Exploring the Encoding Scheme of the Brain by Generating Images on Axes in a Common Space

Jackson Gao

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-157
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-157.html

May 12, 2023

Acknowledgement

I would like to thank my research advisor Prof. Yi Ma for always encouraging me to improve as a student researcher and showing me the big picture of many research areas, and Prof. Doris Tsao for guiding and proposing possible directions of this project. I am also thankful to Yuelin Shi and Dasheng Bi who taught me neuroscience and did online experiments for data collection in this project, and Xili Dai who provided technical support and many ideas of using deep learning models. Support from BAIR lab is also acknowledged.

Exploring the Encoding Scheme of the Brain by Generating Images on Axes in a Common Space

Jackson Gao

Abstract

Deep neural networks for object classification have been used to build a map of low-dimensional object space in the primate IT cortex to describe images of general objects. Corresponding responses of IT cells to these objects show that single IT cells project them onto axes of this space [1]. Through a similar process, it has been found that single cells are tuned to specific face axes that describe specific facial features [2]. In this work, we use deep neural networks to build a common space for both faces and non-face objects. By generating images along axes in this space that would cause the maximal response of cells, we aim to answer the question of whether the encoding scheme of face cells could be used to encode non-face objects and produce meaningful code for general objects.

1 Introduction

In this project, we use several deep learning models that consist of both an encoder and a generator to explore the encoding scheme of the brain. In particular, we want to know the mechanism face cells use to distinguish faces and non-face objects as well as understanding the code of faces and objects given by the brain in general. This section includes definitions and hypotheses.

1.1 Definition of Face Cells

Face cells are neurons that give high responses when face images are shown as stimuli. As shown in Figure 1, they give high response to human faces, medium response to face-like objects, and low response to other objects.

1.2 Problem and Hypotheses

How does the brain process and memorize faces and non-face objects given the behavior of face cells? We propose two hypotheses.

1.2.1 Hypothesis 1

The encoding scheme used by all cells, including face cells, is the same. In this case, face cells still give meaningful responses to non-face objects even though these responses are low. In other words, there is no non-linear gate to classify faces and non-face objects before the signals of an image reach face cells.

1.2.2 Hypothesis 2

The signals of an image are first sent to other types of cells. These cells then classify the image as a face or a non-face object. Faces will be sent to face cells for finer face feature processing, and non-face objects will be processed by other cells. In other words, there is some non-linear gate

Preprint. Under review.

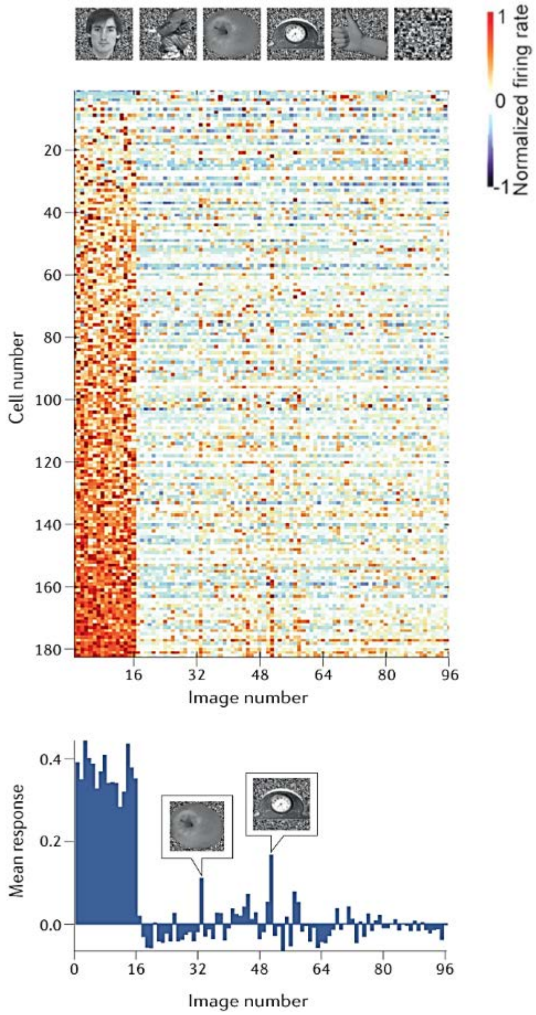


Figure 1: Responses of face cells to image stimuli. The first 16 images are faces, and the last 80 images are non-face objects.

to classify faces and non-face objects either before the signals reach face cells or during the local recurrent processing within face cells, and the encoding scheme used by face cells is unique.

2 Methodology

2.1 Setup and Notation

We use deep learning models that consist of both an encoder $\mathcal{E}$ and a generator $\mathcal{D}$ to represent the brain's encoding process and verify the validity of such representation using several metrics. Given the image matrix $X_{n\times c\times h\times w}$ where n is the number of images and c, h, w are the dimensions. We present these images as stimuli and record the response matrix $Y_{n\times m}$ where m is the number of neurons detected during an experiment. Let the feature matrix $\widetilde{Z}_{n\times d} = \mathcal{E}(X)$ contain feature representations of the images, and the PCA feature matrix $Z_{n\times k} = U_k\Sigma_k$ where $\widetilde{Z}_{n\times d} = U\Sigma V^T$ is the SVD, U_k is the truncated $n \times k$ matrix U, and Σ_k is the truncated $k \times k$ singular value matrix Σ. We train a linear regression $\min_W \|\widetilde{Z} \cdot W - Y\|_2^2$ to obtain the axes matrix $W_{k\times m}$ whose columns are vectors in the k-dimensional PCA feature space. Denote the i-th column of W as W_i.

2.2 Interpretation of Axis

Recall that m is the number of neurons and W contains m vectors in the PCA feature space. For neuron i, W_i is an axis with the following property: For points in the PCA feature space, taking a fixed step along the direction of axis W_i will result in the maximal increase in the predicted response of neuron i. In other words, under the condition that 1) the chosen model has an encoder and a generator that are well-formed and 2) there is indeed a linear mapping between the PCA feature space and the measured responses of neurons, then generating an image $x^* = \mathcal{D}(inv_{PCA}(z^*))$ from the feature z^* that is far away from the origin on axis W_i will trigger the maximal response of neuron i.

2.3 Experiment Design and Interpretation of Result

The image matrix X contains human faces X_{face} and non-face objects X_{obj}. We denote the corresponding PCA feature matrices as Z_{face} and Z_{obj}. When the linear regression is trained with only face features and responses, we denote the learned axes matrix as W_{face}; When the linear regression is trained with only non-face object features and responses, we denote the learned axes matrix as W_{obj}. For each neuron i, we can obtain two points far enough from the origin in the PCA feature space corresponding to two axes $W_{face,i}$ and $W_{obj,i}$. We can then generate two images via inverse PCA and the generator, which should cause the maximal response of neuron i. If these two images look similar, then we have evidence that supports hypothesis 1; If these two images look different, then we have evidence that supports hypothesis 2.

2.4 Verification of Assumptions

We have two main assumptions: 1) The mapping between the PCA feature space and the measured responses of neurons is linear; 2) The encoder and the generator of the chosen model are well-formed. For the first assumption, we use R^2 score as the metric. For the second assumption, we test the model on three aspects: reconstruction, generation, and self-consistency.

2.4.1 R2 Score

Given the PCA feature matrix Z, the learned axes matrix W, the response matrix Y, and the validation PCA feature matrix Z_{val}, we can compute the predicted response matrix $\widehat{Y} = Z_{val}W$. The R^2 score for neuron j is defined as follows:

$$\begin{aligned} R^2(j) &= 1 - \frac{SS_{res}(j)}{SS_{tot}(j)} \\ SS_{res}(j) &= \sum_i (Y_{ij} - X_{ij})^2 \\ SS_{tot}(j) &= \sum_i (Y_{ij} - \bar{Y}_j)^2 \\ \bar{Y}_j &= \frac{1}{n}\sum_i Y_{ij} \end{aligned}$$

R^2 normally ranges from 0 to 1 but can yield negative values. In the best case where $\widehat{Y}$ exactly matches Y, $R^2 = 1$; A baseline that always predicts $\widehat{Y}_0$ will have $R^2 = 0$; Predictions worse than the baseline will have a negative R^2.

2.4.2 Reconstruction

The chosen model's encoder $\mathcal{E}$ and generator $\mathcal{D}$ should be able to reconstruct the images in X reasonably well. We compute the reconstructed images $\widehat{X}$ as follows and compare with X:

$$\begin{aligned} \widetilde{Z} &= \mathcal{E}(X) = U\Sigma V^T \\ \widehat{X} &= \mathcal{D}(U_k \Sigma_k V_k^T) \end{aligned}$$

where $U\Sigma V^T$ is the SVD of $\widetilde{Z}$, U_k is the truncated $n \times k$ matrix U, Σ_k is the truncated $k \times k$ singular value matrix Σ, and V_k is the truncated $d \times k$ matrix V.

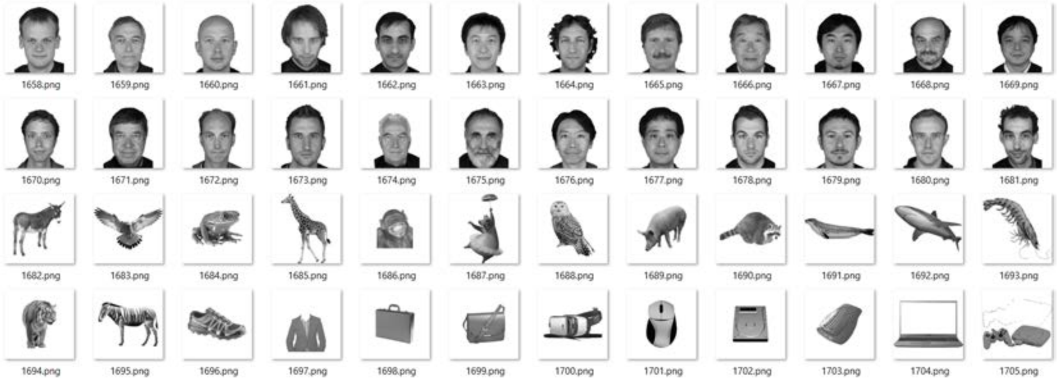


Figure 2: Our dataset contains single grayscale faces and objects with white background

2.4.3 Generation

Any generated image x^* described in Section 2.2 and 2.3 should be of reasonable quality. In the best case, the image should be natural and look like a face or an existing object. In addition, recall that each neuron i has two axes $W_{face,i}$ and $W_{obj,i}$. Images generated from points on different axes should be diverse.

2.4.4 Self-Consistency

The PCA features $\widehat{Z}$ of the reconstructed images $\widehat{X}$ should be close to the PCA features Z of the original images X. First, we compute $\widehat{Z}$ as follows:

$$\widehat{Z} = \mathcal{E}(\widehat{X})V_k$$

where V_k is the truncated $d \times k$ matrix from the SVD of $\mathcal{E}(X) = U\Sigma V^T$.
Then, we project each image's corresponding $\hat{z}$ and z onto an axis w and compute the projection lengths as follows:

$$Proj_w(\hat{z}) = \frac{\langle \hat{z}, w \rangle}{\|w\|}$$

$$Proj_w(z) = \frac{\langle z, w \rangle}{\|w\|}$$

We compare $Proj_w(\hat{z})$ and $Proj_w(z)$ with the distribution of projection lengths of all images as a measurement of self-consistency.

3 Experiments

3.1 Dataset

In the experiments, we use a small dataset of 3392 grayscale images consisting of 2000 human faces and 1392 other objects such as utensils, symbols, animals, and fruits. Examples are shown in Figure 2. Images in this dataset are represented as the image matrix X in Section 2.1 and are not used to train any models in Section 3 unless specifically mentioned.

3.2 PCA

Due to the small size of our dataset and the relatively large feature space to which different models' encoders $\mathcal{E}$ map the data, we use the first 50 principal components of PCA to further reduce the feature dimension. In other words, we set $k = 50$ in Section 2.1 and 2.4. Additionally, we only use a subset of face images in our dataset to fit the PCA in order to prevent face features from dominating

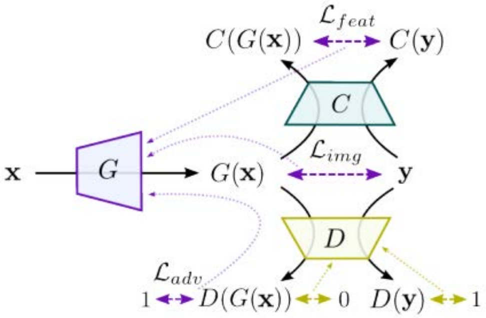


Figure 3: Model structure and training loss of DeepSim

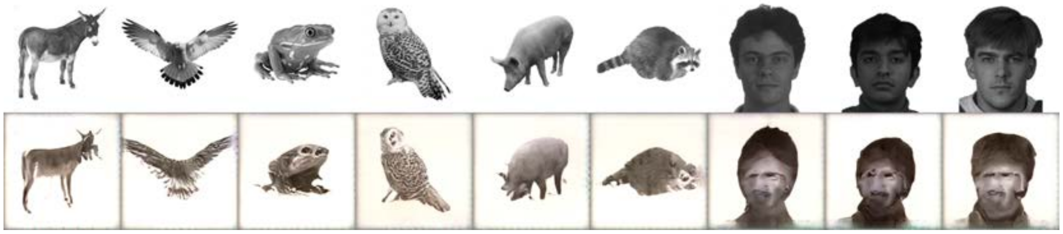


Figure 4: Reconstruction result of DeepSim, upper images are original, lower images are reconstructed

the first 50 principal components. In particular, we use all 1392 object images and 278 face images such that the ratio is roughly $5:1$. Lastly, we verified that using PCA features rather than those given by encoders $\mathcal{E}$ indeed gives much better R^2 scores.

3.3 Models

We choose a variety of models and test their performance on each of the four requirements in Section 2.4.

3.3.1 DeepSim

DeepSim [5] is first introduced in 2016 for the purpose of inverting AlexNet. It consists of a generator $\mathcal{G}$, a discriminator $\mathcal{D}$, and a comparator/encoder $\mathcal{C}$. As shown in Figure 3, training invloves loss in pixel space $\mathcal{L}_{img}$, loss in feature space $\mathcal{L}_{feat}$, and adversarial loss $\mathcal{L}_{adv}$ to make the generated images more natural and realistic. We take AlexNet pretrained on ImageNet and cut the network at the first fully connected layer $fc6$ as our encoder, which gives a feature space of 4096 dimensions, and we use the corresponding pretrined DeepSim generator as our generator in the following experiments.

R2 Score The mean/max R^2 score for DeepSim model is 0.393/0.647 as listed in Table 1. This is already a good result considering the noise in measurement of neuron responses. We use these two scores as our baseline when comparing with other models.

Reconstruction The reconstruction result of randomly chosen images is shown in Figure 4. Some details of the reconstructed animals are lost, and almost all details of the reconstructed faces are lost.

Generation The generation result of points on the face axis or object axis of selected neurons is shown in Figure 5. The generated images' identities are mostly vague, and the actual neuron responses they cause for corresponding neurons are lower than expected when presented as stimuli.

Self-Consistency We interpret DeepSim's self-consistency with Figure 6 and 7. The histogram in both figures shows the distribution of projection lengths of all images' features in the dataset onto the 25th neuron's face axis, and standard deviations of the distribution are indicated by dotted lines. In Figure 6, projection lengths of the original and reconstructed images should be close to each other for

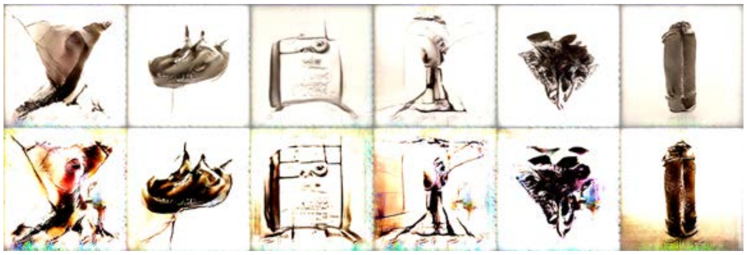


Figure 5: Generation result of DeepSim, lower images are generated at points on an axis that are farther from the origin in the feature space than their corresponding upper images

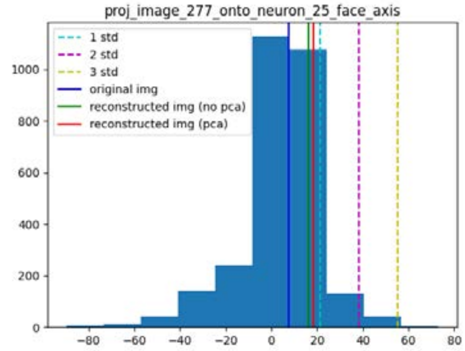


Figure 6: DeepSim's self-consistency measured by projection lengths of an original image and its reconstruction onto a neuron's axis

good self-consistency. The result given by the 277th image does not show strong self-consistency. In Figure 7, projection lengths of images generated at 1 std, 2 std, and 3 std should be close to the corresponding dotted lines for good self-consistency. The result does not show strong self-consistency either.

3.3.2 Stable Diffusion Image Variations Model

Stable Diffusion Image Variations is an image-to-image diffusion model based on latent diffusion models [9]. It is designed for generating variations of a given image. Instead of using CLIP [8] text embedding as guidance for generation, it accepts image embedding from CLIP image encoder τ_θ to gradually denoise normally distributed z_T in a latent space as shown in Figure 8. We take the

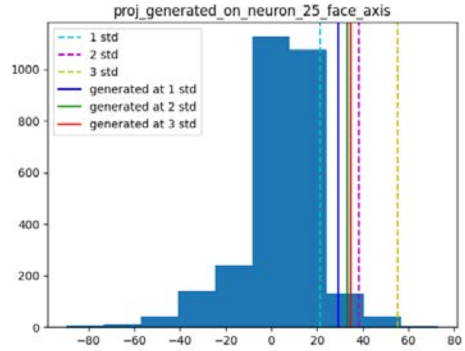


Figure 7: DeepSim's self-consistency measured by projection lengths of images generated at different points on a neuron's axis

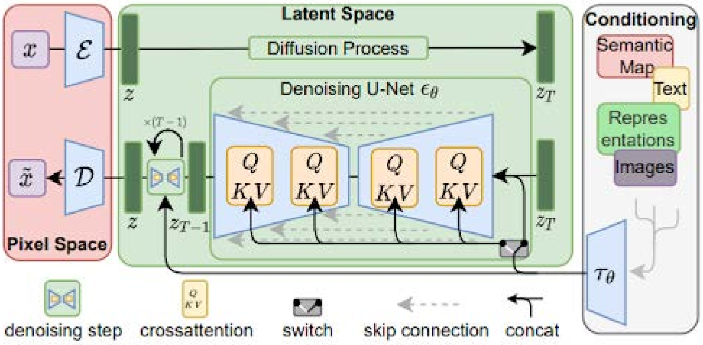


Figure 8: Model structure of Latent Diffusion

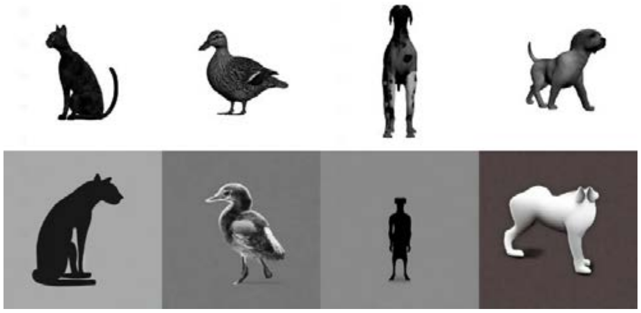


Figure 9: Reconstruction result of Stable Diffusion Image Variations, upper images are original, lower images are reconstructed

pretrained CLIP image encoder τ_θ as our encoder, which gives a feature space of 768 dimensions, and we use the pretrained diffusion model's denoising U-Nets together with the pretrained autoencoder's decoder that brings the denoised z from latent space to pixel space as our generator in the following experiments.

R2 Score The mean/max R^2 score for Stable Diffusion Image Variations is 0.105/0.631 as listed in Table 1. Comparing to that given by DeepSim, the mean score is lower, and the max score is almost the same. Since we care more about the max score because not all neurons gave meaningful responses when measured, we conclude that Stable Diffusion Image Variations passes the R^2 score test, and we proceed to the other tests.

Reconstruction The reconstruction result of randomly chosen images is shown in Figure 9. The model can only reconstruct images up to species, and it does not always recover the exact pose. Many details are also inconsistent.

Generation The generation result of 48 points on an axis of a selected neuron is shown in Figure 10. The generated images are single objects with simple geometric shapes, and there is no observable pattern in them. Since the result does not meet our expectation that the generated images should have clear identities, we do not proceed with this model.

3.3.3 Stable Diffusion Image-to-Image Text-Guided Model

Stable Diffusion Image-to-Image Text-Guided Model is another variant of latent diffusion models [9]. It is designed for modifying a given image under the guidance of a text prompt to change style or add details to the original image. Different from Stable Diffusion Image Variations in which the original images are input to τ_θ, this model keeps part of (usually the first half of) the diffusion process and inputs the original images to the incomplete diffusion process to obtain z_T, which then goes through the corresponding denoising U-Nets under the guidance of the text prompt as shown

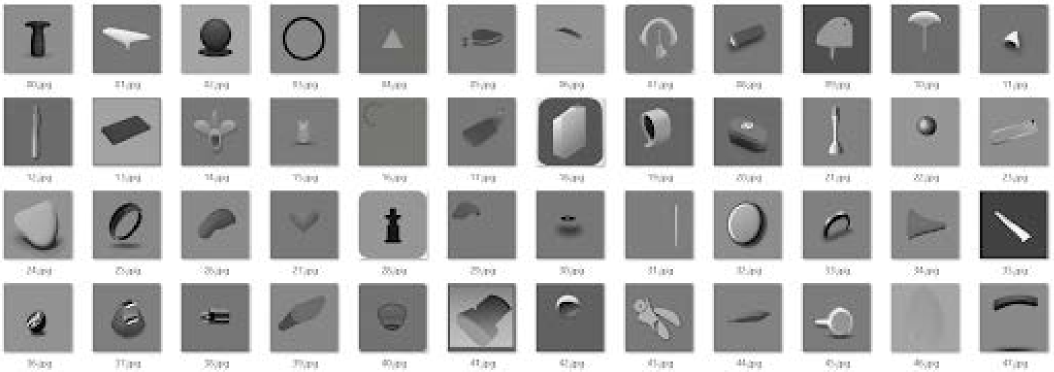


Figure 10: Generation result of Stable Diffusion Image Variations, images are listed in order of increasing distance between the origin and points on an axis to generate from in the feature space

in Figure 8. We take the pretrained autoencoder's encoder $\mathcal{E}$ as our encoder, which gives a feature space of $4 \times 25 \times 25 = 2500$ dimensions, and we use the pretrained diffusion model's first half of diffusion process, the corresponding denoising U-Nets with empty text prompt, and the pretrained autoencoder's decoder as our generator in the following experiments.

R2 Score The mean/max R^2 score for Stable Diffusion Image-to-Image model is -0.036/0.013 as listed in Table 1. Neither the mean score nor the max score is high enough for us to proceed with this model. The pretrained autoencoder's encoder $\mathcal{E}$ for the latent diffusion model does not give a good feature space for our task.

3.3.4 Masked Autoencoder

Masked Autoencoders [6] are trained to predict missing patches in an image. We take the pretrained MAE's encoder as our encoder, which gives a feature space of 768 dimensions if we extract the CLS embedding or $768 \times (num\_patches + 1)$ dimensions if we include all embeddings, and we use the pretrained MAE's decoder as our generator in the following experiments.

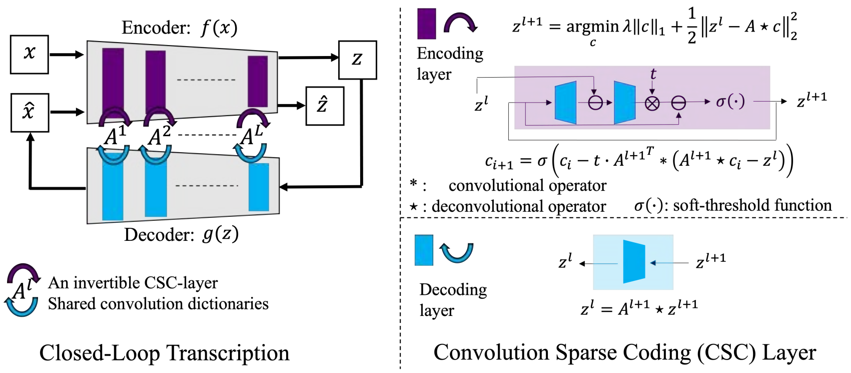
R2 Score If we use the extracted CLS embedding as our feature space, then the mean/max R^2 score for Masked Autoencoder is 0.412/0.653 as listed in Table 1. Both scores are the highest among all models, but the CLS embedding alone is not enough to recover the original image since patch information is stored in corresponding patch embeddings. However, if we use all embeddings, both R^2 scores drop to around 0. We cannot proceed from here unless we find a way to do reconstruction and generation using the CLS embedding only.

3.3.5 CTRL

CTRL [4] is a new computational framework for learning a structured generative model consisting of an encoder and a decoder. It is capable of learning a both discriminative and generative representation for multi-class and multi-dimensional data. The classification performance of the learned encoder and the visual quality of the learned decoder have been proved to be competitive and often better than exising methods based on GAN and VAE. CTRL-CSC [3], as shown in Figure 11, is a variant of CTRL that blends in convolutional sparse coding and dictionary learning to improve the reconstruction result and self-consistency of CTRL. We take CTRL-CSC pretrained on ImageNet and use its encoder and decoder as our encoder and generator in the following experiments.

R2 Score The mean/max R^2 score for CTRL is 0.217/0.552 as listed in Table 1. The mean/max R^2 score for CTRL-CSC is 0.196/0.542. These scores are a little lower compared to DeepSim but should be good enough given that CTRL is still under development.

Reconstruction The reconstruction result of randomly chosen images is shown in Figure 12. The reconstructed images are almost identical to the original images, and the result is the best among all models.



Encoder: $f(x)$
x
$\hat{x}$
z
$\hat{z}$
A^1
A^2
A^L
Decoder: $g(z)$
A^l
An invertible CSC-layer
Shared convolution dictionaries
Closed-Loop Transcription
Encoding layer
$z^{l+1} = \underset{c}{\operatorname{argmin}} \lambda \|c\|_1 + \frac{1}{2}\|z^l - A \star c\|_2^2$
t
z^l
$\sigma(\cdot)$
z^{l+1}
$c_{i+1} = \sigma\left(c_i - t \cdot {A^{l+1}}^T * \left(A^{l+1} \star c_i - z^l\right)\right)$
$*$: convolutional operator
$\star$: deconvolutional operator
$\sigma(\cdot)$: soft-threshold function
Decoding layer
z^l
z^{l+1}
$z^l = A^{l+1} \star z^{l+1}$
Convolution Sparse Coding (CSC) Layer

Figure 11: Model structure of CTRL-CSC

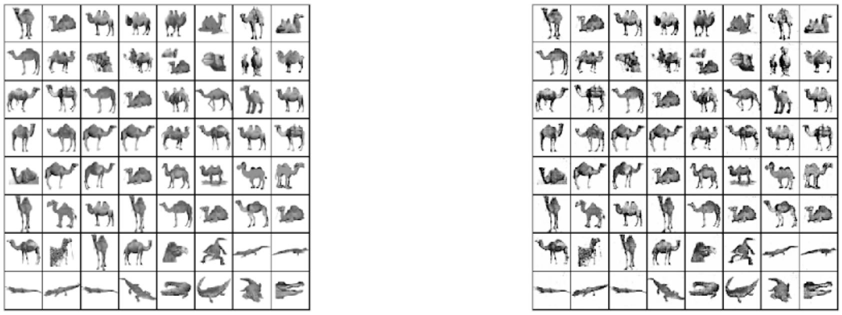


Figure 12: Reconstruction result of CTRL-CSC, left images are original, right images are reconstructed

Generation The generation result of 16 points on an axis of a selected neuron is shown in Figure 13. Images generated by CTRL-CSC look as if we were looking at something through a crystal ball, and their identities are not clear. The actual neuron responses they cause for corresponding neurons are also lower than expected when presented as stimuli.

4 Discussion

Conclusion In this project, we tested five models that consist of both an encoder and a generator, including DeepSim, Stable Diffusion Image Variations, Stable Diffusion Image-to-Image, Masked Autoencoder, and CTRL, in hope that one of them could help push our understanding of the encoding scheme of the brain. Unfortunately, all of them failed at least one test among R2 score, reconstruction, generation, and self-consistency. DeepSim had good R2 score but only okay reconstruction. It lacked strong generative power to generate natural and realistic images, and it did not possess good

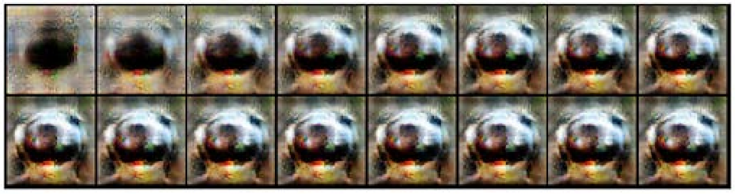


Figure 13: Generation result of CTRL-CSC, images are listed in order of increasing distance between the origin and points on an axis to generate from in the feature space

Table 1: Mean and Max Score of All Neurons' R2 Scores Using Different Models

Model Name (Encoder Name)	**Mean R2**	**Max R2**
DeepSim (AlexNet)	0.393	0.647
Diffusion Image Variations (CLIP ViT)	0.105	0.631
Diffusion Image-to-Image (VAE)	-0.036	0.013
Masked Autoencoder (ViT)	0.412	0.653
CTRL (CTRL Encoder)	0.217	0.552
CTRL CSC (CTRL CSC Encoder)	0.196	0.542

self-consistency. For Stable Diffusion models, generation depends on two variables, the normally distributed z_T to denoise from and the embedding output by CLIP encoder as guidance. The former usually determines the details of the generated images, and the latter usually determines the semantics of the generated images. For Stable Diffusion Image Variations, we treated the feature space given by CLIP encoder as our feature space, and we had no control over z_T, so the details of the reconstructed images were very different from the original images. For Stable Diffusion Image-to-Image, we treated the latent space given by the latent diffusion's pretrained autoencoder as our feature space, and we had no control over the semantics in text guidance, so the model would not be good at generating meaningful images from random points in its latent space. As to Masked Autoencoder, it seemed that the more class information is captured in data representation in our feature space, the higher R^2 score will be because the CLS embedding gave the highest R^2 scores, whereas patch embeddings gave very low R^2 scores. Finally, the CTRL framework was conceptually the best fit for the purpose of this project. It had good R^2 score, perfect reconstruction, good self-consistency. However, its generative power is to be improved. We look forward to improvement in the future.

Limitations This project is limited by computational resources. Models such as Diffusion Autoencoders [7] and Representation Learning with Diffusion Models [10] have not been tested due to a lack of checkpoints for models pretrained on suitable datasets. These variants of Diffusion models allow for simultaneous control over both details and semantics of the generated images and are thus worth a try.

Future Work Apart from generating optimal images from points on the axes that trigger strong responses, there are other ways in which we could gain evidence for either of the hypotheses. One alternative is the overlapping distribution method, which involves using an encoder-generator model to generate non-face objects that have the same distribution as faces in the feature space. This allows us to reduce the multimodal problem.

5 Acknowledgments

I would like to thank my research advisor Prof. Yi Ma for always encouraging me to improve as a student researcher and showing me the big picture of many research areas, and Prof. Doris Tsao for guiding and proposing possible directions of this project. I am also thankful to Yuelin Shi and Dasheng Bi who taught me neuroscience and did online experiments for data collection in this project, and Xili Dai who provided technical support and many ideas of using deep learning models. Support from BAIR lab is also acknowledged.

References

[1] Pinglei Bao, Liang She, Mason McGill, and Doris Y. Tsao. A map of object space in primate inferotemporal cortex. *Nature*, 583(7814):103–108, 2020.

[2] Le Chang and Doris Y. Tsao. The code for facial identity in the primate brain. *Cell*, 169(6), 2017.

[3] Xili Dai, Ke Chen, Shengbang Tong, Jingyuan Zhang, Xingjian Gao, Mingyang Li, Druv Pai, Yuexiang Zhai, XIaojun Yuan, Heung-Yeung Shum, Lionel M. Ni, and Yi Ma. Closed-loop transcription via convolutional sparse coding, 2023.

[4] Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, and Yi Ma. CTRL: Closed-loop transcription to an LDR via minimaxing rate reduction. *Entropy*, 24(4):456, mar 2022.

[5] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks, 2016.

[6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[7] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation, 2022.

[8] 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.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[10] Jeremias Traub. Representation learning with diffusion models, 2022.