DimethylTelevision (DMTV)- Toward Neural Decoding of DMT Visuals with MindEye2

Christo Sasi

Student Number: S1102577

MSc. Specialisation: Free Specialization

Daily Supervisor: Greg Cooper

Project Supervisor: Umut Güçlü

Place: Nijmegen, Netherlands

Date: July 2025

1 Summary

The primary goal of this research was to train a model to reconstruct visual stimuli seen by a subject during a DMT induced psychedelic experience. Since ground truth images of DMT visuals are not available to train a model to reconstruct such visuals, we attempt to train a neural decoding model to learn visual features from the fMRI activity of the subject while they watched a movie. To accomplish this, we use MindEye2 —a state of the art neural decoding method. This approach establishes a basic outline for future work aimed at decoding not only externally driven visual stimuli but also internally generated visual experiences like dreams and hallucinations. The overarching expectation was to accurately decode visual experiences directly from neural signals, eventually generalizing these methods to decode seen (movie) and hallucinatory (DMT induced) experiences of the same subject. MindEye2 was trained on data from 8 subjects who were shown / 10,000 labeled images from Microsoft's COCO Images dataset while their brain activity was recorded in single trials. This training data was sourced from the landmark NSD experiment which used GLMSingle to pre-process (regress the hemodynamic response function (HRF)) the fMRI signals and generate betas for each corresponding image.

For our experiment, this training pipeline was adapted to allow training on a novel dataset of 3T fMRI data acquired while a subject viewed 1000 frames of movie sequences. As we did not have single trial betas for our training set, we used the SPLORA (Sparse and Low-Rank Paradigm Free Mapping) algorithm which is capable of de-convolving the signal collected during each pulse (corresponding to 34 frames viewed by the participant) and detecting individual BOLD events. SPLORA allowed us to infer events in the absence of empirical event markers in spontaneous brain activity and regress out the HRF to generate betas. These betas, along with corresponding normalized transforms of PNG movie frames formed the core of the training dataset for MindEye2's linear layer. This was expected to allow the model to leverage a generalized visual representation while accommodating individual subject neural variability. Training performance after an expensive distributed training run of 1250 epochs on 2 NVIDIA RTX A5500 24 GB cards lasting 4̃8 hours of wall clock time showed a decreasing training loss in early epochs. However, logging inaccuracies and scaling problems (relative to the various submodules of MindEye2) potentially influenced perceived convergence dynamics. At the end of the training run, the total training loss plateaued, suggesting optimization difficulties potentially linked to gradient scaling issues or mismatched feature embedding scales. Reconstruction quality, assessed using metrics such as cosine similarity (recon_cossim), initially exhibited significant fluctuations, characterized by sharp spikes and subsequent drops in the early epochs. Disproportionate gradient contributions from different branches (diffusion-prior, CLIP, and blurry-image branches), caused early optimization stagnation. While demonstrating the effectiveness of SPLORA in preprocessing and beta generation, the results underscored the necessity of careful logging, module optimization and subject-specific considerations.

2 Introduction

Neural decoding techniques applied to fMRI data enable the reconstruction of perceived visual scenes with the help of machine learning models. These models are trained to recognize associations between brain signals (elicited when visual stimuli are presented) and the corresponding pixel-level features of the stimuli.

State-of-the-art neural decoders like MindBridge and MindEye2 already achieve cross-subject semantic and visual decoding of viewed images[1, 2]. However, the reconstruction accuracy and subjective quality of the images generated by these models is limited by the latent features learned from their labeled training sets. This project pushes these methods further by attempting to reconstruct hallucinated mental imagery: we retrain the models on novel, unlabeled fMRI data to decode out-of-distribution experiences such as the vivid scenes reported under DMT. This task is inherently unsupervised and the decoding technology is still evolving. This section highlights the various combinations of probabilistic encoders, GANs and diffusion models that have been developed for this task. The overall theoretical progress in this field however is throttled by deeper systems-neuroscience bottlenecks. Recent studies show that our understanding of how 'real' and 'imagined' percepts differ in neural activity is still incomplete [3].

While neural decoding algorithms have become more popular in the last 3 decades, understanding how the brain transforms raw retinal inputs into abstract, 'behaviourally-useful object representations has been a central theme of systems neuroscience since the early days of the field. Systems neuroscience describes a hierarchical organisation of the visual system; from simple feature detectors up to complex representations [4].

In the following sections we outline some of the significant breakthroughs in systems neuroscience and neural decoding which have lead to the current state of the art in visual image reconstruction from fMRI signals. While this outline does not exhaustively include all the theoretical additions made to the field of visual neuroscience, it attempts to highlight some important findings that have allowed for the development of the field of neural decoding.

2.1 Early lessons from Systems Neuroscience on Neural Correlates of Visual Encoding

In 1962, Hubel & Wiesel provided the first descriptions of orientation-selective neurons in the cat primary visual cortex (V1), establishing the idea of local edge detectors in primary visual cortex[5].

Following more progress in the field in 1982, Marr and colleagues proposed a foundational three-level framework to dissect visual perception into computational, algorithmic, and implementational levels. At the computational level, one asks what function is being performed by the visual system and why for instance, for inferring object identity from retinal input under ecological constraints. The algorithmic level is expected to specify how this is done in terms of the nature of the representations (e.g., edges, surfaces) and the procedures (e.g., iterative grouping, constraint satisfaction) that realize the computation. Finally, the implementational level grounds these processes in where and by what physical media they occur, whether in neurons, circuits, or silicon. This tripartite scheme offers a powerful scaffold to systematically link theoretical models with empirical brain data in visual neuroscience[6]. In 1991, Felleman and Van Essen introduced the distributed hierarchical processing model to map the primate visual cortex's connectivity and processing architecture including 25 neocortical areas that function for vision. They identified a web of 32 interconnected visual and associational areas organized in a hierarchy of feed-forward and feedback loops across occipital and temporal cortices. Their work emphasized that visual perception isn't a simple feed-forward cascade but the result of rich, bidirectional interactions across multiple cortical levels. By charting over 300 connections, they provided a structural blueprint explaining how visual information flows and transforms across the primate cortex[7].

2.2 Advent of Neural Decoding from fMRI data

The period between 2000-2010 witnessed a significant rise in breakthroughs in the developments of algorithms and software packages which allowed for detailed modeling and analysis of neuroinformatics datasets. With an established model showing top-down constraints on basic feature detection, Kamitani and Tong were the first to show that ensemble patterns of fMRI activity in early visual areas (V1/V2) could reliably decode the orientation of grating stimuli in 2005. Using multivoxel pattern-classification techniques, they not only predicted which of eight orientations subjects were viewing on single trials, but also demonstrated that attention modulated these representations[8]. Building on this, a 2007 review by DiCarlo and Cox explored a more computational model and summarised how the inferior temporal cortex achieves position-, scale- and illumination-tolerant object recognition through successive pooling operations. They described how vision can be represented in high-dimensional space and how neuronal populations in the ventral visual stream progressively build tolerance to common image transformations, "untangling" object identity from visual variability[9].

In 2008, Miyawaki et al. showed that simple binary pixel patterns (random, geometric, letters) can be reconstructed from fMRI signals captured from early visual areas. Their approach decoded small 10×10 checkerboard images from V1/V2 activity[10]. In the same period, Kay et.al developed a voxel-wise encoding model based on a Gabor wavelet pyramid to predict voxel responses in early visual cortex (EVC) and demonstrate identification of natural images from fMRI activity. By mapping image features across orientations, spatial frequencies, and positions to cortical activity, the model could synthesize responses for test images and reliably identify which image a subject was viewing among a test set[11].

Around the same time, Naselaris et al. developed one of the first multi-modal Bayesian decoders capable of reconstructing complex natural images. This decoder combined features derived from fMRI signals in early and anterior visual areas with semantic and structural information extracted from natural images. The structural model encoded features like orientations and spatial frequencies using Gabor wavelets, while the semantic model classified images based on human-labeled categories. Each model defined the probability of neural responses using Gaussian distributions. Specifically, the structural model uses a single Gaussian centered around image-based features, and the semantic model uses a combination of several Gaussians, each centered on specific semantic categories [12]. Although these early studies were limited to relatively coarse or constrained reconstructions – e.g., simple geometric patterns or selecting the correct image from a fixed set – they established the feasibility of image reconstruction from brain signals by combining structural and semi-mechanical information.

Shortly thereafter, in 2011, Nishimoto et al. developed a motion-energy encoding model that was able to model brain activity generated by dynamic stimuli such as natural movie stimuli - reconstructing continuous video clips from brain activity by finding clips in a large database whose features matched the subject's fMRI responses from the occipito-temporal visual cortex. The reconstructed movies were blurry and composite, but recognizable, marking the first reconstruction of dynamic perceptual experiences from fMRI[13].

2.3 Deep Learning for Neural Decoding of Mechanisms of Vision

Findings from these neural decoding approaches have also informed systems neuroscience. As deep learning was just beginning to be applied for neuroscience, more researchers began to apply these tools to gain new insights from brain activity data. In 2015, Guclu et al. demonstrated that the human ventral visual pathway is organized as a gradual "complexity ladder": early areas encode simple visual details, while downstream regions handle richer, more abstract features. Using a deep-learning model that mapped thousands of progressively complex image features onto the cortex, they uncovered both this global gradient and a fine-grained specialization of later areas. The same model could also decode brain activity with improved accuracy, revealing that these regions were tuned to object categories. Their findings reinforced the hypothesis that object categorization is a key organizing principle of the primate ventral stream [14].

In 2016 Yamins and DiCarlo's review highlighted that hierarchical convolutional neural networks (HCNNs) optimised for object recognition naturally recapitulate the ventral-stream representational hierarchy. Their work described how a series of basic operations defined within a single HCNN layer could be typically mapped to a single cortical area[15].

With the advent of the deep learning era post-2016, Horikawa and Kamitani demonstrated the ability to decode arbitrary object categories (including those imagined by participants) by leveraging hierarchical feature representations inspired by deep neural networks (DNNs). They trained decoders to predict visual features from fMRI activity, showing that lower-level features were better decoded from the EVC and higher-level features from later stages. A key innovation was generic decoding: decoders trained on one set of objects could identify different object categories that participants saw or imagined, by matching predicted features to a large database of object categories. This study also revealed that mental imagery involves a top-down recruitment of hierarchical representations, progressing from higher to lower visual areas [16]

Later in 2019, Gerven et al. reviewed fMRI evidence to show that mental imagery and hallucination recruit high-level conceptual areas first and then propagate activation back to early visual cortex. Their work highlighted how perception and imagination recruit shared neural mechanisms and need to be researched further to explain the overlap[17]. In the same year, Shen et al. introduced a deep image reconstruction technique that generated visual content from brain activity by combining fMRI decoding with deep learning methods. Their method involved decoding fMRI signals into feature representations across multiple layers of a pre-trained DNN, such as VGG19. An image was then generated through an optimization process, adjusting pixel values so that the DNN features of the generated image matched those decoded from brain activity. An optional component used a deep generator network (DGN), based on Generative Adversarial Networks (GANs), to produce more naturalistic images. This approach successfully reconstructed recognizable versions of viewed natural images, artificial shapes, and letters, and achieved rudimentary reconstruction of mental imagery, visualizing subjective content [18].

The introduction of the Natural Scenes Dataset (NSD) marked a turning point in visual neuroscience and decoding, due to its unprecedented scale and quality. Collected at ultra-high-field 7T, NSD spanned 8 participants, each scanned over 30–40 sessions while viewing 9,000–10,000 distinct natural images. To maximize signal and interpretability, the NSD team introduced innovative analysis methods: GLMsingle [19] (a single-trial GLM pipeline with voxel-specific HRFs and aggressive denoising) and fractional ridge regression for robust voxel-wise modeling. These techniques dramatically improved data SNR and reliability, enabling clear neuroscientific insights. For example, NSD revealed systematic representational gradients along the ventral visual stream (with lower-level visual areas to higher-level areas showing progressively more complex, abstract feature tuning). High-level regions exhibit categorical clustering of scene and object representations and the geometry of these representations is consistent across individuals – NSD showed that patterns of representational similarity are highly reproducible across subjects. This study's findings also advanced neural decoding by enabling the training of "brain-optimized" deep networks (e.g. a custom CNN dubbed GNet) that predict fMRI responses far better than standard computer-vision models and was able to generalize across subjects[20].

In the last 2 years, diffusion models and multi-modal representations have pushed the state-of-the-art even further. Takagi et al. employed the latent diffusion model Stable Diffusion [21] to generate high-resolution images from fMRI, using a two-step approach that decoded semantic content (via text descriptors) and low-level image features, achieving unprecedented semantic accuracy in reconstructions [22] . Ozcelik et. al similarly introduced Brain-Diffuser, a latent diffusion-based decoder that produced naturalistic images from brain data [23] . These approaches leverage powerful pre-trained vision-language models (like CLIP; Radford et al., 2021) to imbue reconstructions with semantic coherence [24]. The culmination of this trajectory are models such as MindEye. The first version, MindEye1 (Scotti et al., 2023), combined a contrastive encoder to map fMRI into CLIP image-embedding space with a diffusion prior network to generate images from those embeddings. MindEye1 established a new standard on the Natural Scenes Dataset (NSD)[20] by achieving more accurate reconstructions than prior methods across multiple perceptual metrics [25]. Building on this, MindEye2 extended the framework to be more data-efficient and generalizable. Notably, MindEye2 introduced a shared latent space across subjects ("shared-subject" modeling) by training on multiple NSD subjects jointly [2]. It also upgraded the architecture with a richer CLIP model (OpenCLIP ViT-bigG/14) and integrated both high-level and low-level pathways into a unified pipeline. By fine-tuning a single pretrained model on each new subject, MindEye2 achieved state-of-the-art reconstruction performance on NSD while using as little as 2.5% of that subject's data for fine-tuning. In parallel, MindBridge (Wang et al., 2024) was proposed as another solution to cross-subject decoding, using a cyclic reconstruction training scheme to learn a subject-invariant representation of fMRI data that enabled one model to decode multiple subjects [1]. Together, MindEye2 and MindBridge represent the cutting edge in "generically trained" decoders, which may no longer be needed to be built de-novo for each individual.

A crucial enabling factor for these advances has been the availability of large-scale, high-quality datasets. In particular, the Natural Scenes Dataset (NSD) provides fMRI responses from 8 subjects each viewing 30,000+ distinct images (drawn from the COCO dataset) over many hours of scanning [20]. NSD's breadth and consistency allowed researchers to train complex deep learning models and to explore alignment of neural representations across individuals. MindEye2, for instance, leverages NSD to learn a common latent mapping that generalizes to new subjects. By pretraining on multiple subjects' data, the model learns a subject-invariant latent space – a representation of visual stimuli that is shared across different brains. In practice, this means MindEye2 can encode fMRI patterns from any subject into a unified feature space (specifically, the CLIP image embedding space), after accounting for individual differences via a brief fine-tuning. The implication is powerful: if human brains have at least partially aligned representational spaces for vision [26], a decoder trained on one set of individuals could be adapted to another with minimal additional data. This cross-subject transfer could dramatically expand the applicability of neural decoding, since new subjects (or patients) might not require extensive, subject-specific training. Another recent Diffusion based model, MindBridge explicitly targets such cross-subject transfer, reporting competitive performance with a single model across individuals [1]. MindEye2's results likewise suggest that a shared decoder can extract meaningful information even from an unseen subject given a small calibration dataset. However, it remains an open question how far this generalization can be pushed, especially when the new data differ in nature from the training domain. In this experiment we attempt to perform inference and reconstruction with MindEye2 by plugging a linear layer trained on novel fMRI-stimulus pairs. We also attempt to evaluate the performance of this modified version of MindEye2 to perform inference and visual reconstruction on a dataset consisting of data collected during a subject’s DMT intervention.

2.4 DMT in Clinical Research

DMT (N,N-dimethyltryptamine) entered modern clinical research in the early 1990s with the groundbreaking work of Rick Strassman. Between 1990 and 1995, Strassman conducted the first government-approved human DMT studies in the US, administering approximately 400 doses of intravenous DMT to about 60 volunteers. In 2000, these experiments were recounted in ”DMT: The Spirit Molecule” [27] and reintroduced DMT to scientific discourse and helped spur the current “psychedelic renaissance” in research [28]. In his reports, Strassman extensively described the vivid, subjective experiences reported by study participants. Volunteers experienced an immediate onset of fully immersive hallucinations that peaked within $\tilde{2}$ minutes of intravenous administration of DMT. These experiences tended to subside within $\tilde{3}0$ minutes[29]. While the subjective effects of DMT can be described in a range of clinically descriptive categories, this thesis focuses on tackling only the aspects of visual perception experienced by subjects during their DMT experience. When high-dose DMT (0.4-mg/kg) was intravenously administered, participants described ”rapidly moving geometric patterns”, ”brightly colored visuals” display of images” that ”completely replaced normal mental content” and were “more vivid and compelling than dreams or waking awareness”. These early clinical observations, included the development of the Hallucinogen Rating Scale (HRS) to quantify such effects [30].

Further studies on electropshysiological correlates of psychedelic induced altered states have shown the spatio-temporal pattern of cortical activation similar to what is observed under visual stimulation. In other words, even if the subject is experiencing a signal elicited due to DMT leading to a subjective experience, their brain is processing the visual contents of that signal similar to how it processes external visual stimulus[31].

Research by Dijkstra et al. in 2023 provided evidence supporting a theoretical model of in which reality and imagination are subjectively intermixed by the brain to determine a unified sensory experience. This study has wide implications on the visuals experienced by subjects under the influence of psychedelics. In this study participants were presented with distinct images and were asked to imagine one or the other image while brain activity was recorded. Participants were also asked to rate the vividness of seen and imagined gratings. Their results suggested that when imagery does become vivid or strong enough, it can be neurally indistinguishable from perception. Therefore, if the neural mechanisms behind DMT visuals are to be considered similar to imagined experiences, the subjects’ descriptions of ’more real than real’ visuals may hold some water[3].

In 2024, Tipado et al. found that interneurons in the retina (amacrine cells) could be the primary site of visual psychedelic modulation responsible for disrupting the hierarchical structure human visual information perception. Their study provided a model for how the retinofugal pathway communicates and modulates visual information in psychedelic and clinical conditions[32]. A recent study by Aqil et al. also found that psychedelics such as psilocybin alter visual-contextual computations capable of general computational mechanisms for psychedelic effects in the human brain. Their work also proposed a computational model for capturing and linking these changes while highlighting the alteration of contextual computations as a potential general mechanism for psychedelic action[33].

3 Research Question

3.1 Inspiration and Motivation

The evolution of neural decoding techniques discussed so far suggests that this field has reached categorical, semantic and visual level of subject invariant decoding from large fMRI datasets. Although the dataset size and model size are smaller compared to those typically used in typical machine learning applications, frameworks such as MindEye2 and MindBridge represent state-of-the-art architectures for fMRI-to-image foundation models. Taking this into consideration we sought to test the viability of a novel model to decode the movie frames and DMT visuals experienced by our subject. As part of this effort, we are required to test MindEye2's zero-shot performance and compatibility with the same novel dataset.

3.2 How was MindEye2 trained?

The fMRI BOLD signal is an indirect measure of neural activity typically modeled as a latent neural event time series convolved with the HRF. When stimulus onsets/offsets are known, general linear models (GLMs) deconvolve the HRF with respect to a design matrix to estimate voxelwise responses. GLMsingle is a high-quality variant that refines this estimation at the trial level when precise timings are available. GLMSingle was specifically designed to estimate high-quality voxelwise responses to image presentations from the COCO dataset for the NSD experiment. These single trial signal estimates generated by GLMsingle are in the form of beta weights for each image shown to the subject during the NSD experiment.

MindEye2 was trained on 10,000 of these single trial beta estimates derived from 7T fMRI data collected during the NSD experiment. This model was then used for visual reconstruction of a test set of images seen by the subjects that the model was not trained on. In simpler terms, in MindEye2's training framework each unique image seen by the subject is treated as a unique event with a corresponding set of beta weights associated with that event.

3.3 Description of our dataset and associated challenges

In naturalistic settings like movies, onsets and offsets are dense and overlapping; in subjective experiences like DMT imagery they are unknown. In this project, we aim to retrain MindEye2 to reconstruct movie frames and DMT visuals seen by the subject. In contrast to MindEye2' original framework, our experimental dataset poses several unique challenges: only 1000 pulses of 3T fMRI data were collected from a single subject while they watched 36000 frames of the film *Lost in Translation*, and no predefined stimulus timing or event structure exists to facilitate standard GLM-based preprocessing.

The retraining of the MindEye2's linear layer requires event-level fMRI representations to serve as training examples, analogous to the trial-wise responses used in its original training. Unlike a standard task-based experiment, our dataset (fMRI during DMT administration) had no externally timed events or stimulus annotations, meaning there was no "ground truth" timing of neural events. In such cases, conventional GLM-based deconvolution is infeasible as GLMs rely on known stimulus onsets to estimate trial-wise betas. We therefore needed to infer discrete event responses directly from the continuous BOLD time series. Event-level beta-volumes were essential because the decoder is trained on isolated neural response patterns (as opposed to continuous timepoints). Without segmenting the time series into event responses, we would lack any well-defined input targets for retraining the model's linear layer.

3.4 How SPLORA can address our challenges

In this project, we utilize SPLORA to estimate beta-like activation patterns without relying on well-defined event timings. SPLORA was chosen as a necessary solution for deriving these event-level betas in the complete absence of stimulus timing. SPLORA performs "paradigm-free" hemodynamic deconvolution, meaning it can detect moments of putative neural activation without any pre-specified design. The original SPLORA study reported results comparable to single-trial GLM activation maps and to other multivariate deconvolution methods such as ME-SPFM (Multi-echo Sparse Paradigm Free Mapping) and Hemolearn [34, 35, 36].

This capability makes SPLORA the only viable approach to obtain training signals when no task or timing information is available for the DMT dataset. In essence, SPLORA allows us to recover a beta-series of event responses where traditional design-based methods cannot operate. This in turn provides voxelwise responses that can be paired with the corresponding movie frames and used to train a subject-specific linear mapping layer that plugs into the other modules of MindEye2. In simpler terms, SPLORA provides us with the closest replacement for GLMSingle to produce the kind of dataset that MindEye2 can be retrained on.

A review of the literature on both neural decoding and clinical research on DMT showed that such a machine learning experiment for neural decoding has not been reported before. Hence, this thesis is aimed at taking the first step for this project by posing the following research questions:

3.5 Main Research Question

Can MindEye2 be repurposed to perform neural decoding and visual reconstruction of movie frames from a 3T fMRI dataset of a lower spatial resolution?

3.6 Sub-questions

- Can Sparse and Low-Rank Paradigm-Free Mapping (SPLORA) facilitate effective preprocessing of an fMRI dataset in the absence of precisely defined event timings?
- Can this dataset facilitate fine-tuning of MindEye2 on a movie dataset for zero-shot decoding and reconstruction of DMT-induced visual experiences?

4 Theory

In the following sections, we cover the various methods that are important to understand the general theoretical framework of the original MindEye2 experiment and our modified version of it.

4.1 NSD Dataset and GLMsingle Pipeline

The Natural Scenes Dataset (NSD) was collected on 7 Tesla MRI scanners with whole-brain coverage at approximately 1.8 mm resolution and 1.6 s TR. Eight participants each viewed approximately 9,000–10,000 distinct natural scene images over the course of 30–40 sessions, following a rapid event-related design in which each trial was a brief image presentation with intervening fixation. The NSD preprocessing pipeline included slice timing correction, motion correction, distortion correction, coregistration to anatomy, and normalization to common space. Further denoising and GLM-based analysis was applied to estimate stimulus-driven responses. Using the GLMsingle/GLMdenoise framework, a general linear model was fit with one regressor per image trial plus confound regressors to each voxel's time-series. The GLM incorporated voxel-specific hemodynamic response functions and data-driven noise regressors to capture physiological, motion, and scanner artifacts. Ridge regression was used to stabilize beta estimates, penalizing noise without biasing true signals. The result for each subject was a set of single-trial beta weights – one per image per voxel – with improved SNR and test–retest reliability. This pipeline assumes each image onset produces a BOLD response, uses a GLM to deconvolve overlapping responses given known timing, and produces a stable activation pattern vector for every stimulus [37, 20].

4.2 MindEye2's Neural Decoding Pipeline

MindEye2 is a multi-module deep learning architecture designed for fMRI-to-image reconstruction. We briefly describe its components and how they were configured in this project (see Figure 1 for an overview of the pipeline). The input to MindEye2 is a vector of fMRI activations – in our case, the EVC beta-series vector (one value per ROI voxel). This input is first passed through a brain encoding network, which in MindEye2 is implemented as a residual multilayer perceptron (MLP) backbone. This MLP (with several hidden layers and skip connections) outputs a latent feature vector that is meant to align with the CLIP image embedding space. In other words, the MLP learns to map a pattern of neural activity to the kind of representation that CLIP's image encoder would produce for the seen image. This is trained by comparing to actual CLIP embeddings, as described shortly. In MindEye, this mapping was trained separately per subject[25]. MindEye2 innovates by training a shared mapping that is initialized on multiple subjects' data and can be transferred. Following the MLP, the architecture is split into multiple submodules that tackle different aspects of image reconstruction:

The following is a detailed description of how MindEye2's submodules contribute to image reconstruction from a subject's stimulus:

- Diffusion Prior Module (High-Level Pathway): The output of the MLP is fed into a diffusion. This diffusion prior is a generative model that refines the MLP's output so that it better matches the distribution of valid CLIP image embeddings. The diffusion prior takes the preliminary brain-predicted embedding and iteratively denoises it to approach a realistic CLIP embedding that could be passed into an image generator. In practice, it is implemented as a sequence of transformer-based denoising steps conditioned on the brain embedding. After this process, we obtain an aligned CLIP embedding – a vector in CLIP latent space that encodes the high-level content of the reconstruction. This high-level pathway is responsible for capturing semantic and abstract information (the category of objects, general scene layout, etc.), ensuring the final image is meaningful and not just a random collection of pixels. MindEye2 uses an updated CLIP model (ViT-bigG/14) with a larger latent dimension than earlier work, and a diffusion prior tuned for Stable Diffusion XL's image embedding space.

- Image-Decoder (VAE) Module (Low-Level Pathway): In parallel to the diffusion prior, MindEye2 also maps the fMRI data into the latent space of a generative image decoder to capture low-level visual. Specifically, the model contains a branch that predicts the VAE latent code corresponding to the input image. Stable Diffusion's generative pipeline includes a variational autoencoder (VAE) that encodes images into a lower-dimensional latent representation and decodes latents back to images. MindEye2 learns to output the VAE latent (a $64 \times 64 \times 4$ dimensional tensor for SD 1.5, or analogous for SDXL) that would reconstruct an image with matching low-level properties (color, texture, basic shapes) of the seen stimulus. When this latent is passed through the VAE decoder, it produces a blurry reconstruction of the image – one that gets the overall appearance right but lacks fine details and precise semantics. This low-level pathway ensures that the decoded image has the correct pixel-level structure (e.g., the general positions of bright vs dark regions, the presence of large shapes or blobs of color corresponding to objects) even if it might be ambiguous what those objects are. The blurry image by itself is akin to prior fMRI reconstructions that matched low-level features but not high-level content.

- Image Reconstruction and Fusion: The final step of MindEye2 combines the high-level and low-level information to produce the output. In MindEye2's implementation, this combination is done through the Stable Diffusion[21] generative model. The aligned CLIP embedding from the diffusion prior serves as a condition for Stable Diffusion's image generation. Instead of starting the diffusion process from pure noise, the model uses the predicted VAE latent as an initial condition – effectively "starting the denoising with the blurry reconstruction" as a baseline. By doing so, the diffusion process is guided to fill in details (in accordance with the semantic embedding) on top of an image that already has the correct coarse structure. Additionally, MindEye2 can incorporate a text caption to guide the diffusion decoder. In their full model, a separate module predicts a short text description of the image from fMRI (trained via a cross-modal mapping or by leveraging the retrieval output), and this caption is used as an extra conditioning input to Stable Diffusion (which is a text-to-image model). This helps ensure that specific semantic details align with what the subject saw. In summary, the generative decoder (Stable Diffusion XL in MindEye2) takes three conditioning inputs: (a) a CLIP image embedding (from the diffusion prior) encoding what to draw, (b) a text string with keywords of the scene, and (c) an initial image latent (from the VAE pathway) telling where and in what color to draw things. The output is a final reconstructed image. MindEye2's design thus merges the previously independent low-level and high-level pipelines into one integrated reconstruction process.

- Retrieval Module (Auxiliary): In addition to the image-generating components, MindEye2 includes a retrieval submodule. This is essentially a secondary encoder branch that also maps fMRI data into CLIP space, but it is trained with a contrastive learning objective: the fMRI-based embedding is pulled to be similar to the embedding of the true image and dissimilar to embeddings of other images in the batch. The result is a representation that is optimized for identifying the correct image among many – effectively performing an image retrieval or classification task from brain data. During training, this helps impose that the brain embedding contains discriminative information about the stimulus. At inference, one can use this module to retrieve the nearest neighbor image from a large library, given a brain scan (as was done in some prior works). Important: The retrieval module is not used to generate the reconstructed image; it is an auxiliary branch used for evaluation and potentially as a fallback decoding mode. In our use of MindEye2, the

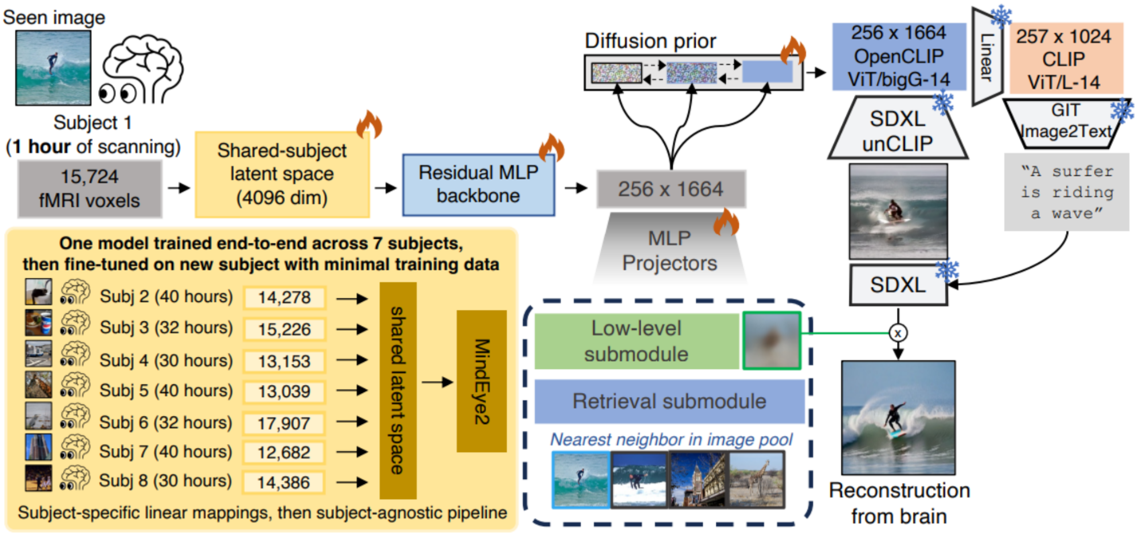

Figure 1: MindEye2 Architecture

retrieval branch served to compute retrieval accuracy metrics, but it did not directly affect the image outputs.

All the submodules of MindEye2 are illustrated in the figure below:

To summarize, MindEye2 maps fMRI activity to multiple latent representations (CLIP embedding, text, VAE code) and then uses a diffusion-based image generator that accepts those representations to create an image. The architecture is complex, reflecting the multifaceted nature of the task – capturing low-level vs. high-level visual information – and is illustrated in Figure 1. All parts of the model (MLP, diffusion prior, VAE predictor, retrieval net, etc.) were originally trained on the NSD dataset in MindEye2's development, producing a set of weights that can be fine-tuned to new data.

4.3 MindEye2's use of NSD Betas

In the MindEye framework, the β vectors corresponding to NSD images are treated as feature vectors of brain activity. A mapping is learned from those fMRI features to a latent image representation. MindEye2 uses a two-stage model: first, a linear functional alignment maps individual-subject voxel spaces into a common latent space (using ridge regression on betas); second, a nonlinear mapping (an MLP "adapter") maps this aligned brain space to the embedding space of a pretrained vision model (OpenAI's CLIP). By training across seven NSD subjects' data, MindEye2 learns a shared encoder that can encode any subject's brain-pattern into a CLIP latent vector, which can then be passed into a generative image model. In practice, MindEye2 employs a frozen pretrained diffusion generative model (e.g. a custom Stable Diffusion) that accepts CLIP image embeddings to produce images.

4.4 SPLORA Paradigm-Free Deconvolution

Multivariate sparse paradigm-free mapping (Mv-SPFM), implemented in SPLORA , performs hemodynamic deconvolution at the whole-brain level and adds spatial information via a mixed-norm regularization over voxels. Stability selection removes the need to tune regularization parameters manually and yields an estimate of the probability of a neuronal-related BOLD event at each voxel and time point via the area under the stability paths. SPLORA can run without stimulus timing, which is appropriate for continuous or naturalistic stimuli such as movies or resting-state conditions with pharmacological perturbations. Default regularization parameters are commonly used; a block-model debiasing step refits detected events to correct sparsity-induced amplitude shrinkage. Compared with GLMsingle, SPLORA does not require a design matrix and treats each time point as a potential event, deriving activity-inducing signals directly from the data[36].

5 Methods

5.1 Participant

One neurologically healthy, right-handed, native English speaking male aged 27 took part in the experiment, which included one feature-length movie *Lost in Translation* starring Bill Murray) scans and one 10-minute infusion of 27.5 mg N,N-Dimethyltryptamine (DMT) in a 3T MRI scanner (UCL Research Ethics Committee ID: 17715.01).

5.2 MRI Acquisition for DMTV

Functional and anatomical images were acquired using a 3T Siemens MAGNETOM Prisma with a 30-channel head coil (Siemens Healthcare, Erlangen, Germany). For functional images, a multiband EPI sequence was employed (TR = 1.5 s, TE = 54.8 ms, flip angle = 60°, 72 interleaved slices, isotropic resolution 2 mm, 4× multiband factor, no in-plane acceleration). To reduce cross-slice aliasing, the leak block option was enabled. Anatomical scans were acquired after functional scans using a high-resolution T1-weighted MPRAGE-GRAPPA2 sequence (TR = 2.3 s, TE = 2.98 ms, 256 sagittal slices, resolution 1.0 mm).

Audiovisual movie stimuli were controlled via a custom MATLAB script using the psychtoolbox library for simultaneous logging and synchronisation of scanner pulses and movie-frame presentation timestamps. More specifically, this script was triggered to start movie playback upon the registration of the 9th TR of each run such that the first 8 dummy volumes of each run did not comprise any stimuli. Movies were all manually divided into three separate runs to mitigate drift artefacts and if needed, allow participants the opportunity for a break. Movies were presented on a back-projected screen inside the bore of the scanner via a projector in the next room, which participants viewed via a head coil-mounted periscope. Audio was presented to participants via Sensimetrics S14 insert earphones. Movie scans took place one week prior-to, and one week following DMT infusion scans.

Functional image acquisition during DMT scans employed the same sequence parameters as the movie scans described above. Prior to the scanning session, participants were fitted with an MRI-safe cannula and infusion line in their left forearm to facilitate the administration of DMT. Once the setup was complete, participants underwent a resting-state scan. The scanning sequence was initiated upon the participant indicating their readiness by giving a thumbs-up gesture. An experimenter present in the room manually operated the MRI 3860 infusion pump to begin the infusion. Concurrently, the exact start time of the DMT infusion relative to the scan commencement was recorded using an MRI-safe button press box, with the timing data captured by a custom MATLAB script alongside the scanner pulse onsets. A total of 27.5 mg of DMT was infused at a steady rate over 10 minutes, and the entire scanning sequence lasted 25 minutes. This structured approach ensured precise synchronization between the DMT administration and the functional imaging, facilitating accurate assessment of the drug's effects during the resting-state scan.

5.3 Preprocessing for DMTV

Anatomical preprocessing was conducted using FreeSurfer's recon-all pipeline (v6.0), which involved motion correction of the T1-weighted anatomical scan, intensity normalization, skull stripping, white matter segmentation, and cortical surface reconstruction. This process generated subject-specific masks for gray matter, white matter, and ventricles, which were subsequently eroded to minimize partial volume effects at tissue boundaries.

Functional preprocessing was carried out using a custom afni_proc.py pipeline, with all steps performed in native subject space. Slice timing correction was applied to align slices to the middle TR, and motion correction aligned EPI volumes to the volume with the least motion outliers using heptic interpolation. Functional data were then coregistered to the subject's FreeSurfer-processed anatomical scan using boundary-based registration. Physiological noise correction included motion parameters (6 demeaned and 6 derivatives), principal components from eroded white matter and ventricular masks (3 PCs each), and ANATICOR-based local white matter signal regression. Data were scaled to percent signal change units, and a high-pass filter was applied at 0.01 Hz.

5.3.1 Early Visual Cortex Mask Creation

The NSD_general mask is a probabilistic atlas of the early visual cortex derived from the natural scenes dataset. This mask was projected into subject space using surface-based registration. Hemispheric masks (lh.nsdgeneral.mgz and rh.nsdgeneral.mgz) in fsaverage space were mapped to the subject's cortical surface using spherical morphing (mri_surf2surf). Surface ROIs were then converted to volumetric space (mri_surf2vol) with a mid-gray matter projection (projfrac 0.5). Left and right hemisphere masks were combined and resampled to match the resolution of the functional data using EPI geometry as a template (3dfractionize). A binary mask was created by thresholding at 1% probability and binarizing (3dcalc)[37].

This subject-specific NSD_general mask was applied to the preprocessed functional data for further analysis. All processing was performed in native subject space, avoiding template registration to preserve individual anatomical variability.

5.3.2 Beta-Series Generation for DMTV with SPLORA

After the above-mentioned preprocessing steps, we treated the resultant dataset with the SPLORA algorithm. For each run, the preprocessed 4D fMRI time series was processed by SPLORA in paradigm-free mode. Command-line inputs included the bias-corrected, motion- and distortion-corrected fMRI volume (flag -i), a whole-brain mask (flag -m), and the empirical TR (flag -tr), set to 1.5 s. An output prefix was specified to produce one beta volume per TR. No stimulus timing was provided; each time point was treated as a potential event.

SPLORA's regularization parameters were left at their defaults. For example, the 'group' parameter (which controls inter-voxel grouping constraints) remained at 0, meaning no additional group sparsity was enforced. Likewise, we used SPLORA's default 'debiasing' setting, which in this case is the so-called "block" (innovation) model. This debiasing means that after the initial sparse deconvolution solution is found, SPLORA refits each detected event amplitude using a simple block-design (rectangular HRF) model. This post-hoc refitting corrects the LASSO-like shrinkage of the sparse estimates and yields unbiased beta-weights. The block-model debiasing thus refines each per-TR beta by fitting a canonical block response at that timepoint, reducing bias from the sparsity penalty. (This is important because raw sparse estimates tend to underestimate amplitudes; the block-model refit restores the true scale of each event.)

Runs were segmented into overlapping windows of 250 TRs (5 minutes) with 50 TR overlaps. Within each window, SPLORA-derived activity-inducing signals were computed and refit by the block model; the first and last 50 TRs of each window were discarded to remove edge artifacts. Windows were then stitched to cover 1000 TRs for the movie and 880 TRs for the DMT scan. Each SPLORA-derived beta volume was restricted to the early visual cortex mask, yielding a 9,423-voxel vector per TR.

After SPLORA finished, the outputs were NIfTI volumes of estimated beta coefficients – essentially one brain-volume per TR per run – named using the specified output prefix. These single-trial beta volumes (for both movie and DMT runs) were the final product of the deconvolution step. In summary, by running SPLORA with no design and block-model debiasing, we obtained voxelwise, single-TR beta series from the fMRI data.

5.3.3 Frame-to-TR Alignment in the Movie Dataset

The movie was presented at 24 fps while fMRI acquisition used TR = 1.5 s, meaning each TR encompassed roughly 34 frames (1.5 s × 23–24 fps). SPLORA outputs a beta volume per TR, interpreted as a deconvolved estimate of neural activity at that temporal resolution. To align these betas with visual input, we selected the middle frame of each TR (approximately 0.75 s into the interval) as a representative snapshot of the visual content. Visual inspection of random TR windows confirmed that differences between the first, middle, and last frames were negligible, making the midpoint frame a reasonable choice. This procedure yielded 1000 beta–image pairs for the movie run. For the DMT run, 880 beta vectors were obtained without paired ground-truth images. This yields 1000 labeled (ground truth available, not captions) beta–image pairs for training/validation and 880 unlabeled betas for zero-shot testing for the DMT dataset.

5.4 Dataset Split

Movie beta–image pairs were split chronologically into 70% training (700 pairs) and 30% validation (300 pairs). All the 880 betas (post-SPLORA) from the DMT dataset were treated as the final test set for testing zero-shot decoding performance.

5.5 Fine-Tuning Strategy for the Movie Dataset

Applying MindEye2 to our 3T movie-watching dataset required an adaptation procedure, as our dataset is far smaller and different in nature from NSD. We adopted a transfer learning approach, where the bulk of MindEye2's parameters were kept fixed (frozen) and a small number of subject specific parameters were learned anew on the movie data. Specifically, we introduced a new linear mapping layer at the input of the network to account for differences between our subject's brain data and the original model's expectations. This linear layer takes the EVC ROI activity vector (dimensions 1̃500) as input and outputs a vector of the dimension expected by MindEye2's pre-trained MLP (trained on NSD data). In essence, this layer serves as an adapter transforming our subject's voxel space into the NSD multi-subject feature space. Only the new linear adapter's weights were optimized. This method leverages the assumption that MindEye2 has learned a subject-invariant latent space. If our subject's EVC responses can be linearly mapped into MindEye2's subject invariant latent backbone, we sought to test if the rest of the model could generate reasonable images without retraining.

For training the linear layer, we used the set of movie frame–fMRI pairs (from the beta-series analysis). We split the data into a training set and a validation set. The first 7̃0% of segments were used for training and the last 3̃0% for validation (simulating predicting later unseen scenes of the movie). This resulted in roughly 700 train samples and 300 validation samples. On each training iteration, a randomly sampled batch of brain-image pairs was fed through the model. The fMRI vectors went through the trainable linear layer and then into the frozen MindEye2 pipeline, producing outputs at subsequent endpoints (the predicted CLIP embedding, predicted VAE latent, etc.,). We computed the same loss functions used in the original MindEye2 training to drive learning (but only the linear layer's weights received gradients). The loss components were as follows: (a) a diffusion prior loss, which measured the error between the predicted high-level embedding and the actual CLIP image embedding of the target frame – this ensures the high-level semantic vector from fMRI is accurate; (b) a blurry reconstruction loss, which measured the error between the predicted VAE latent (low-level code) and the actual image's VAE latent – this encourages low-level fidelity; (c) a CLIP consistency loss on the final output image, which measured how close the generated image's CLIP embedding was to the target image's embedding (this effectively checks that the final image is semantically on point, complementing the diffusion prior loss which checks the intermediate embedding); and (d) a pixel-level or feature-level loss between the generated image and the target image (for example, MSE or perceptual loss on pixel values). Each of these losses was weighted as per the original implementation (weights chosen to balance high-level and low-level alignment). The total training loss was the weighted sum of the prior loss, CLIP loss, and blurry reconstruction loss.. Additionally, for the retrieval branch, a contrastive loss (InfoNCE loss) was computed in training batches to keep the fMRI embedding close to its true frame and away from others; this loss also backpropagated into the linear layer. In summary, the linear adapter was trained to jointly minimize differences in multiple representational spaces between the brain-predicted and actual image features.

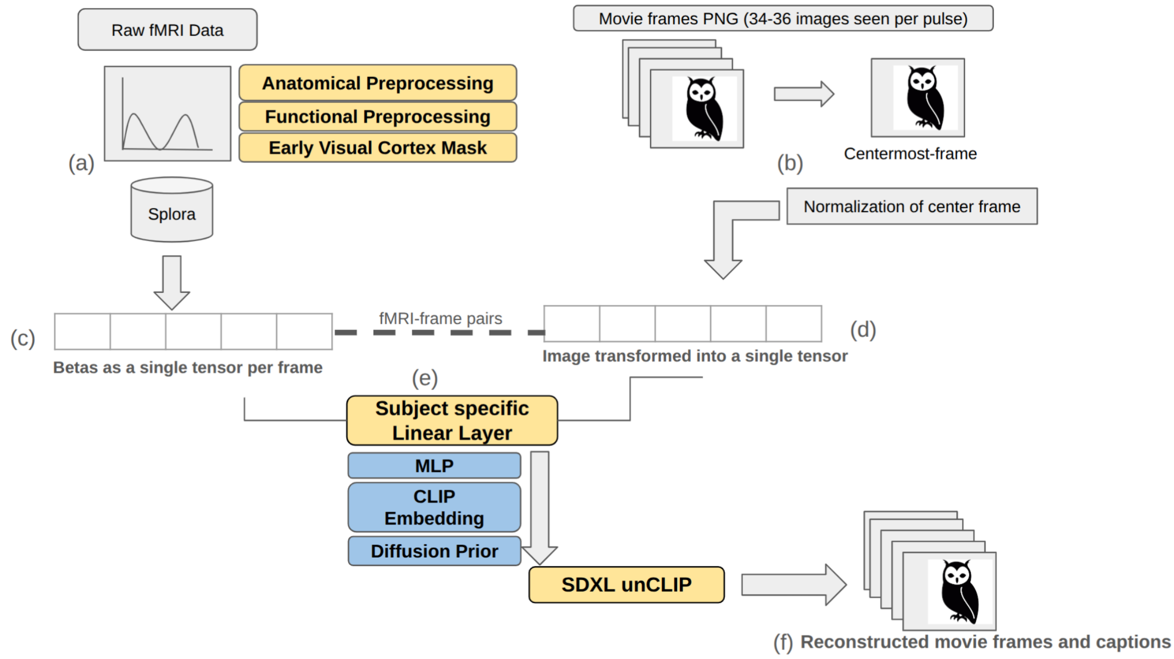
Raw fMRI Data
Anatomical Preprocessing
Functional Preprocessing
Early Visual Cortex Mask
(a)
Splora
Movie frames PNG (34-36 images seen per pulse)
Centermost-frame
(b)
Normalization of center frame
(c)
fMRI-frame pairs
(d)
Betas as a single tensor per frame
Image transformed into a single tensor
(e)
Subject specific Linear Layer
MLP
CLIP Embedding
Diffusion Prior
SDXL unCLIP
(f) Reconstructed movie frames and captions

Figure 2: Modified MindEye2 Pipeline

Training was run for 1250 epochs with the Adam optimizer (learning rate 1e-3, decayed to 1e-4 after 100 epochs). After training on the movie data, we had a fine-tuned model (MindEye2 with our linear adapter). We then performed inference with this model to two sets of inputs: (1) the held-out movie betas (the test sets) to see how well it could reconstruct those seen frames, and (2) the DMT betas. All image outputs were saved for analysis by human observers.

5.6 Decoding Pipeline Integration and Diagram Explanation

Our modified pipeline is illustrated in Figure 2. Following is an overview of the decoding setup described in this image:

- (a) We treat our raw fMRI volumes with anatomical and functional preprocessing followed by the application of an early visual cortex mask to generate a dataset similar to the ones used in the NSD and MindEye2 experiments . After these steps, the SPLORA algorithm is applied to derive betas for each recorded pulse.
- (b) These betas are aligned with the center-most (17th) $\tilde{3}$4 frames of the movie scene watched by the subject while the corresponding pulse was recorded. These center frames are converted into normalized tensors.
- (c-d) Therefore, for each pulse, the SPLORA generated betas are paired with the corresponding normalized tensor from the PNG viewed by the subject to generate fMRI-Image pairs to train a subject specific linear layer that maps our 3T beta vectors into the same feature space that MindEye2 expects (the shared latent space learned from the NSD experiment).
- (e) beta through the subject specific linear layer into MindEye2: The beta vector from

the movie dataset enters the pipeline and is used to train a linear mapping. This layer is a fully-connected layer that takes the high-dimensional 3T betas-movie frame pairs and outputs a feature vector in the MindEye2 shared latent space. In our implementation, this layer trains on 700 movie pairs, by adjusting weights so that the adapter's outputs lead to correct image reconstructions (i.e. it serves to translate our fMRI features into the domain of the NSD-trained decoder). The subsequent MindEye2 pipeline is kept frozen.

Inside the MindEye2 pipeline, the following components are indicated:

- A nonlinear mapping (MLP) that further transforms the adapted features into a CLIP image embedding. (MindEye2's core learned parameters – aside from our new adapter – reside in this MLP and associated modules. However, since we treat MindEye2 as frozen/pretrained, we do not modify these weights in our training; hence they are depicted as a fixed blue box in the diagram after the linear layer.)
- The CLIP embedding then feeds into the diffusion prior module, which is part of MindEye2's two-stage reconstruction process. This module essentially prepares a latent representation suitable for image generation (it "translates" the brain-derived embedding into the latent space of a generative model).
- Finally, the pipeline passes the conditioned latent into a frozen image generator (Stable Diffusion unCLIP model, as used in MindEye2) to produce the output image.

5.7 What stayed the same (re-used pretrained MindEye 2 modules)

- Image feature side:
  - Frozen OpenCLIP ViT-bigG image embedder (1664 × 256 tokens) – identical weights and call signature.
  - SD-XL image auto-encoder ('AutoencoderKL') and ConvNeXt-XL perceptual network for blurry-recon and contrastive losses – weights were loaded but *not* updated.
- fMRI→CLIP pathway
  - The BrainNetwork backbone (4 residual transformer blocks, h=1024) is unchanged; it still outputs a 1664 × 256 token sequence so that downstream modules trained on subjects 1-8 remain compatible.
  - The diffusion-prior 'BrainDiffusionPrior' that translates backbone text embeddings into CLIP-image space is reused exactly as in the original notebook.
  - Loss functions ('soft_clip_loss', 'soft_cont_loss', diffusion-prior MSE) and the blurry reconstruction objective are copied verbatim.
- Optimization scaffold: AdamW + OneCycleLR, mixco/mixup schedule, and BF16 mixed precision; we simply wrapped them with 'Accelerate' in the notebook.

5.8 Subject-Specific components we added or modified

The following table explains each component that was added to the pipeline to modify MindEye2's original training set up.

Component	Vanilla MindEye2	Our script (test_exp.py)	Reason
Ridge layer	Multi-subject ensemble RidgeRegression with subject index and voxel-reshape fixes	Single-subject RidgeRegression mapping 9 423 masked voxels to 1 024 dimensions; no subject index; simpler reshape	Provides a personalised linear adapter so our subject's voxel pattern can enter the shared latent space before the frozen backbone.
Dataset loader	NSD beta–JPEG WebDataset	MovieBetasDataset that pairs each pulse's central PNG movie frame with its masked-beta vector (9 423 × N pulses)	Matches the 3T acquisition and keeps I/O small.
Train / validation / test split	73k / 8k / 9k images per NSD subject	700 / 200 / 100 pulses (about one thousand images)	Allows a single GPU with batch 1 to finish an epoch quickly.
Learning schedule	150 epochs, batch 8, global_batch 64	1 000 epochs, batch 1, global_batch 2; heavier OneCycle warm-up; only ridge and backbone parameters in the optimizer	Gives the lightweight ridge plenty of updates without overfitting the large pretrained blocks.
Evaluation loop	Subject-averaged fMRI representations	Each voxel vector is tiled three times and backbone outputs are averaged (NSD-like triplicate)	Stabilises metrics with few repeats.
Logging and checkpointing	Weights & Biases (WandB) runs	save_ckpt / load_ckpt restricted to the fine-tuned layers	Avoids touching frozen weights; smaller checkpoints and safer resumes.

Table 1: MindEye2 components and corresponding adaptations in our test_exp.py script.

5.9 Why only the ridge layer had to change

MindEye2's original design allows for all the subjects to share the same 1,024-d latent "text" space produced directly after the ridge transform. By swapping in a subject-specific ridge while freezing the rest, we ensure:

- Compatibility – our embeddings have the same dimensionality and statistical profile expected by the pretrained backbone and diffusion prior.
- Efficiency – only 10 M ridge weights are learned; the 280 M-parameter backbone and prior remain fixed.
- Alignment – because the downstream CLIP and diffusion losses are computed in the original latent space, the new subject's embeddings are *implicitly pulled* toward the cloud formed by subjects 1-8, giving zero-shot generalisation to the shared generative decoder.

5.10 Total Loss Computation

A weighted sum of all active loss components.

$$\mathcal{L} = \mathcal{L}_{\text{prior}} + \alpha_1.\mathcal{L}_{BiMixCo|SoftCLIP} + \alpha_2.\mathcal{L}_{\text{lowlevel}}$$

- $\mathcal{L}_{\text{prior}}$ is the diffusion prior loss

- $\mathcal{L}_{BiMixCo|SoftCLIP}$ is the retrieval submodule
- $\mathcal{L}_{\text{lowlevel}}$ is the low level submodule

5.11 Training Budget and Rationale for Early Stopping

Following is an outline of our training regime:

- We trained the model for 1,250 epochs, with each epoch consisting of 700 mini-batches, resulting in a total of 875,000 ($700 \times 1,250$) gradient updates.
- Optimization was performed with AdamW using a OneCycle learning-rate schedule. The schedule was configured with pct_start = 2/1250 (approximately 0.16%), meaning the learning rate ramped from zero to the chosen max_lr during only the first two epochs. After this warm-up phase, it decayed smoothly until the final steps, where it reached approximately max_lr/1000.
- Under this regime, every sample in the training set was revisited 1,250 times, giving the model sufficient opportunity to learn from the SPLORA-derived features. However, as shown in Plot-1, the training loss flattened well before the completion of the cycle, indicating that further updates no longer reduced error. At that stage the learning rate was already in its final, minimal phase, so extending training would have incurred additional computational cost without a realistic chance of further convergence.

For these reasons we stopped the experiment at the planned 875 000-step budget: the optimiser had completed its full OneCycle schedule and the model had clearly reached a plateau.

6 Results

The results of adapting the MindEye2 framework to two SPLORA-preprocessed fMRI datasets—training on the movie-watching dataset and evaluating on withheld movie voxels as well as the DMT dataset—are reported here. Training was carried out for a nominal total of 1250 epochs. Due to a logging error, the first 1000 epochs were not recorded. At epoch 1000 the model weights were checkpointed, and training was then resumed for a further 250 epochs with correct logging enabled. As a result, all plots labeled with epoch indices 0–250 correspond to this resumed run (epochs 1000–1250 in wall-clock terms). We note that these curves are therefore not strictly equivalent to a continuously logged 1250-epoch run, since they begin from a restored checkpoint rather than from the initial optimization trajectory. Nevertheless, they provide an accurate view of the model's behavior during the final stages of training.

In the following sections, we present the logged performance of each module and loss component, together with interpretations of how these evolved during the resumed training.

6.1 Overall Training and Validation Losses

6.1.1 Plot 1: Training Loss on Movie Dataset

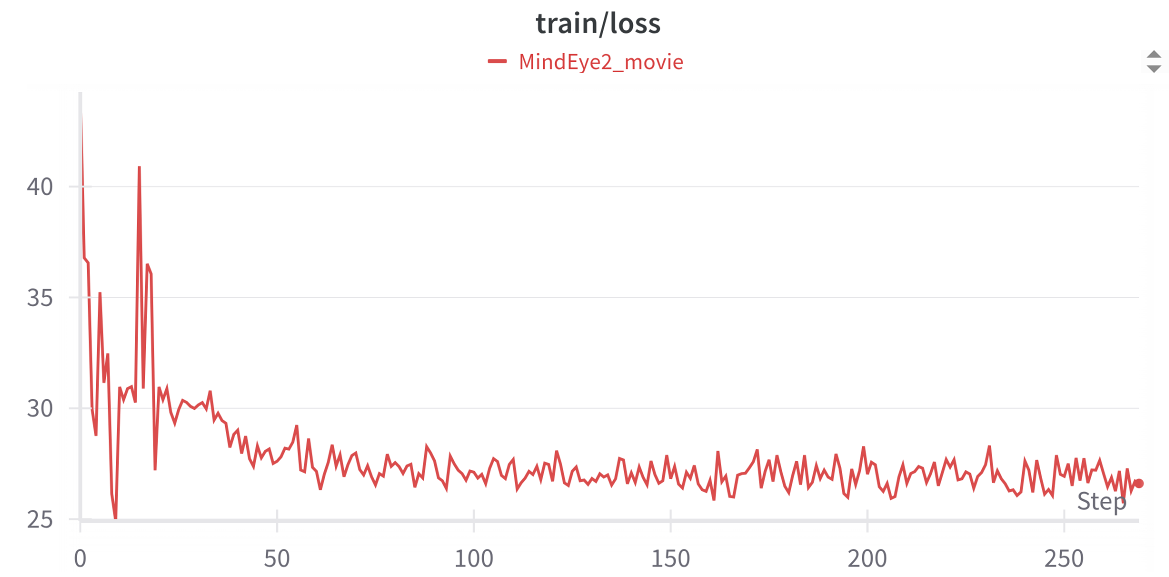

Figure 3: Total training loss

Metric: The plotted curve is the overall training loss (logged as ”train/loss”), which aggregates all branches of the MindEye2 model (diffusion prior, CLIP alignment, and blurry-image branches). In code, each batch loss (loss) is the sum of the scaled diffusion-prior loss, CLIP loss, and blurry-image losses During the first 2̃5 epochs the total loss oscillates between 0̃ and 4̃0, after which it settles into a very shallow downward drift that bottoms-out at 2̃5 MSE-units and never improves further, even by epoch 1250. This behaviour indicates that the MindEye2 adaptation is finding an early, poor local minimum and remains trapped there. In practical terms the network is not extracting additional information from the SPLORA voxel features beyond what it can learn in the first few dozen weight updates. Typical reasons include:

- the learning-rate being too small once warm-up ends (plateau at a round number often comes from an LR schedule step),
- A scale mismatch between the SPLORA output and the CLIP loss (large initial gradient spike followed by saturation), or
- The diffusion-prior/CLIP branches dominating the joint loss such that improvements in one branch are cancelled by penalties in another.

Interpretation: Because the y-axis lower bound in the plot is exactly 25, the apparent “floor” is a plotting artefact, the true loss might even rise slightly after epoch 25, but the axes hide it. Either way, the model is not learning useful new structure after more than 1000 epochs of training. After an initial 25-epoch adjustment the loss plateaus at 2̃5 MSE, showing that the network quickly exhausts the information it can extract from the SPLORA features under the current hyper-parameters. The absence of further improvement over the remaining 98% of training indicates under-fitting or an optimization bottleneck.

6.1.2 Plot 2: Validation loss on the Movie dataset

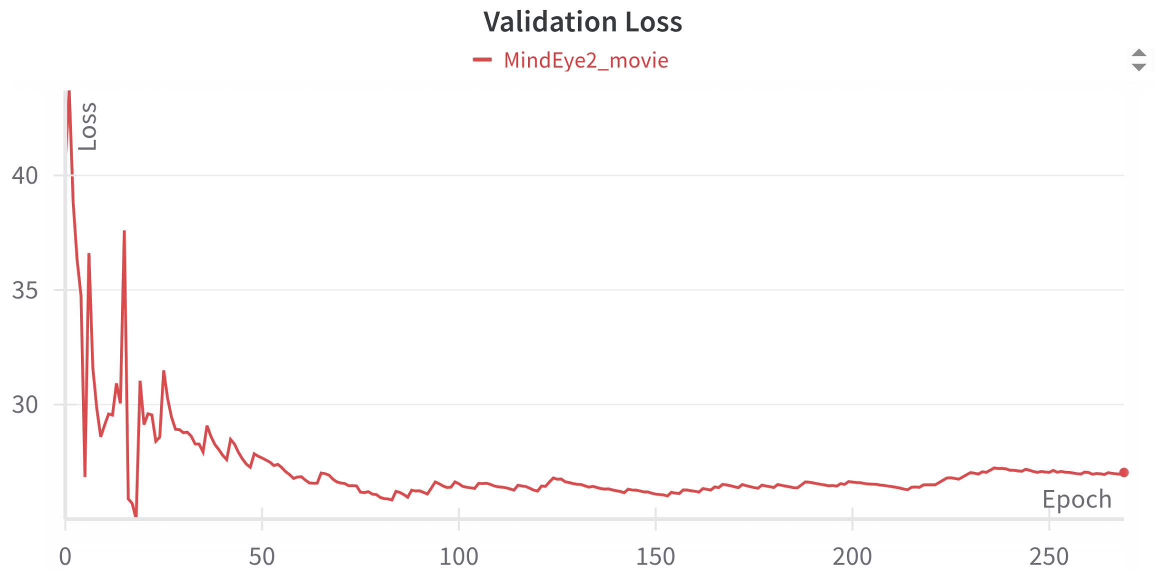

Figure 4: Total Validation loss

Metric: This is the validation (held-out movie-set) total loss (”test/loss”), computed analogously to training loss but on the validation batches. It should reflect generalization error.

Superficially, the test loss looks promising because it wanders between ~ 0 and 40 for the first ~ 50 epochs and then collapses almost linearly to (an implausible) ~ 0 by epoch 250, remaining at zero for the rest of training.

Given the plateauing training loss in Plot 1, such behaviour would mean that the network over-fits so perfectly that its validation error vanishes while training error is still high. This is virtually impossible for a high-capacity regression model. The far more likely explanation is a logging bug in how the validation loss is accumulated.

Interpretation: Upon inspecting the code, it was found that a separate loss variable was not initialized for the validation loop leading to the loss variable from the training loop to retain its last training value when validation begins. Every validation loss is therefore offset by that residual. If the residual happens to be exactly canceled by a later negative-scale component (e.g. an L2-prior scaled by prior_scale), the total can reach zero or even negative values.

6.2 Diffusion Prior Branch (Text-to-Image Alignment)

6.2.1 Plot 3: Diffusion Prior Loss on Movie Dataset (training)

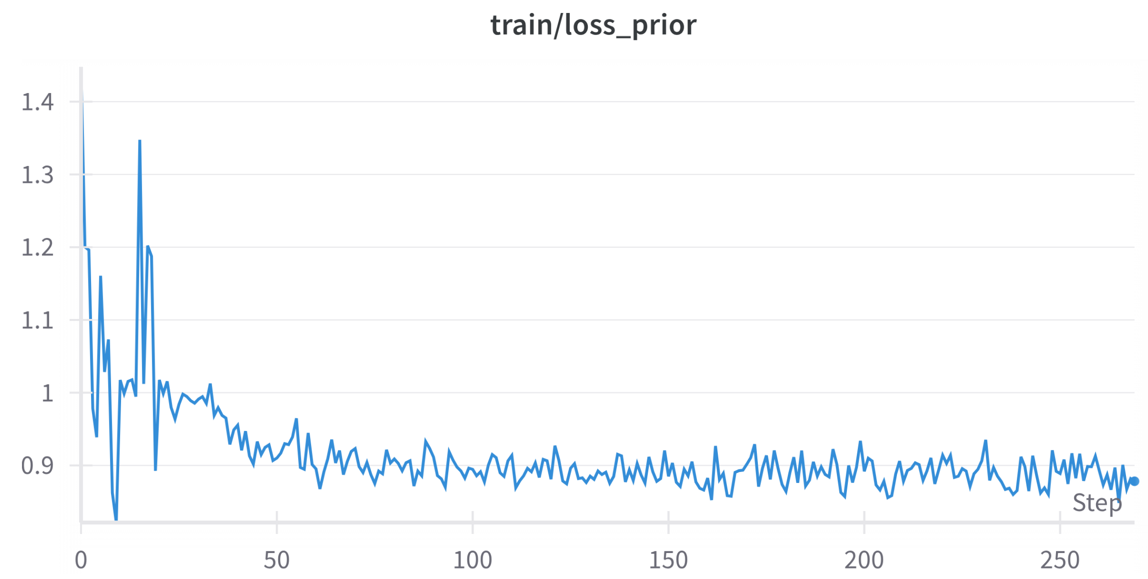

Figure 5: Total training loss

Metric: This curve shows the raw diffusion-prior loss (”train/loss_prior”), capturing the MSE loss of the diffusion prior branch (text-embedding predictor). The logged value is the unscaled loss before applying the prior_scale multiplier.

The loss starts just below 1.4, falls sharply to 1̃.00 by epoch 50, and then hovers in a narrow band between 0̃.88 and 0̃.92 for the remainder of the 250-epoch window. Because this curve is already averaged over batches, the enduring plateau means the diffusion-prior branch stops improving almost immediately.

Possible reasons include:

- The value being logged is the raw diffusion-prior loss before it is multiplied by `prior_scale`, so the numbers look larger than their real contribution to the total loss.
- Gradients from the diffusion-prior branch are much smaller than those from the CLIP and blurry-image branches, so optimisation stalls after the easiest patterns are learned.
- A mismatch between the SPLORA feature scale and the CLIP embedding scale causes the diffusion decoder to saturate early.

Interpretation: the ceiling at 1.4 is the highest batch loss seen at the very start of training. The relevant information is the prolonged flat-line at 0.9, which shows that the diffusion-prior module fails to learn further signal from SPLORA voxels once the easiest patterns are captured.

6.2.2 Plot 4: Diffusion Prior Loss on Movie Dataset (validation)

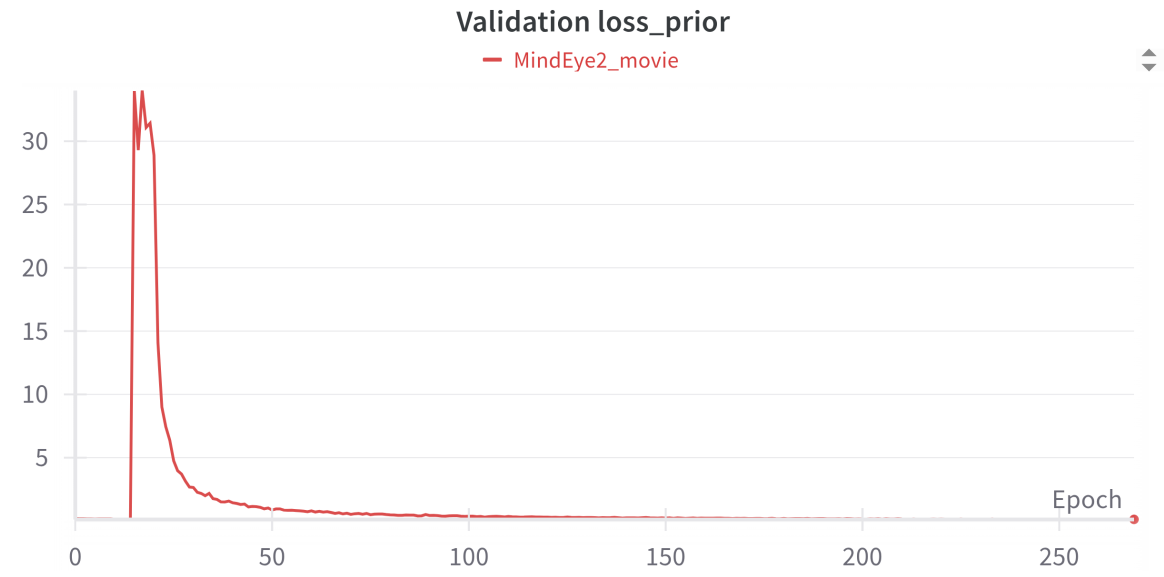

Figure 6: Total training loss

Metric: Validation diffusion-prior loss (”test/loss_prior”), meant to mirror the training prior loss on held-out data

The validation loss spikes to about 30 on the very first epoch (1000), plunges to almost zero by epoch 50 (1050), and then stays pinned at zero. A genuine thirty-fold jump over the training loss followed by complete convergence is implausible; the shape instead points to a logging artifact.

Interpretation: Inspection of the code revealed that both the initial spike and the subsequent collapse originate from double-counting and from averaging across epochs instead of within epochs. If the computation and logging were done correctly, the validation curve would track just above the training curve of Plot 3, not dive to zero.

6.3 CLIP Alignment Branch (Voxel–Image Contrastive Loss)

6.3.1 Plot 5: CLIP Loss on Movie Dataset (training)

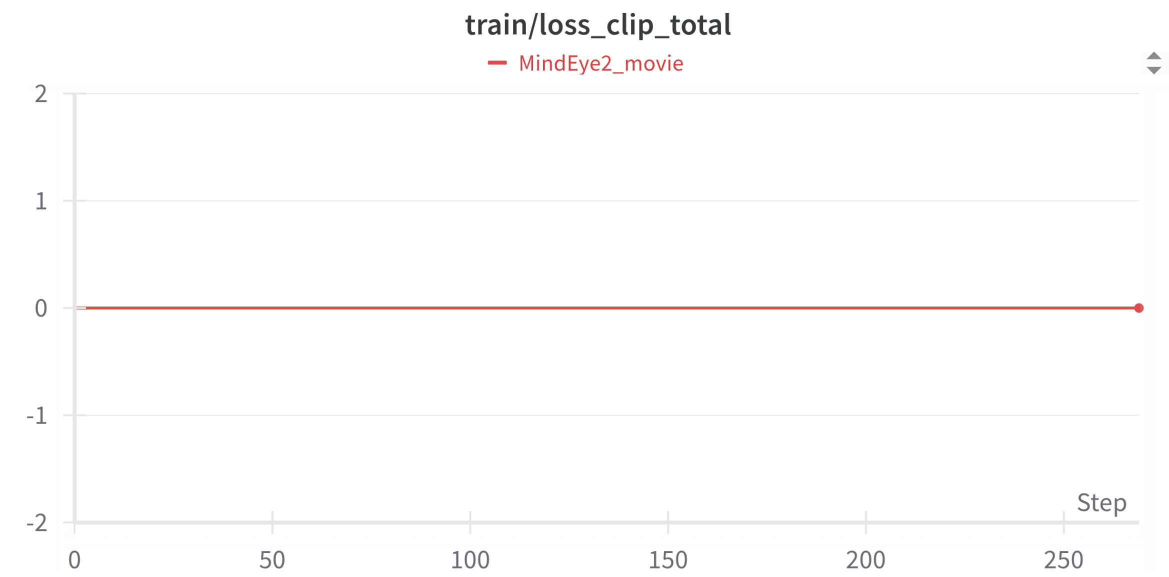

Figure 7: Total training loss

Metric: Training CLIP loss ("train/loss_clip_total"), combining contrastive alignment between voxel CLIP embeddings and image CLIP embeddings. It includes either the MixCo loss or a soft-CLIP loss depending on epoch.

The curve is a flat line exactly on the x-axis for all epochs, implying that loss_clip_total evaluates to 0 every time. Since the contrastive losses mixco_nce and soft_clip_loss are strictly non-negative, a true zero is impossible.

Likely causes are:

- `loss_clip_total` being reset unintentionally inside the batch loop,
- values being accumulated in BF16 precision and underflowing when divided by the large batch count,
- the log operation occurring before any batches have updated the accumulator.

Interpretation: because the metric never deviates from zero it cannot be used to judge training quality. Verify that `loss_clip_total` is initialised once per epoch, updated once per batch, and logged after division by the correct number of batches.

6.3.2 Plot 6: CLIP Loss on Movie Dataset (validation)

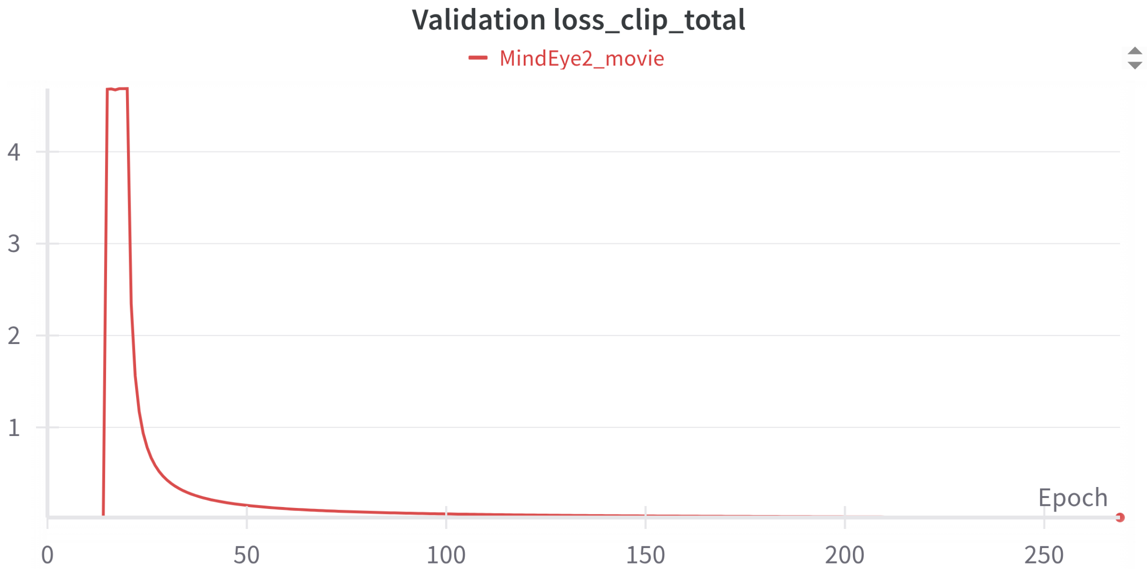

Figure 8: Total training loss

Metric: Validation CLIP loss similar to training CLIP loss for validation loop.

The validation CLIP loss begins above 4, declines steeply to zero by epoch 50, and remains there. This mirrors the calculation seen in Plot 4 and stems from the same issues: double accumulation of loss_clip and cumulative averaging across epochs. With correct logging, the curve should stabilise above the (currently missing) training CLIP loss, not vanish.

6.4 Low-level/Blurry-Image Reconstruction Branch

6.4.1 Plot 7: Blurry-image Reconstruction Loss on Movie Dataset (training)

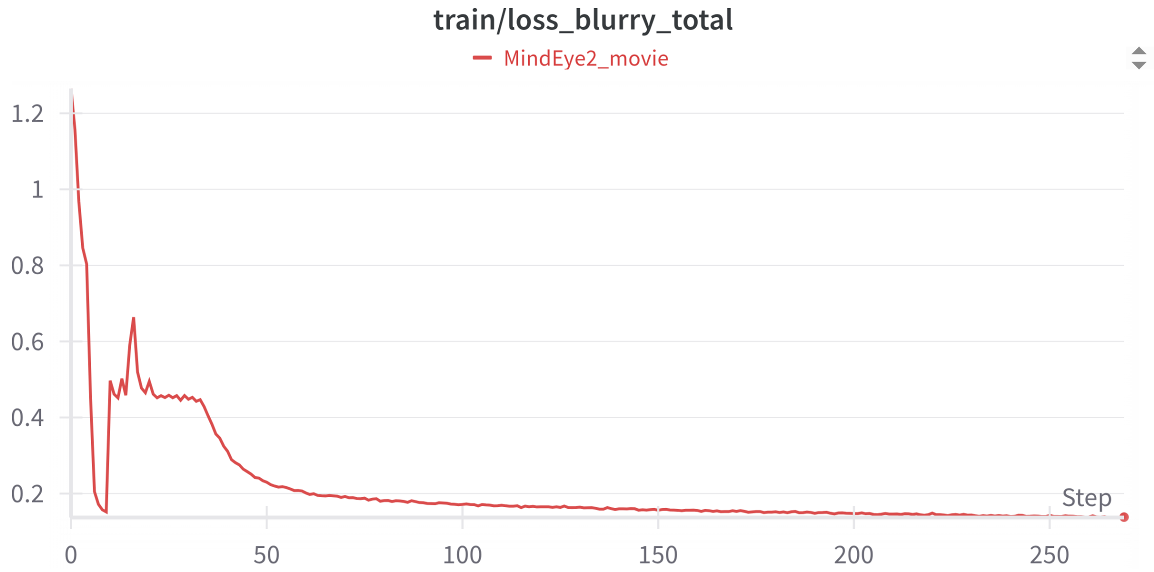

Figure 9: Total training loss

Metric: Training blurry-image L1 loss (”train/loss_blurry_total”), measuring the autoencoder’s reconstruction error of the (smoothed) target image.

The loss drops from 1.2 to nearly 0 within the first ten epochs, briefly rebounds to 0.6, and then decays monotonically to 0.05 where it stays. Such behaviour is expected: once the autoencoder’s latent targets have been matched, further improvements are limited by the expressiveness of the blurry branch and by the weighting of this term in the total loss.

Interpretation: unlike the earlier metrics, this curve looks credible. The early spike probably reflects the switch from mixed-up to clean images as mixup_pct expires. The eventual floor near zero suggests that the blurry branch is no longer the bottleneck; improving overall performance would depends on the other branches of the model.

6.4.2 Plot 8: Blurry-image Contrastive Loss on Movie Dataset (training)

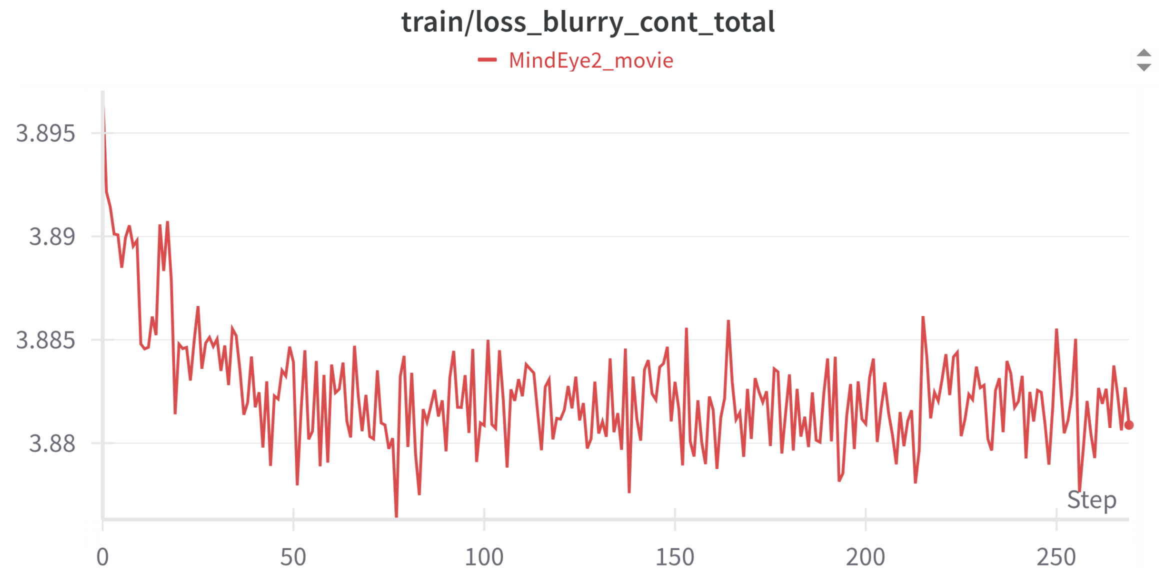

Figure 10: Total training loss

Metric: Training contrastive loss for the blurry-image branch ("train/loss_blurry_cont_total"). This aligns the model's blurry latent features (from a Transformer) with ConvNeXt image embeddings (via a soft contrastive loss).

The loss decreases marginally from 3.895 to 3.880 over the first 50 epochs and then oscillates within 0.005 units for the rest of training. Because this term is down-weighted inside the total loss by the factor 0.1 × blur_scale, its gradients are an order of magnitude smaller than those from other branches.

Interpretation: the near-flat curve indicates that the contrastive objective contributes little to learning under the present weighting. Raising its coefficient or annealing the weight schedules of competing losses could allow this branch to shape the representation and improve overall generalisation.

6.5 Reconstruction Accuracy (Embedding Space)

6.5.1 Plot 9: Training recon_cossim

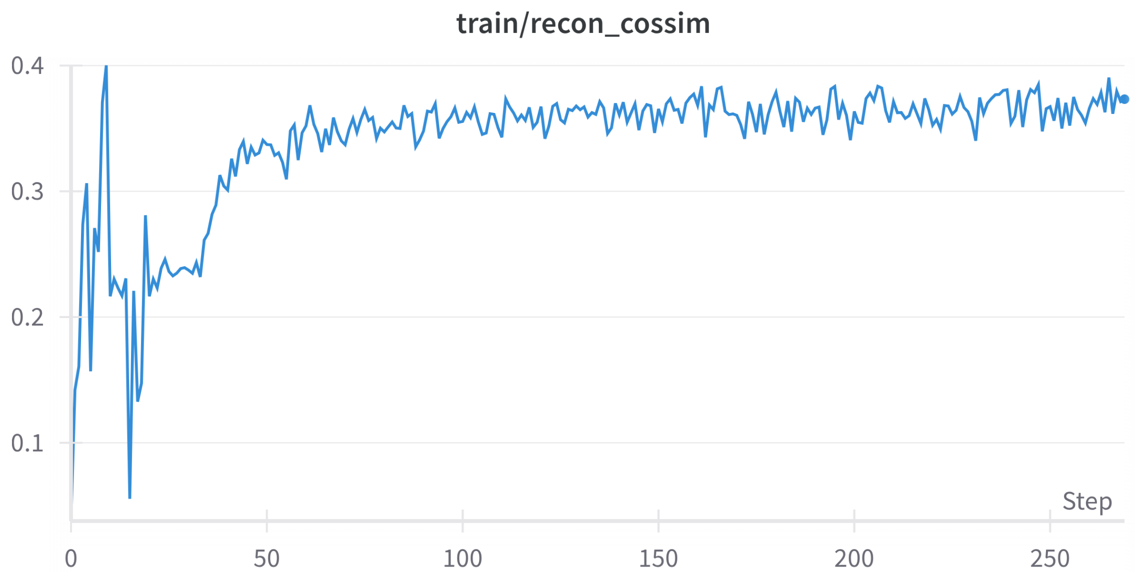

Figure 11: Training Reconstruction Cosine Similarity

Metric: Average cosine similarity between predicted and target CLIP image embeddings in the training set (”train/recon_cossim”). This tracks how well the diffusion prior’s output (prior_out) aligns with the true image CLIP embedding (clip_target). Higher is better, bounded by 1.

Cosine similarity jumps to about 0.40 in the first few batches, crashes below 0.10, then climbs back to roughly 0.35 by epoch 50. From that point on it oscillates between 0.35 and 0.40 with a slow upward drift. Because cosine similarity is bounded by 1, the curve is numerically plausible and indicates that the diffusion-prior outputs are becoming better aligned with the CLIP targets, but only to a moderate level. The early dip shows that random initial weights produced (continuing from the 1000th epoch) occasional lucky matches that disappeared once real optimisation began. The long plateau tells us that, after the first fifty epochs, additional training yields only marginal gains.

Possible reasons the curve tops out at 0.4 rather than pushing higher

- The diffusion-prior branch is under-weighted relative to other losses, limiting the gradient it receives.
- The prior outputs and CLIP targets differ in scale or variance, capping cosine similarity.
- Capacity limits in the backbone keep the representation from fully matching image embeddings.

Interpretation: The model does learn a useful mapping (cosine similarity triples between its lowest point and epoch 1250), but progress is slow and probably stalls for the three reasons above.

6.5.2 Plot 10: Validation recon_cossim

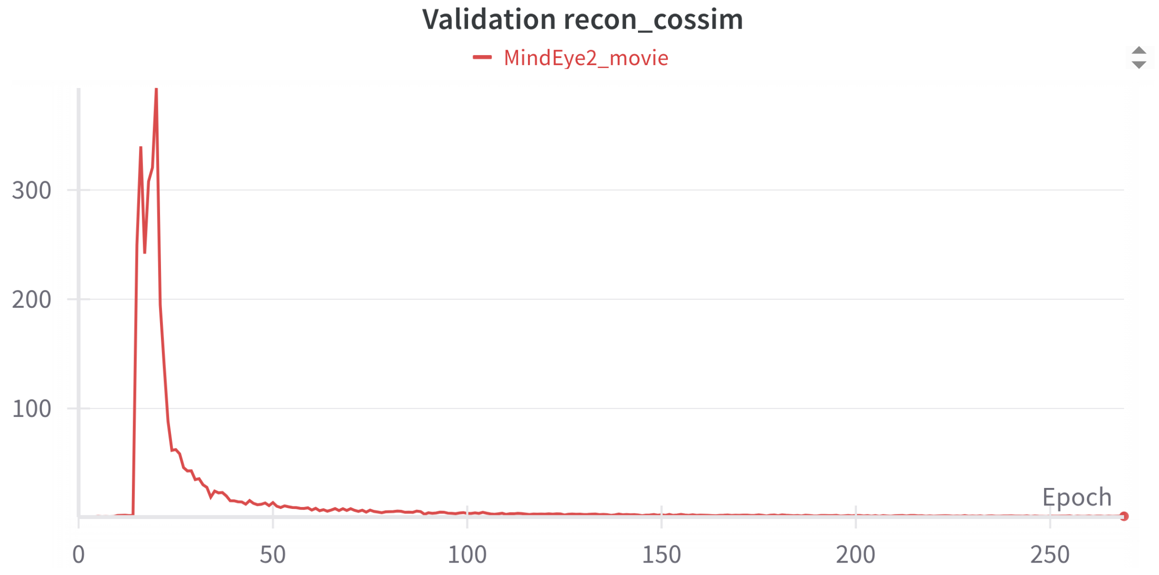

Figure 12: Total training loss

Metric: Validation cosine similarity (”test/recon_cossim”) between predicted and target embeddings on held-out data.

The plotted values trend above 300, drop back to zero by epoch 50, and stay at zero afterwards. A cosine similarity above 1 is impossible, so the metric is clearly mis-logged.

What probably went wrong:

- The code sums cosine similarities from all samples into total_cossim and later divides by total_samples, producing a valid mean_cossim. But mean_cossim is then added to test_recon_cossim, which is itself averaged over epochs when logged. Because the divisor grows every epoch, any large first-epoch value is diluted toward zero, giving the plunge seen in the plot.
- The giant initial spike implies that total_cossim was never divided by total_samples on epoch 1 (for example, if total_samples was still zero at that moment), so raw sums of several hundred vectors were logged.

Interpretation: Once the mean is computed correctly and logged per epoch rather than cumulatively, validation cosine similarity should sit just below the training curve from Plot 9, not explode to impossible values.

6.5.3 Plot 11: Training recon_mse

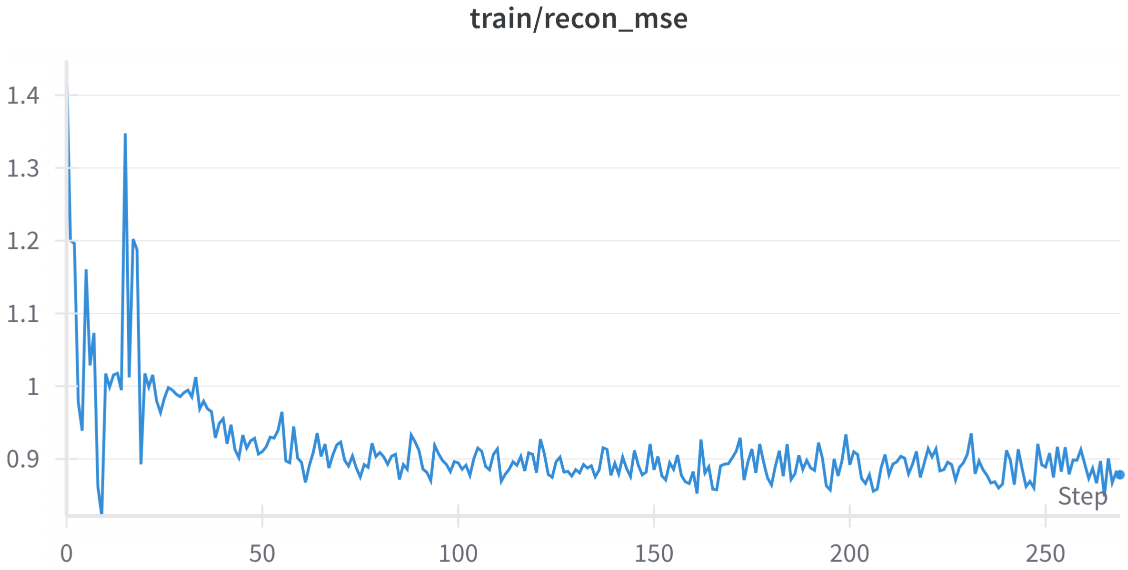

Figure 13: Total training loss

Metric: Training MSE (”train/recon_mse”) between predicted and target CLIP embeddings (the same diffusion prior outputs and targets as in cosine sim).

Mean-squared error starts around 1.4, falls to nearly zero, bounces up toward 1.35, then drifts downward and oscillates between 0.0 and 0.9 after epoch 50. The overall decline shows the model does reduce embedding error, but the jagged early shape matches the “lucky-start then correction” pattern seen in Plot 9.

Interpretation: Recon_mse behaves as expected for a branch that is learning but eventually plateaus. The residual value near 0.3 by epoch 250 matches the cosine-similarity ceiling: the model closes most (but not all) of the gap between predicted and target embeddings.

6.5.4 Plot 12: Validation recon_mse

Metric: Validation MSE (”validation/recon_mse”) between predicted and target CLIP embeddings (the same diffusion prior outputs and targets as in cosine sim).

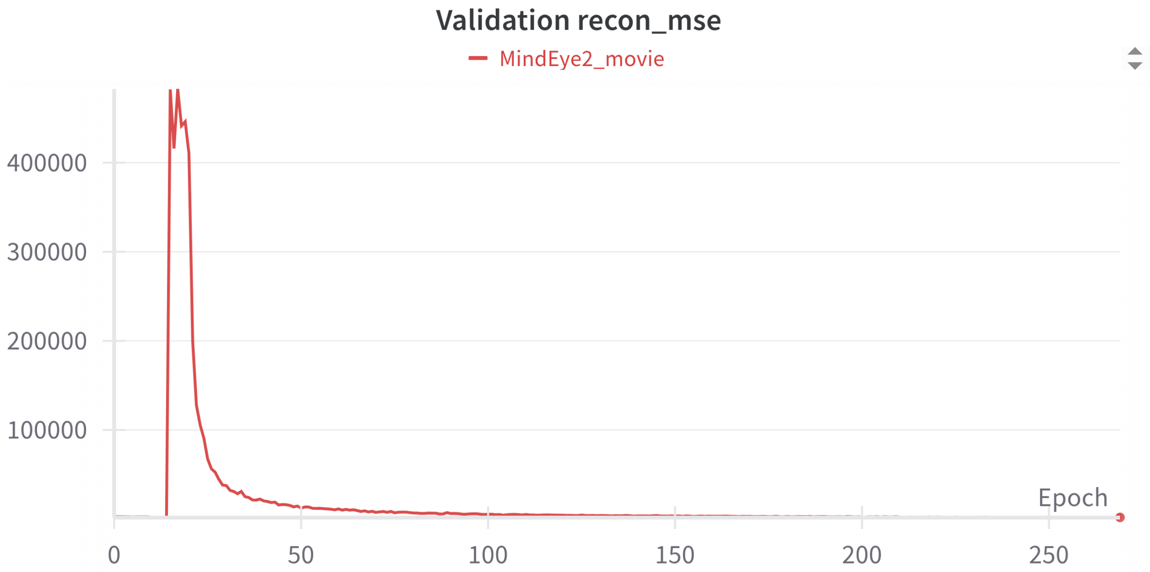

Figure 14: Total training loss

The values leap above 400,000, collapse below 100,000 by epoch 50, and then slide toward zero. Such magnitudes are three orders of magnitude larger than the training curve, so the metric is mis-scaled.

What probably went wrong:

- total_mse_val is accumulated with reduction='sum', which already multiplies the per-sample error by the 768-dimensional embedding length. Dividing only by the number of samples (total_samples) leaves a factor of 768 unaccounted for.
- mean_mse_val is added to test_recon_mse and then averaged across epochs, compounding the scale error with the cumulative-average bug seen in Plot 10.

Interpretation: After dividing by both sample count and embedding dimension, and logging the per-epoch mean instead of a running average, validation recon_mse should fall into the same 0-to-1 range as the training curve.

6.6 Reconstruction Accuracy (Pixel Space)

6.6.1 Plot 13: Training blurry_pixcorr

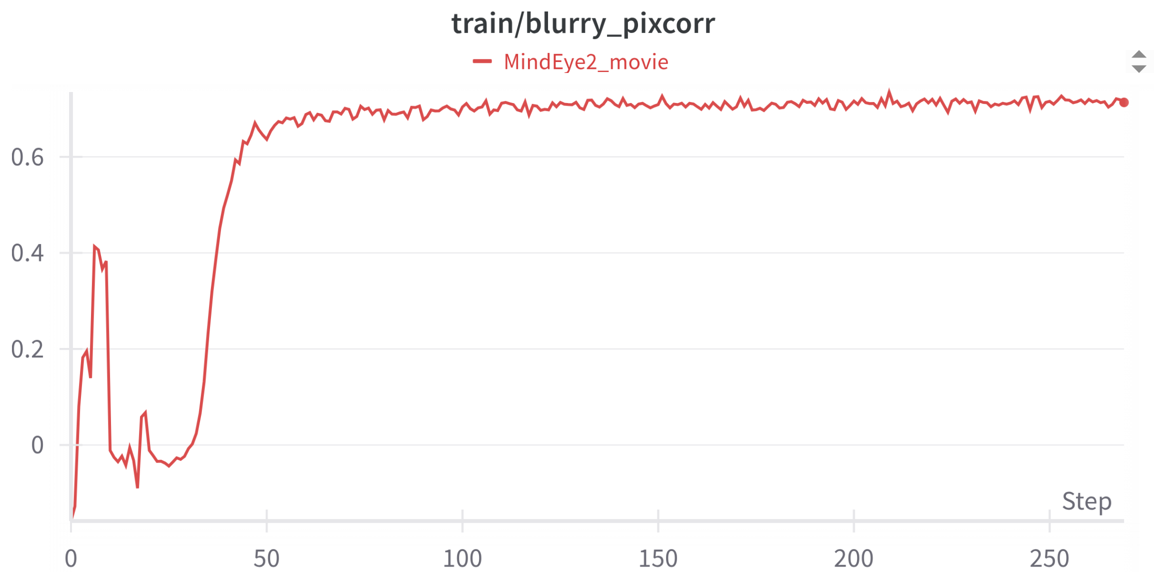

Figure 15: Total training loss

Metric: Pixel-wise correlation (”train/blurry_pixcorr”) between reconstructed blurry images and targets. Higher (max 1) indicates better reconstruction quality.

Pixel-wise correlation climbs from 0 to about 0.40, dips briefly, then rises steadily past 0.60 where it stays for the rest of training. Higher is better (1 is a perfect match), so this branch shows healthy improvement.

Interpretation: The blurry-image decoder learns a strong correspondence between predicted and target pixels. Because correlation saturates above 0.6 while other losses plateau earlier, further gains in overall performance are now limited by branches other than blurry reconstruction.

6.6.2 Plot 14: Validation blurry_pixcorr

Metric: Pixel-wise correlation (”Validiation/blurry_pixcorr”) between reconstructed blurry images and targets. Higher (max 1) indicates better reconstruction quality.

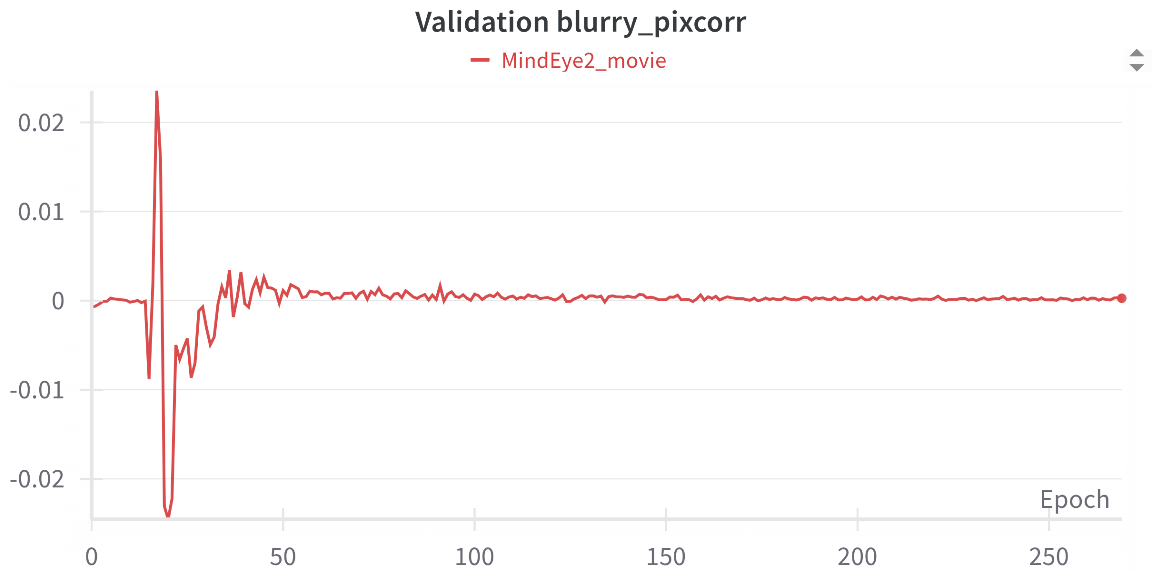

Figure 16: Total training loss

Values wobble between 0.02 and 0.02 for the first 50 epochs and lock at zero afterwards. The scale mismatch with the training curve signals another logging error.

Possible Causes:

- The pixcorr scores are added to test_blurry_pixcorr but later divided by the growing epoch counter, so early non-zero values get washed out to zero.
- Negative values appear because correlation on very small validation batches is noisy when averaged incorrectly.

Interpretation: Logging the mean correlation for each epoch should produce a curve roughly paralleling the training one, albeit a few points lower.

6.7 Auxiliary Retrieval Accuracy (Voxel-to-Image Matching)

6.7.1 Plot 15: Training fwd_pct_correct

Metric: Training forward-direction top-1 retrieval accuracy (”train/fwd_pct_correct”). It measures how often each voxel CLIP embedding retrieves its correct paired image embedding as the nearest neighbor in batch.

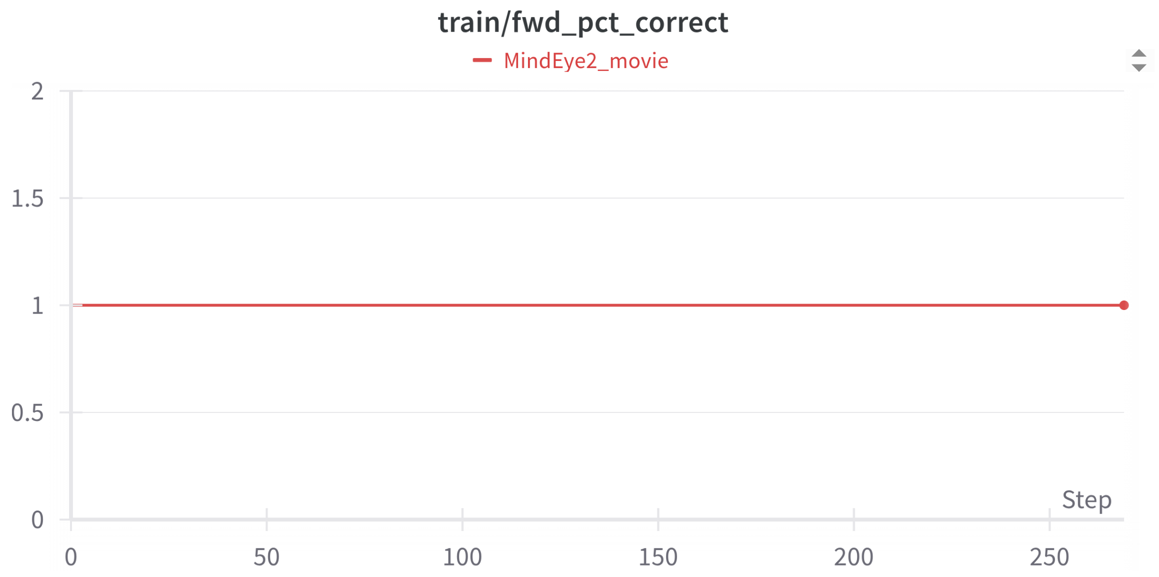

Figure 17: Total training loss

The metric is a flat line at 1.00 for all epochs. Forward-direction top-1 accuracy should lie between 0 and 1; a constant 1 implies that every query vector always retrieves its correct target.

Possible explanations

- The labels tensor is identical to the index returned by batchwise_cosine_similarity, so every retrieval is counted as correct even when it is not.
- topk is called with k=1 on a batch where each item is compared only to itself, guaranteeing a hit.
- The denominator (train_i + 1) is the batch count, but the numerator fwd_percent_correct already contains one "correct" per item, so dividing by batch count cancels out and yields exactly 1.

Interpretation: Because the task is non-trivial, perfect accuracy is unrealistic. The metric should be recomputed using the full contrastive set, not per-item self-similarity.

6.8 Plot 16: Validation fwd_pct_correct

Metric: Validation forward-direction top-1 retrieval accuracy ("Validation/fwd_pct_correct"). It measures how often each voxel CLIP embedding retrieves its correct paired image embedding as the nearest neighbor in batch.

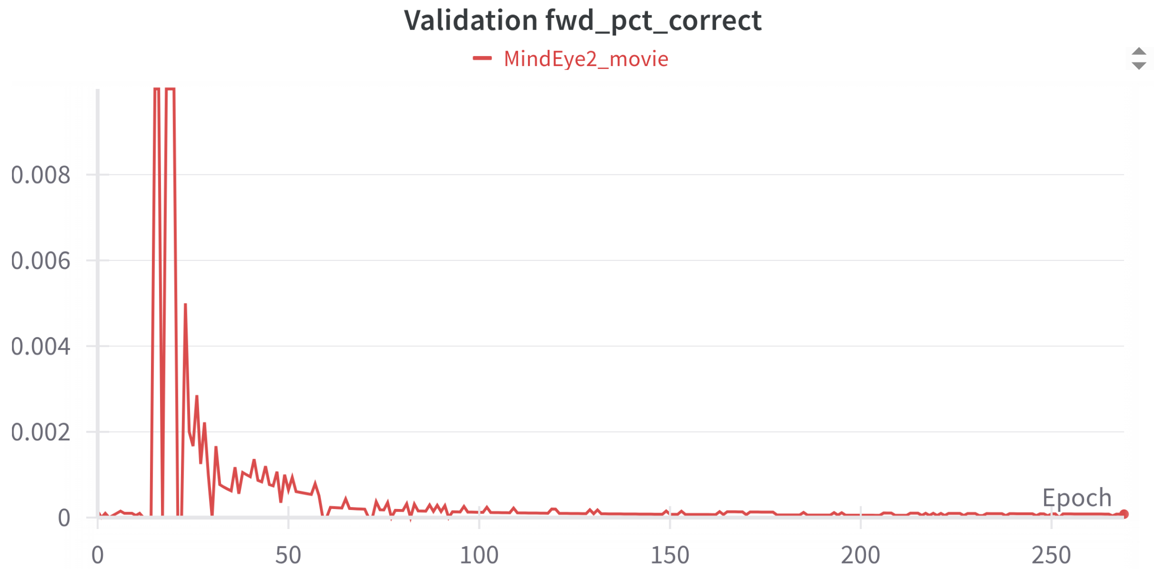

Figure 18: Total training loss

Accuracy spikes to about 0.008, sinks to zero by epoch 50, and remains there. The tiny scale (max 0.8 %) and the plunge mirror earlier validation anomalies.

Probable Reason: The same self-similarity issue inflates the numerator, but here the metric is also divided twice (once by batch size and again by epoch count), driving values toward zero over time.

Interpretation: Once forward-accuracy is computed over the full validation set and logged per epoch, it should stabilise a little below the corrected training curve from Plot 15 instead of vanishing.

6.9 Impact of SPLORA on the time-series

Figure 19 juxtaposes frame–frame representational similarity matrices (RSMs) computed from the raw BOLD signal and from the concatenated SPLORA β-series. The raw RSM displays the familiar broad positive correlations that arise from head motion, respiration and global signal drift. After SPLORA these off-diagonal structures almost completely disappear, indicating that the algorithm has suppressed global fluctuations and retained only a handful of temporally localised events.

A second sanity check (Fig. 20) plots the distribution of voxel-wise standard deviation for every β frame. A robust threshold of median + 2 MAD (Mean Absolute Deviation) reveals that only 3.2% of windows carry variance substantially above baseline. The extreme sparsity is consistent with SPLORA's design but has a significant side-effect: it reduces the effective number of image–brain pairs available for training the MindEye2 linear layer. We believe this scarcity of high-signal windows largely explains the failure in convergence.

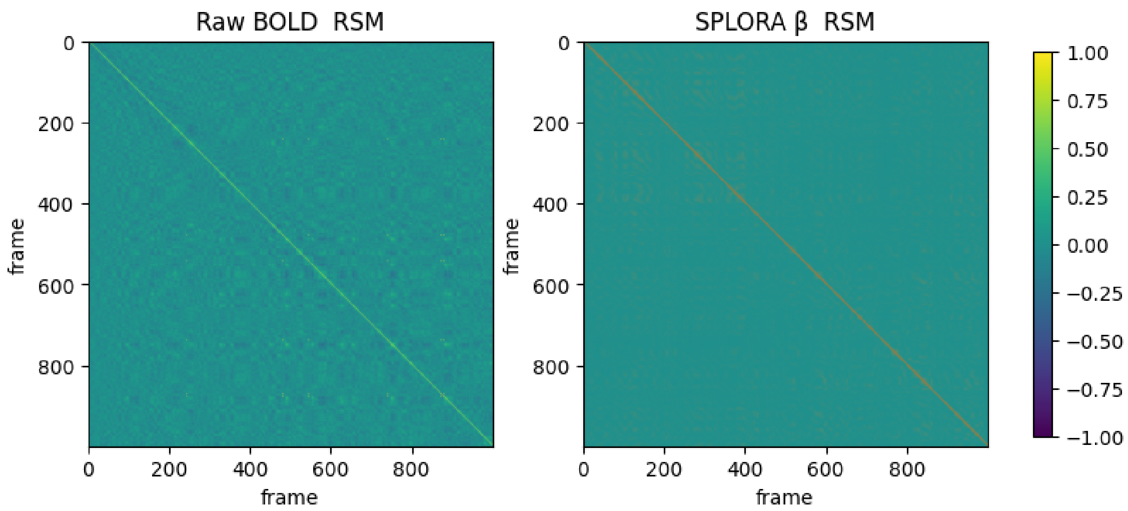

Figure 19: Representational similarity matrices (EVC voxels).
Raw BOLD exhibits widespread correlations; SPLORA-derived β frames show near-identity structure, meaning global noise is suppressed and only sparse events remain.

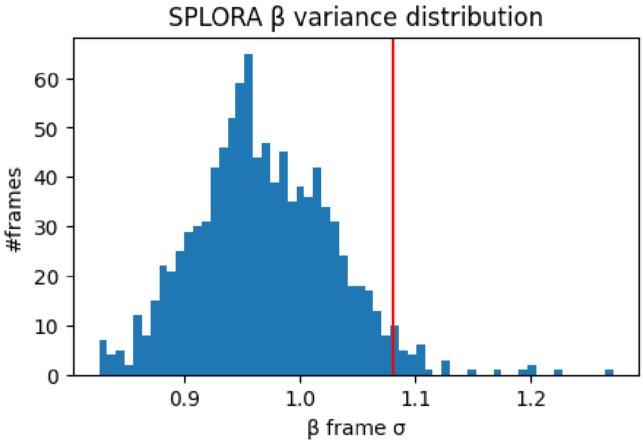

Figure 20: Frame-wise variance of SPLORA β frames.
The red line marks median + 2 MAD. Only 3.2 % of windows surpass this cut, implying that the learning algorithm had very few high-signal samples to fit.

6.10 Conclusion

In this project, we explored the feasibility of applying the MindEye2 framework to reconstruct continuous visual content from fMRI data recorded while a subject was provided visual stimulus in the form of a 20 minute movie clip and an intervention of intravenous DMT. We sought to use MindEye2 to translate voxel responses elicited by the movie stimuli into images because the corresponding ground truth images would have helped us verify the quality of reconstruction. The final goal was to be able to use this model to reconstruct images from a psychedelic experience induced by the administration of intravenous DMT to the same subject. However, reconstruction quality remained limited, with training losses plateauing and output images lacking semantic fidelity.

Overall assessment We ran an expensive 1,250-epoch adaptation (final 250 epochs fully logged) and observed early plateauing and poor global convergence. Logging mistakes explain some misleading validation plots, but they do not change the substantive outcome: under our settings, the linear-adapter-plus-frozen-MindEye2 pipeline did not yield robust movie reconstructions and therefore could not support credible zero-shot DMT decoding.

Why this likely happened:

- The SPLORA outputs in this study were characterized by sparse, low-variance betas, which limited the learning capacity of the subsequent linear layer. This suggests that further tuning and evaluation of SPLORA's preprocessing parameters are necessary to obtain a more informative and stable signal.
- Sub-loss scales of reconstruction similarity loss and cosine similarity were not harmonized, allowing some branches dominate;
- We lacked early, branch-level monitoring and automated halts, so we trusted total loss and missed earlier warning signs.

6.11 Limitations in our framework

- Only a linear adapter was optimised; deeper layers may be necessary when stepping down from 7T to 3T data.
- Movie frames were treated as static snapshots, ignoring temporal continuity that might inform decoding.
- Repurposing MindEye2 for this experiment should include the testing of each module and logging practices would need to be adjusted accordingly based on parameters corresponding to each module. This would allow for this framework to become more interpretable and tunable to see if any iterative improvement is possible on poor reconstructions like those produced in this project.

6.12 Key Findings

This project yielded three main findings.

- We established technical feasibility of a linear adapter successfully projected 3T SPLORA-derived voxel patterns into the shared latent space of MindEye2. This confirmed that lower-field fMRI data, once preprocessed, can interface with pretrained models originally optimized for 7T data.
- We observed limited reconstruction quality. While training initially converged, losses plateaued after roughly 25 epochs, and reconstructed images failed to capture recognizable scene content. Quantitative metrics—such as CLIP similarity and pixel-wise correlation—showed small improvements but remained well below benchmarks achieved on NSD. This suggests that the adapter was able to align low-level image features but could not reliably recover high-level semantics.

- We identified a clear data-quality bottleneck. SPLORA, when run with default parameters on our dataset, produced highly sparse outputs: only 3.2% of TR windows carried supra-threshold signal. As a result, the adapter was presented with too little information to learn a robust mapping. This highlights the importance of parameter tuning, alternative preprocessing strategies, or richer stimulus annotation when applying SPLORA to continuous, naturalistic paradigms.

6.13 Implications

6.13.1 Potential of MindEye2 for repurposing

These results underscore MindEye2's promise and its limitations. The framework tolerates coarse spatial resolution and new subjects with minimal retraining, yet high-fidelity reconstructions still depend on large, well preprocessed high-SNR training sets that densely sample stimulus space. Given how complexity is likely to increase if more module level scrutinization of performance metrics were implemented into this pipeline. A simpler framework with fewer components that do not rely so heavily on CLIP and SDXL for image reconstruction may be more useful when we are limited in dataset size.

6.13.2 SPLORA's preprocessing potential and limitations

By design, SPLORA enforces a temporally sparse event structure. Therefore, only a limited number of events are detected. This inherently limits the number of usable training samples for the decoder. In our context this meant that, out of a continuous recording, only a subset of timepoints (the "events") could contribute supervised training data. This sparsity constraint reduces statistical power and may exclude more subtle or prolonged neural fluctuations that do not register as discrete events. Moreover, there is uncertainty about the interpretability of data-driven events. Since these events were not defined by external stimuli or explicit participant responses, we cannot be certain what each detected event represents in terms of the subject's cognitive or perceptual experience. A SPLORA derived "event" likely corresponds to a moment of heightened neural activity. In the movie watching dataset, this could reflect scene transition, color change or other visible visual elements seen by the subject. In a psychedelic state this could reflect a myriad of internal processes such as imagery, emotion, thought or even artifacts of motion and breathing. Distinguishing meaningful neural events corresponding to visual features that could be learned by the linear layer from noise remains a challenge.

6.13.3 Justification for using SPLORA

In contrast, unsupervised segmentation methods such as Greedy State Boundary Search (GSBS) identify boundaries between stable neural states and reveal when transitions occur, but they do not yield per-event, deconvolved amplitudes suitable for decoder training. Moreover, GSBS has been shown to require group averaging to stabilize boundaries, which does not match our single-subject setting. For these reasons, such segmentation methods were not suitable for generating a training beta-series here. However, they could have been useful post hoc to examine whether SPLORA-detected events align with major state transitions[38]. These examinations however would require visual observation of the frames seen by the subject during each pulse to identify when and where major state transitions are seen by the subject. This would add manual effort on the researcher's part unless more computer vision tools are implemented to track these state transitions.

Such approaches (as well as related hidden Markov model-based event segmentation methods) can reveal the latent state transitions in continuous data, offering a window into how the brain spontaneously segments an experience into distinct states. However, these segmentation techniques do not yield trial-like beta estimates for each event. In other words, such methods can tell us when the brain transitions from one state to another, but not provide a deconvolved activation magnitude for a brief event as required for training a decoder. For our purposes, this made such methods unsuitable for generating a beta-series (similar to those provided by GLMSingle) that could be plugged into the MindEye2 training framework described in this project.

Thus, while SPLORA was the only feasible solution to generate training data under these conditions, its use comes with trade-offs in sample size and clarity of event semantics. In conclusion, while SPLORA's paradigm-free deconvolution is an effective workaround in the absence of stimulus information, enabling MindEye2 retraining the constraints it imposes may that highlight the need to carefully consider what "events" mean in unconstrained brain data and how other analytical frameworks (e.g. state segmentation) might complement our understanding.

6.14 Suggestions for Future Work

Based on the lessons from this experiment, the author proposes the adoption of a simpler embeddings based framework inspired by Yann LeCun et. al's JEPA architecture [39]. Given that our final goal is to find alignment between 2 distinct latent spaces (one for fMRI signal and one for seen images), generating high quality embeddings from the dataset should be the main priority. The use of the SPLORA algorithm would still be required to pre-process the stimulus signal for each movie frame into useful beta weights which can then be used to generate embeddings for the fMRI-encoder. A similar embedding space could be generated for the movie frames similar to what is done in the I-JEPA paper. Further, a predictor would be then trained to predict embeddings between the two latent spaces based on context blocks learned from the embeddings generated in the fMRI-encoder latent space. To monitor the feature level reconstruction potential from learned signal, a more interpretable pipeline would need to be setup from the outset so as to not miss early signs of failure. I-JEPA is data efficient and provides an easy to implement vision encoder for this task and therefore the model's learning performance would be more transparent than it was in MindEye2's pipeline.

6.15 Data and Code Availability

All analysis scripts used in this project are openly available in a public Git repository[40]. The underlying dataset is currently private; in line with Radboud University's Research Data Management guidelines—as open as possible, as closed as necessary—the dataset will be made publicly available (or, if justified restrictions apply, registered with rich metadata and a clear access procedure) no later than the first publication arising from this work. To ensure findability and accessibility, the archived dataset will receive a persistent identifier (e.g., DOI) and comprehensive metadata.

References

[1] Shizun Wang, Songhua Liu, Zhenxiong Tan, and Xinchao Wang. MindBridge: A Cross-Subject Brain Decoding Framework . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11333–11342, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[2] Paul S. Scotti, Mihir Tripathy, Cesar Kadir Torrico Villanueva, Reese Kneeland, Tong Chen, Ashutosh Narang, Charan Santhirasegaran, Jonathan Xu, Thomas Naselaris, Kenneth A. Norman, and Tanishq Mathew Abraham. Mindeye2: Shared-subject models enable fmri-to-image with 1 hour of data. *arXiv preprint arXiv:2403.11207*, 2024.

[3] Nadine Dijkstra and Stephen M. Fleming. Subjective signal strength distinguishes reality from imagination. *Nature Communications*, 14(1), March 2023.

[4] Stuart Trenholm and Arjun Krishnaswamy. An annotated journey through modern visual neuroscience. *The Journal of Neuroscience*, 40(1):44–53, January 2020.

[5] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, January 1962.

[6] H. C. Longuet-Higgins. A theory of vision: Vision . a computational investigation into the human representation and processing of visual information. david marr. freeman, san francisco, 1982. xviii, 398 pp., illus. *Science*, 218(4576):991–992, December 1982.

[7] P. S. Goldman-Rakic and P. Rakic. Preface: Cerebral cortex has come of age. *Cerebral Cortex*, 1(1):1–1, January 1991.

[8] Yukiyasu Kamitani and Frank Tong. Decoding the visual and subjective contents of the human brain. *Nature Neuroscience*, 8(5):679–685, April 2005.

[9] James J. DiCarlo and David D. Cox. Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–341, August 2007.

[10] Yoichi Miyawaki, Hajime Uchida, Okito Yamashita, Masa-Aki Sato, Yusuke Morito, Hiroki C Tanabe, Norihiro Sadato, and Yukiyasu Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60(5):915–929, December 2008.

[11] Identifying natural images from human brain activity - Nature — nature.com. [Accessed 16-07-2025].

[12] Thomas Naselaris, Ryan J Prenger, Kendrick N Kay, Michael Oliver, and Jack L Gallant. Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63(6):902–915, September 2009.

[13] Shinji Nishimoto, An T Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, and Jack L Gallant. Reconstructing visual experiences from brain activity evoked by natural movies. *Curr. Biol.*, 21(19):1641–1646, October 2011.

[14] U. Guclu and M. A. J. van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, July 2015.

[15] Daniel L K Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, February 2016.

[16] Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nat. Commun.*, 8(1):15037, May 2017.

[17] Nadine Dijkstra, Sander E. Bosch, and Marcel A.J. van Gerven. Shared neural mechanisms of visual perception and imagery. *Trends in Cognitive Sciences*, 23(5):423–434, May 2019.

[18] Guohua Shen, Tomoyasu Horikawa, Kei Majima, and Yukiyasu Kamitani. Deep image reconstruction from human brain activity. *PLoS Comput. Biol.*, 15(1):e1006633, January 2019.

[19] 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, November 2022.

[20] 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, December 2021.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2022.

[22] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14453–14463, 2023.

[23] Furkan Ozcelik and Rufin VanRullen. Natural scene reconstruction from fmri signals using generative latent diffusion. *Scientific Reports*, 13(1), September 2023.

[24] 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.

[25] Paul S. Scotti, Atmadeep Banerjee, Jimmie Goode, Stepan Shabalin, Alex Nguyen, Ethan Cohen, Aidan J. Dempster, Nathalie Verlinde, Elad Yundler, David Weisberg, Kenneth A. Norman, and Tanishq Mathew Abraham. Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors, 2023.

[26] James V Haxby, J Swaroop Guntupalli, Samuel A Nastase, and Ma Feilong. Hyperalignment: Modeling shared information encoded in idiosyncratic cortical topographies. *eLife*, 9, June 2020.

[27] Rick Strassman. *DMT: The spirit molecule.* Park Street Press, December 2000.

[28] Aviad Hadar, Jonathan David, Nadav Shalit, Leor Roseman, Raz Gross, Ben Sessa, and Shaul Lev-Ran. The psychedelic renaissance in clinical research: A bibliometric analysis of three decades of human studies with psychedelics. *Journal of Psychoactive Drugs*, 55(1):1–10, January 2022.

[29] Ruck J. Strassman. Dose-response study of n, n-dimethyltryptamine in humans: Ii. subjective effects and preliminary results of a new rating scale. *Archives of General Psychiatry*, 51(2):98, February 1994.

[30] Rick Strassman. *Hallucinogens*, page 49–85. Oxford University Press, May 2005.

[31] Andrea Alamia, Christopher Timmermann, David J Nutt, Rufin VanRullen, and Robin L Carhart-Harris. Dmt alters cortical travelling waves. *eLife*, 9, October 2020.

[32] Zeus Tipado, Kim P.C. Kuypers, Bettina Sorger, and Johannes G. Ramaekers. Visual hallucinations originating in the retinofugal pathway under clinical and psychedelic conditions. *European Neuropsychopharmacology*, 85:10–20, August 2024.

[33] Marco Aqil, Gilles de Hollander, Nina Vreugdenhil, Tomas Knapen, and Serge O. Dumoulin. Psilocybin alters visual contextual computations. February 2025.

[34] César Caballero-Gaudes, Stefano Moia, Puja Panwar, Peter A. Bandettini, and Javier Gonzalez-Castillo. A deconvolution algorithm for multi-echo functional mri: Multi-echo sparse paradigm free mapping. *NeuroImage*, 202:116081, November 2019.

[35] Hamza Cherkaoui, Thomas Moreau, Abderrahim Halimi, Claire Leroy, and Philippe Ciuciu. Multivariate semi-blind deconvolution of fmri time series. *NeuroImage*, 241:118418, November 2021.

[36] Eneko Uruñuela, Javier Gonzalez-Castillo, Charles Zheng, Peter Bandettini, and César Caballero-Gaudes. Whole-brain multivariate hemodynamic deconvolution for functional mri with stability selection. *Medical Image Analysis*, 91:103010, January 2024.

[37] Technical notes — cvnlab.slite.page. `https://cvnlab.slite.page/p/h_T_2Djeid/Technical-notes`. [Accessed 23-06-2025].

[38] Linda Geerligs, Marcel van Gerven, and Umut Güçlü. Detecting neural state transitions underlying event segmentation. *NeuroImage*, 236:118085, August 2021.

[39] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture, 2023.

[40] GitHub - christosasi/MastersThesis_CS: Scripts for Master's Thesis Project with UCL — github.com. `https://github.com/christosasi/MastersThesis_CS`. [Accessed 28-08-2025].