
Appendix

Anonymous Author(s)

Affiliation

Address

email

1 More Details about the Baseline of ZEBRA

The Baseline employs the ViT-based fMRI encoder from fMRI-PTE [1], which is pretrained on the UK Biobank dataset [2]. This encoder transforms the fMRI scan into a shared latent space, where a diffusion prior network then converts the latent brain embeddings into vision features for image generation using Stable Diffusion.

The Baseline consists of three main components:

1. **fMRI Encoder:** This module processes the input fMRI data and transforms it into a unified 2D brain activation map [1], resulting in a single-channel image $x \in \mathbb{R}^{256 \times 256}$. The fMRI encoder then maps this image into a latent representation $E \in \mathbb{R}^{B \times L \times C_1}$, where B is the batch size, L is the number of tokens, and C_1 is the brain feature dimension.
2. **Latent Representation Conversion:** The latent brain embedding E is subsequently transformed into a CLIP-compatible embedding $F \in \mathbb{R}^{B \times L \times C_2}$ to provide guidance for the reconstruction process.
3. **Diffusion Prior Network:** As in *MindEye2* [3], we employ a diffusion prior [4] to map the fMRI-CLIP embedding F to a reconstructed OpenCLIP image embedding F_y corresponding to the visual stimulus.

The training of the Baseline involves three key losses:

1. **Contrastive Loss on CLIP Text Embeddings:** This loss, denoted $\mathcal{L}_{\text{CLIP}_t}$, is calculated between the predicted CLIP text embedding F^t and the ground truth F_y^t .
2. **Contrastive Loss on CLIP Image Embeddings:** The loss $\mathcal{L}_{\text{CLIP}_i}$ is computed between the predicted CLIP image embedding F and the ground truth F_y^i .
3. **Diffusion Prior Loss:** The loss $\mathcal{L}_{\text{prior}}$ is used to train the diffusion prior network to minimize the reconstruction error.

Both $\mathcal{L}_{\text{CLIP}_t}$ and $\mathcal{L}_{\text{CLIP}_i}$ are implemented as the BiMixCo loss, which aligns fMRI signals x and corresponding image embeddings y using a bidirectional contrastive loss and MixCo data augmentation, as detailed below.

The MixCo procedure involves mixing two independent fMRI signals. For each fMRI signal x , we randomly sample another fMRI signal x_m corresponding to a different index m . The two signals are then mixed using a linear combination:

$$x^* = \text{mix}(x, x_m) = \lambda \cdot x + (1 - \lambda)x_m, \quad (1)$$

where x^* represents the mixed fMRI signal and λ is a hyperparameter sampled from a Beta distribution. The ridge regression module then maps x^* to a lower-dimensional space, yielding $x^{*'}$, from which the embedding F is obtained using the MLP, i.e., $F = \mathcal{E}(x^{*'})$.

33 The BiMixCo loss function is formulated as:

$$\begin{aligned}
\mathcal{L}_{\text{BiMixCo}} = & -\frac{1}{2L} \sum_{i=1}^L \lambda_i \cdot \log \frac{\exp(\text{sim}(\mathbf{F}_i, \mathbf{y}_i)/\tau)}{\sum_{k=1}^L \exp(\text{sim}(\mathbf{F}_i, \mathbf{y}_k)/\tau)} \\
& -\frac{1}{2L} \sum_{i=1}^L (1 - \lambda_i) \cdot \log \frac{\exp(\text{sim}(\mathbf{F}_i, \mathbf{y}_{m_i})/\tau)}{\sum_{k=1}^L \exp(\text{sim}(\mathbf{F}_i, \mathbf{y}_k)/\tau)} \\
& -\frac{1}{2L} \sum_{j=1}^L \lambda_j \cdot \log \frac{\exp(\text{sim}(\mathbf{F}_j, \mathbf{y}_j)/\tau)}{\sum_{k=1}^L \exp(\text{sim}(\mathbf{F}_j, \mathbf{y}_j)/\tau)} \\
& -\frac{1}{2L} \sum_{j=1}^L \sum_{\{l|m_l=j\}} (1 - \lambda_j) \cdot \log \frac{\exp(\text{sim}(\mathbf{F}_l, \mathbf{y}_j)/\tau)}{\sum_{k=1}^L \exp(\text{sim}(\mathbf{F}_l, \mathbf{y}_j)/\tau)},
\end{aligned} \tag{2}$$

34 where \mathbf{F} represents the OpenCLIP embeddings for the image y .

35 The Diffusion Prior network is used to transform the fMRI embedding \mathbf{F} into the reconstructed
36 OpenCLIP image embeddings of stimulus \mathbf{F}_y . The objective is to minimize the mean squared error
37 (MSE) between the predicted and target embeddings, formulated as:

$$\mathcal{L}_{\text{Prior}} = \mathbb{E}_{\mathbf{F}_y, \mathbf{F}, \epsilon \sim \mathcal{N}(0,1)} \|\epsilon(\mathbf{F}) - \mathbf{F}_y\|^2. \tag{3}$$

38 2 More Details about Metrics

39 To evaluate reconstruction quality, we adopt a comprehensive set of low-level and high-level metrics.

40 On the low-level side, we include four metrics: pixel-wise correlation, Structural Similarity Index
41 Measure (SSIM) [5], AlexNet(2), and AlexNet(5). Pixel-wise correlation and SSIM are computed by
42 averaging the similarity scores between each reconstructed image and its ground-truth counterpart.
43 AlexNet(2) and AlexNet(5) assess semantic similarity by measuring the two-way classification accu-
44 racy based on features extracted from the 2nd and 5th layers of a pre-trained AlexNet, respectively.

45 For high-level evaluation, we extract features using several pre-trained models. EffNet-B and SwAV
46 metrics are calculated by averaging the feature distance between reconstructions and ground-truth
47 images using EfficientNet-B1 [6] and SwAV-ResNet50 [7]. In contrast, the Inception [8] and CLIP [9]
48 metrics reflect the accuracy of two-way classification using the corresponding high-level features.

49 3 Comparison with Baselines across Subjects

50 Table 1 presents a detailed comparison of ZEBRA with NeuroPictor*, and our baseline across subjects
51 1, 2, 5, and 7 from the Natural Scenes Dataset. The results are reported under a zero-shot setting,
52 where NeuroPictor* refers to our reimplementaion pretrained on the remaining seven subjects
53 without any subject-specific finetuning. Across both low-level (PixCorr, SSIM) and high-level
54 metrics (AlexNet, Inception, CLIP, etc.), ZEBRA consistently outperforms the zero-shot baselines,
55 especially on semantic metrics like Alex(5) and CLIP accuracy. Notably, its performance remains
56 stable across subjects, indicating strong generalization ability. The final row reports the averaged
57 scores across all four subjects, further confirming the effectiveness of ZEBRA in both perceptual
58 quality and semantic fidelity.

59 References

- 60 [1] X. Qian, Y. Wang, J. Huo, J. Feng, and Y. Fu, “fmri-pte: A large-scale fmri pretrained transformer encoder
61 for multi-subject brain activity decoding,” *arXiv preprint arXiv:2311.00342*, 2023.
- 62 [2] K. L. Miller, F. Alfaro-Almagro, N. K. Bangerter, D. L. Thomas, E. Yacoub, J. Xu, A. J. Bartsch, S. Jbabdi,
63 S. N. Sotiropoulos, J. L. Andersson, *et al.*, “Multimodal population brain imaging in the uk biobank
64 prospective epidemiological study,” *Nature neuroscience*, vol. 19, no. 11, pp. 1523–1536, 2016.
- 65 [3] P. S. Scotti, M. Tripathy, C. K. T. Villanueva, R. Kneeland, T. Chen, A. Narang, C. Santhirasegaran, J. Xu,
66 T. Naselaris, K. A. Norman, and T. M. Abraham, “Mindeye2: shared-subject models enable fmri-to-image
67 with 1 hour of data,” in *Proceedings of the 41st International Conference on Machine Learning, ICML’24*,
68 JMLR.org, 2024.
- 69 [4] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation
70 with clip latents,” *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

Table 1: Results of NeuroPictor*, our baseline, and ZEBRA for each subject, compared against representative methods under different training regimes. All results are averaged over subjects 1, 2, 5, and 7 from the Natural Scenes Dataset. “NeuroPictor*” denotes our implementation in a zero-shot setting, pretrained on the other 7 subjects.

Method	Low-Level				High-Level			
	PixCorr↑	SSIM↑	Alex(2)↑	Alex(5)↑	Incep↑	CLIP↑	Eff↓	SwAV↓
<i>Subject 1</i>								
NeuroPictor*	0.069	0.305	73.1%	75.4%	63.6%	66.8%	0.910	0.583
Our baseline	0.089	0.325	72.5%	74.7%	64.7%	63.2%	0.891	0.579
ZEBRA	0.153	0.384	76.1%	81.8%	73.4%	72.3%	0.814	0.490
<i>Subject 2</i>								
NeuroPictor*	0.061	0.303	73.0%	75.2%	62.4%	65.9%	0.938	0.590
Our baseline	0.079	0.323	72.4%	74.5%	63.5%	62.3%	0.918	0.586
ZEBRA	0.135	0.382	76.0%	81.6%	72.0%	71.3%	0.839	0.496
<i>Subject 5</i>								
NeuroPictor*	0.049	0.290	69.7%	74.6%	62.7%	66.7%	0.922	0.599
Our baseline	0.063	0.309	69.1%	73.9%	63.8%	63.1%	0.903	0.595
ZEBRA	0.119	0.365	72.6%	80.9%	72.4%	72.1%	0.825	0.503
<i>Subject 7</i>								
NeuroPictor*	0.049	0.290	69.9%	73.7%	61.2%	64.7%	0.986	0.655
Our baseline	0.064	0.309	69.3%	73.0%	62.2%	61.2%	0.966	0.650
ZEBRA	0.117	0.370	73.8%	80.4%	71.0%	70.4%	0.869	0.534
<i>Average</i>								
NeuroPictor*	0.057	0.297	71.4%	74.7%	62.5%	66.0%	0.939	0.607
Our baseline	0.074	0.316	70.8%	74.0%	63.5%	62.5%	0.920	0.602
ZEBRA	0.131	0.375	74.6%	81.2%	72.2%	71.5%	0.837	0.506

- [5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [6] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.
- [7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.
- [8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.