# DreamScape: Denoising CLIP Embeddings with User Images for Improved Visualization Reconstruction

DEVAN SHAH

## 1 INTRODUCTION

In the last six months, the scientific community has witnessed tremendous efforts dedicated to the transformation of cerebral visual representations and mental imagery into tangible images. Systems such as DreamDiffusion [3], Second Sight [5], MindEye [10], NeuroImagen [6], Brain-Diffuser [7], among others introduced in 2023, have demonstrated notable advancements in the field of visualization. Nevertheless, the process of accurately capturing and interpreting the visualizations of a user remains a formidable challenge, primarily due to the challenge of gathering accurate ground-truth labels. Thus researchers must train their models on adjacent tasks, such as reconstructing what a user is seeing not visualizing, with the end goal of converting their models to visualization.

Reese Kneeland, the author of Second Sight, discusses, "the primary purpose of my work was to lay the groundwork for extending these sorts of models to mental imagery." Similar to these authors, we now aim to go beyond reconstructing what a user is seeing to reconstructing what a user is visualizing. This project introduces a system that will aid these models in the complex task of dream reconstruction and recovering what a user is visualizing, applicable to both fMRI and EEG embeddings.

## 2 CHALLENGES IN THE FIELD

A major challenge in reconstructing user visualizations is model accuracy. Transfering these models to visualization reconstruction, a task they are not trained on, will only lead to more noise and error. Current models struggle with additional noise, and even state-of-the-art (SOTA) EEG-to-image models achieve only 85% ImageNet categorical accuracy [6]. This is a remarkable result but lacks the fine-grained accuracy required for non-generic dream reconstruction. SOTA fMRI models perform more accurately but still require more fine-grained results. For instance, the presence of a person in a dream may have significantly different connotations depending on the person.

Accurate visualization reconstruction would lead to an improved model of cognition and a more comprehensive understanding of the brain, with numerous practical applications. These include measuring dream content for persons with PTSD, facilitating non-verbal communication for aphasia patients, aiding therapy, and advancing Alzheimer's research. As mobile EEG devices become more affordable, strong results on dream analysis with EEG readings could have a widespread impact with benefits accessible to ordinary individuals.

## 3 METHODOLOGY

We employ a vector database [8] to extend current image reconstruction tools with a database of user-specific images. This approach will allow systems to achieve an increased understanding of the user they are operating on. By augmenting traditional systems with user photo data, we aim to convert categorical accuracy into identifying precise moments important to the user (i.e., where MindEye recognizes a brown dog, we recognize the user's puppy Frodo). Thus using user history, we enable improved denoising of brain data for improved visualization reconstruction.

Author's address: Devan Shah, ds6237@princeton.edu.

Moreover, our system is fast and lightweight, leading to, when employed in conjunction with an image store, an abundance of applications within brain, image, and text cross-modality transfer. By storing embedding results, we can identify trends in user thought patterns, find underlying structure in otherwise unrelated thoughts, and provide summaries of user thoughts. We will provide the necessary code to add user images and deploy the system.

## 4 CORE SYSTEM OVERVIEW

Our system goal is to produce a denoising-module that can be added to the end of a DreamDiffusion or MindEye type system. The output of these systems is a CLIP embedding [9] and thus our module will denoise the output CLIP embedding using user information. To do this, we aim to estimate a set of potential user visualizations, and improve embedding accuracy from MindEye or DreamDiffusion if the user is visualizing an item in this set, and we aim to have minimal impact if the user is not. Thus, this system presents upside if the user visualizes predictable items and minimal downside if not.

The primary motivation behind our system is to view the space of potential user interests, and thus potential user visualizations, as a sparse subset of the entire object space that we can estimate given user information. Being output from noisy models, visualization embeddings may point to an object that clearly has no relevance to the user when a nearby object in CLIP space has greater relevance and is more likely what the user was visualizing.

By storing relevant user moments in a vector database, we can now estimate the set of potential user visualizations. We gather this set of relevant moments by collecting user images and storing the CLIP embeddings of those images in a vector database, with potential modifications we discuss later. To find the most relevant images, we additionally employ current user location and the location the image was taken.

When given a CLIP embedding representing a user visualization and the location of the user, we perform a nearest neighbor search on the database of user images to identify the most relevant images. For sufficiently related images, we then perform a weighted-averaging of their CLIP embedding vectors and the initial visualization CLIP embedding to result in a user-adjusted embedding, which we then output.

## 5 SYSTEM DESIGN

To collect relevant user images, we leverage the Google Photos API and the Flickr API to download a user's images and to download images from locations relevant to the user (i.e. sights they may see frequently but not photograph).

To search for relevant user images given the visualization embedding and current location, we test two different techniques.

(1) DreamScapeST1 - We store the location an image was taken as metadata and filter the results of a nearest neighbor search using the metadata.
(2) DreamScapeST2 - We test a deep encoder that increases cosine similarity for embeddings from nearby locations.

For each system, when giving an embedding that may represent a user's visualization, we use the relevant system and vector search (for $k$ neighbors) to find user images with similar CLIP embeddings. For CLIP image embeddings where the cosine similarity with the system-modified visualization embedding exceeds a certain threshold, denoted $\alpha$, we average the image embedding into the initial thought visualization embedding to prod in this direction.

**Search Technique #1**: By adding a metadata tag, we can restrict search by location, finding the most relevant image taken at a given location and thresholding with $\alpha$ to determine whether the user is likely visualizing of something similar to the image. We encode images with the CLIP model, and thus we search for the most relevant image at a given

location via comparing the CLIP embeddings from user images and the user visualization embedding.

**Search Technique # 2**: We train a neural network to modify CLIP embeddings with location information. The network takes in input of size 514, representing the CLIP embedding of 512 dimensions and the latitude and longitude, and after two linear layers with ReLU activation, outputs a new embedding of size 512. The model was developed with PyTorch and trained for 9 hours on over $7,000,000$ random vectors from a multivariate normal distribution and on various locations to learn the function $f$ satisfying:

$$\cos(f(v, loc), f(v', loc')) = \cos(v, v') + 0.15e^{-||loc - loc'||_1}$$

Where $v, v'$ are vectors with dimension 512, $loc, loc'$ are vectors in $\mathbb{R}^2$ representing the latitudes and longitudes of their respective location, $||\cdot||_1$ is the $\ell_1$ norm, and $\cos(v, v') = \frac{|v \cdot v'|}{||v||_2 ||v'||_2}$. For each user image, we take the CLIP embedding of that image and pass it through this location model, storing the resulting output in the vector database and storing the initial CLIP embedding with that vector as metadata.

Then, when searching for relevant images for a given visualization embedding, we use the location model to generate a location-imbued vector and search the vector database for the most relevant user images. If the cosine similarity exceeds $\alpha$, we then average the CLIP embeddings (not the location-imbued embeddings) of those user images with the initial visualization embedding, and use the result as our output. As we do not need to filter with metadata and the location model is fast, this technique offers a noticeable increase in speed.

## 6 EXPERIMENT RESULTS

We perform several tests to validate our system. For our testing, we use EEG data and fMRI data, with the data provided by the DreamDiffusion and MindEye projects. We perform our initial tests using a simulated high-accuracy EEG embedding model and subsequently perform tests with a modified-MindEye model on fMRI test data. System code and tests are available *here*.

### 6.1 Zoo Simulation Tests

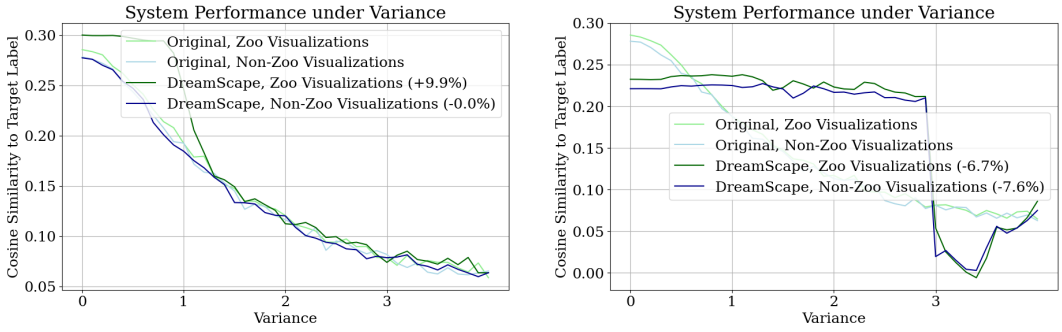
In this setting, we assume the user is visiting a zoo in New York City. This user typically travels around New York and thus has many images from New York, and we will test the models performance when the user has zoo related visualizations and non-zoo related visualizations, and their images consists of photos from that zoo and additional photos from elsewhere.

To test DreamScapeST1, we use a test image set of 368 images corresponding to the user's photos after we restrict location to being nearby (i.e. prior photos the user took at the zoo). We use 32 samples of EEG data representing user visualizations for our experiment, with 16 being on objects a user may have previously seen at the zoo and thus there are similar photos in the test image set, and 16 being on objects that do not have similar photos in the test image set, which we denote non-zoo visualizations.

To test DreamScapeST2, we keep the same 32 visualization samples, but we use a user image set of 3356 images corresponding to many user photos, with a 10% subset being prior photos the user took at the zoo. The photos previously taken at the zoo have their location chosen from a Gaussian distribution around the user, whereas other photos are chosen from a wider distribution centered outside the zoo.

For the DreamScapeST1 test, each of the previously seen objects is related to ~ 8% of the test image set, and for the DreamScapeST2 test, only ~ 0.9% of the test image set are related to each of the previously seen objects. The EEG and image data is gathered from the DreamDiffusion team, and the images are from ImageNet.

We simulate a perfect EEG-to-CLIP encoder and use our system with thresholds of $\alpha = 0.6$ and $k = 3$. To simulate the high noise on these models when adapted for visualization, we test the system by adding Gaussian noise, with varying variance, to the CLIP embedding vector. For Table 1, we test the subject accuracy of the nearest image (by cosine distance of CLIP embedding) both with and without our system correcting the embedding. Finding the nearest image is performed by searching a 611, 283 image sample from the *COYO-700M* dataset stored on a Pinecone Vector Database Index. For Figure 1, we measure the alignment between the output embedding of the visualization and the CLIP encoding of the visualization's label, testing both DreamScapeST1 and DreamScapeST2 against a system without the modules.




(a) DreamScape ST1 CLIP Alignment between output and target

(b) DreamScape ST2 CLIP Alignment between output and target

Fig. 1. Measuring DreamScape alignment with different search techniques

| | Var: 0 | Var: 0.25 | Var: 0.75 | Var: 1.5 | Var: 2 |
|---|---|---|---|---|---|
| DS Zoo | 15/16 | 15/16 | 16/16 | 8/16 | 4/16 |
| DS Non-Zoo | 16/16 | 14/16 | 12/16 | 3/16 | 2/16 |
| Zoo | 15/16 | 16/16 | 11/16 | 5/16 | 1/16 |
| Non-Zoo | 16/16 | 13/16 | 14/16 | 3/16 | 3/16 |

Table 1. DreamScape ST1 (DS) Subject Accuracy compared to Original System

The tests for the DreamScapeST1 system showcase strong results. We have that, based on human annotation of subject accuracy of nearest neighbor images, the model improves semantic content of the embeddings of zoo-related visualizations without noticeable cost to non-zoo related visualizations.

Figure 1a and 1b showcase CLIP alignment. To measure the results, we CLIP encode the text "An image of [label]" and measure the cosine similarity with the output embedding. For the DreamScapeST1 model, the average nearly

10% improvement is a remarkable result, as showcased in 1a. We note the improvements are especially pronounced when the noise has variance below 1, and this may be due to CLIP embeddings having coordinates lying primarily within $-1$ and 1, and thus applying noise near or above that magnitude is extremely lossy. However, as evident by 1b, DreamScapeST2 underperforms, especially with increased variance where leveraging location data is more pertinent. This is likely as, to increase model training speed, I trained the location model on data from a different distribution than the CLIP distribution. The model may then be specific to the multivariate normal distribution it was trained upon and not extend to different distributions. Additionally, the target function is likely difficult to train on, as the additional $0.15e^{-x}$ term may only marginally impact loss and be easy to ignore, whereas simply returning the exact input CLIP vector, or a worse reconstruction of it, and ignoring location is an easy local minimum.

### 6.2 Recall with noised fMRI Data

For this test, we noise the fMRI brain data directly by adding Gaussian noise with a given variance to simulate the noise in visualization. We then apply a modified MindEye model, with the modifications exhibited in the attached code and made entirely to reduce memory, on the noised data to produce CLIP embeddings. We then apply DreamScapeST1 with $\alpha = 0.5$ and $\alpha = 0.7$ and a user image dataset of size 101 (after filtering by location). The ground-truth visualized objects each have 2 similar images in the user image dataset. We then search the *LAION-5B* dataset for the nearest image by CLIP cosine similarity and display it. For prior tests, we employed CLIP Base Patch 32, which has size 512 embeddings, yet for this test we employ CLIP Large Patch 14, which has size 768 embeddings, as this is employed in the original MindEye code.

We show select examples in Figure 2. The full results on 1530 images spanning no module, DreamScapeST1 with $\alpha = 0.7$, and DreamScapeST1 with $\alpha = 0.5$, all under varying levels of noise, are available for download at the following link *Image Files*. The full test spanned 10 distinct images under noise varying from 0.0 to 5.0.

In comparing images, the DS 0.5 and DS 0.7 module appear to outperform the system with no module, and the DS 0.5 module experiences higher performance at low variance than the DS 0.7 module before underperforming DS 0.7 at greater variance. These results suggest $\alpha$ may need to be tuned for a given model for maximal performance in that specific situation.

## 7 ADDITIONAL SYSTEMS

In producing highly accurate embeddings and gaining access to an individual's thought data, we outline and code additional systems built upon accurate denoising infrastructure.

**Adapting Embedding Space**: Through the addition of bias vectors, we can add vectors to the user image database that represent shifts in object clustering and effectively alter the CLIP embedding space to incorporate user attributes. An individual user may have associations different from the CLIP object associations, and thus by leveraging EER (and fMRI) model capabilities of predicting emotion [1, 2] and optional user-reinforcement, our system can add bias vectors to the vector database. When searching for image vectors in the user index, we first search for nearby bias vectors and add a displacement vector corresponding to the type of bias to the search vector. This encodes differences in how a user associates certain objects compared to the default for the CLIP embedding space. This replaces the need to fine-tune a CLIP model for each user. Prior results [4] have shown that CLIP embeddings can hold such semantic and sentiment data.

I will give an example of the benefit of such a system. In the general CLIP embedding space, animals typically perceived as "cute" may fall near each other. Yet, a traumatic experience between the user and a dog may lead to the

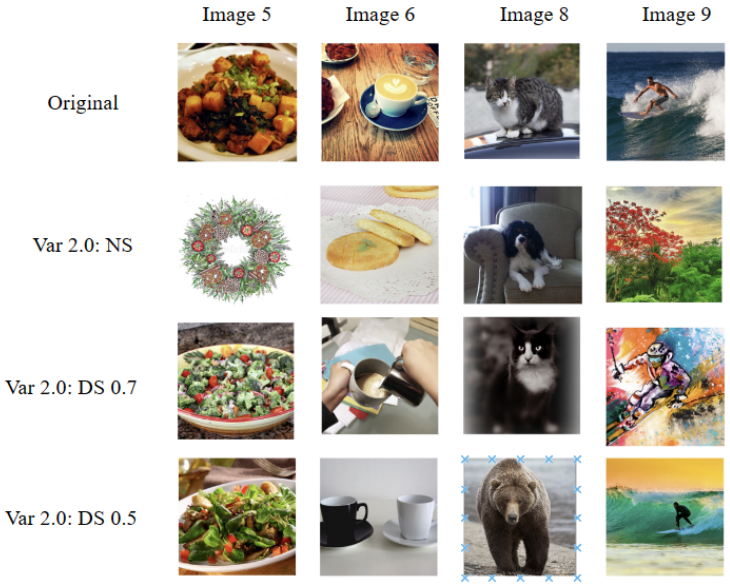



Fig. 2. Examples of Retrieval with No Module (NS), DreamScapeST1 at $\alpha = 0.7$ (DS 0.7), and DreamScapeST1 at $\alpha = 0.5$ (DS 0.5) on fMRI data with Gaussian noise of variance 2.0 added to each coordinate.

user viewing dogs as more different from other "cute animals". By measuring this reaction and adding a corresponding bias vectors, when the system is inputted a visualization embedding vector that noisily appears to involves dogs, it can add the displacement context to the search vector, leading vector search to shift away from the cute animal cluster, as despite the noise, the user was likely not thinking of a cute animal.

**Thought Location**: To navigate thought content, we store visualization embeddings and associated locations in a vector database. We can then search using CLIP vectors-which may encode text, images, or other thoughts-and find the most recent times or locations we had thoughts with the searched content, enabling multi-modal thought search. We will develop this dataset of user visualization embeddings and location data as a user uses our system, if they choose to opt in.

**Summary**: To summarize our thoughts, we can store thought embeddings in a list, and leverage an annotated image dataset, such as COYO-700M, to associate words to each thought via vector search on the annotated image dataset. Thus, we associate a sequence of thoughts to descriptive text, and can use a Large Language Model to synthesize the resulting phrases into a summary of thought content.

**Tracking Thought Content**: Given any term or emotion, we can CLIP embed the text for that term or emotion and measure cosine similarity of prior thoughts, or thoughts at a given location, with the text embedding. This allows us to measure how the user's thoughts align with certain sentiment or how frequently the user thinks of certain objects, and how this may change over time and location.

## 8 CONTINUED WORK

We designed an accurate model to denoise user visualizations using user image data, and presented applications and implications of such a system. However, despite several successes in our work, there were still significant setbacks. Notable, DreamScapeST2 was unable to properly encode location information into a CLIP embedding. We additionally were unable to run this system using a mobile EEG headset. For future work, I aim to train an embedding model on mobile EEG systems for a live and accessible system and additionally test on DreamDiffusion, not just MindEye. For improved embeddings, I will use a custom similarity function, rather than cosine similarity, to allow more straightforward encoding of location information into a CLIP embedding.

## 9 ACKNOWLEDGEMENTS

I would like to thank CareYaya for donating a Muse 2 EEG device for this project and providing both funding and expertise in moving this project forward.

I would also like to thank Dr. Paul Scotti and Dr. Ken Norman for their help. Dr. Norman was instrumental in his help of understanding the challenges of current dream reconstruction approaches, and Dr. Scotti aided me in allowing the MindEye model to work on a limited-memory device, such as my 3070Ti GPU.

## REFERENCES

[1] Arsalan, A., and Majid, M. A study on multi-class anxiety detection using wearable eeg headband. *Journal of Ambient Intelligence and Humanized Computing 13*, 12 (2022), 5739–5749.

[2] Arsalan, A., Majid, M., Butt, A. R., and Anwar, S. M. Classification of perceived mental stress using a commercially available eeg headband. *IEEE Journal of Biomedical and Health Informatics 23*, 6 (2019), 2257–2264.

[3] Bai, Y., et al. Dreamdiffusion: Generating high-quality images from brain eeg signals. *arXiv 2306.16934* (2023).

[4] Bustos, C., et al. On the use of vision-language models for visual sentiment analysis: A study on clip. *arXiv 2310.12062* (2023).

[5] Kneeland, R., et al. Second sight: Using brain-optimized encoding models to align image distributions with human brain activity. *arXiv preprint arXiv:2306.00927 1*, v1 (Jun 2023).

[6] Lan, Y.-T., Ren, K., Wang, Y., Zheng, W.-L., Li, D., Lu, B.-L., and Qiu, L. Seeing through the brain: Image reconstruction of visual perception from human brain signals, 2023.

[7] Ozcelik, F., and VanRullen, R. Natural scene reconstruction from fmri signals using generative latent diffusion, 2023.

[8] Pinecone. Pinecone: Vector database for vector search, 2023.

[9] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.

[10] Scotti, P. S., et al. Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors. *arXiv 2305.18274* (2023).