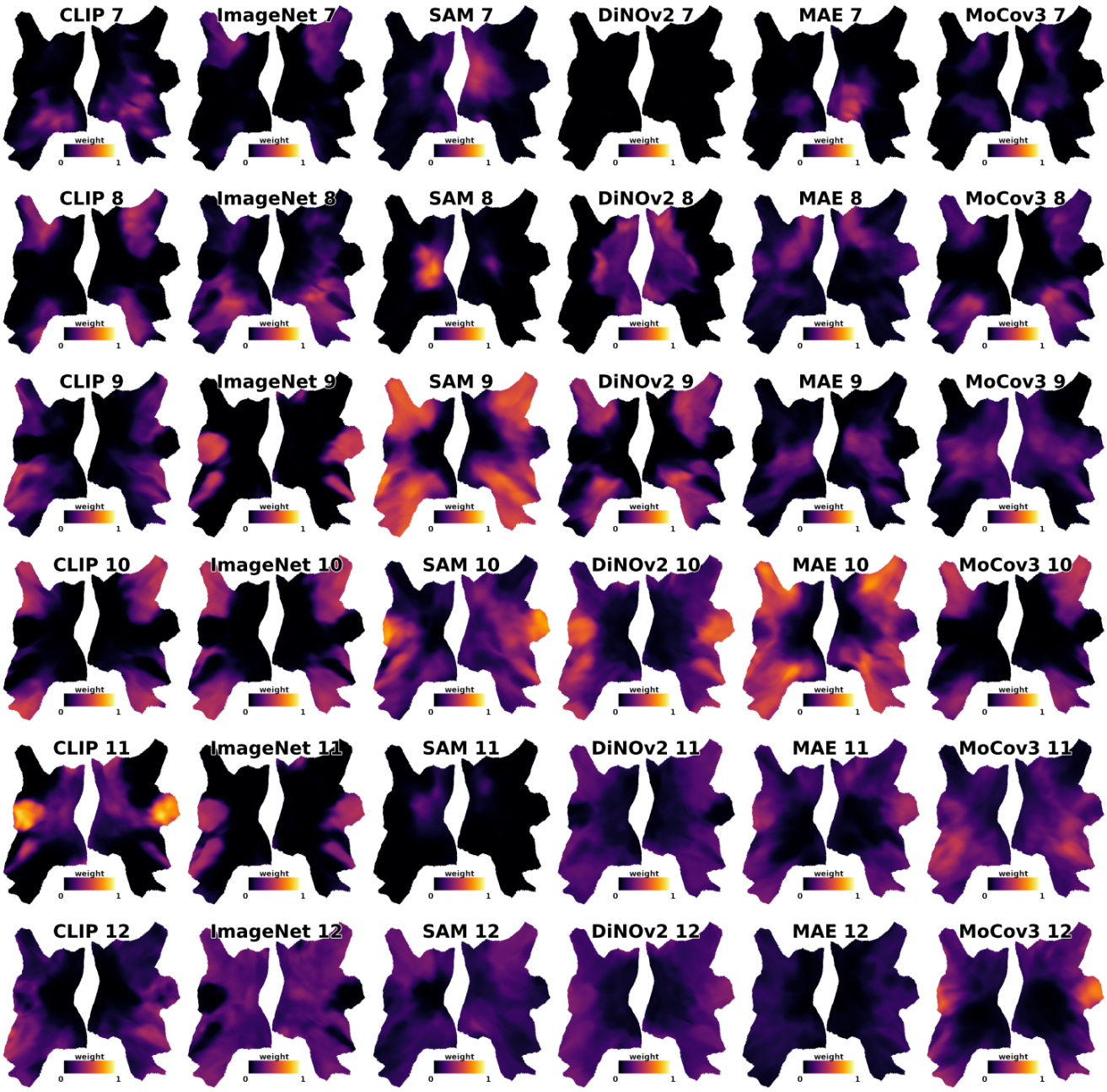


Figure 15. **Raw Layer Selector weights (Part 2)**. Layer 7 to 12 of ViT base size 12-layer models. The number tailing model name is the layer index.

Figure 16. **Raw Layer Selector weights (Part 3)**. Stable Diffusion model. SD(E): 8 encoder layers (fixed at T=40). SD(D): 12 decoder layers (fixed at T=40). SD(T*): 50 time steps (fixed at decoder layer no. 6). The number tailing model name is the layer or time step index.

D.3. Network Hierarchy and Model Sizes

In this section, we expand the main results Section 4.3 brain-layer alignment display to include more size variants. Details for pre-trained models, including layer, width, input size, patch size, and training data, are in Appendix C.

CLIP CLIP models showed increasing brain-net alignment as they scaled up both data and size. Both early and late layers in larger CLIP models are more selected by the brain. Notable, CLIP (M) and CLIP (S) were trained with the same model size but $\times 10$ smaller training samples, CLIP (S) showed low confidence selection for the whole visual brain and only the late layers were more selected.

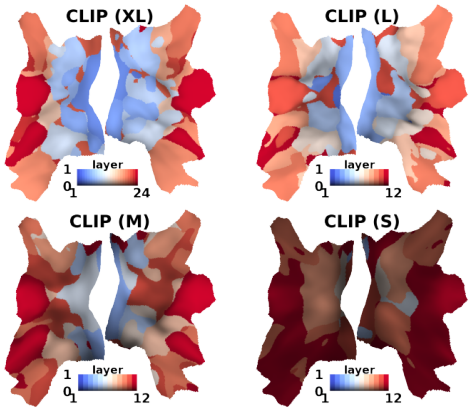


Figure 17. **CLIP Brain-Net Alignment**. XL to M are size and data variants, M and S are the same size but have smaller training data.

SAM SAM models showed decreasing brain-net alignment as they scaled up sizes. Larger SAM models' late layers were not selected; SAM's early layers were not selected in all model sizes. The uncertainty of selection went up in the early visual cortex for larger SAM models.

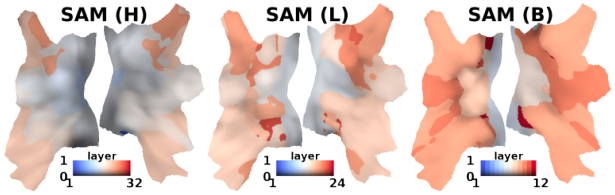


Figure 18. **SAM Brain-Net Alignment**. Size variants, same training data.

ImageNet ImageNet models showed decreasing brain-net alignment as the size scales up. Base size ImageNet model's early and late were both selected, larger size ImageNet model's early layers were not selected.

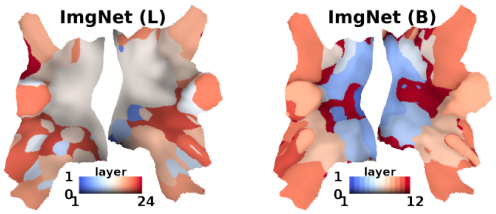


Figure 19. **ImageNet Brain-Net Alignment**. Size variants, same training data.

DiNOv2 DiNOv2 models showed decreasing brain-net alignment when scaled up. Larger DiNOv2 models' early layers were less selected, only the last 1/4 of the layers were selected for the gigantic model. The first 1/2 of the layers were not selected for DiNOv2 models of all sizes.

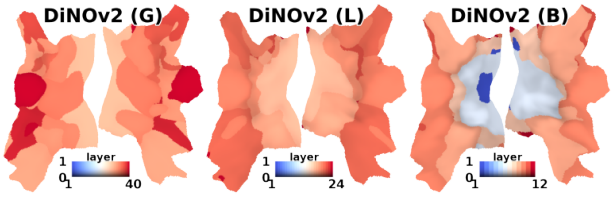


Figure 20. **DiNOv2 Brain-Net Alignment**. Size variants, same training data.

MAE MAE models showed increasing brain-net alignment from base to large, decreasing from large to huge. MAE's early layers were not selected for the base size model, selected for the large and huge size models. MAE's late layers were not selected for the huge size model, selected for the base and large models. The huge model had more separation of semantic brain regions.

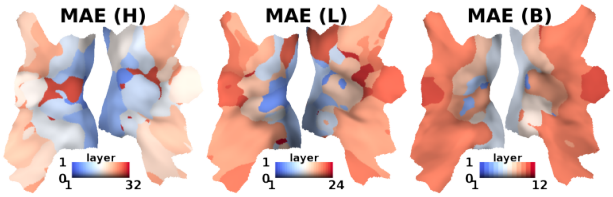


Figure 21. **MAE Brain-Net Alignment**. Size variants, same training data.

MoCov3 MoCov3 showed decreasing brain-net alignment as size scales up. MoCov3's late layers were more selected for small and base size models, and MoCov3's late layers were significantly less selected for large size models.

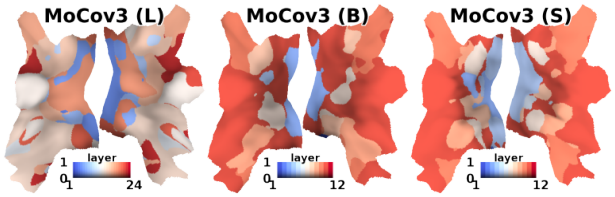


Figure 22. **MoCov3 Brain-Net Alignment**. Size variants, same training data.

D.4. Fine-tuned Model

In the main results Section 4.4 , we attached an MLP prediction head to the last layer class token and fine-tuned the whole model to ISIC and EuroSAT tasks. In our main results, we found CLIP to maintain its computation layouts after fine-tuning while SAM and DiNOv2 re-wired their late layers and surfer from catastrophic forgetting.

Brain score after fine-tuning In this appendix, we quantitatively compare brain score before and after fine-tuning. In Figure 23. Brain score of CLIP dropped from 0.131 to 0.115 after fine-tuning, DiNOv2 dropped from 0.128 to 0.085, SAM dropped from 0.111 to 0.086. The fact that CLIP dropped less brain score further support the observation that CLIP maintain computation layouts.

Fine-tune last layer performance In this appendix, we further reported the fine-tuning performance score in Table 14. CLIP had the best performance overall. Interestingly, SAM and DiNOv2 also had competitive performance despite their late layers being mostly re-wired (Section 4.4). We found the fine-tuning performance score does not correlate to the changes in brain alignments.

	Fine-tuned Accuracy ↑			
Dataset	CLIP	MAE	SAM	DiNOv2
ISIC (±0.008)	**0.640**	0.589	***0.627***	0.622
EuroSAT (±0.004)	**0.954**	0.936	0.885	***0.946***

Table 14. Fine-tuned performance score. Average of 10 runs. The whole model is fine-tuned with the prediction head attached to the last layer class token.

Grid search on which layer to fine-tune In the main results, we stated that "ISIC requires low-level features", we verify this statement in this appendix. In this experiment, we ran a grid search that attached the prediction head to each layer, layers before the prediction layer are trained, and layers after the prediction layer are discarded. In Figure 24, on ISIC, we found CLIP layer 7 reached peak performance, and other models also peaked at mid-to-late layers;

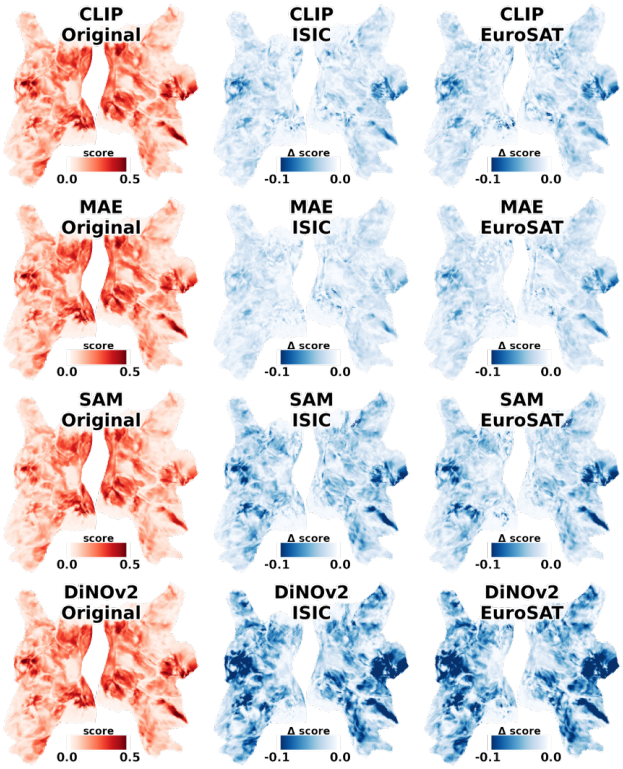


Figure 23. Brain score before and after fine-tuning on small datasets (ISIC, EuroSAT). Brain score of CLIP dropped less compare to DiNOv2 and SAM. CLIP suffer less from catastrophic forgetting.

on EuroSAT, all models' performance peaked at the last or second-last layer. Overall, the ISIC task relies on low-level features, EuroSAT task relies on high-level features.

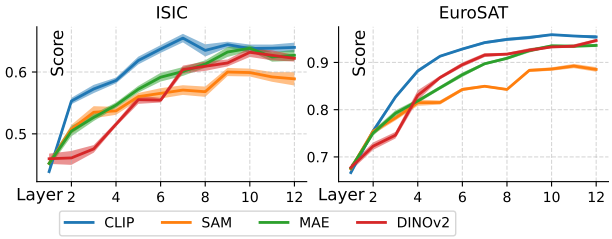


Figure 24. Grid search of fine-tuned layer. Average of 10 runs. The prediction head is attached to one layer, later layers are discarded.

D.5. Channels and Brain ROIS

In the main results Section 4.5 , we displayed the top selected channel in latent image space for V1 and FFA. In this appendix, we further display all ROIs: V1, V2, V3, V4, EBA, FBA, OFA, FFA, OPA, PPA, OWFA, and VWFA. Results are in Figure 29 to 34. Methods and pseudocode are in Appendix E.2.

Comparing across all ROIs, we found:

- from V1 to V4, features become increasingly abstract
- EBA captures the body but not including face, EBA has two global patterns for whether a human is present
- FBA captures body including face, FBA has two global patterns for whether a human is present
- OFA segments out object and background
- FFA reacts to face centered at the eyeball, FFA has two global patterns for whether a human is present
- OPA segments out the central object but not peripheral objects
- PPA reacts uniformly to the whole image
- OWFA segments out the object and background
- VWFA has two global patterns for whether a human is present

Comparing across all models, we found:

- SAM's V1 to V4 features have finer segmentation of objects, SAM's EBA activates less on bodies, SAM's OPA does not capture the global layout, and SAM's OWFA does not capture abstract representation.
- MAE's EBA activates less on bodies, MAE's FBA activates more on bodies and faces.
- CLIP's V4 has less activation on the central object, CLIP's EBA reacts to human bodies FBA reacts to animal bodies.
- DiNOv2's V3 showed grid structure, DiNOv2's EBA and FBA react to human bodies but less to animal bodies.

E. Methods Details

E.1. NSD Data Processing Details

We used the officially released GLMsingle [17] beta3 preparation of the data, the pre-processing pipeline consists of motion correlation, hemodynamic response function (HRF) selection for each voxel, nuisance regressor estimation via PCA, and finally, a general linear model (GLM) is fit independently for each voxel with selected HRF and nuisance regressor. In addition to the officially released preprocessing, we applied session-wise z-score to each voxel independently [10]. We used the official release of the data on FreeSurfer average (brain surface) space. There's a total of 327,684 vertices for the whole cerebral cortex and subcortical regions, we only used 37,984 vertices in the visual cortex defined by the 'nsdgeneral' ROI. We used coordinates of vertices in inflated brain surface space.

E.2. Model and Visualization Pseudocode Code

Model (FactorTopy) Pseudocode Listing 1 presents a PyTorch-style pseudocode for our main *FactorTopy* model. The factorized selectors in Equation 1 are implemented as separate MLPs with `tanh`, `softmax`, and `sigmoid` activation functions, respectively. `pe` is sinusoidal positional encoding.

Channel Clustering We clustered selected channels (linear regression weights $\boldsymbol{w}$) into 20 clusters in primary results Section 4.5 and Appendix D.5. The procedure for clustering is: 1) use kernel trick $\bar{\boldsymbol{w}} = \boldsymbol{w}^T\boldsymbol{w}$, $\boldsymbol{w} \in \mathbb{R}^{D\times N}$ where D is channel dimension, N is the number of voxels. 2) use k-means clustering on $\bar{\boldsymbol{w}}$ with euclidean distance, k=1000. 3) use Agglomerative Hierarchical Clustering on the k-means centroids, euclidean distance, and Ward's method, iterative merge until resulting in 20 clusters.

Channel Visualization Pseudocode In the main results Section 4.5 and Appendix D.5, we visualized the top selected channel in image space for brain ROIs. The motivation for the image space visualization is to plot the top selected channel for an ROI of voxels; voxels' linear regression weights are functioning as '*channel selection*' that answers "*which channels best predict my brain response?*".

In a single voxel case, we can 1) obtain local tokens $\mathbb{R}^{D\times H\times W}$ by summing features $\mathbb{R}^{L\times D\times H\times W}$ with layer selector weight $\hat{\boldsymbol{\omega}}_i \in \mathbb{R}^L$, 2) sum local tokens $\mathbb{R}^{D\times H\times W}$ with regression weight $w_i \in \mathbb{R}^D$, output a greyscale image $\mathbb{R}^{1\times H\times W}$.

To extend to an ROI of voxels, we 1) summed local tokens from all layer $\mathbb{R}^{L\times D\times H\times W}$ by ROI-average layer selector weights $\hat{\boldsymbol{\omega}}_* = \frac{1}{|roi|}\sum_{i\in roi}\hat{\boldsymbol{\omega}}_i$, where $|roi| = N'$, output $\mathbb{R}^{D\times H\times W}$, 2) applied PCA to reduce linear regres-

```
### FactorTopy model ###
# x: Tensor, [B, 3, 224, 224], B := batch size
# coord: Tensor, [N, 3], N := number of voxels

## 1. backbone
local_tokens, global_tokens = backbone(x)
# local_tokens: dict, {layer: [B, C, H, W]}
# global_tokens: dict, {layer: [B, C]}

## 2a. downsample, (H, W) -> (8, 8)
local_tokens = downsample(local_tokens)

## 2b. layer-unique bottleneck, C -> D
for layer in layers:
    local_tokens[layer] = bottle_neck[layer](
        local_tokens[layer]) # [B, D, 8, 8]
    global_tokens[layer] = bottle_neck[layer](
        global_tokens[layer]) # [B, D]

## 3. multi-selectors
space = tanh(space_mlp(pe(coord))) # [N, 2]
layer = softmax(layer_mlp(pe(coord))) # [N, L]
scale = sigmoid(scale_mlp(pe(coord))) # [N, 1]

## 4. get v
# get v_local
v_local = bilinear_interpolate(
    local_tokens, space) # [B, N, D, L]
# sum v_local and v_global
v_global = stack(global_tokens).repeat(1, N)
    # [B, N, D, L]
v = v_local * (1-scale) + v_global * scale
    # [B, N, D, L]
# sum over layers
v = (v * layer).sum(dim=-1) # [B, N, D]

## 5. voxel-specific linear regression
y = (v * w).mean(dim=-1) + b # [B, N]
```

Listing 1. PyTorch-style pseudocode of our methods FactorTopy.

sion weights $\mathbb{R}^{D\times N'}$ along the dimension of number of voxels N', output $\mathbb{R}^{D\times 3}$, 3) applied top 3 PC weights to local tokens to reduce the channel dimension D of local tokens, output RGB image $\mathbb{R}^{3\times H\times W}$. A complete pseudocode is in Listing 2.

E.3. Training Details

Hardware and Wall-clock We conducted experiments on a mixture of Nvidia A6000 and RTX4090 GPUs. Features of the pre-trained model are pre-computed and cached. We used bottleneck dimension $D = 128$; increasing D will significantly increase computation intensity as the number of brain voxels (vertices) is large (37,984). A full data (22K data samples) model converges in 1 to 3 hours for 12 to 40 layer models respectively. A partial data (3K data samples) 12-layer model converges in 30 minutes.

```
### top channel visualization ###
# x: Tensor, [3, 224, 224], batch size is 1
# coord: Tensor, [N, 3], N := number of voxels
# roi_mask: Tensor, [N], boolean, sum = N'

## 1. backbone
local_tokens, global_tokens = backbone(x)
# local_tokens: dict, {layer: [C, H, W]}

## 2b. layer-unique bottleneck, C -> D
for layer in layers:
    local_tokens[layer] = bottle_neck[layer](
        local_tokens[layer]) # [D, H, W]
local_tokens = stack(local_tokens)
    # [L, D, H, W]

## 3. multi-selectors
layer = softmax(layer_mlp(pe(coord[roi_mask])))
    # [N', L]

## 4. sum local_tokens by ROI
layer_weights = layer.mean(0) # [L]
local_tokens = sum(layer_weights * local_tokens)
    # [D, H, W]

## 5. PCA on linear regression weights
_w = w[:, roi_mask] # [D, N']
_pc_w = pca(_w) # [D, 3]

## 6. RGB image
image = _pc_w.t() @ local_tokens # [3, H, W]
```

Listing 2. PyTorch-style pseudocode for channel visualization.

Optimizer and Training Recipe For training brain encoding models, we used the AdamW optimizer, batch size 8, learning rate 1e-3, betas (0.9, 0.999), and weight decay 1e-2. We trained for 1,000 steps per epoch, with an early stopping of 20 epochs. Models reached maximum validation score at 40,000 to 60,000 steps, and the multi-selectors in our methods became stable after 10,000 steps. For each model, we saved the top 10 validation checkpoints and used ModelSoup [21] to average the best validation checkpoints and greedily optimize the score on the test set. We did not apply any data augmentation, existing data augmentation is not useful for brain encoding because the prediction target (brain) is not transformed alongside the input image.

Loss and Regularization We used smooth L1 loss (beta=0.1) with an additional decaying regularization term on layer selector $\hat{\omega}^{layer}$. The motivation for regularization is the use of softmax activation function in layer selector MLP leads to vanishing gradient at one-hot output, layer selector converges to a singular selection for all voxels (Figure 25) if with insufficient regularization,

$$
\begin{aligned}
loss_{reg} &= -\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum_{l=1}^{L}\hat{\omega}_{i,l}^{layer}\log\hat{\omega}_{i,l}^{layer}}{\sum_{l=1}^{L}\frac{1}{L}\log\frac{1}{L}}\right) \\
decay &= max(0, 1-\frac{step_i}{step_{total}}) \\
loss &= loss_{l1} + \lambda * loss_{reg} * decay
\end{aligned} \tag{3}
$$

N is number of voxels, L is number of layers, λ is set to 0.1, $step_i$ is the current training step, $step_{total}$ is total steps of linear decaying. In Table 15, we ran a grid search of $step_{total}$ and concluded to use a total decay step of 6000; the same total decay step is set for all models. Figure 25 shows the resulting brain-to-layer mapping when trained with less regularization decay steps. When trained with insufficient regularization, layer selection converges to a local minimum (Table 15) of selecting only the last layer (Figure 25).

It's worth noting that it's possible to optimize the performance score by searching optimal decay steps for every model. However, we use entropy as a confidence measurement (Equation 2) in our experiments. The regularization term impacts the resulting confidence value, thus, we set the same total decay step (6000) for all models to avoid unfair comparison of confidence measurement.

	Decay Steps, Brain Score $R^2\uparrow$ ($\pm$ 0.001)			
Model	2000	4000	6000	8000
CLIP	0.093	0.128	***0.131***	**0.132**
DiNOv2	0.113	**0.126**	**0.126**	***0.125***

Table 15. Performance score w.r.t. total decay steps for regularization term. Grid search with CLIP and DiNOv2 base size 12-layer model. Average of 3 runs.

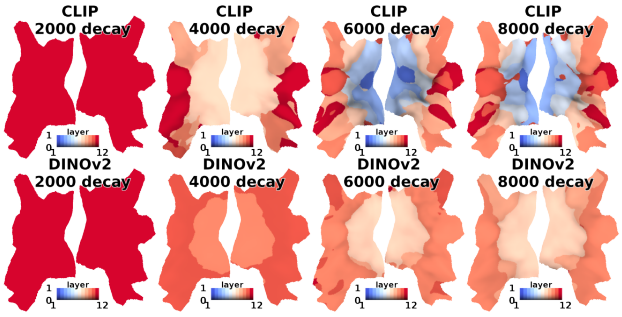


Figure 25. Layer selector output w.r.t. regularization decay total steps, number is total steps.

F. Related Work: State-of-the-art Methods

In the main text Section F.1, we compared our methods against the state-of-the-art methods' most salient design, but not their original methods. In this appendix, we discuss the competition-winning approaches in detail and explain the motivation for comparing their most salient design but not their original methods.

Experiment Setting Our experiment setting is different from the competition-winning methods. They build an ensemble of ROI-unique models, there's less demand for voxel-wise feature selection in ROI-unique models because voxels in the same ROI select similar features. However, we build one all-ROI model that covers all visual brain voxels, and the local similarity and global diversity of voxels emphasized the importance of factorized and topology-constrained feature selection introduced in this work. Overall, existing work use pre-defined ROIs and ensemble of ROI-unique models, we build one all-ROI model.

Past Algonauts competition-winning methods used an ensemble with a grid search of layers [6, 10]. The best single-layer model outperforms averaging or concatenating multiple layers. We aim to build a single all-ROI model that dynamically selects layers for voxels in every ROI. In Figure 26, we verified that our layer selector weights matched the grid search score of single-layer models.

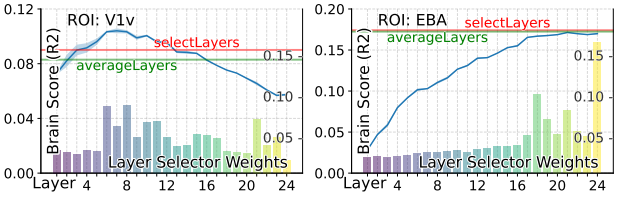


Figure 26. Grid search matched layer selector weights. Right y-axis is selector weights, and left y-axis is prediction score. Blue curve: per-layer model grid search score; Red line: score with layer selection; Green line: score with uniform layer average. Grid search with class token only.

Algonauts 2021 winner, patchToken The Algonauts 2021 challenge was hosted with another 3T fMRI dataset [14] but not NSD. The Algonauts 2021 dataset lacks the high data quality that NSD has, the lower data quality limited the effect of novel model building. The Algonauts 2021 competition winners (the top 3 methods are summarized in supplementary of [14]) used primitive methods that do not select unique features for each voxel but compress the flattened patches into one feature vector for all voxels. All voxels use the same feature vector. The most salient design in Algonauts 2021 is the patch compression module, patchToken, we re-implemented their methods (Table 3, patchToken) and found that patchToken methods achieved suboptimal performance on the high-quality dataset NSD.

GNet by NSD Before the Algonauts 2023 challenge, for the NSD dataset, the commonly used state-of-the-art method is GNet introduced by the NSD authors [2]. GNet introduced a layer-specific spatial pooling field for each voxel, which is a non-factorized and non-topology-constrained feature selection for each voxel. The original GNet was an end-to-end CNN trained from scratch, later studies [7] swapped the image backbone model with frozen state-of-the-art pre-trained ViT models to increase the performance. In our comparison we used a frozen CLIP XL model for all the models, so it's not the original GNet. The most salient design in GNet is the layer-specific spatial pooling field, we re-implemented the spatial pooling field design and compared it with our methods (Table 3, GNetViT). Notably, GNetViT requires quadratic memory and computation because of the unique $L \times H \times W$ spatial pooling field for each voxel.

Algonauts 2023 first place Our methods is an extension of the Algonauts 2023 winning methods Memory Encoding Model (Mem) [22]. Mem used topology constraints but only partially factorized the feature selection (they are missing the scale axis). In our methods, we further introduced fully factorized feature selection. There are some major differences between our work and their settings: 1) We only consider one image as input, Mem used extra information including past 32 images, behavior response, and time information, extra information led to a shocking 10% challenge score boost. 2) We build one single all-ROI model, Mem builds an ensemble of ROI-unique models. 3) We ran the training only once, Mem used dark knowledge distillation and ran the training twice. 4) We only used voxels in the visual brain, Mem additionally used voxels outside the visual brain to increase data samples. 5) We only used one subject for training, Mem trained a shared backbone for all 8 subjects. Mem's is partially factorized (without scale axis) and topology-constrained feature selection, we included Mem's most salient design in our ablation study in Table 3 "- no scale sel".

Algonauts 2023 second place For the second place winning methods of the Algonauts 2023 challenge [1], the most important factors to their winning are: 1) they used extra information including behavior response and time information, which led to a 4% challenge score boost, 2) they built ROI-unique models ensemble, and 3) they trained on 8 subjects. For the methods, they used a Detection Transformer (DETR) style attention mask with ROIs as queries. Their feature selection is voxel-shared but ROI-specific and also image-specific, not factorized or topology constrained. The DETR-style attention mask requires quadratic computation resources, their methods is possible to run for 36 ROIs as queries but impossible for 37,984 voxels as queries. We did not include the transformer methods in the comparison because their methods fundamentally rely on pre-defined ROIs and are unable to scale up to voxels. The closest comparison to this transformer method is GNetViT.

Algonauts 2023 third place For the third place winning methods of the Algonauts 2023 challenge [15], the most important factors that contributed to their winning are: 1) they ensembled 6 backbone models, 2) they pre-trained models on all ROI and all subjects, then fine-tuned ROI-unique models for each subject, and 3) they used a bag of training tricks. Their method used the same feature vector for voxels in the same ROI, similar to the patchToken methods in Algonauts 2021. However, it remains unclear how they compressed the $L \times C \times H \times W$ feature into one feature vector. We did not include this method in the comparison because the feature compression module is unclear, the closest comparison is classToken and patchToken.

F.1. Performance and Complexity

Previous state-of-the-art brain encoding approaches made diverse choices on image encoders and feature selections. We re-implemented them to avoid unfair comparison by keeping their most salient design choices but swapping them in standard components. We used CLIP-XL [9, 12] backbone for the image encoder for all methods.

There are three distinct types of feature selections. **1)** The simplest way is to leverage the class tokens, classToken, by taking it from each layer, $\mathbb{R}^{L \times C}$, applies a layer-unique transformation to $\mathbb{R}^{L \times D}$, and average pools across the layers to obtain a $\mathbb{R}^{D}$ feature vector. **2)** The second way, patchComp, extracts information from the patch image token, allowing finer pixel region selection: flattened features first along the spatial dimension $H \times W$ for each layer and fed $\mathbb{R}^{C \times H \times W}$ to a layer-unique-MLP that compressed it to a $\mathbb{R}^{D}$ feature vector. **3)** Finally, in the style of GNet [2], we construct a layer-specific 2D selection mask to pool $\mathbb{R}^{L \times D \times H \times W}$ into a vector of $\mathbb{R}^{L \times D}$, followed by pooling layers to obtain a $\mathbb{R}^{D}$ feature vector. In the ablation study of our network (*FactorTopy*), we created several

versions each by replacing one of the factorized selectors in layer, space, and scale and with average pooling. We also created a more robust version by sampling three times in the space selection. Comparison results are reported in Table 16.

Method	**Time**‡	**MACs**	**Brain Score R^2 ↑ (± 0.001)**			
			all	**V1v**	**V3v**	**EBA**
classToken	×1	×1	0.100	0.085	0.075	0.173
patchToken [14]	×1	×1	0.122	0.176	0.163	0.165
GNetViT [2]	×94	×17	0.124	0.174	0.146	0.174
FactorTopy (Ours)	×3	×1.2	**0.132**	**0.205**	**0.179**	**0.175**
- w/o topology	×3	×1.4	0.130	0.197	0.176	0.174
- no layer sel	×3	×1.2	0.125	0.181	0.162	0.174
- no space sel	×3	×1.2	0.117	0.094	0.089	0.175
- no scale sel	×3	×1.2	0.131	0.201	0.177	0.175
+ multiple sample	×7	×1.6	0.134	0.207	0.182	0.176

Table 16. **Performance Ablation**. Average of 3 runs. ‡: wall-clock.

Figure 27. Brain-to-Network alignment trained with limited data samples. Base size models, number of samples marked in brackets.

G. Limited Training Samples

Practical use of our brain-to-network mapping tool for network visualization requires our network to be trained efficiently. Using data scaling experiments shown in Figure 28 and Figure 27, we conclude that teaching our model with 3K sample images (30 minutes on RTX4090) offers a good trade-off. Our topological constraints and factorized feature

selection (*FactorTopy*) scales better to less training data.

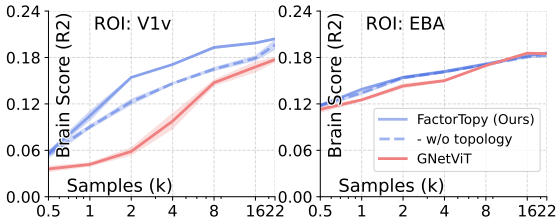
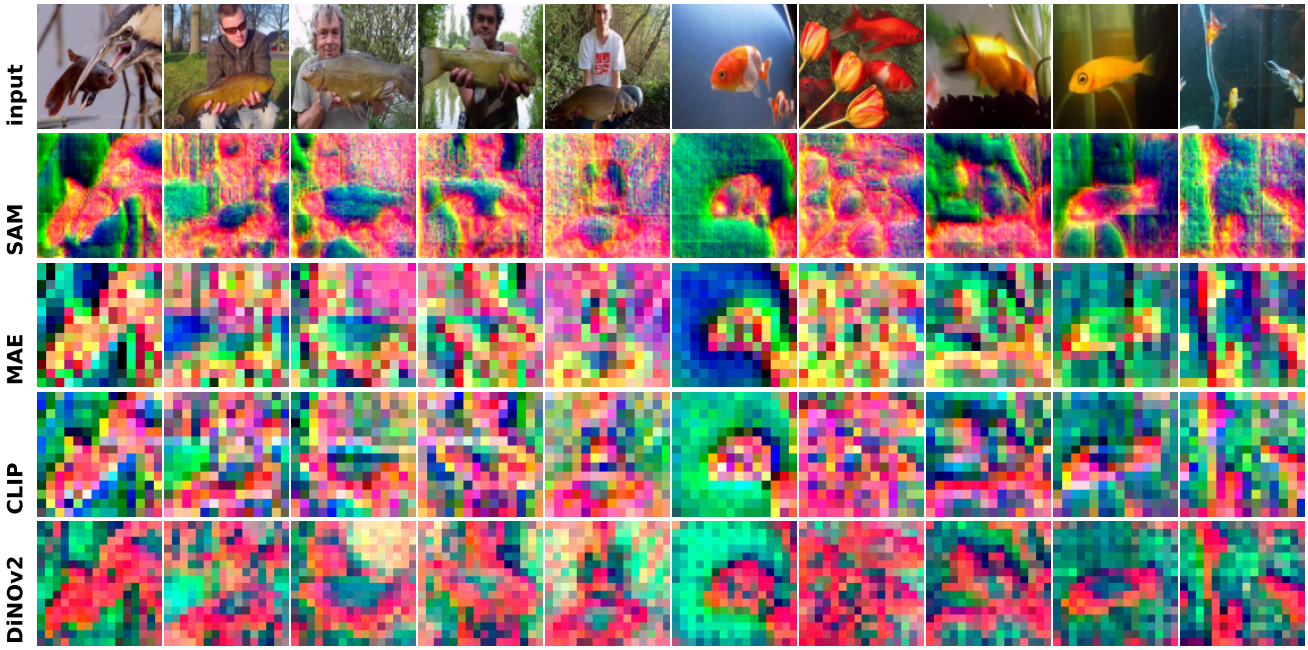


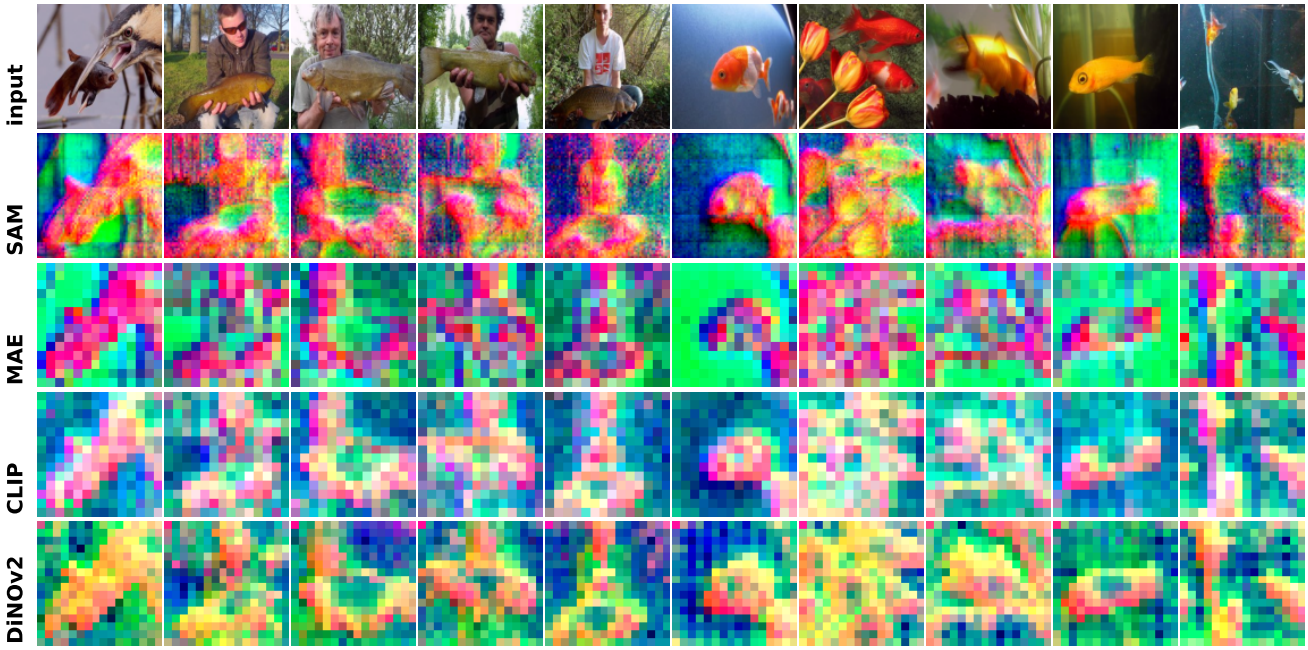
Figure 28. Performance w.r.t. training data sample, in log scale.



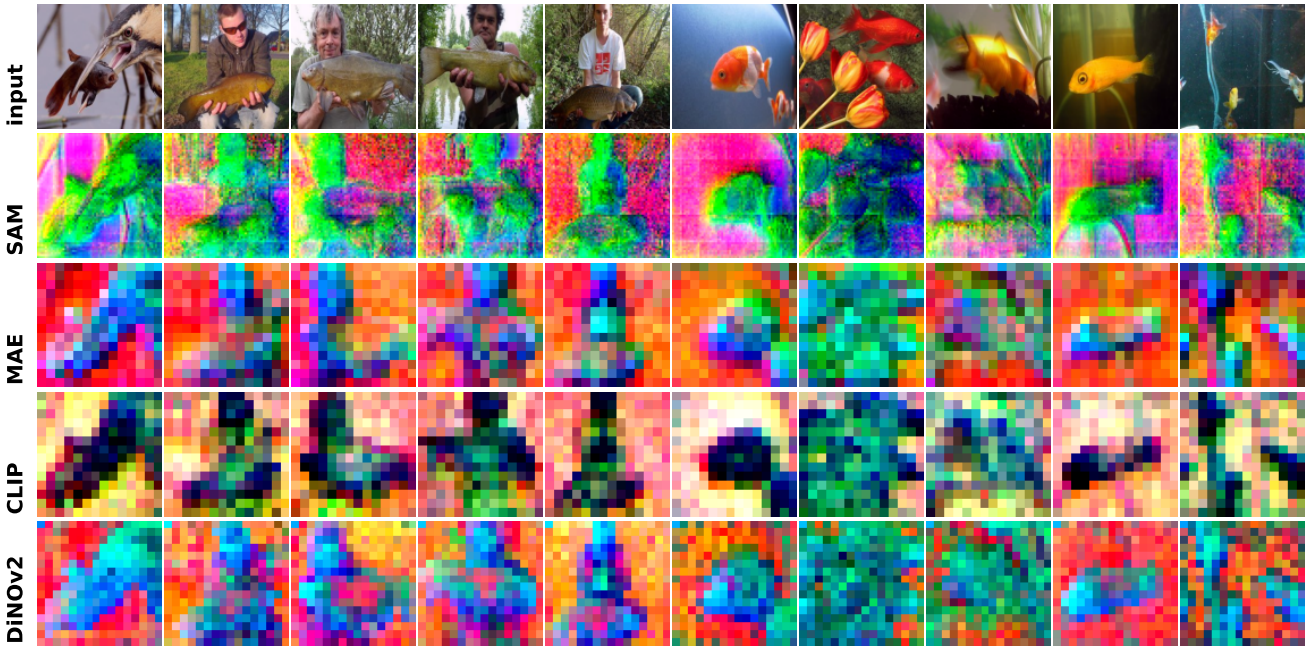
(a) **V1** (early visual)

(b) **V2** (early visual)

Figure 29. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).



(a) **V3** (early visual)



(b) **V4** (mid-level)

Figure 30. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).

(a) **EBA** (body)

(b) **FBA** (body)

Figure 31. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).

(a) **OFA** (face)

(b) **FFA** (face)

Figure 32. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).

(a) **OPA** (navigation)

(b) **PPA** (scene)

Figure 33. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).

(a) **OWFA** (words)

(b) **VWFA** (words)

Figure 34. Top 3 selected channels for voxels in one brain ROI (methods in Appendix E.2, findings in Appendix D.5).

References

[1] Hossein Adeli, Sun Minni, and Nikolaus Kriegeskorte. Predicting brain activity using Transformers, 2023. Pages: 2023.08.02.551743 Section: New Results. 15

[2] 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, J. Benjamin Hutchinson, Thomas Naselaris, and Kendrick Kay. A massive 7T fMRI dataset to bridge cognitive neuroscience and artificial intelligence. *Nature Neuroscience*, 25(1):116–126, 2022. Number: 1 Publisher: Nature Publishing Group. 1, 15, 16

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments, 2021. arXiv:2006.09882 [cs]. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers, 2021. arXiv:2104.14294 [cs]. 3

[5] Xinlei Chen, Saining Xie, and Kaiming He. An Empirical Study of Training Self-Supervised Vision Transformers, 2021. arXiv:2104.02057 [cs]. 3

[6] R. M. Cichy, K. Dwivedi, B. Lahner, A. Lascelles, P. Iamshchinina, M. Graumann, A. Andonian, N. A. R. Murty, K. Kay, G. Roig, and A. Oliva. The Algonauts Project 2021 Challenge: How the Human Brain Makes Sense of a World in Motion, 2021. arXiv:2104.13714 [cs, q-bio]. 14

[7] Colin Conwell, Jacob S. Prince, Kendrick N. Kay, George A. Alvarez, and Talia Konkle. What can 1.8 billion regressions tell us about the pressures shaping high-level visual representation in brains and machines?, 2023. Pages: 2022.03.28.485868 Section: New Results. 15

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. ISSN: 1063-6919. 3

[9] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets, 2023. arXiv:2304.14108 [cs]. 3, 15

[10] A. T. Gifford, B. Lahner, S. Saba-Sadiya, M. G. Vilas, A. Lascelles, A. Oliva, K. Kay, G. Roig, and R. M. Cichy. The Algonauts Project 2023 Challenge: How the Human Brain Makes Sense of Natural Scenes, 2023. arXiv:2301.03198 [cs, q-bio]. 12, 14

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners, 2021. arXiv:2111.06377 [cs]. 3

[12] Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. 3, 15

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything, 2023. arXiv:2304.02643 [cs]. 3

[14] Benjamin Lahner, Kshitij Dwivedi, Polina Iamshchinina, Monika Graumann, Alex Lascelles, Gemma Roig, Alessandro Thomas Gifford, Bowen Pan, SouYoung Jin, N. Apurva Ratan Murty, Kendrick Kay, Aude Oliva, and Radoslaw Cichy. BOLD Moments: modeling short visual events through a video fMRI dataset and metadata, 2023. Pages: 2023.03.12.530887 Section: New Results. 15, 16

[15] Xuan-Bac Nguyen, Xudong Liu, Xin Li, and Khoa Luu. The Algonauts Project 2023 Challenge: UARK-UAlbany Team Solution, 2023. arXiv:2308.00262 [cs]. 15

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, 2023. arXiv:2304.07193 [cs]. 3

[17] Jacob S. Prince, Ian Charest, Jan W. Kurzawski, John A. Pyles, Michael J. Tarr, and Kendrick N. Kay. Improving the accuracy of single-trial fMRI response estimates using GLMsingle. *eLife*, 11:e77599, 2022. 12

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. arXiv:2103.00020 [cs]. 3

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, 2022. arXiv:2112.10752 [cs]. 4

[20] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models, 2022. arXiv:2210.08402 [cs]. 4

[21] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy

without increasing inference time, 2022. arXiv:2203.05482 [cs]. 13

[22] Huzheng Yang, James Gee, and Jianbo Shi. Memory Encoding Model, 2023. arXiv:2308.01175 [cs]. 15

[23] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT Pre-Training with Online Tokenizer, 2022. arXiv:2111.07832 [cs]. 3