LEARNING VISUAL CORRECTIONS FROM EEG

Anonymous authors
Paper under double-blind review



Figure 1: **Neural signals, visual corrections.** Given a source image and EEG recorded while a participant views a target image, EEGRAFT learns visual corrections without text instructions or edit masks at inference.

ABSTRACT

EEG recorded during visual perception carries information about viewed objects and their attributes, offering a basis for neural guidance of image editing. The challenge is to translate this information into precise semantic and appearance changes while preserving unrelated source content. We introduce EEGRAFT, which learns to align neural and visual representations to guide image editing from EEG. The editor's source-conditioned visual states query the aligned EEG summary and unpooled features through separate attention pathways, adapting neural guidance throughout generation through time-gated residual updates. At inference, the editor requires only a source image and EEG recorded while viewing the target, without access to the target image, text instructions, or edit masks. To support systematic study of this task, we construct NEVE, a benchmark covering appearance and semantic editing with 29,275 source–target pairs and 2.71 million EEG trials, by generating source images while preserving the measured correspondence between the original viewed stimuli and their EEG recordings. Our method outperforms six baselines across all six aggregate metrics, reducing C-LPIPS by 22.7% and increasing the recovery–preservation score R-Edit from 0.386 to 0.585 relative to the strongest baseline for each metric. With the editor and source image fixed, correctly paired EEG produces more accurate edits than mismatched or disabled EEG conditioning. We hope this work advances the use of neural signals for visual content creation and interaction.

1 INTRODUCTION

Visual perception encodes the objects and attributes of the external world in neural activity (Horikawa & Kamitani, 2017). EEG recorded during image viewing carries information about the perceived content, which can be recovered for visual recognition and generation (Song et al., 2024; Bai et al., 2024). Image editing is a central task in computer vision that enables users to modify objects and their attributes while preserving the surrounding image, supporting visual creation and interaction (Brooks et al., 2023; Hertz et al., 2022). The visual information in EEG raises the possibility of guiding these modifications with neural responses to a viewed target. Such a capability

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

would extend image editing to neural guidance and provide a concrete test of whether information decoded from visual perception can support controlled changes to images. This requires the edits to match the target's semantics and appearance while preserving source content unrelated to the intended changes. The challenge is to translate the visual information in noisy EEG (Song et al., 2024) into precise changes to the semantics and appearance of the source image.

In this paper, we introduce EEGRAFT, which learns to align EEG with visual representations, translating neural activity during visual perception into controlled changes in image semantics and appearance. Our central idea is to let the editor's visual state, informed by the source image, determine how it reads target-related information from EEG. We provide a visually aligned EEG summary together with unpooled encoder features, retaining access to neural features beyond the compact summary. The visual states query these two representations through separate attention pathways. The EEG representations remain fixed during generation, while their readouts vary with the evolving visual states and guide the editor through time-gated residual updates. Alongside flow matching (Lipman et al., 2022) and target supervision, an auxiliary prediction-difference loss encourages the change between one-step endpoint estimates with and without EEG to match the source–target difference in latent space, with the source and noise held fixed. At inference, EEGRAFT requires only a source image and EEG, without the target image, text instructions, or edit masks.

To investigate how visual information encoded in neural activity can support controlled image editing, we construct NEVE (Neural Evidence for Visual Editing). Starting from image-EEG pairs (Gifford et al., 2022; Zhu et al., 2024; Xu et al., 2025), we generate modified source images while retaining the viewed stimuli as editing targets. This construction introduces the visual differences required for editing while preserving the measured correspondence between each target and its EEG recording. NEVE covers appearance changes and semantic changes, comprising 29,275 image pairs with 2.71 million EEG trials. We partition each editing subset by target concept and exclude test stimuli and their recordings from EEG alignment and training of the corresponding editor. Evaluation measures recovery of target content in regions requiring modification, together with preservation of source content elsewhere.

Under this benchmark, EEGRAFT outperforms six baselines across all six aggregate metrics, reducing C-LPIPS by 22.7% and C-MAE by 32.8% relative to the strongest baseline for each metric. PixCorr increases from 0.502 to 0.752, while the recovery–preservation score R-Edit improves from 0.386 to 0.585. Ablations show that combining the EEG summary and unpooled features improves all six metrics over either pathway alone. With the trained editor and source image fixed, correctly paired EEG produces more accurate edits than mismatched recordings or disabled EEG conditioning. Compared with mismatched EEG, correct pairing improves C-SSIM from 0.383 to 0.420. These findings support the contribution of target-related EEG information to controlled changes in image semantics and appearance. Our main contributions are:

- We introduce EEGRAFT, which learns visual corrections through source-conditioned EEG readout. Its dual readout adapts neural guidance to the evolving image state while retaining access to features before pooling.
- We construct NEVE, a benchmark of 29,275 source–target pairs for appearance and semantic editing, with held-out stimuli and EEG recordings.
- We compare with six baselines and provide fixed-source EEG controls to assess editing accuracy and the contribution of correctly paired recordings.

2 RELATED WORK

Visual Decoding and Neural Image Manipulation. Visual decoding recovers perceptual information from neural recordings at different representational levels. Early fMRI work reconstructed images through explicit encoding models and image priors (Kay et al., 2008; Miyawaki et al., 2008; Naselaris et al., 2009); later approaches predicted features in pretrained visual spaces and coupled them to image generators (Shen et al., 2019; Takagi & Nishimoto, 2023; Scotti et al., 2023). EEG decoding follows a similar progression, from direct generative mappings to retrieval and reconstruction through aligned visual embeddings (Kavasidis et al., 2017; Song et al., 2024; Bai et al., 2024; Li et al., 2024). Recent work explores general-purpose EEG encoders (Jiang et al., 2024), hierarchical visual alignment (Zheng et al., 2026), and autoregressive visual decoding (Dai et al., 2026). These works ask what images or visual representations can be recovered from recordings. Related

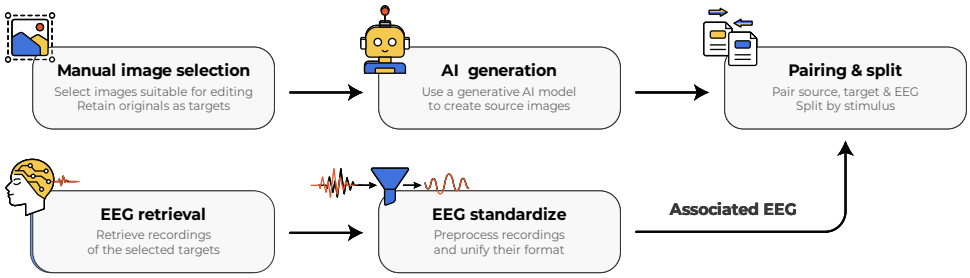

Figure 2: **NEVE construction pipeline.** A generative image model modifies manually selected stimulus images into sources. The original stimuli remain the targets and retain recorded EEG. After EEG standardization, the resulting source–EEG–target triplets are partitioned by target concept per editing subset, keeping all stimuli from a concept and their recordings in the same split.

work uses neural signals to modify existing images. EEG has been used to learn semantic editing directions (Davis et al., 2022; De La Torre-Ortiz & Ruotsalo, 2024), while fMRI-conditioned and multimodal methods combine neural signals with image or editing context (Yu et al., 2025b;a; Zhou et al., 2026). In our setting, EEG is recorded while participants view the desired target, whereas the source image is supplied separately. The recording therefore describes the target rather than an explicit edit operation. The remaining question is how target-related EEG can guide precise visual changes while preserving source content unrelated to editing.

Source-Conditioned Image Editing and Flow Models. Image editing methods differ in how they preserve the source while introducing the desired change. Earlier image-to-image and latent-space approaches learn mappings or semantic directions over the source representation (Isola et al., 2017; Zhu et al., 2017; Patashnik et al., 2021). Diffusion-based methods use source noising, inversion, attention control, or feature reuse during denoising to preserve unedited content (Meng et al., 2021; Mokady et al., 2023; Hertz et al., 2022; Tumanyan et al., 2023). Instruction-trained editors and lightweight conditioning modules make the editing interface learnable (Brooks et al., 2023; Zhang et al., 2023; Ye et al., 2023). These methods typically specify edits through language, masks, reference images, or other structured controls. We infer the desired change from target-viewing EEG relative to the source. We therefore retain the pretrained editor's native source conditioning and add neural readout after source and generation features interact. This design builds on flow matching and rectified flow, which learn vector fields for continuous-time image generation (Lipman et al., 2022; Liu et al., 2022; Esser et al., 2024). Recent flow-based systems extend this formulation to practical image editing and in-context generation (Kulikov et al., 2025; Labs et al., 2025; Wu et al., 2025). Incorporating EEG into pretrained editors while explicitly supervising its contribution to visual changes remains an open challenge.

3 Benchmark

We construct NEVE, a benchmark for EEG-guided image editing. Each example pairs a source image with a viewed target and its EEG recording. We generate source images through constrained modifications of selected stimuli (Sec. 3.1), keeping the originals as targets and preserving their EEG pairings. We evaluate on held-out concepts using change-region and source-relative measures (Sec. 3.2) to distinguish target recovery from similarity already present between source and target.

3.1 Task Formulation and Benchmark Construction

Problem formulation. Each example is a triplet $(I_s, \mathbf{X}, I_y)$, where I_s is the source image, I_y is the target image, and $\mathbf{X}$ is the EEG recorded while a participant views I_y. Given the source, EEG, and sampled generation noise $\boldsymbol{\epsilon}$, an editor f_θ produces:

$$\hat{I} = f_\theta(I_s, \mathbf{X}; \boldsymbol{\epsilon}). \tag{1}$$

The goal is to use target-related information in $\mathbf{X}$ to correct the source–target differences while preserving content that does not require editing. The target I_y provides supervision during training and a reference for evaluation. At inference, the editor receives only I_s and $\mathbf{X}$, without text instructions.

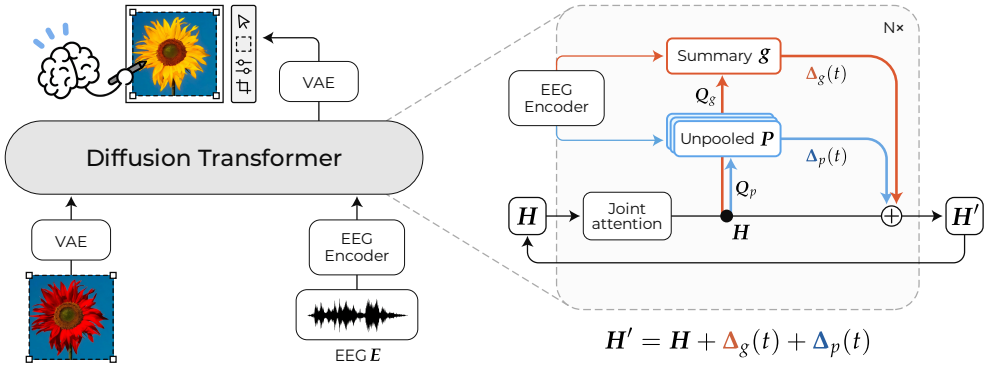

Figure 3: **EEGRAFT architecture.** Source-conditioned visual states query the EEG summary and unpooled features through separate attention pathways, whose time-gated readouts provide residual updates. Here, $\Delta_b(t)$ abbreviates $\Delta_b(\mathbf{H}, \mathbf{M}_b, t)$ for $b \in \{g, p\}$. The native feed-forward sublayer following $\mathbf{H}'$ is omitted for clarity.

Benchmark design. We obtain target–EEG pairs $(I_y, \mathbf{X})$ from THINGS-EEG2 (Gifford et al., 2022), EEG-ImageNet (Zhu et al., 2024), and Alljoined-1.6M (Xu et al., 2025). For each manually selected target I_y, a generative image model creates a modified source I_s, preserving the original target and its recorded EEG (Figure 2). The benchmark covers two editing settings: appearance edits change visual attributes while preserving object identity and spatial structure, whereas semantic edits change the object category and its characteristic form. In both settings, prompts require editing only the main foreground object or coherent group while preserving unrelated scene content. Attribute changes must be plausible for the object category, and replacement categories must fit the scene. When modifying a group, the same change is applied to all members. These constraints aim to reduce source-only shortcuts from implausible edits or unchanged group members that reveal the target. To combine recordings across resources, we standardize EEG to a common electrode layout and stimulus-locked temporal grid, retaining supplied preprocessing for processed releases. NEVE contains 29,275 source–target pairs and 2,714,765 EEG trials. The appearance subset has 14,486 training and 1,600 test pairs; the semantic subset has 11,989 training and 1,200 test pairs. Construction details appear in Appendix A.

3.2 Evaluation Protocol

Benchmark split. We partition the appearance and semantic subsets separately at the target-concept level. Within each subset, stimuli of the same concept and their EEG recordings remain in one split, ensuring concept-disjoint training and test partitions. Stimuli shared across recording resources receive consistent assignments within that subset. The shared EEG aligner excludes all test targets and their associated recordings from either subset across recording resources: a target held out for one task cannot enter alignment through the other. Each editor is trained only on its training partition. Appendix A provides details on dataset partitioning and shared EEG alignment.

Evaluation metrics. We emphasize source–target change regions because unchanged source content can dominate whole-image scores. These regions are derived from the RGB differences between I_s and I_y, independently of the generated image $\hat{I}$, and are shared by all methods. Within these regions, C-LPIPS (Zhang et al., 2018) (↓), C-MAE (↓), and C-SSIM (Wang et al., 2004) (↑) measure perceptual distance, absolute RGB error, and local structural similarity to the target, respectively. T-LPIPS (↓) and PixCorr (↑) measure whole-image perceptual distance and RGB pixel correlation; PixCorr does not measure absolute color accuracy. R-Edit (↑) additionally balances change-region recovery against increased target error in low-change regions, relative to source copying. Appendix B defines the metrics and their computation. We train separate editors for appearance and semantic editing. Overall scores weight the two subsets by their test-set sizes.

4 Method

We learn to edit a source image using EEG recorded while a participant views a target image. Our method, EEGRAFT, builds on a pretrained flow-based editor (Sec. 4.1), with an overview shown in

Figure 3. We first learn a visually aligned EEG summary while retaining the encoder's unpooled tokens (Sec. 4.2). With the EEG encoder frozen, the editor uses its current source-conditioned visual state to query both representations (Sec. 4.3). During training, we compare one-step endpoint estimates with and without EEG while keeping the source and noise fixed. An auxiliary loss matches this prediction difference to the source–target difference in latent space, supplementing flow matching and target supervision. Image-space losses further supervise the required edits using decoded one-step estimates, without unrolling the sampler (Sec. 4.4).

4.1 Source-Conditioned Flow Editing

Let E and D denote the frozen VAE encoder and decoder (Rombach et al., 2022), with source latent $\mathbf{z}_s = E(I_s)$ and target latent $\mathbf{z}_y = E(I_y)$. Conditional flow matching (Lipman et al., 2022; Esser et al., 2024) uses the training path $\mathbf{x}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{z}_y$, where $t \in [0,1]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with target velocity $\mathbf{v}^\star = \mathbf{z}_y - \boldsymbol{\epsilon}$. The source latent provides a condition rather than an endpoint of this path. We train the source- and EEG-conditioned velocity field with

$$\mathcal{L}_{\text{flow}} = \mathbb{E}[\mathcal{E}_{\text{vel}}(\mathbf{v}_\theta(\mathbf{x}_t, t; \mathbf{z}_s, \mathbf{X}), \mathbf{v}^\star)], \tag{2}$$

where the expectation is over training examples, noise, and time, and $\mathcal{E}_{\text{vel}}$ measures squared velocity error with normalization specified in Appendix C.2. At inference, we integrate the learned field from $\tilde{\mathbf{x}}_0 = \boldsymbol{\epsilon}$ and decode the endpoint as $\hat{I} = D(\tilde{\mathbf{x}}_1)$. We initialize the editor from OmniGen2 (Wu et al., 2025), retaining its native source conditioning. Separate projections and refiners embed source and generation latents, whose tokens then interact through joint attention. The resulting source-conditioned generation states provide the queries for EEG readout.

4.2 Visually Aligned EEG Representations

We learn a visually aligned EEG summary and retain the encoder's unpooled features as a second representation. The LaBraM-base encoder F_ϕ (Jiang et al., 2024) produces EEG tokens $\mathbf{P}$, and a linear regression head A_ψ maps their mean-pooled representation to a summary $\mathbf{g}$. A frozen ViT-H image encoder Φ_{vis} supplies the alignment target $\mathbf{y}$ from the paired image I_y:

$$\mathbf{P} = F_\phi(\mathbf{X}), \qquad \mathbf{g} = A_\psi(\text{MeanPool}(\mathbf{P})), \qquad \mathbf{y} = \Phi_{\text{vis}}(I_y), \tag{3}$$

where MeanPool averages over EEG tokens. Following visual alignment in EEG decoding (Dai et al., 2026), we train these representations with feature regression and a symmetric multi-positive objective. For a minibatch of B EEG–image pairs, the positive set $\mathcal{P}_i$ contains all examples sharing the stimulus of example i, including repeated trials or trial averages. Same-stimulus examples remain positive matches rather than being treated as negatives (Khosla et al., 2020). For each direction ν, EEG-to-image or image-to-EEG, π_{ij}^ν is the probability assigned to candidate j for query i, normalized over the minibatch candidates. The regression and contrastive terms are:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B}\sum_{i=1}^{B}\text{MSE}(\mathbf{g}_i, \mathbf{y}_i), \quad \mathcal{L}_{\text{con}} = -\frac{1}{2B}\sum_{i=1}^{B}\sum_{\nu}\log\sum_{j\in\mathcal{P}_i}\pi_{ij}^\nu. \tag{4}$$

where MSE is the element-averaged squared error. The complete alignment objective is:

$$\mathcal{L}_{\text{EEG}} = \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}, \tag{5}$$

with λ_{reg} and λ_{con} controlling the relative weights. After alignment, we freeze F_ϕ and A_ψ and supply both $\mathbf{g}$ and $\mathbf{P}$ to the editor. Appendix C.1 defines the matching probabilities.

4.3 Source-Conditioned EEG Readout

The same target can require different corrections to different source images. We therefore use the editor's source-conditioned visual state to read both EEG representations. The *Summary* pathway reads $\mathbf{g}$, while the *Unpooled* pathway reads $\mathbf{P}$. Learned projections Π_g and Π_p map these representations to separate EEG memories:

$$\mathbf{M}_g = \Pi_g(\mathbf{g}), \qquad \mathbf{M}_p = \Pi_p(\mathbf{P}). \tag{6}$$

Subscripts g and p denote the *Summary* and *Unpooled* pathways, respectively. Π_p acts token-wise. Both memories are computed once per recording before sampling. For a single editor block (index

omitted), the native joint-attention update $\mathcal{J}_t$ jointly processes generation tokens $\mathbf{Z}_x$ and source tokens $\mathbf{Z}_s$. Its generation-token output, including the native residual update, is:

$$\mathbf{H} = \mathcal{J}_t(\mathbf{Z}_x, \mathbf{Z}_s). \tag{7}$$

For pathway $b \in \{g, p\}$, RMS normalization operators N_b^H and N_b^M act on $\mathbf{H}$ and $\mathbf{M}_b$, respectively. Each attention head $h \in \mathcal{H}$ forms queries from the visual state and keys and values from the EEG memory:

$$\mathbf{Q}_{b,h} = N_b^H(\mathbf{H})\mathbf{W}_{b,h}^Q, \quad \mathbf{K}_{b,h} = N_b^M(\mathbf{M}_b)\mathbf{W}_{b,h}^K, \quad \mathbf{V}_{b,h} = N_b^M(\mathbf{M}_b)\mathbf{W}_{b,h}^V. \tag{8}$$

Each head attends only to its own pathway's memory:

$$\mathbf{A}_{b,h} = \mathrm{softmax}\Big(\mathbf{Q}_{b,h}\mathbf{K}_{b,h}^\top / \sqrt{d_h}\Big). \tag{9}$$

Here, d_h is the query/key width of head h, and $\top$ denotes transpose. Softmax operates row-wise over the n_b tokens in $\mathbf{M}_b$; neither source tokens nor the other pathway's memory enters this set. The head outputs are concatenated along the feature axis, projected back to the visual width by $\mathbf{W}_b^O$, and scaled by a learned scalar time gate $a_b(t)$:

$$\Delta_b(\mathbf{H}, \mathbf{M}_b, t) = a_b(t)\,\mathrm{Concat}_{h\in\mathcal{H}}[\mathbf{A}_{b,h}\mathbf{V}_{b,h}]\mathbf{W}_b^O. \tag{10}$$

Pathway outputs update $\mathbf{H}$. Gates use fixed time features $\mathbf{e}(t)$, learned coefficients $\boldsymbol{\gamma}_b$, and bias κ_b:

$$\mathbf{H}' = \mathbf{H} + \sum_{b\in\{g,p\}} \Delta_b(\mathbf{H}, \mathbf{M}_b, t), \qquad a_b(t) = \sigma(\boldsymbol{\gamma}_b^\top \mathbf{e}(t) + \kappa_b), \tag{11}$$

where σ is the logistic sigmoid. The native feed-forward sublayer then processes $\mathbf{H}'$. Both pathways query the same visual state, using independent readout parameters and gates across pathways and blocks. The source informs the EEG queries, while each pathway normalizes attention over its EEG memory. The memories remain fixed during sampling, while their readout varies with visual state and time. Appendix D specifies the memory projections, normalization layers, and time features.

4.4 Training Objectives and Inference

We measure the EEG-induced prediction change through paired evaluations of the current editor at $t = 0$, with and without both EEG pathways. The evaluations share the source, any active LoRA dropout masks, and standard Gaussian noise $\boldsymbol{\epsilon}_0$ drawn independently of the flow-matching noise:

$$\mathbf{v}_c = \mathbf{v}_\theta(\boldsymbol{\epsilon}_0, 0; \mathbf{z}_s, \mathbf{X}), \qquad \mathbf{v}_o = \mathbf{v}_\theta^{\mathrm{off}}(\boldsymbol{\epsilon}_0, 0; \mathbf{z}_s). \tag{12}$$

The one-step endpoint estimates $\hat{\mathbf{z}} = \boldsymbol{\epsilon}_0 + \mathbf{v}_c$ and $\hat{\mathbf{z}}_o = \boldsymbol{\epsilon}_0 + \mathbf{v}_o$ satisfy $\hat{\mathbf{z}} - \hat{\mathbf{z}}_o = \mathbf{v}_c - \mathbf{v}_o$. We supervise the conditioned endpoint with $\mathcal{L}_{\mathrm{end}}$ and match the prediction difference to $\mathbf{z}_y - \mathbf{z}_s$ through the auxiliary loss $\mathcal{L}_\Delta$:

$$\mathcal{L}_{\mathrm{end}} = \mathbb{E}[\mathcal{E}_{\mathrm{vel}}(\mathbf{v}_c, \mathbf{z}_y - \boldsymbol{\epsilon}_0)], \tag{13}$$

$$\mathcal{L}_\Delta = \mathbb{E}[\mathrm{mean}\,\rho_\beta(\mathbf{v}_c - \mathrm{sg}(\mathbf{v}_o) - (\mathbf{z}_y - \mathbf{z}_s))]. \tag{14}$$

Here, sg stops gradients, ρ_β is the elementwise SmoothL1 penalty with threshold $\beta > 0$, and mean averages over latent elements. Expectations are over training examples and noise. The EEG-disabled reference is recomputed from the current editor and detached in $\mathcal{L}_\Delta$.

We supervise decoded one-step target estimates with $\mathcal{L}_{\mathrm{dec}}$, using spatial weights derived from each training pair to emphasize regions requiring edits. Appearance editing uses noise-endpoint estimates and includes a source-preservation term; semantic editing uses sampled auxiliary times without this additional term. Neither requires unrolling the sampler (Appendix C.3). The total editing loss is

$$\mathcal{L}_{\mathrm{edit}} = \mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{end}}\mathcal{L}_{\mathrm{end}} + \lambda_\Delta(k)\mathcal{L}_\Delta + \lambda_{\mathrm{dec}}(k)\mathcal{L}_{\mathrm{dec}}, \tag{15}$$

where k is the optimizer-step index and λ_{end} weights endpoint supervision. The schedules for $\lambda_\Delta(k)$ and $\lambda_{\mathrm{dec}}(k)$ are specified in Appendix E. Training updates the memory projections, EEG readout parameters, time gates, and backbone LoRA (Hu et al., 2021), while the EEG encoder, regression head, VAE, and original backbone weights remain frozen.

We generate edits from the source image and EEG through Euler integration of the velocity field, keeping the source latent and EEG memories fixed and decoding only the final latent. Inference requires no target images, text instructions, edit masks, or training-derived spatial weights (Appendix K).

Table 1: **EEG-guided image editing on NEVE.** T-LPIPS and PixCorr measure whole-image agreement with the target; C-LPIPS, C-MAE, and C-SSIM measure agreement in source–target change regions. R-Edit jointly assesses target error and excess error in low-change regions. All methods use the same benchmark averaging protocol. Best results are in **bold**.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.897	0.881	0.368	0.190	0.046	0.275
LoongX	0.952	1.070	0.293	0.198	0.166	0.337
MindPainter	0.671	0.751	0.295	0.187	0.291	0.350
Brain-Supervised	0.415	0.626	0.339	0.291	0.454	0.386
MindArt	0.462	0.663	0.358	0.252	0.467	0.371
PVS-EEG	0.397	0.620	0.330	0.286	0.502	0.377
EEGraft (Ours)	**0.278** -0.119	**0.479** -0.141	**0.197** -0.096	**0.420** +0.129	**0.752** +0.250	**0.585** +0.199

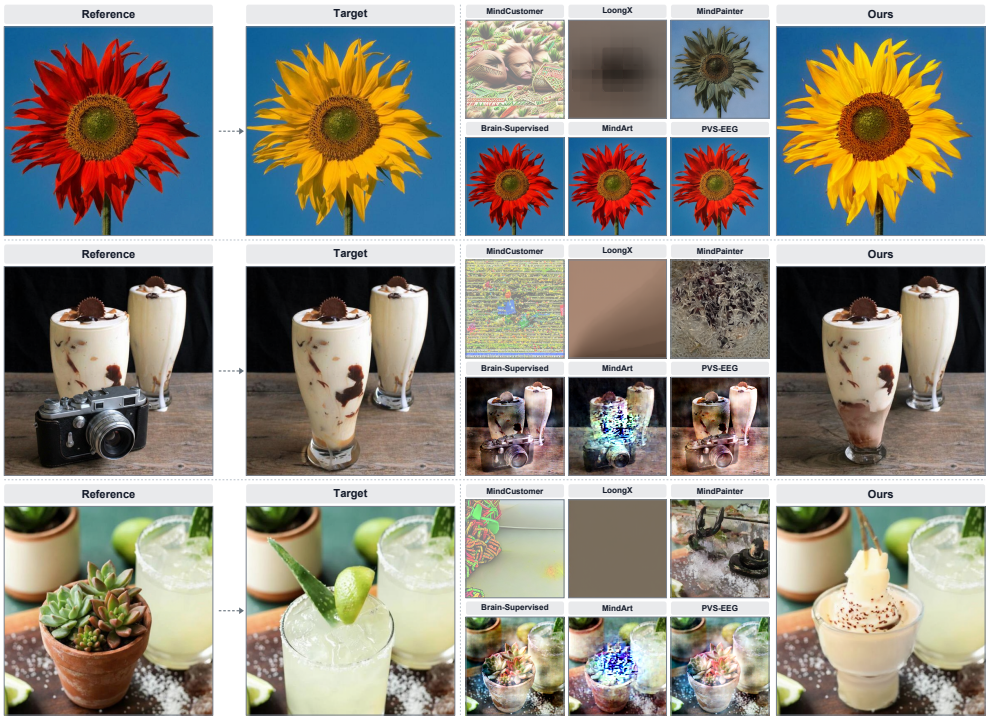

Figure 4: **Visual corrections from EEG.** From left to right, the source image, the viewed target, outputs from six baselines, and the EEGRAFT output. See Appendix L for more baseline settings.

5 EXPERIMENTS

We evaluate EEGRAFT on NEVE's held-out appearance and semantic test sets against six neural image-editing baselines under a unified evaluation protocol (Sec. 5.1), assessing editing accuracy (Sec. 5.2) and source preservation (Sec. 5.3). We then assess the contribution of correctly paired EEG through fixed-editor controls and examine the roles of the EEG readout pathways and training objectives through component ablations (Sec. 5.4).

5.1 EXPERIMENTAL SETUP

Dataset and evaluation. We evaluate on 1,600 appearance and 1,200 semantic test pairs from NEVE, following the target-concept splits and metrics in Sec. 3.2. Within each subset, we average the five per-trial metrics over all available EEG trials and compute R-Edit from summed errors. Overall scores weight the subsets by their test-pair counts (1,600:1,200); see Appendix B for details.

Table 2: **Effect of EEG conditioning on editing.** With the editor and source fixed, we compare Correct EEG, Null EEG, and Mismatched EEG.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
Null EEG	0.310	0.520	0.242	0.386	0.696	0.521
Mismatched EEG	0.307	0.512	0.245	0.383	0.692	0.532
Correct EEG	**0.278** $^{-0.029}$	**0.479** $^{-0.033}$	**0.197** $^{-0.045}$	**0.420** $^{+0.034}$	**0.752** $^{+0.056}$	**0.585** $^{+0.053}$

Baselines. We compare with six neural image-editing methods: MindCustomer (Yu et al., 2025a), MindPainter (Yu et al., 2025b), Brain-Supervised (Davis et al., 2022), PVS-EEG (De La Torre-Ortiz & Ruotsalo, 2024), MindArt (Chen et al., 2024), and LoongX (Zhou et al., 2026). All baselines use target-viewing EEG, retaining their EEG encoders where available and otherwise using our aligned encoder. Appendix L details adaptations and differences from their original neural inputs and editing tasks.

Implementation. We initialize the editor from OmniGen2 (Wu et al., 2025) and use a visually aligned LaBraM-base EEG encoder (Jiang et al., 2024). We train separate editors for appearance and semantic editing, keeping the aligned EEG encoder frozen. Inference uses 28 Euler steps. Appendices E and K specify the trainable components, optimization settings, and sampling details.

5.2 Target Recovery

EEGraft achieves the best aggregate C-LPIPS, C-MAE, and C-SSIM scores on Neve (Table 1). Compared with PVS-EEG, the strongest baseline on C-LPIPS, EEGraft reduces C-LPIPS from 0.620 to 0.479 (22.7%). It also reduces C-MAE from 0.293 to 0.197 (approximately 32.8%) compared with LoongX, the strongest baseline on this metric. These gains indicate closer agreement with the target within regions that require editing. The advantage also holds at the whole-image level, with T-LPIPS of 0.278 and PixCorr of 0.752. Results on the retained official test stimuli and participant-wise results for ten participants are provided in Appendices A and I, respectively.

Figure 4 distinguishes two editing errors: retaining source attributes that should change, and modifying the image without recovering the required foreground content or scene structure. For example, several baselines retain the red sunflower, whereas EEGraft recovers the target's yellow petals while preserving the blue background. These results also span distinct original conditioning settings: LoongX uses instruction-based multimodal recordings (Zhou et al., 2026), MindPainter uses fMRI for masked painting (Yu et al., 2025b), and Brain-Supervised learns GAN-latent semantic directions from EEG (Davis et al., 2022). NEVE instead requires target-perception EEG to guide recovery of a specific viewed target relative to the source, beyond producing a plausible image or semantic change. Appendix L details these task and observation differences.

5.3 Source Preservation

EEGraft achieves stronger target recovery with less added error in low-change regions than all six baselines (Figure 5). We measure added error relative to the source image's existing target error in these regions. Higher target improvement indicates a greater reduction in change-region target error, while higher preservation odds indicate less added error in low-change regions. MindPainter and LoongX achieve greater target improvement than PVS-EEG and Brain-Supervised, but lower preservation odds. The joint recovery–preservation score R-Edit reaches 0.585, compared with 0.386 for Brain-Supervised, the strongest baseline, and 0.377 for PVS-EEG. Appendix B provides the metric and axis definitions.

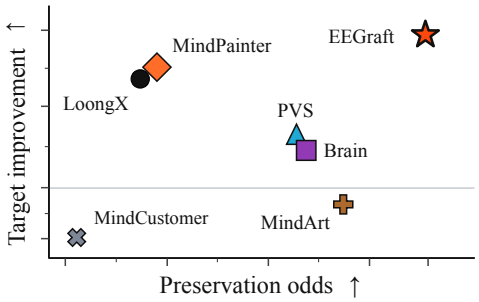

Figure 5: **Target recovery and source preservation.** Higher values indicate greater target improvement; positions farther right indicate better preservation.

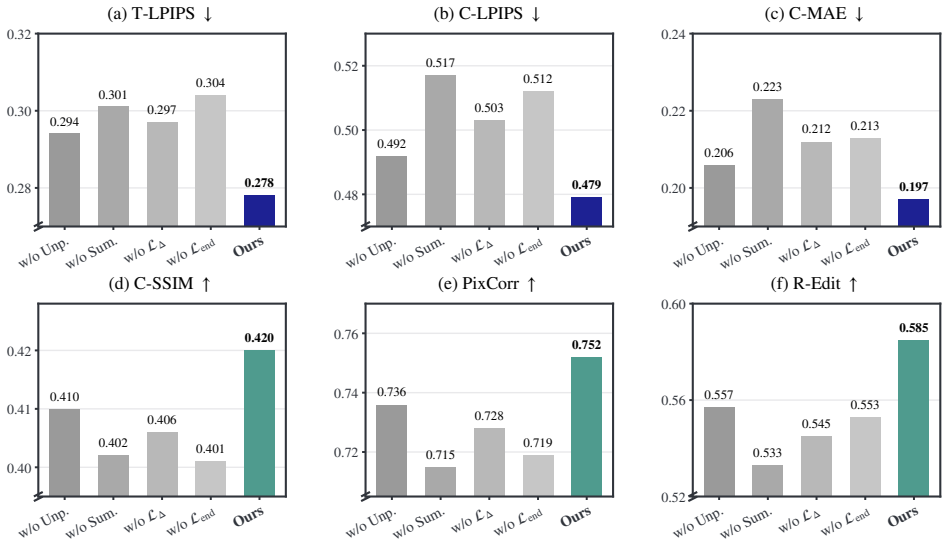

Figure 6: **Component ablations.** We compare the full model with variants that omit the Unpooled pathway, the Summary pathway, the prediction-difference loss $\mathcal{L}_\Delta$, or the endpoint loss $\mathcal{L}_{\text{end}}$.

5.4 Ablation Study

Contribution of Paired EEG. With the trained editor, source, and evaluation target fixed, we compare *Null EEG*, which bypasses both readouts but still generates an image rather than copying the source; *Mismatched EEG*, a recording of another stimulus from the same participant; and *Correct EEG*, the target-paired recording. Mismatched EEG keeps both pathways active but breaks recording–target correspondence. Null and Mismatched EEG show no consistent ordering across metrics (Table 2). Correct EEG leads on all six metrics, reducing C-LPIPS from 0.512 to 0.479 and C-MAE from 0.245 to 0.197 relative to Mismatched EEG, while raising R-Edit from 0.532 to 0.585. With conditioning active in both cases, this advantage supports a contribution from recording–target correspondence beyond enabling EEG conditioning. Appendix G reports additional cross-participant mismatch controls.

Component Ablations. Figure 6 reports ablations of the training objectives and EEG pathways. Removing $\mathcal{L}_\Delta$ raises C-LPIPS and C-MAE from 0.479 and 0.197 to 0.503 and 0.212, and lowers R-Edit from 0.585 to 0.545; removing $\mathcal{L}_{\text{end}}$ yields 0.512, 0.213, and 0.553, respectively. Both removals degrade all six metrics. These results support prediction-difference supervision as a complement to target supervision. Summary-only outperforms Unpooled-only on all six metrics. Combining both pathways improves every metric over either alone, reducing C-LPIPS from 0.492 to 0.479 and C-MAE from 0.206 to 0.197 relative to Summary-only, confirming the added value of unpooled features. Inference measurements show larger residual updates from Summary and smaller, closely aligned updates from Unpooled (Appendix H). Appendix G provides further ablations.

6 Conclusion

We introduced EEGraft for editing image semantics and appearance using EEG recorded while a participant views a target image. The method aligns EEG with visual representations and reads neural information through source-conditioned visual states, guiding generation without text instructions or edit masks. On Neve, our benchmark for appearance and semantic editing, EEGraft improves target recovery and source preservation over the compared methods. With the editor and source fixed, correctly paired EEG produces more accurate edits than mismatched or disabled conditioning, supporting the contribution of target-related neural information to controlled image editing.

Limitations. Fine textures, local markings, and subtle color differences are not always recovered reliably. Our evaluation uses target-viewing EEG and generated source images; broader applicability to real-world editing and interactive use remains to be established. We hope this work encourages more precise and reliable neural guidance for visual creation and interaction.

AI USE STATEMENT

We used generative AI tools for source-image generation and manuscript editing. To construct NEVE, we used `gemini-3-pro-image` to generate source images by applying constrained visual modifications to selected stimulus images. The original stimuli served as targets, and their associated EEG recordings came from existing EEG–image datasets. Neither the target images nor the EEG recordings were generated by AI. We also used generative AI tools to improve the clarity, grammar, and readability of portions of the manuscript. The authors reviewed all AI-generated images and AI-assisted text and take responsibility for the final paper, including its AI-assisted content.

REFERENCES

Yunpeng Bai, Xintao Wang, Yan-Pei Cao, Yixiao Ge, Chun Yuan, and Ying Shan. Dreamdiffusion: High-quality eeg-to-image generation with temporal masked signal modeling and clip alignment. In *European Conference on Computer Vision*, pp. 472–488. Springer, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Jiaxuan Chen, Yu Qi, Yueming Wang, and Gang Pan. Mind artist: Creating artistic snapshots with human thought. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27197–27207. IEEE, 2024.

Sicheng Dai, Hongwang Xiao, Shan Yu, and Qiwei Ye. Autoregressive visual decoding from eeg signals. *arXiv preprint arXiv:2602.22555*, 2026.

Keith M Davis, Carlos De La Torre-Ortiz, and Tuukka Ruotsalo. Brain-supervised image editing. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18459–18468. IEEE, 2022.

Carlos De La Torre-Ortiz and Tuukka Ruotsalo. Perceptual visual similarity from eeg: Prediction and image generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11146–11155, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Alessandro T Gifford, Kshitij Dwivedi, Gemma Roig, and Radoslaw M Cichy. A large and rich eeg dataset for modeling human visual object recognition. *NeuroImage*, 264:119754, 2022.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, 8(1):15037, 2017.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

Wei-Bang Jiang, Liming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous eeg data in bci. In *International Conference on Learning Representations*, volume 2024, pp. 16405–16426, 2024.

Isaak Kavasidis, Simone Palazzo, Concetto Spampinato, Daniela Giordano, and Mubarak Shah. Brain2Image: Converting brain signals into images. In *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1809–1817, 2017. doi: 10.1145/3123266.3127907.

Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, 2008.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730. IEEE, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Dongyang Li, Chen Wei, Shiying Li, Jiachen Zou, Haoyang Qin, and Quanying Liu. Visual decoding and reconstruction via eeg embeddings with guided diffusion. *arXiv preprint arXiv:2403.07721*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

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, 2008. doi: 10.1016/j.neuron.2008.11.004.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

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, 2009. doi: 10.1016/j.neuron.2009.09.006.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Paul Scotti, Atmadeep Banerjee, Jimmie Goode, Stepan Shabalin, Alex Nguyen, Aidan Dempster, Nathalie Verlinde, Elad Yundler, David Weisberg, Kenneth Norman, et al. Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors. *Advances in Neural Information Processing Systems*, 36:24705–24728, 2023.

Guohua Shen, Tomoyasu Horikawa, Kei Majima, and Yukiyasu Kamitani. Deep image reconstruction from human brain activity. *PLOS Computational Biology*, 15(1):e1006633, 2019. doi: 10.1371/journal.pcbi.1006633.

Yonghao Song, Bingchuan Liu, Xiang Li, Nanlin Shi, Yijun Wang, and Xiaorong Gao. Decoding natural images from eeg for object recognition. In *International conference on learning representations*, volume 2024, pp. 47648–47665, 2024.

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)*, pp. 14453–14463. IEEE, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1921–1930, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv e-prints*, pp. arXiv–2506, 2025.

Jonathan Xu, Ugo Bruzadin Nunes, Wangshu Jiang, Samuel Ryther, Jordan Pringle, Paul Steven Scotti, Arnaud Delorme, and Reese Kneeland. Alljoined-1.6 m: A million-trial eeg-image dataset for evaluating affordable brain-computer interfaces, 2025.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

Muzhou Yu, Shuyun Lin, Lei Ma, Bo Lei, and Kaisheng Ma. Mindcustomer: Multi-context image generation blended with brain signal. In *Forty-second International Conference on Machine Learning*, 2025a.

Muzhou Yu, Shuyun Lin, Hongwei Yan, and Kaisheng Ma. Mindpainter: Efficient brain-conditioned painting of natural images via cross-modal self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Jiawen Zheng, Haonan Jia, Ming Li, Yuhui Zheng, Yufeng Zeng, Yang Gao, and Chen Liang. Learning brain representation with hierarchical visual embeddings. *arXiv preprint arXiv:2602.07495*, 2026.

Pengfei Zhou, Jie Xia, Xiaopeng Peng, Wangbo Zhao, Zilong Ye, Zekai Li, Suorong Yang, Jiadong Pan, Yuanxiang Chen, Ziqiao Wang, et al. Neural-driven image editing. *Advances in Neural Information Processing Systems*, 38:100465–100505, 2026.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

Shuqi Zhu, Ziyi Ye, Qingyao Ai, and Yiqun Liu. Eeg-imagenet: An electroencephalogram dataset and benchmarks with image visual stimuli of multi-granularity labels. *arXiv e-prints*, pp. arXiv–2406, 2024.

A NEVE CONSTRUCTION

Source–target pairing. We construct NEVE using the stimulus images and EEG recordings in THINGS-EEG2 (Gifford et al., 2022), EEG-ImageNet (Zhu et al., 2024), and Alljoined-1.6M (Xu et al., 2025). We manually select suitable stimulus images and use `gemini-3-pro-image` to generate source images from them. The original stimulus remains the target I_y and retains its recorded EEG $\mathbf{X}$; the generated image serves as the source I_s. Each editing example is thus $(I_s, \mathbf{X}, I_y)$, with EEG corresponding to the viewed target rather than the generated source.

Source-image generation. We generate each source image by applying a constrained modification to a selected stimulus image, while retaining the original stimulus as the target. Appearance edits change the color of the main foreground object or coherent group; semantic edits replace it with a distinct, scene-compatible category. The task summaries below specify the permitted changes and the content to preserve.

Appearance Edits: Task Summary

Identify the main foreground object or coherent group of similar objects. Change only its appearance by assigning a clearly different, natural, and category-typical color. For a group, apply the same color consistently to all members. Preserve semantic identity, instance count, shape, pose, geometry, texture, and spatial arrangement, together with the background, viewpoint, framing, lighting, and unrelated objects. Keep shading and material appearance natural after recoloring. The color change should be immediately visible, while avoiding subtle shade changes, implausible colors, partial recoloring, or modifications outside the selected target.

Semantic Edits: Task Summary

Identify the main foreground object or coherent group of similar objects. Replace only this target with a semantically distinct, realistic, and scene-compatible category that is also clearly distinguishable in coarse form without relying on color alone. For a group, apply one replacement category consistently to all members. Preserve approximate count, location, scale, and group arrangement, together with the background, viewpoint, framing, lighting, and unrelated objects. Allow the silhouette, part structure, and local geometry to change as required by the new category, while avoiding subtype substitutions, recoloring-only edits, hybrid remnants, or unnecessary scene changes.

The coral text marks constraints introduced to make source generation both realistic and tightly controlled. They reduce unnatural edits that could let pretrained editors recover the target from visual priors alone, while also preventing residual cues from unchanged instances of the same object group.

EEG standardization. For THINGS-EEG2 and Alljoined-1.6M, we apply a zero-phase FIR band-pass filter at 0.1–75 Hz and a 50 Hz notch filter to the continuous EEG. We extract epochs from 200 ms before to 1000 ms after stimulus onset and perform baseline correction by subtracting each channel's mean over the pre-stimulus $[-200, 0]$ ms interval. We then retain the first 1000 ms after stimulus onset and downsample to 200 Hz, yielding 200 temporal samples per channel. EEG-ImageNet is used in its released, preprocessed form. All recordings are expressed in μV and mapped to the THINGS-63 scalp grid. Matching electrodes are copied directly, and missing locations are filled by spherical-spline interpolation. The resulting inputs are 63×200 tensors. For the 500 ms EEG-ImageNet trials, unobserved temporal bins are zero-padded and tracked by an observation mask. Interpolation and padding standardize the input representation without adding independent EEG measurements.

Dataset size and partitioning. The appearance subset contains 14,486 training and 1,600 test source–target pairs, while the semantic subset contains 11,989 training and 1,200 test pairs. In total, NEVE comprises 29,275 pairs and 2,714,765 EEG trials from 46 participants. The appearance and semantic subsets are selected and partitioned separately, since a stimulus need not be suitable for both editing tasks. Within each subset, partitioning is performed at the target-concept level: all stimuli belonging to the same concept and all associated EEG recordings remain in one split. Stimuli shared across recording resources receive consistent assignments within that subset. No target concept appears in both the training and test partitions of either subset.

Shared EEG alignment. The shared EEG aligner is trained on the union of non-test target–EEG data from the two editing subsets, with all test targets and their associated recordings from either subset excluded. This exclusion applies across editing subsets and recording resources: a target held out for one task cannot enter alignment through the other task. After alignment, the EEG encoder and regression head are frozen, and the two editors are trained separately on their respective training partitions.

Official-subset evaluation. The Full test set retains all 200 images from the original official test split as the Official subset. This preserves the stimulus selection of the original visual-decoding protocol, including images with varied suitability for editing. Table 3 reports the Official-subset comparison; Table 1 reports the Full test set. Both use the same metric definitions and averaging protocol. EEGraft's advantage extends to the retained official stimuli, showing that the improvement is not restricted to the additional evaluation images in NEVE.

Table 3: **Quantitative comparison on the combined Official test sets.** The evaluation combines appearance-editing and semantic-editing pairs. Subset scores are weighted by the number of test pairs, with R-Edit computed within each subset before averaging. Best results are shown in bold. Superscripts show EEGraft minus the best-performing baseline for each metric.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.918	0.884	0.371	0.172	0.060	0.267
LoongX	0.939	1.080	0.295	0.179	0.212	0.330
MindPainter	0.673	0.742	0.295	0.173	0.270	0.340
Brain-Supervised	0.438	0.637	0.324	0.278	0.446	0.375
MindArt	0.514	0.675	0.347	0.226	0.460	0.358
PVS-EEG	0.419	0.633	0.319	0.269	0.496	0.366
EEGraft (Ours)	**0.314**$^{-0.105}$	**0.518**$^{-0.115}$	**0.228**$^{-0.067}$	**0.361**$^{+0.083}$	**0.685**$^{+0.189}$	**0.527**$^{+0.152}$

B EVALUATION

Evaluation regions. For evaluation example i, let $I_{s,i}$, $I_{y,i}$, and $\hat{I}_i$ denote the source, target, and generated image, preserving the notation in Sec. 3.1. Images are read as RGB, resized to 512×512 with PIL bicubic interpolation, and represented in FP32 on $[0, 1]$. For pixel p and RGB channel c, define the source–target difference $d_i(p)$ and its smoothed map $\tilde{d}_i$ as:

$$d_i(p) = \frac{1}{3}\sum_{c=1}^{3} |I_{y,i}(p,c) - I_{s,i}(p,c)|, \qquad \tilde{d}_i = \mathrm{Gaussian}_{\sigma=1} * d_i, \tag{16}$$

where $*$ denotes convolution, $\mathrm{Gaussian}_{\sigma=1}$ is the smoothing kernel, and that kernel has standard deviation $\sigma = 1$ pixel, reflective boundaries, and truncation at four standard deviations. Let η_i denote the change threshold, M_i the resulting change region, and K_i its complement in the image domain. We use:

$$\eta_i = \max\left(\frac{3}{255}, \mathrm{Otsu}_{256}(\tilde{d}_i)\right), \qquad M_i = \mathrm{Dilate}_{\mathrm{disk}(3)}[\tilde{d}_i > \eta_i], \qquad K_i = \neg M_i. \tag{17}$$

Here, Otsu_{256} is Otsu thresholding with 256 bins; the comparison is strict. $\mathrm{Dilate}_{\mathrm{disk}(3)}$ denotes binary dilation with a radius-3 disk and zero-valued boundaries; $\neg$ denotes the binary complement. We use a binary mask and its set of active pixels interchangeably; $|M_i|$ denotes the number of active pixels. These regions depend only on the source and target RGB images, are shared by all methods, and are distinct from the latent-derived training weights.

Perceptual and structural metrics. T-LPIPS uses the calibrated AlexNet LPIPS v0.1 model with scalar output. The network is frozen and evaluated in FP32 with TF32 disabled, batch size 24, and inputs mapped to $[-1, 1]$. C-LPIPS switches the same network to spatial output. Let $d_{\mathrm{LPIPS},i}(p)$ denote its full-image distance map at pixel p. Then:

$$\text{T-LPIPS}_i = \mathrm{LPIPS}(\hat{I}_i, I_{y,i}), \qquad \text{C-LPIPS}_i = \frac{1}{|M_i|}\sum_{p \in M_i} d_{\mathrm{LPIPS},i}(p). \tag{18}$$

Feature extraction uses the complete images, so regional values retain the surrounding context within the feature receptive fields.

C-SSIM uses `skimage.metrics.structural_similarity` with RGB channel axis 2, data range 1, an 11×11 Gaussian window with standard deviation 1.5, population covariance, and parameter values `K1`$= 0.01$ and `K2`$= 0.03$. Let $\mathrm{SSIMmap}(\hat{I}_i, I_{y,i})_{p,c}$ denote the returned local structural-similarity value at pixel p and channel c. We average this map over M_i and all RGB channels:

$$\text{C-SSIM}_i = \frac{1}{3|M_i|} \sum_{p \in M_i} \sum_{c=1}^{3} \mathrm{SSIMmap}(\hat{I}_i, I_{y,i})_{p,c}. \tag{19}$$

The returned whole-image scalar is discarded, and the map is used without additional border cropping. This is single-scale SSIM. PixCorr measures whole-image pixel correlation between $\hat{I}_i$ and $I_{y,i}$.

Reference-relative error. For a nonempty pixel region R and RGB images U, V, the regional mean absolute error is:

$$\mathrm{MAE}_R(U, V) = \frac{1}{3|R|} \sum_{p \in R} \sum_{c=1}^{3} |U(p, c) - V(p, c)|. \tag{20}$$

The per-image change-region error is:

$$\text{C-MAE}_i = \mathrm{MAE}_{M_i}(\hat{I}_i, I_{y,i}). \tag{21}$$

Let $\mathrm{Collateral}_i$ denote the nonnegative increase in target error outside the change region, relative to the source's existing error:

$$\mathrm{Collateral}_i = \max\Big(\mathrm{MAE}_{K_i}(\hat{I}_i, I_{y,i}) - \mathrm{MAE}_{K_i}(I_{s,i}, I_{y,i}), 0\Big). \tag{22}$$

An improvement in the low-change region incurs zero collateral cost. For a valid evaluation index set $\mathcal{S}$ within an editing subset, the reference-relative score is:

$$\text{R-Edit}(\mathcal{S}) = \frac{\sum_{i \in \mathcal{S}} \mathrm{MAE}_{M_i}(I_{s,i}, I_{y,i})}{\sum_{i \in \mathcal{S}} \left[\mathrm{MAE}_{M_i}(I_{s,i}, I_{y,i}) + \text{C-MAE}_i + \mathrm{Collateral}_i\right]}. \tag{23}$$

When the summed source change-region error is positive, R-Edit lies in $(0, 1]$, and copying the source gives 0.5. Exceeding this reference requires the reduction in change-region target error to outweigh the collateral cost. Regional metrics are undefined for an empty change region. R-Edit additionally requires a nonempty preservation region and is computed over examples with all required errors defined. Errors are summed before forming the ratio; a zero denominator returns an undefined score without adding an epsilon.

Reference points and aggregation. The errors are summed before taking the ratio; per-image ratios are not averaged. Collateral cost is clipped separately for each example, so improvements on one image cannot cancel added error on another. When the summed source-copy error is positive, reproducing every target gives a score of 1, and:

$$\begin{aligned} \text{R-Edit}(\mathcal{S}) > \tfrac{1}{2} \quad \Longleftrightarrow \quad & \sum_{i \in \mathcal{S}} \left[\text{C-MAE}_i + \mathrm{Collateral}_i\right] \\ & < \sum_{i \in \mathcal{S}} \mathrm{MAE}_{M_i}(I_{s,i}, I_{y,i}). \end{aligned} \tag{24}$$

The reference is source copying, not random generation, and R-Edit is not a success rate. For fixed examples and masks, a higher score corresponds to lower summed output C-MAE and collateral cost. C-LPIPS, C-SSIM, T-LPIPS, and PixCorr are complementary measurements and do not enter this score.

The main benchmark aggregation weights the appearance and semantic subset scores by their test-set sizes (1,600 and 1,200 pairs, respectively). R-Edit is computed within each subset before this weighted averaging, not from an average of per-image ratios.

Target-recovery and preservation visualization. Figure 5 visualizes the two components of the recovery–preservation trade-off separately. For an evaluation set $\mathcal{S}$, define

$$A = \sum_{i \in \mathcal{S}} \mathrm{MAE}_{M_i}(I_{s,i}, I_{y,i}), \qquad E = \sum_{i \in \mathcal{S}} \text{C-MAE}_i, \qquad C = \sum_{i \in \mathcal{S}} \text{Collateral}_i.$$

Here, A is the change-region error of source copying, E is the corresponding error of the edited outputs, and C is the summed increase in target error over the low-change regions. With this notation, R-Edit $= A/(A + E + C)$.

The vertical axis in Figure 5 measures the relative reduction in change-region target error:

$$\text{Target Improvement} = \frac{A - E}{A} = 1 - \frac{E}{A}.$$

A positive value therefore means that the edited output is closer to the target than source copying within the regions that require correction; zero corresponds to no improvement over the source, and a negative value indicates increased change-region error.

For preservation, we first define the normalized quantity

$$p_{\text{pres}} = \frac{A}{A + C}.$$

We plot its odds,

$$\text{Preservation Odds} = \frac{p_{\text{pres}}}{1 - p_{\text{pres}}} = \frac{A}{C},$$

for $C > 0$; when $C = 0$, the odds are interpreted as $+\infty$. Because A is fixed across methods evaluated on the same examples, larger preservation odds correspond to less collateral error relative to the scale of the required correction. These axes are used only to visualize the two aspects of the trade-off; the reported benchmark metric remains R-Edit.

C Training Details

We expand the alignment and editing objectives without changing the notation of Sec. 4. Optimization settings and objective schedules are reported in Appendix E.

C.1 Multi-Positive EEG Alignment

For the minibatch in equation 4, let s_{ij} denote the compatibility logit between EEG summary $\mathbf{g}_i$ and image feature $\mathbf{y}_j$. The matching probabilities are:

$$\pi_{ij}^{\mathrm{E}\to\mathrm{V}} = \frac{\exp(s_{ij})}{\sum_{\ell=1}^{B} \exp(s_{i\ell})}, \qquad \pi_{ij}^{\mathrm{V}\to\mathrm{E}} = \frac{\exp(s_{ji})}{\sum_{\ell=1}^{B} \exp(s_{\ell i})}. \tag{25}$$

Here, ℓ indexes candidates. In the first expression, EEG i queries the image candidates; in the second, image i queries the EEG candidates. The positive set $\mathcal{P}_i$ is determined by stimulus identity, including the paired example itself. The objective in equation 4 takes the logarithm after summing the positive probabilities, rather than averaging separate positive log-probabilities. Regression targets the feature of the exact paired stimulus. The ViT-H encoder is frozen.

C.2 Latent Training Objectives

Velocity error. For arrays $\mathbf{a}$ and $\mathbf{b}$ with n scalar elements, let a_j and b_j denote their elements indexed by j. The mean squared error and its energy-normalized version are:

$$\begin{aligned} \mathrm{MSE}(\mathbf{a}, \mathbf{b}) &= \frac{1}{n} \sum_{j=1}^{n} (a_j - b_j)^2, \\ \mathrm{NMSE}(\mathbf{a}, \mathbf{b}) &= \frac{\mathrm{MSE}(\mathbf{a}, \mathbf{b})}{\mathrm{sg}\left(\max\left(\frac{1}{n}\sum_{j=1}^{n} b_j^2, 10^{-4}\right)\right)}. \end{aligned} \tag{26}$$

The stop-gradient operator sg is the same as in equation 14. For both flow and endpoint supervision,

$$\mathcal{E}_{\text{vel}}(\mathbf{a}, \mathbf{b}) = \begin{cases} \text{NMSE}(\mathbf{a}, \mathbf{b}), & \text{appearance edits,} \\ \text{MSE}(\mathbf{a}, \mathbf{b}), & \text{semantic edits.} \end{cases} \tag{27}$$

Per-example errors are averaged over the examples assigned to each objective. Flow supervision uses equation 2; endpoint supervision uses equation 13 with independent noise $\boldsymbol{\epsilon}_0$. Time distributions and batch assignments are given in Table 4.

Prediction-difference penalty. For a scalar residual u, the SmoothL1 function in equation 14 is:

$$\rho_\beta(u) = \begin{cases} u^2/(2\beta), & |u| < \beta, \\ |u| - \beta/2, & |u| \geq \beta. \end{cases} \tag{28}$$

It is applied elementwise and then averaged over latent elements. Because both endpoint estimates use the same noise, their difference is $\mathbf{v}_c - \mathbf{v}_o$. For a paired evaluation, the endpoint and prediction-difference objectives act on the residuals

$$\mathbf{r}_y = \hat{\mathbf{z}} - \mathbf{z}_y,$$
$$\mathbf{r}_\Delta = (\hat{\mathbf{z}} - \text{sg}(\hat{\mathbf{z}}_o)) - (\mathbf{z}_y - \mathbf{z}_s).$$

The first measures absolute target error, while the second compares the EEG-induced prediction change with the desired source–target displacement. The online reference is computed by the EEG-disabled editor rather than set to the source latent. Stop-gradient directs the auxiliary update through the conditioned evaluation. Updates to the shared backbone LoRA, including those through the conditioned branch, can change subsequent reference predictions. Gradient handling and forward-pass modes are specified in Appendix E.3. The relative weights of the two objectives are specified in Table 4.

C.3 Decoded Image Supervision

One-step estimates. Let τ denote an auxiliary generation time and $\hat{\mathbf{z}}_\tau$ the corresponding one-step endpoint estimate. The decoded prediction J is:

$$\hat{\mathbf{z}}_\tau = \mathbf{x}_\tau + (1 - \tau)\mathbf{v}_\theta(\mathbf{x}_\tau, \tau; \mathbf{z}_s, \mathbf{X}), \qquad J = D(\hat{\mathbf{z}}_\tau), \tag{29}$$

where $\mathbf{x}_\tau$ at time τ. The reconstructed target and source references are $J_y = D(\mathbf{z}_y)$ and $J_s = D(\mathbf{z}_s)$. Auxiliary times and noise reuse are specified in Appendix E.3.

Image-space supervision is applied to this one-step estimate rather than an unrolled sampling trajectory. The prediction and its references use the same VAE decoder.

Spatial weights. Let p index a latent spatial location and c a latent channel. The channel-averaged absolute source–target difference $d_z(p)$ and its bounded transform $a(p)$ are:

$$d_z(p) = \text{mean}_c\, |\mathbf{z}_y(c, p) - \mathbf{z}_s(c, p)|, \qquad a(p) = 1 - \exp\left[-\frac{d_z(p)}{\max(\text{mean}_p\, d_z(p), 10^{-4})} \right]. \tag{30}$$

Here, mean_c and mean_p average over latent channels and spatial locations, respectively. For appearance editing, the edit map w_e applies 3×3 max pooling to a, bilinear upsampling to the image resolution, and normalization to unit spatial sum. The low-change map w_b is constructed from the inverse edit distribution and normalized separately.

Semantic editing uses 3×3 max pooling, 5×5 smoothing, interpolation, and clipping of the latent-difference map. Let $w_{\text{difference}}$ denote the resulting normalized difference weights and w_{uniform} the uniform spatial distribution at the same resolution. The semantic edit map is:

$$w_e = \tfrac{1}{2} w_{\text{difference}} + \tfrac{1}{2} w_{\text{uniform}}. \tag{31}$$

These soft weights are used only in training. Evaluation uses the RGB-derived binary regions defined in Appendix B.

Pixel and color discrepancies. For generic RGB images U and V, let $U^{(r)}$ and $V^{(r)}$ denote their versions at resolution $r \times r$, and let w be a spatial weight map, with $w^{(r)}$ its resized and normalized version at that resolution. In image-space expressions, p indexes pixels and c indexes RGB channels. Let ρ_{RGB} denote the elementwise RGB SmoothL1 penalty, using the form in equation 28 with the RGB implementation's threshold. The weighted RGB discrepancy is:

$$\mathcal{R}_w^{(r)}(U,V) = \sum_p w^{(r)}(p) \frac{1}{3} \sum_{c=1}^{3} \rho_{\mathrm{RGB}}\left(U^{(r)}(p,c) - V^{(r)}(p,c)\right). \tag{32}$$

At each supervised resolution, write $\mathbf{o}_U(p) \in \mathbb{R}^3$ for the OKLab coordinates of $U^{(r)}(p)$, $\mathbf{h}_U(p) \in \mathbb{R}^2$ for their ab components, and $\chi_U(p) = \|\mathbf{h}_U(p)\|_2$ for chroma. The resolution index is suppressed on these color coordinates. With $\cos(\cdot,\cdot)$ denoting cosine similarity and $\|\cdot\|_2$ the Euclidean norm, the color terms are:

$$\mathcal{L}_{\mathrm{OKLab}}^{(r)} = \sum_p w_e^{(r)}(p) \|\mathbf{o}_J(p) - \mathbf{o}_{J_y}(p)\|_2, \tag{33}$$

$$\mathcal{L}_{\mathrm{hue}}^{(r)} = \frac{\sum_p w_e^{(r)}(p) \chi_{J_y}(p) [1 - \cos(\mathbf{h}_J(p), \mathbf{h}_{J_y}(p))]}{\sum_p w_e^{(r)}(p) \chi_{J_y}(p)}, \tag{34}$$

$$\mathcal{L}_{\mathrm{chroma}}^{(r)} = \sum_p w_e^{(r)}(p) |\chi_J(p) - \chi_{J_y}(p)|. \tag{35}$$

Hue is weighted by target chroma as well as the edit map. Appearance RGB losses use decoded values in $[-1,1]$, with conversion to $[0,1]$ inside the OKLab transformation.

Complete decoded objective. Let $\mathcal{D}_{\mathrm{vis}}(J, J_y; w_e)$ denote the configuration-specific target discrepancy and λ_{keep} the source-preservation coefficient. The complete decoded-image loss is:

$$\mathcal{L}_{\mathrm{dec}} = \mathcal{D}_{\mathrm{vis}}(J, J_y; w_e) + \lambda_{\mathrm{keep}} \mathcal{R}_{w_b}^{(512)}(J, J_s). \tag{36}$$

The first term measures target agreement; the second penalizes changes to the source in low-change regions. Each expression is evaluated per example and then averaged over the decoded-supervision examples. The preservation term is omitted when $\lambda_{\mathrm{keep}} = 0$.

Appearance edits. Let $\mathcal{L}_{\mathrm{coarse}}^{\mathrm{app}}$ denote unweighted mean RGB SmoothL1 error between $J^{(4)}$ and $J_y^{(4)}$. The appearance target discrepancy is:

$$\begin{aligned}\mathcal{D}_{\mathrm{vis}}(J, J_y; w_e) = \mathcal{R}_{w_e}^{(512)}(J, J_y) + 0.50\mathcal{L}_{\mathrm{OKLab}}^{(512)} + 0.25\mathcal{L}_{\mathrm{hue}}^{(512)} \\ + 0.25\mathcal{L}_{\mathrm{chroma}}^{(512)} + 0.25\mathcal{L}_{\mathrm{coarse}}^{\mathrm{app}}.\end{aligned} \tag{37}$$

We set $\lambda_{\mathrm{keep}} = 0.10$. Thus, RGB and color agreement are supervised at the image resolution, with an additional coarse-scale term and the source-preservation term in equation 36.

Semantic edits. Let $\mathcal{L}_{\mathrm{appearance}}$ combine RGB and learned-feature agreement, and let $\mathcal{L}_{\mathrm{color}}$ combine regional RGB and color agreement. The semantic target discrepancy is:

$$\mathcal{D}_{\mathrm{vis}}(J, J_y; w_e) = 0.10\mathcal{L}_{\mathrm{appearance}} + 0.12\mathcal{L}_{\mathrm{color}}, \qquad \lambda_{\mathrm{keep}} = 0. \tag{38}$$

Here, $\mathcal{L}_{\mathrm{RGB}}$ averages unweighted mean RGB absolute errors at resolutions 16, 32, and 64. The feature loss $\mathcal{L}_{\mathrm{DINO}}$ measures soft local DINO patch matching, while $\mathcal{L}_{\mathrm{CLIP}}$ measures global CLIP feature agreement; both are specified below. Their combination is:

$$\mathcal{L}_{\mathrm{appearance}} = 0.10\mathcal{L}_{\mathrm{RGB}} + 0.60\mathcal{L}_{\mathrm{DINO}} + 0.30\mathcal{L}_{\mathrm{CLIP}}. \tag{39}$$

Let $\mathcal{L}_{\mathrm{coarse}}^{\mathrm{sem}}$ denote unweighted mean RGB absolute error at resolution 4. Regional RGB and color discrepancies are averaged over resolutions 16 and 64:

$$\begin{aligned}\mathcal{L}_{\mathrm{color}} = \frac{1}{2} \sum_{r \in \{16,64\}} \left[0.25\mathcal{R}_{w_e}^{(r)}(J, J_y) + 0.50\mathcal{L}_{\mathrm{OKLab}}^{(r)} + 0.25\mathcal{L}_{\mathrm{hue}}^{(r)} + 0.25\mathcal{L}_{\mathrm{chroma}}^{(r)}\right] \\ + 0.25\mathcal{L}_{\mathrm{coarse}}^{\mathrm{sem}}.\end{aligned} \tag{40}$$

For global feature agreement, let Φ_{CLIP} denote the frozen CLIP ViT-L/14 image encoder. We use

$$\mathcal{L}_{\mathrm{CLIP}} = \tfrac{1}{2}\left[1 - \cos(\Phi_{\mathrm{CLIP}}(J), \Phi_{\mathrm{CLIP}}(J_y))\right]. \tag{41}$$

DINO soft local matching. We extract patch features from frozen DINOv2-base, removing CLS and register tokens, and restore their spatial layout. For each grid side length $s \in \{4, 8\}$, adaptive average pooling produces an $s \times s$ feature map. Flattening and feature-wise ℓ_2 normalization give predicted features $\mathbf{f}_i$ and target features $\mathbf{f}_j^y$, where i and j index the pooled spatial positions. Let $\mathbf{q}_i$ and $\mathbf{q}_j$ denote their grid coordinates. With $\mathbf{1}[\cdot]$ denoting an indicator, $\|\cdot\|_\infty$ the maximum-coordinate norm, and $\mathrm{clamp}(\cdot, 0, 2)$ clipping to $[0, 2]$, the permitted-neighborhood indicator $\mathcal{A}_{ij}$ and matching cost C_{ij} are:

$$\mathcal{A}_{ij} = \mathbf{1}[\|\mathbf{q}_i - \mathbf{q}_j\|_\infty \le 1], \qquad C_{ij} = \frac{\mathrm{clamp}(1 - \mathbf{f}_i^\top \mathbf{f}_j^y, 0, 2)}{2} + 0.05 \frac{\|\mathbf{q}_i - \mathbf{q}_j\|_2^2}{s^2}. \tag{42}$$

The neighborhood contains up to 3×3 positions and is restricted to valid coordinates at the grid boundary. We suppress the grid-size index on features and matching quantities. With temperature 0.03, the logits Z_{ij} and directional matching probabilities $\pi_{ij}^{\rightarrow}$ and $\pi_{ij}^{\leftarrow}$ are:

$$Z_{ij} = \begin{cases} -C_{ij}/0.03, & \mathcal{A}_{ij} = 1, \\ -\infty, & \mathcal{A}_{ij} = 0, \end{cases} \qquad \pi_{ij}^{\rightarrow} = \mathrm{softmax}_j(Z_{ij}), \quad \pi_{ij}^{\leftarrow} = \mathrm{softmax}_i(Z_{ij}). \tag{43}$$

The rightward probabilities normalize over target positions for each predicted position; the leftward probabilities normalize over predicted positions for each target position. The expected matching loss at grid size s and its multi-scale average are:

$$\mathcal{L}_{\mathrm{DINO}}^{(s)} = \frac{1}{2}\left[\frac{1}{s^2}\sum_i \sum_j \pi_{ij}^{\rightarrow} C_{ij} + \frac{1}{s^2}\sum_j \sum_i \pi_{ij}^{\leftarrow} C_{ij}\right], \qquad \mathcal{L}_{\mathrm{DINO}} = \tfrac{1}{2}\left(\mathcal{L}_{\mathrm{DINO}}^{(4)} + \mathcal{L}_{\mathrm{DINO}}^{(8)}\right). \tag{44}$$

The two directional normalizations are independent. We do not apply Sinkhorn normalization or an extra entropy penalty. Gradients pass through both the softmax probabilities and matching costs. Finite-temperature matching permits a small positive loss even for identical, spatially varying feature maps. The matching loss is used within $\mathcal{L}_{\mathrm{appearance}}$ in equation 39; its execution settings are given in Appendix E.3.

D Architecture Details

This section details the EEG representations and source-conditioned EEG readout introduced in Secs. 4.2 and 4.3. Training settings are provided in Appendix E.

D.1 EEG Representations and Memory Projections

LaBraM-base outputs unpooled features $\mathbf{P} \in \mathbb{R}^{63\times200}$. Mean pooling followed by a linear head produces the summary $\mathbf{g} \in \mathbb{R}^{1024}$. The entries of $\mathbf{P}$ are learned encoder features, not the recorded EEG samples in $\mathbf{X}$. The Summary projection Π_g first applies non-affine LayerNorm and a $1024 \to 4096$ linear map. It then reshapes the output into four tokens and applies a second non-affine LayerNorm, producing $\mathbf{M}_g \in \mathbb{R}^{4\times1024}$. The Unpooled projection Π_p applies LayerNorm, followed by $200 \to 1024$ and $1024 \to 1024$ linear maps with GELU between them, and a final RMSNorm. This produces $\mathbf{M}_p \in \mathbb{R}^{63\times1024}$. The memory lengths in equation 9 are therefore $n_g = 4$ and $n_p = 63$. Neither projection uses the source image; source information enters the EEG readout through the queries in equation 8.

D.2 Visual Backbone and Neural Conditioning

The OmniGen2 backbone operates on 512×512 images represented by $16 \times 64 \times 64$ VAE latents. It contains 32 Transformer blocks with a hidden width of 2520. Separate projections and refiners process the source and generation latents, while three-axis rotary positional embeddings encode image identity and spatial coordinates. The backbone supplies source information through its native reference conditioning, denoted by $\mathcal{J}_t$ in equation 7.

Each EEG attention pathway has an internal width of 256 and eight heads. The query, key, value, and output projections are bias-free. Both N_b^H and N_b^M use RMSNorm, and attention dropout is

set to zero. The two pathways query their respective EEG memories from the same visual state $\mathbf{H}$ in parallel, with separate parameters and softmax normalizations. Their residual updates are added before the native feed-forward sublayer. EEG readout parameters are not shared across Transformer blocks. The scalar gates in equation 11 use the time-feature vector:

$$\mathbf{e}(t) = \left[t,\ t^2,\ \sin(\pi t),\ \cos(\pi t),\ \sin(2\pi t)\right]^\top. \tag{45}$$

Each pathway in each block has its own gate coefficients $\boldsymbol{\gamma}_b$ and bias κ_b. Appendix E.1 specifies the initialization of the gates and projections.

E Implementation Details

This section describes the trainable parameters, initialization, optimizers, and training schedules. Architecture and objective definitions are provided in Appendices D and C, respectively.

E.1 Trainable Components and Initialization

During EEG alignment, we update only the rank-8 LoRA parameters in LaBraM and the linear regression head. The original LaBraM weights and the ViT-H image encoder remain frozen. The alignment loss in equation 5 uses $\lambda_{\text{reg}} = 8$ and $\lambda_{\text{con}} = 2$. After alignment, we freeze the EEG encoder, including its LoRA adapters, and the regression head.

During editor training, we update the Summary and Unpooled memory projections, both EEG attention pathways and their normalization parameters, the time gates, and the backbone LoRA parameters. The VAE and the original backbone weights remain frozen.

The attention output matrices $\mathbf{W}_b^O$ are initialized from a normal distribution with standard deviation 10^{-4}. Gate coefficients $\boldsymbol{\gamma}_b$ are initialized to zero, and gate biases κ_b to -2. Backbone LoRA uses rank 32 and dropout 0.05. Its scaling parameter is set to 32 for semantic editing.

E.2 Optimization and Training Schedules

We train separate editors for appearance and semantic editing, using the same architecture and frozen EEG encoder. Both configurations use the objective in equation 15. Configurations use weight decay 0.01, optimizer epsilon 10^{-8}, and global gradient-norm clipping at 1. Semantic training excludes one-dimensional parameters and normalization parameters from weight decay. The base learning rates are 2×10^{-5} for Summary attention, 3×10^{-5} for the Summary memory projection, 10^{-5} for Unpooled attention and its memory projection, and 10^{-6} for backbone LoRA. Table 4 lists the remaining settings.

Appearance training accumulates gradients over four groups of three examples per optimizer step: two groups from THINGS-EEG2, one from Alljoined-1.6M, and one from EEG-ImageNet. Participants rotate within each resource. Seventy percent of groups contain distinct images from a single concept; the remaining groups contain images from different concepts. The 2400-step schedule begins with 200 warmup steps, followed by cosine decay to 15% of the base learning rates. We report results from the checkpoint at step 2400. Semantic training begins with 100 warmup steps, followed by cosine decay to 15% of the base learning rates at step 1200. The rates then increase smoothly to 30% during steps 1201–1300 and decay by a cosine schedule to 15% at step 3000.

E.3 Auxiliary Supervision

We recompute the EEG-disabled reference in equation 14 from the current editor without gradient tracking. For semantic training, we temporarily put the editor in evaluation mode for both paired forward passes, while retaining gradients through the EEG-conditioned branch. For appearance training, we keep LoRA dropout active and reuse each LoRA module's dropout mask across the two passes.

Appearance decoded supervision reuses the $t = 0$ endpoint estimate $\hat{\mathbf{z}}$. Semantic decoded supervision uses independent auxiliary noise and samples time from the distribution in Table 4. We reconstruct the target and source references from cached VAE latents without gradient tracking. For

Table 4: **Editor training configurations.** Batch sizes count examples per optimizer step. Each objective is averaged over its own examples. The step index is k; δ_0 is a point mass at zero, $\mathcal{U}$ denotes a uniform distribution, $\mathbf{1}[\cdot]$ is an indicator, and mod denotes the integer remainder. The decoded coefficient multiplies the complete $\mathcal{L}_{\text{dec}}$.

Setting	Appearance	Semantic
Editing steps	2400	3000
Optimizer	AdamW8bit	Fused AdamW
Adam moment coefficients	$(0.9, 0.95)$	$(0.9, 0.999)$
Flow batch	12; four groups of three	72
Endpoint / difference batch	All 12 examples	Stratified subset of 24
Decoded-supervision batch	All 12 on active steps	Subset of 16; microbatch 4
Flow time distribution	$0.1\delta_0 + 0.9\mathcal{U}(0, 1)$	$\mathcal{U}(0, 1)$
λ_{end}	0.30	0.10
$\lambda_{\Delta}(k)$	0.20	$0.02 \min(1, k/200)$
Difference threshold β	1.0	1.0
Decoded-supervision time	$\tau = 0$	$\tau \sim \mathcal{U}(0.1, 0.6)$
$\lambda_{\text{dec}}(k)$	$0.48\,\mathbf{1}[k \bmod 4 = 0]$	$\mathbf{1}[k \bmod 2 = 0]$

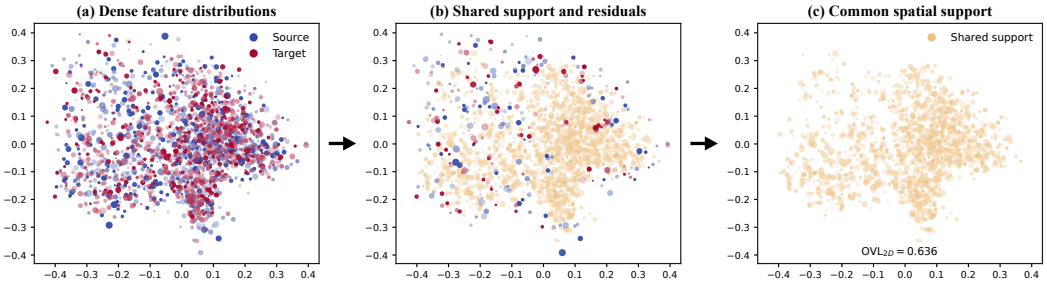

Figure 7: **Source–target feature overlap.** Frozen ViT-H patch features are independently ℓ_2-normalized and jointly projected to 2D with PCA. **(a)** Source and target distributions. **(b)** Shared-support cells occupied by both distributions. **(c)** Shared support alone, with $\text{OVL}_{2D} = 0.636$.

the predicted image, gradients pass through the frozen decoder and any frozen visual encoders used by the loss. The decoded-image feature path runs in FP32.

Appearance decoded supervision is applied every fourth step, starting at step 4. Its multiplier is $0.12 \times 4 = 0.48$ on active steps, giving a nominal frequency-averaged coefficient of 0.12. For semantic editing, the feature and color losses are applied every second step to the same 16 examples and share the same one-step estimate. Their coefficients on active steps are 0.10 and 0.12, respectively, with no frequency compensation. The corresponding nominal frequency-averaged coefficients are therefore 0.05 and 0.06. The DINO coefficient is $0.10 \times 0.60 = 0.06$ on active steps, or 0.03 after accounting for frequency. Neither configuration uses a separate auxiliary-gradient cap or a mismatched-EEG ranking loss.

F Why evaluate change regions?

A feature-space view provides complementary intuition for change-region evaluation. We jointly project normalized ViT-H patch features from source and target images into 2D and divide the space into a 40×40 grid. Their overlap is measured by:

$$\text{OVL}_{2D} = \sum_b \min\left(\frac{N_s(b)}{N_s}, \frac{N_t(b)}{N_t}\right). \tag{46}$$

Figure 7 gives $\text{OVL}_{2D} = 0.636$, showing substantial shared feature support despite source–target differences. This helps explain why whole-image metrics can be dominated by already shared content, whereas change-region metrics focus on the visual differences that require correction. The

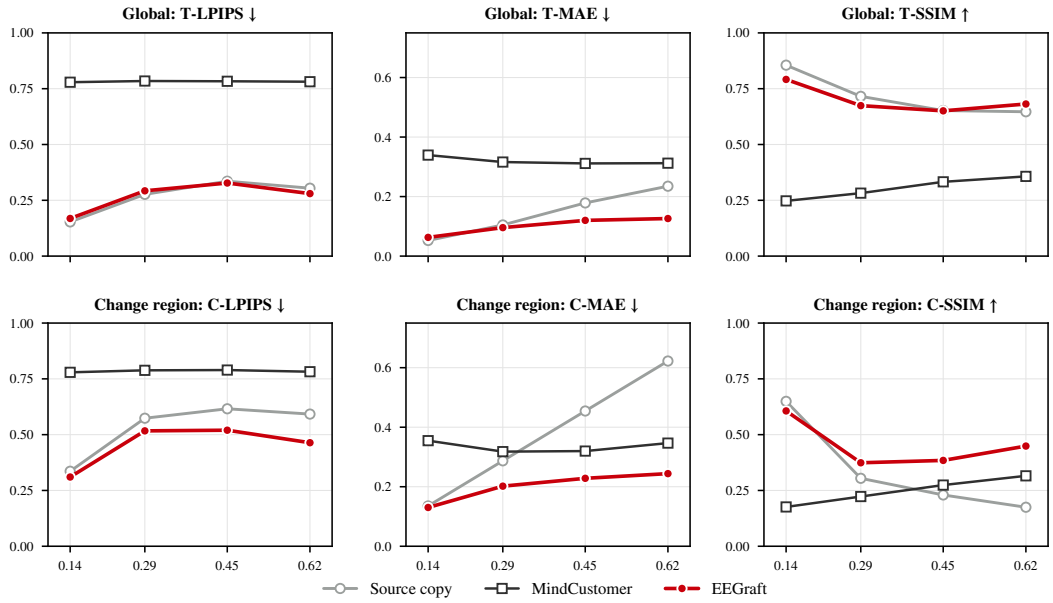

Figure 8: **Whole-image and change-region evaluation.** The top row reports T-LPIPS, T-MAE, and T-SSIM; the bottom row reports their change-region counterparts. Gray, red, and black curves denote source copying, EEGraft, and MindCustomer, respectively. Regional evaluation exposes the uncorrected source–target differences that whole-image agreement obscures.

overlap is descriptive of the 2D feature projection and does not represent the fraction of unchanged image area.

Whole-image agreement does not by itself establish that an edit has been completed. When much of the source already agrees with the target, copying the source retains that agreement while leaving the required correction unresolved. For RGB error, let T-MAE_i denote the whole-image mean absolute error for example i. A region contributes to this mean in proportion to its area. With M_i and K_i denoting the change and low-change masks, $|M_i|$ and $|K_i|$ their pixel counts, and MAE_{K_i} the mean RGB error in K_i, this relation is:

$$\text{T-MAE}_i = \frac{|M_i|\,\text{C-MAE}_i(\hat{I}_i) + |K_i|\,\text{MAE}_{K_i}(\hat{I}_i, I_{y,i})}{|M_i| + |K_i|}. \tag{47}$$

Thus, low error over a large low-change region dilutes an unresolved editing error in a smaller change region. C-MAE instead averages within the region that needs correction.

Figure 8 illustrates the consequence across the plotted source–target difference levels. Source copying remains competitive under whole-image measures because the retained scene contributes to the score. The regional pixel and structural measures make the uncorrected differences explicit: the source-copy reference deteriorates as those differences increase, while EEGraft recovers more of the target within the change region. Regional evaluation therefore tests completion of the edit rather than agreement inherited from the source.

C-LPIPS, C-MAE, and C-SSIM provide complementary evidence of perceptual, absolute RGB, and local structural recovery. They do not replace the whole-image measures, which assess the completed image, or R-Edit, which additionally charges for increased target error in low-change regions. All regional comparisons use the same output-independent masks. The mask construction and metric implementations are given in Appendix B.

G Architecture Ablations

EEG correspondence. We first examine whether matching the EEG recording to the target improves editing accuracy. Figure 9compare correctly paired EEG with two mismatch controls and

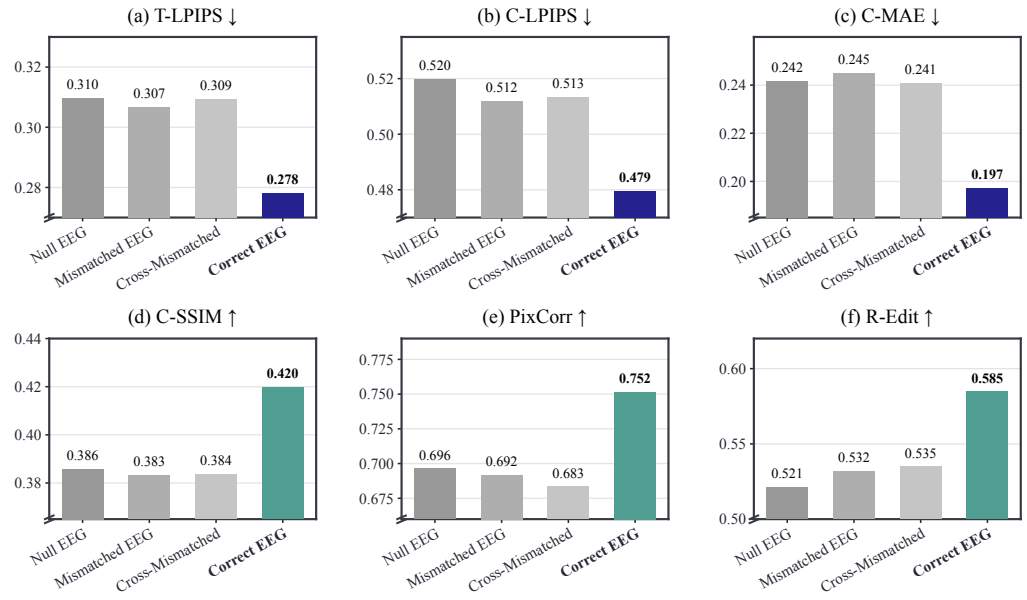

Figure 9: **Effect of mismatched EEG conditioning.** We compare four inference settings: EEG conditioning disabled (*Null EEG*), mismatched EEG from the same subject (*Mismatch*), mismatched EEG from a different subject (*Cross-Mismatch*), and correctly paired EEG (*Correct EEG*). Correct EEG achieves the best performance across all six metrics. Both mismatch controls remain below the correctly paired condition, while their relative differences are small and vary across metrics.

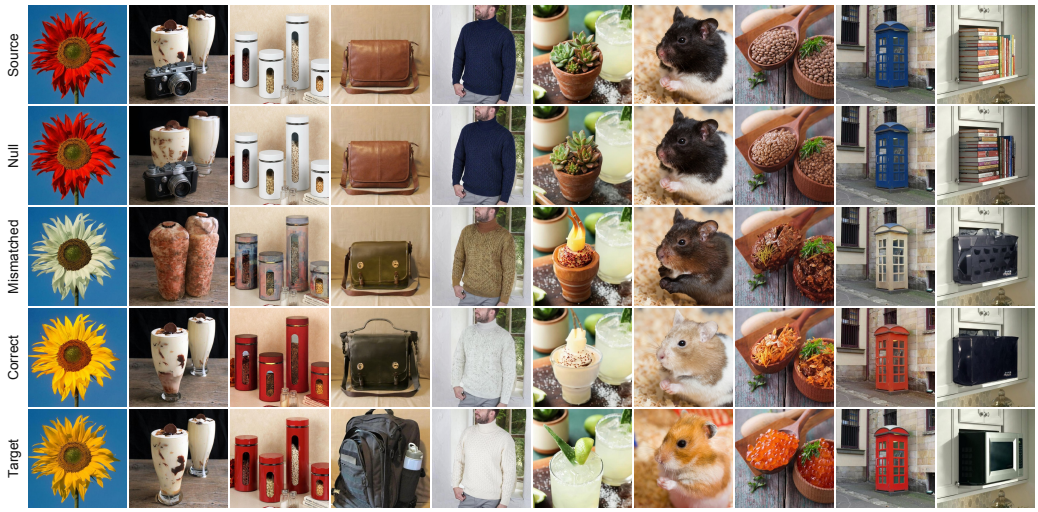

Figure 10: **Effect of EEG conditioning.** With the trained editor and source image fixed, we compare three conditions: *Null EEG*, which disables both EEG readout pathways; *Mismatched EEG*, which uses a recording of another stimulus from the same participant; and *Correct EEG*, which uses the target-paired recording.

disabled EEG conditioning. (1) *Mismatch* replaces the target-associated recording with EEG from another stimulus of the same participant, preserving participant identity while changing the stimulus correspondence. (2) *Cross-Mismatch* further uses EEG from a different participant and a different concept. Correct EEG gives the best performance across all six metrics in both editing subsets. For the aggregate evaluation, Correct EEG reduces C-LPIPS and C-MAE by approximately 6.4% and 19.6% relative to Mismatched EEG, and by approximately 6.6% and 18.3% relative to Cross-Mismatched EEG. Both comparisons retain active EEG conditioning and measure the advantage of correct target pairing.

Interpreting the controls. Relative to Null EEG, Mismatched EEG improves aggregate C-LPIPS from 0.520 to 0.512 but worsens C-MAE from 0.242 to 0.245. Natural-image EEG contains common stimulus-locked responses alongside stimulus-dependent variation; Alljoined-1.6M, one of our source datasets, reports both (Xu et al., 2025). By shared response structure, we mean patterns associated with visual stimulation across recordings, not information identifying the particular target. Correct and Mismatched EEG pass through the same aligned encoder, and their memories are read in the context of the source and current generation state. A plausible explanation is that shared components in these representations support source-conditioned updates even when target pairing is broken. The controls establish the additional benefit of correct pairing but do not isolate the effect of these shared components. These controls do not separate the contributions of the learned readout, source information, and visual priors. Mismatch and Cross-Mismatch are alternative unpaired-input controls, not ordered levels of corruption. Their ordering varies across metrics and editing subsets, while Correct EEG outperforms both. Null EEG still produces an image through the generator; it is distinct from the source-copy reference used by R-Edit.

Training without EEG. We also compare against a source-only editor trained and evaluated without EEG (Table 5). Unlike the inference-time Null EEG ablation, this baseline is optimized without an EEG input. Across 1,600 appearance and 1,200 semantic editing samples, EEGraft reduces the weighted-average C-LPIPS from 0.632 to 0.479 and improves all six reported metrics.

Table 5: Comparison with a source-only baseline trained and evaluated without EEG.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
Trained w/o EEG	0.389	0.632	0.317	0.329	0.592	0.480
EEGraft (Ours)	**0.278** $^{-0.111}$	**0.479** $^{-0.153}$	**0.197** $^{-0.120}$	**0.420** $^{+0.091}$	**0.752** $^{+0.160}$	**0.585** $^{+0.105}$

Summary capacity control. Removing the Unpooled pathway also reduces the conditioning capacity available to the editor. To test whether the degradation of the Summary-only variant is simply caused by this reduction, we construct an additional Summary-large variant. Summary-only retains the original Summary design, whereas Summary-large still uses only the Summary pathway but expands its output representation to match the EEG conditioning-token budget of the dual-readout setting. Relative to Summary-only, Summary-large increases T-LPIPS and C-LPIPS by 0.002 and 0.003, respectively, leaves C-MAE unchanged, increases C-SSIM by 0.001, and decreases PixCorr by 0.002. Thus, increasing the capacity of the Summary pathway does not recover the performance lost when the Unpooled pathway is removed. This control suggests that the benefit of the dual readout is not explained simply by the reduced conditioning capacity of Summary-only, but instead depends on retaining access to EEG features before pooling.

H Inference-Time Behavior of Neural Grafting

We examine the residual updates contributed by the Summary and Unpooled pathways during inference. Update magnitude is measured by the root mean square (RMS) of the residual elements; cosine similarity measures the agreement of their directions. Figure 11(a) divides each pathway's update magnitude by that pathway's mean. The resulting co-variation shows that both pathways respond across samples and Transformer layers, but the normalization does not imply equal absolute strength. Panel (b) provides that comparison: the Summary update is 3.54 times larger on average than the Unpooled update. The mean cosine similarity of the two residuals is 0.985.

These measurements characterize an asymmetric contribution rather than two unrelated update directions. The Summary pathway contributes the dominant update, and the Unpooled pathway adds a smaller update in a closely aligned direction. This behavior complements the ablations in Appendix G: a smaller residual is not redundant, because removing access to unpooled features reduces the full model's overall editing accuracy, and increasing summary capacity does not recover that benefit.

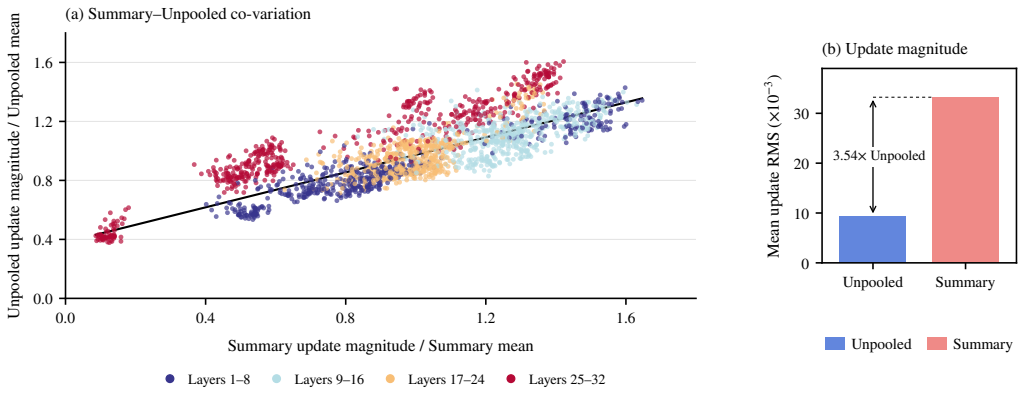

Figure 11: **Inference-time behavior of the dual EEG readout.** (a) Summary and Unpooled update magnitudes co-vary across samples and Transformer layers after branch-wise normalization; colors indicate layer depth. (b) In absolute scale, the Summary residual is $3.54\times$ larger on average than the Unpooled residual.

I SUBJECT-WISE RESULTS

Tables 6–15 provide the participant-level breakdown for Sub-01 through Sub-10 using the main evaluation protocol. EEGraft consistently improves both whole-image agreement and change-region fidelity across these participants, complementing the benchmark-level results.

Table 16 reports the EEG encoder's Top-1 and Top-5 image retrieval accuracy for the first ten subjects in our benchmark, using a 200-image candidate set. We provide both per-subject results and the macro-average across subjects.

Table 6: Quantitative results for Sub-01.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.900	0.896	0.352	0.216	0.062	0.279
MindPainter	0.683	0.764	0.297	0.196	0.281	0.343
Brain-Supervised	0.394	0.613	0.339	0.307	0.474	0.387
PVS-EEG	0.379	0.604	0.326	0.304	0.523	0.379
MindArt	0.434	0.639	0.350	0.272	0.481	0.374
LoongX	0.946	1.071	0.287	0.219	0.172	0.336
EEGraft	**0.268** -0.111	**0.456** -0.148	**0.174** -0.113	**0.472** +0.165	**0.772** +0.249	**0.609** +0.222

Table 7: Quantitative results for Sub-02.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.903	0.894	0.398	0.195	0.041	0.264
MindPainter	0.677	0.750	0.290	0.201	0.285	0.347
Brain-Supervised	0.379	0.605	0.333	0.325	0.471	0.392
PVS-EEG	0.361	0.599	0.328	0.321	0.519	0.383
MindArt	0.424	0.645	0.350	0.285	0.489	0.377
LoongX	0.961	1.063	0.299	0.214	0.169	0.326
EEGraft	**0.265** -0.096	**0.452** -0.147	**0.177** -0.113	**0.472** +0.147	**0.786** +0.267	**0.602** +0.210

Table 8: Quantitative results for Sub-03.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.897	0.889	0.376	0.183	0.061	0.277
MindPainter	0.636	0.741	0.290	0.191	0.309	0.365
Brain-Supervised	0.400	0.614	0.337	0.282	0.479	0.392
PVS-EEG	0.386	0.609	0.332	0.276	0.531	0.380
MindArt	0.454	0.645	0.364	0.246	0.476	0.372
LoongX	0.959	1.066	0.283	0.192	0.172	0.345
EEGraft	**0.279** -0.107	**0.457** -0.152	**0.191** -0.092	**0.431** +0.149	**0.759** +0.228	**0.595** +0.203

Table 9: Quantitative results for Sub-04.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.898	0.889	0.369	0.198	0.052	0.263
MindPainter	0.683	0.777	0.301	0.190	0.289	0.330
Brain-Supervised	0.366	0.600	0.322	0.350	0.510	0.387
PVS-EEG	0.355	0.598	0.318	0.339	0.558	0.375
MindArt	0.418	0.641	0.340	0.302	0.516	0.368
LoongX	0.956	1.081	0.285	0.207	0.160	0.323
EEGraft	**0.258** -0.097	**0.444** -0.154	**0.186** -0.099	**0.487** +0.137	**0.789** +0.231	**0.580** +0.193

J Performance across Edit Granularities

Visual edits in NEVE span different levels of change granularity. We organize the benchmark into two broad editing regimes: appearance edits and semantic edits. Appearance edits operate mainly on lower-level visual properties while largely preserving semantic identity and spatial structure. Semantic edits operate at a higher level of visual abstraction, modifying the semantic content and potentially inducing broader structural changes.

K Inference Settings

Inference uses 28 Euler steps to integrate the learned velocity field from standard Gaussian noise. Let N be the number of steps and $0 = t_0 < t_1 < \cdots < t_N = 1$ the sampling grid, with j indexing an update. The numerical step is:

$$\tilde{\mathbf{x}}_{t_{j+1}} = \tilde{\mathbf{x}}_{t_j} + (t_{j+1} - t_j)\, \mathbf{v}_\theta(\tilde{\mathbf{x}}_{t_j}, t_j; \mathbf{z}_s, \mathbf{X}). \tag{48}$$

We set $N = 28$, encode the source once, and compute both EEG memories before the updates. These conditions remain fixed while the generation state changes. The final latent is decoded once to obtain $\hat{I}$. Target images, construction prompts, and training-derived spatial weights are not inference inputs.

L Baseline Settings and Interpretation

Comparison scope. The baseline results evaluate application to NEVE's target-perception EEG task, using the common metrics and aggregation in Sec. 3.2. The original methods differ in neural measurements and editing formulations, as summarized below.

Neural observations. fMRI-based methods such as MindPainter condition generation on voxel-wise hemodynamic measurements, whereas EEG provides electrical measurements over scalp electrodes and time (Yu et al., 2025b; Song et al., 2024). Using an EEG encoder adapts the input interface but does not reproduce voxelwise fMRI observations. LoongX uses a diffusion transformer

Table 10: Quantitative results for Sub-05.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.903	0.883	0.358	0.183	0.062	0.286
MindPainter	0.662	0.760	0.300	0.177	0.295	0.366
Brain-Supervised	0.435	0.651	0.347	0.263	0.425	0.390
PVS-EEG	0.417	0.645	0.338	0.253	0.469	0.382
MindArt	0.489	0.693	0.372	0.234	0.440	0.373
LoongX	0.971	1.078	0.299	0.189	0.139	0.347
EEGraft	**0.295**$^{-0.122}$	**0.504**$^{-0.141}$	**0.205**$^{-0.094}$	**0.384**$^{+0.121}$	**0.704**$^{+0.235}$	**0.588**$^{+0.198}$

Table 11: Quantitative results for Sub-06.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.888	0.877	0.368	0.185	0.056	0.280
MindPainter	0.690	0.764	0.302	0.172	0.273	0.350
Brain-Supervised	0.436	0.644	0.349	0.282	0.461	0.385
PVS-EEG	0.415	0.639	0.341	0.276	0.508	0.378
MindArt	0.479	0.675	0.367	0.243	0.469	0.373
LoongX	0.965	1.069	0.286	0.202	0.152	0.344
EEGraft	**0.288**$^{-0.127}$	**0.472**$^{-0.167}$	**0.186**$^{-0.100}$	**0.436**$^{+0.154}$	**0.752**$^{+0.244}$	**0.599**$^{+0.214}$

with EEG, fNIRS, PPG, and head-motion signals collected during instruction-based editing (Zhou et al., 2026). NEVE instead pairs EEG with a viewed target, making the required edit dependent on interpreting that target information relative to the source.

Editing formulations. In its original setting, Brain-Supervised learns GAN-latent semantic directions from EEG responses to attended features in generated faces (Davis et al., 2022). The original MindPainter task specifies a brain-conditioned masked editing region (Yu et al., 2025b), while MindCustomer blends brain and image contexts for generation (Yu et al., 2025a). NEVE evaluates these mechanisms by recovery of the particular target in the change region and preservation of unrelated source content. This joint criterion distinguishes completing the required edit from changing semantic content or producing a plausible image.

Complementary controls. The source-only editor trained and evaluated without EEG yields aggregate C-LPIPS of 0.632, compared with 0.479 for EEGraft (Table 5). Correct–Mismatched comparisons additionally keep the trained editor fixed while changing recording–target correspondence (Sec. 5.4). These controls assess the contribution of EEG without relying on the output quality of an external baseline.

M Failure Cases: Granularity of Recoverable EEG Guidance

Figure 12 shows representative cases in which the remaining error is concentrated at a finer visual granularity than the dominant edit itself. For example, in the tote-bag case, EEGraft recovers the main appearance change from a dark bag to a light one, but not the exact repeated motif on the target. In the floral example, the output recovers the intended object-level transition to a bouquet while differing in the precise composition and colors of individual flowers. A similar pattern appears in the wind-chime and stapler examples: the output follows the target's broad color change, but does not reproduce every part-level color or the exact target shade. Other examples exhibit analogous discrepancies in local labels, texture, or within-object appearance.

We interpret these cases as a granularity gap between the visual correction required by the target and the neural evidence that is reliably recoverable under the current EEG acquisition and representation. The target recording can provide sufficient evidence for broad semantic or appearance attributes

Table 12: Quantitative results for Sub-07.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.888	0.878	0.378	0.185	0.029	0.273
MindPainter	0.650	0.732	0.293	0.193	0.287	0.352
Brain-Supervised	0.371	0.601	0.334	0.305	0.472	0.391
PVS-EEG	0.353	0.594	0.324	0.302	0.522	0.383
MindArt	0.422	0.634	0.358	0.272	0.478	0.370
LoongX	0.935	1.045	0.293	0.203	0.184	0.339
EEGraft	**0.251** $^{-0.102}$	**0.448** $^{-0.146}$	**0.178** $^{-0.115}$	**0.454** $^{+0.149}$	**0.781** $^{+0.259}$	**0.598** $^{+0.207}$

Table 13: Quantitative results for Sub-08.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.892	0.873	0.362	0.178	0.036	0.253
MindPainter	0.658	0.741	0.285	0.187	0.308	0.334
Brain-Supervised	0.392	0.608	0.308	0.312	0.492	0.375
PVS-EEG	0.377	0.603	0.296	0.307	0.539	0.368
MindArt	0.448	0.648	0.325	0.267	0.507	0.363
LoongX	0.945	1.066	0.284	0.191	0.176	0.317
EEGraft	**0.269** $^{-0.108}$	**0.458** $^{-0.145}$	**0.179** $^{-0.105}$	**0.439** $^{+0.127}$	**0.775** $^{+0.236}$	**0.570** $^{+0.195}$

while leaving finer details less well determined. Exact texture patterns, local markings, per-part color assignments, and small within-color differences can be weak, spatially mixed, or ambiguous in the recoverable EEG representation. When such details are not strongly specified by the neural condition, the pretrained editor can still follow the broader target cue while resolving the remaining appearance using its visual prior.

Importantly, these examples do not imply that fine visual information is absent from neural activity. Rather, they show that such information is not always resolved reliably by the present scalp-EEG recordings, preprocessing, visual alignment, and editor conditioning. This distinction is also reflected qualitatively in Figure 12: the outputs are often consistent with the target at the level of the main correction, even when they miss the exact target-specific realization of that correction. Improving the recoverable granularity of neural guidance – for example through richer neural representations, stronger aggregation across repeated trials, or neural measurements with greater spatial specificity – is an important direction for future work.

N Additional Qualitative Results

Table 14: Quantitative results for Sub-09.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.894	0.864	0.358	0.200	0.032	0.280
MindPainter	0.690	0.772	0.306	0.170	0.278	0.354
Brain-Supervised	0.412	0.641	0.349	0.288	0.443	0.389
PVS-EEG	0.394	0.632	0.338	0.284	0.490	0.381
MindArt	0.457	0.673	0.364	0.252	0.448	0.377
LoongX	0.953	1.084	0.307	0.204	0.143	0.341
EEGraft	**0.277** -0.117	**0.477** -0.155	**0.189** -0.117	**0.442** +0.154	**0.744** +0.254	**0.600** +0.211

Table 15: Quantitative results for Sub-10.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
MindCustomer	0.905	0.877	0.382	0.177	0.048	0.259
MindPainter	0.705	0.770	0.307	0.174	0.273	0.333
Brain-Supervised	0.360	0.577	0.312	0.330	0.506	0.395
PVS-EEG	0.349	0.575	0.308	0.319	0.552	0.381
MindArt	0.413	0.615	0.330	0.289	0.522	0.377
LoongX	0.959	1.076	0.282	0.198	0.185	0.327
EEGraft	**0.239** -0.110	**0.444** -0.131	**0.187** -0.095	**0.462** +0.132	**0.798** +0.246	**0.588** +0.193

Table 16: Top-k retrieval accuracy of the EEG encoder for the first ten subjects in our benchmark. Accuracy (%) is evaluated among 200 candidate images, with chance levels of 0.5% for Top-1 and 2.5% for Top-5.

Subject	Top-1↑	Top-5↑	Subject	Top-1↑	Top-5↑
Sub-01	25.00	53.00	Sub-06	32.50	60.50
Sub-02	17.50	52.00	Sub-07	22.50	51.00
Sub-03	26.50	58.50	Sub-08	37.00	66.50
Sub-04	30.50	60.50	Sub-09	12.00	38.50
Sub-05	23.00	53.50	Sub-10	31.50	65.00
	Macro-average across 10 subjects:		Top-1 = 25.80%	Top-5 = 55.90%	



Figure 12: **Representative failure cases.** Each column shows the source, target, and EEGraft output. Red boxes highlight target-specific details that are not fully recovered. Across these examples, the output often captures the dominant semantic or appearance correction, such as object category, broad color family, or overall structure, while differing from the target in exact motifs, local texture, part-level color, or subtle hue.

Table 17: **Performance across edit granularities.** We report results separately for edits dominated by lower-level visual changes and edits involving higher-level visual transformations. Best results are shown in **bold**.

Method	T-LPIPS ↓	C-LPIPS ↓	C-MAE ↓	C-SSIM ↑	PixCorr ↑	R-Edit ↑
(a) appearance edits on lower-level visual properties						
MindCustomer	0.926	0.907	0.405	0.157	0.030	0.241
LoongX	0.978	1.105	0.315	0.247	0.153	0.303
MindPainter	0.620	0.717	0.298	0.202	0.360	0.332
Brain-Supervised	0.252	0.508	0.355	0.371	0.545	0.393
MindArt	0.288	0.525	0.356	0.333	0.536	0.367
PVS-EEG	0.241	0.502	0.345	0.364	0.605	0.378
EEGraft (Ours)	**0.190** -0.051	**0.347** -0.155	**0.169** -0.129	**0.544** +0.173	**0.848** +0.243	**0.615** +0.222
(b) semantic edits on higher-level visual abstractions						
MindCustomer	0.859	0.847	0.319	0.234	0.068	0.321
LoongX	0.918	1.023	0.263	0.133	0.183	0.383
MindPainter	0.740	0.796	0.291	0.168	0.199	0.373
Brain-Supervised	0.633	0.783	0.317	0.184	0.332	0.377
MindArt	0.693	0.846	0.361	0.143	0.374	0.376
PVS-EEG	0.606	0.777	0.309	0.181	0.365	0.376
EEGraft (Ours)	**0.396** -0.210	**0.656** -0.121	**0.235** -0.028	**0.254** +0.020	**0.623** +0.249	**0.544** +0.161



Figure 13: **Appearance editing results (1/3).**

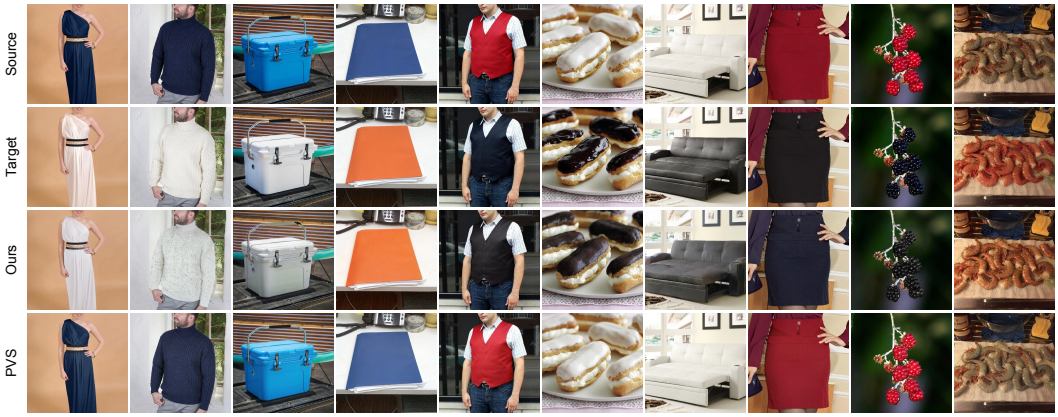

Figure 14: **Appearance editing results (2/3).**

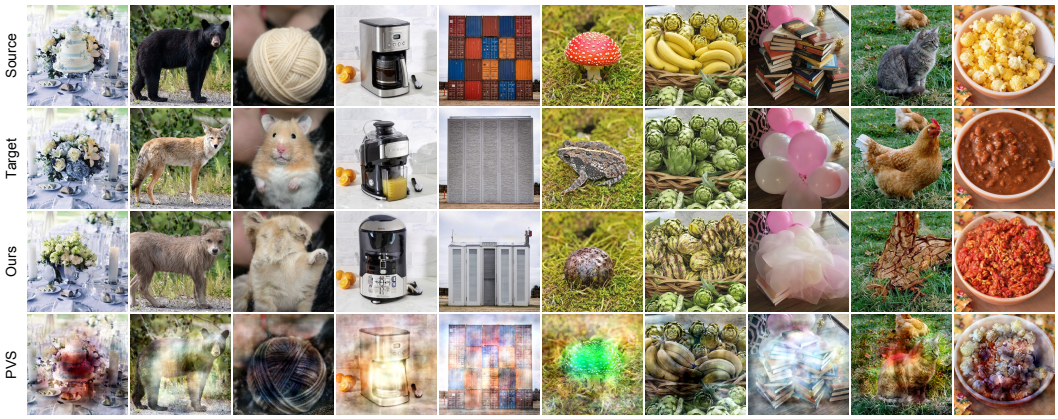

Figure 15: **Semantic editing results (1/3).**

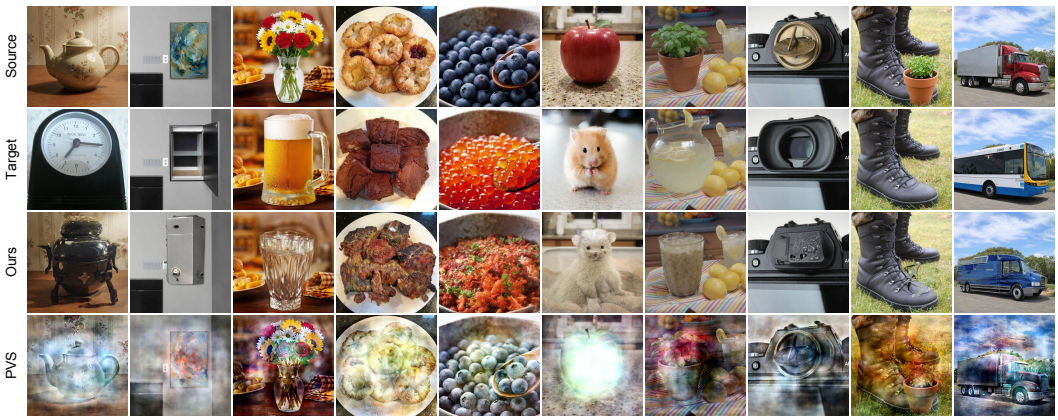

Figure 16: **Semantic editing results (2/3).**