ZEROMAG: ZERO-SHOT MULTIMODAL ADAPTER GENERATION FOR PLUG-AND-PLAY EEG FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

EEG foundation models (EFMs) capture reusable knowledge from large-scale EEG data, while many EEG recordings also include companion physiological signals that provide complementary information beyond the EEG-only interface. The challenge is to preserve this pretrained knowledge while extending the EFM to heterogeneous multimodal recordings through an adaptation inferred from unlabeled target data. We introduce ZeroMAG, a zero-shot multimodal adapter generation framework that extends a frozen EEG encoder and prediction head using unlabeled target recordings, without target labels or target-side optimization. The target datasets are held out from all model training and selection in the ZeroMAG pipeline. ZeroMAG organizes companion modalities around a configuration-invariant adapter, constructs a modality–subject–task condition from unlabeled recordings and task context, and generates adapter weights in a function-constrained latent space learned from source adapters. Across six held-out target datasets and three EFM backbones, ZeroMAG improves balanced accuracy by 7.22 percentage points over EEG-only inference and 4.89 points over direct weight regression, while coming within 0.50 points of supervised multimodal adaptation on average. Ablations further show that removing functional supervision from either representation learning or conditional generation degrades generated-adapter performance, confirming the contribution of both components.

1 INTRODUCTION

Electroencephalography (EEG) is a non-invasive and widely used modality across clinical, cognitive, and brain–computer interface applications (Craik et al., 2019; Wolpaw et al., 2002; Lee et al., 2025). Its large and growing body of mostly unlabeled recordings has enabled EEG foundation models (EFMs) that learn reusable representations across downstream tasks (Kuruppu et al., 2025; Jiang et al., 2024; Wang et al., 2025; 2024a; Ma et al., 2026; Zhou et al., 2025).

EEG is increasingly recorded alongside companion signals that capture complementary physiological information, such as eye movements, muscle activity, and cardiac dynamics (Fig. 1(a)). Between 2016 and 2024, at least 124 deep learning studies combined EEG with EOG, EMG, ECG, or GSR (Lee et al., 2025), and concurrent EEG–fNIRS studies increased markedly after 2016 (Yeung & Chu, 2022); yet the number and type of available signals vary widely across datasets (Fig. 1(b)).

A common approach to exploiting these signals is multimodal physiological or general time-series foundation models (Thapa et al., 2024; Zhang et al., 2024; Fu et al., 2025). However, this often means training a new multimodal backbone instead of reusing the strong EEG representations already learned by existing EFMs, while also fixing the supported modality set in advance. Adapting an existing EFM is cheaper, but new recordings often arrive unlabeled because annotation requires expert review, making supervised adaptation for each target dataset impractical. Label-free adaptation can avoid target labels, but typically still requires target-side optimization (Liang et al., 2020; Wang et al., 2021). This raises the central question of this work:

Can multimodal capabilities be added to an existing EFM without target-side training?

Answering this question requires addressing three challenges.

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.

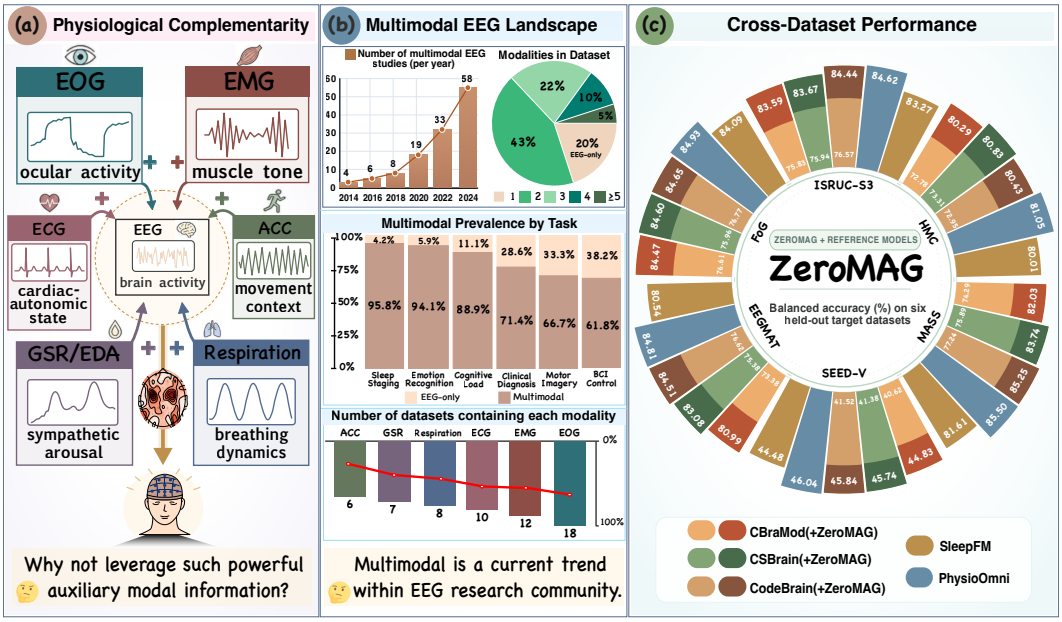

Figure 1: Multimodal EEG landscape and representative results. (a) Physiological information captured by common companion signals. (b) Publication trends, task-wise multimodal prevalence, and companion modality availability, based on prior surveys (Lee et al., 2025; Yeung & Chu, 2022). (c) Balanced accuracy on six held-out target datasets, comparing three EFM backbones with and without ZeroMAG against SleepFM and PhysioOmni.

C1: Heterogeneous modality configurations. EEG datasets differ in their companion signals: sleep recordings often include EOG and EMG, whereas emotion-recognition datasets may include eye movement. An adapter must accommodate these differences while allowing the available signals to contribute complementary information. The challenge is to accommodate arbitrary configurations across datasets within a single unified framework, without per-dataset redesign.

C2: Adaptation needs are target-dependent but labels are often unavailable. Even with the same companion signals, the appropriate adaptation differs across datasets and prediction tasks, so it must be determined for each target. However, target labels are usually unavailable because annotation is costly and requires expert review. The challenge is therefore to infer the target-specific adaptation from unlabeled recordings together with the task context.

C3: Parameter reconstruction does not ensure functional consistency. The generator is trained to reproduce adapters fitted on the source datasets. Two adapters with similar weights can still make different predictions, because the output is much more sensitive to some parameters than to others. A generator trained only to minimize parameter error may therefore produce adapters that are close to the targets in weight space but behave differently. The challenge is to make the generated adapters agree with the source adapters in their predictions.

To address these challenges, we propose ZeroMAG, which treats multimodal extension as personalized parameter generation: given unlabeled target recordings and task context, it directly generates a target-specific adapter while keeping the EFM fixed. A **configuration-invariant adapter** organizes reusable modality-specific blocks around a fixed EEG anchor (C1); a **modality–subject–task condition** captures target-specific information without dataset identifiers (C2); and **function-constrained generation** maps this condition to adapter parameters through a block-aligned latent space learned from fitted source adapters (C3).

Across six held-out target datasets and three EFM backbones, ZeroMAG improves balanced accuracy by 7.22 and 4.89 percentage points over EEG-only inference and direct weight regression, respectively, while coming within 0.50 points of supervised multimodal adaptation on average, and remains competitive with multimodal foundation models that require multimodal pretraining (Fig. 1(c)).

2 PRELIMINARIES

2.1 EEG FOUNDATION MODEL

Let $\mathbf{X}^{\mathrm{EEG}}$ denote an EEG sample. An EEG foundation model consists of an encoder $f(\cdot)$ and a prediction head $h(\cdot)$, so that

$$\mathbf{Z} = f(\mathbf{X}^{\mathrm{EEG}}), \qquad \mathbf{L} = h(\mathbf{Z}) \in \mathbb{R}^K, \tag{1}$$

where $\mathbf{Z}$ is the EEG representation, $\mathbf{L}$ the corresponding prediction logits, and K the number of classes. We directly load the EFM encoder checkpoint and pair it with the compatible prediction head trained on source data for the supported task. Both f and h remain frozen during source adapter fitting, both generator-training stages, and target inference.

2.2 ADAPTER-BASED MULTIMODAL INTEGRATION

Let $\mathcal{A}$ denote the available companion physiological modalities, with corresponding signals $\mathbf{X}^{\mathcal{A}} = \{\mathbf{X}^m\}_{m\in\mathcal{A}}$. To incorporate these signals without modifying the EFM, an adapter $a(\cdot;\omega)$ is inserted between $f(\cdot)$ and $h(\cdot)$:

$$\mathbf{Z}^{\mathcal{A}} = a\left(\mathbf{Z}, \mathbf{X}^{\mathcal{A}}; \omega\right), \qquad \mathbf{L}^{\mathcal{A}} = h\left(\mathbf{Z}^{\mathcal{A}}\right), \tag{2}$$

where $\omega \in \mathbb{R}^{P(\mathcal{A})}$ denotes the adapter parameters, $\mathbf{Z}^{\mathcal{A}}$ the multimodal representation, and $\mathbf{L}^{\mathcal{A}}$ the prediction logits returned by the frozen head $h(\cdot)$.

2.3 PROBLEM FORMULATION

Given a frozen EFM, we aim to adapt it to downstream datasets with a single framework. These datasets differ in which modalities they record and in the tasks they target, so each one requires a different adaptation. We formulate this problem through three objectives.

C1: Define a unified adapter architecture. Let $\mathcal{M}$ denote the set of companion physiological modalities. The adapter should maintain a shared internal organization consisting of modality-specific parameters and shared parameters:

$$\omega = \left(\{\omega^m\}_{m\in\mathcal{A}}, \omega^{\mathrm{sh}}\right), \qquad a\left(\mathbf{Z}, \mathbf{X}^{\mathcal{A}}; \{\omega^m\}_{m\in\mathcal{A}}, \omega^{\mathrm{sh}}\right), \quad \forall\, \mathcal{A} \subseteq \mathcal{M}. \tag{3}$$

Only the modality-specific components associated with $m \in \mathcal{A}$ participate for a given configuration, while the overall adapter organization remains unchanged.

C2: Define a label-free mapping from target data to adapter parameters. Let $\mathcal{R} = \{(\mathbf{X}_i^{\mathrm{EEG}}, \mathbf{X}_i^{\mathcal{A}})\}_i$ denote the unlabeled target samples, and let τ denote the prediction task associated with the frozen EFM. We seek a generator

$$g:\ (\mathcal{R}, \tau) \mapsto \widehat{\omega} \in \mathbb{R}^{P(\mathcal{A})}, \tag{4}$$

where $\widehat{\omega}$ denotes the generated adapter parameters. The generator does not use target labels or dataset identifiers.

C3: Learn the generator from source reference adapters. Let $\mathcal{S}$ be a set of source triples $(\mathcal{R}, \tau, \omega^*)$, where ω^* is a reference adapter fitted on labeled source data. The generator is trained so that its outputs match these references in both parameters and predictions through the frozen EFM:

$$\min_g\ \mathbb{E}_{(\mathcal{R},\tau,\omega^*)\sim\mathcal{S}}\Big[\, \|g(\mathcal{R},\tau) - \omega^*\|^2 + \lambda\, \delta_{\mathrm{fun}}(\omega^*, g(\mathcal{R},\tau)) \,\Big], \tag{5}$$

where δ_{fun} measures the discrepancy between the predictions induced by two adapters through the same frozen EFM. At deployment, g produces the target adapter in a single forward pass, without any target-side optimization.

3 METHODOLOGY

3.1 FRAMEWORK OVERVIEW

Given an unseen multimodal recording and a supported prediction task, ZeroMAG generates a plug-in adapter that adds multimodal capability to a frozen EFM without target labels or target-side optimization. As illustrated in the upper part of Fig. 2, unlabeled target samples are first encoded into a

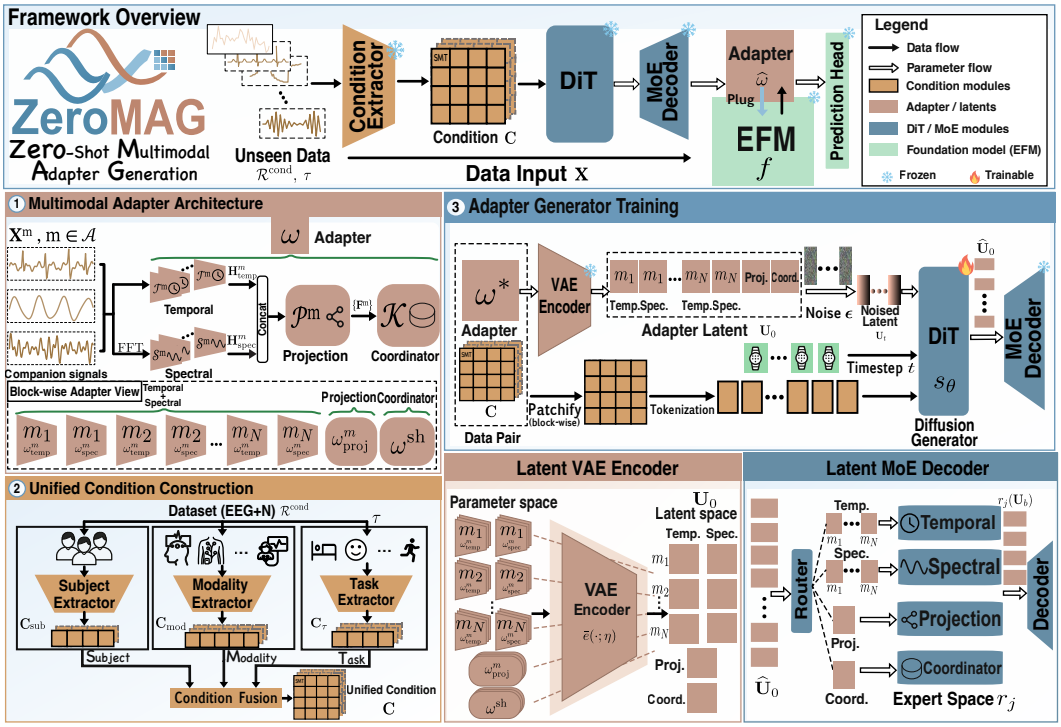

Figure 2: Overview of ZeroMAG. The framework combines a configuration-invariant multimodal adapter, target-derived condition construction, and function-constrained conditional generation for zero-shot adapter synthesis.

recording-specific condition. This condition guides adapter generation in a learned latent space, and a frozen decoder maps the generated latent representation to adapter parameters:

$$\widehat{\omega} = d\left(s_\theta\left(\boldsymbol{\epsilon}, E_{\text{cond}}\left(\mathcal{R}^{\text{cond}}, \tau; \xi\right); \psi\right)\right), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

Here, $\mathcal{R}^{\text{cond}}$ is the unlabeled conditioning subset, τ specifies the task. The condition extractor E_{cond}, conditional diffusion sampler s_θ, and decoder d are parameterized by ξ, θ, and ψ, respectively; $\boldsymbol{\epsilon}$ is the initial Gaussian noise. The sampler s_θ runs 50 deterministic DDIM steps (Song et al., 2020) with the conditional denoiser $\boldsymbol{\epsilon}_\theta$ of Sec. 3.4.

The adapter architecture (Sec. 3.2) defines what is generated, the target condition (Sec. 3.3) specifies what to adapt to, and the two-stage procedure (Sec. 3.4) learns a structured adapter space and conditionally generates within it.

3.2 Multimodal Adapter Architecture

Intuition. To generate adapters for heterogeneous modality configurations with a single generator, the adapters must share a common parameter structure. Without it, each configuration would require its own output space. This suggests a ***1+N*** organization, where the frozen EEG representation serves as the invariant anchor and modality-specific companion branches provide configuration-dependent extensions.

Operationalization. Following the ***1+N*** intuition, we treat the frozen EEG representation as the fixed anchor (1) and attach one modality-specific branch for each available companion modality (N). Regardless of modality type, each companion branch follows the same integration pathway: **Temporal–Spectral Encoding**, **Feature Projection**, and **Shared Coordination**. The first two components extract and align modality-specific features, while the shared coordinator integrates them with the EEG representation.

Temporal–Spectral Encoding. For each available companion modality, a temporal encoder extracts patterns from the signal over time, while a spectral encoder processes its frequency-domain

representation:

$$\begin{aligned}\mathbf{H}^m_{\text{temp}} &= \mathcal{T}^m\left(\mathbf{X}^m;\omega^m_{\text{temp}}\right),\\ \mathbf{H}^m_{\text{spec}} &= \mathcal{S}^m\left(\text{FFT}(\mathbf{X}^m);\omega^m_{\text{spec}}\right), \qquad m\in\mathcal{A}.\end{aligned} \tag{7}$$

Here, $\mathbf{H}^m_{\text{temp}}$ and $\mathbf{H}^m_{\text{spec}}$ are temporal and spectral features. The encoders $\mathcal{T}^m$ and $\mathcal{S}^m$ have parameters ω^m_{temp} and ω^m_{spec}, respectively, and FFT denotes the fast Fourier transform.

Feature Projection. The temporal and spectral features are concatenated and mapped into the EFM representation space, providing a common interface for heterogeneous companion signals:

$$\mathbf{F}^m = \mathcal{P}^m\left(\text{Concat}\left(\mathbf{H}^m_{\text{temp}},\mathbf{H}^m_{\text{spec}}\right);\omega^m_{\text{proj}}\right), \qquad m\in\mathcal{A}. \tag{8}$$

Here, $\mathcal{P}^m$ is the projection block with parameters ω^m_{proj}, and Concat denotes feature concatenation. Its output $\mathbf{F}^m$ is the projected feature of modality m.

Shared Coordination. A shared coordinator $\mathcal{K}$ jointly processes the frozen EEG representation and available companion features to produce the adapted representation:

$$\mathbf{Z}^{\mathcal{A}} = \mathcal{K}\left(\mathbf{Z},\{\mathbf{F}^m\}_{m\in\mathcal{A}};\omega^{\text{sh}}\right). \tag{9}$$

Here, ω^{sh} denotes the shared coordinator parameters. Together, these components define the adapter mapping $a(\mathbf{Z},\mathbf{X}^{\mathcal{A}};\omega)$. Only available companion branches are evaluated, while the block schema and functional block identities remain fixed, satisfying C1.

3.3 Unified Condition Construction

Intuition. At deployment, the source-trained generator infers the adapter from unlabeled target recordings and the supported task. Modality information alone cannot distinguish these cases: recordings with the same modalities may exhibit different subject-specific physiology, while the same physiological information may serve different prediction tasks. This suggests conditioning adapter generation on modality, subject, and task information.

Operationalization. Following this intuition, the condition encoder E_{cond} builds one factor for each source of variation (modality, subject, and task) and combines them through a fusion module F_{cond}:

$$\mathbf{C} = E_{\text{cond}}\left(\mathcal{R}^{\text{cond}},\tau;\xi\right) = F_{\text{cond}}\left([\mathbf{C}_{\text{mod}};\mathbf{C}_{\text{sub}};\mathbf{C}_\tau]\right). \tag{10}$$

Here, $\mathcal{R}^{\text{cond}}$ denotes the unlabeled target subset used for condition construction, τ specifies the prediction task, and $\mathbf{C}$ is the resulting target condition. $\mathbf{C}_{\text{mod}}$, $\mathbf{C}_{\text{sub}}$, and $\mathbf{C}_\tau$ denote the modality, subject, and task factors, respectively, while ξ contains all trainable parameters of E_{cond}.

Factor Construction. Let i index samples in $\mathcal{R}^{\text{cond}}$. For each modality $m\in\{\text{EEG}\}\cup\mathcal{A}$, $\mathbf{V}^m_i$ denotes its sample-level feature and $\mathbf{V}^m$ the corresponding recording-level feature. The three factors are constructed as

$$\begin{aligned}\mathbf{C}_{\text{mod}} &= F_{\text{mod}}\Big(\{\mathbf{V}^m+\mathbf{e}^m_{\text{mod}}\}_{m\in\{\text{EEG}\}\cup\mathcal{A}}\Big),\\ \mathbf{C}_{\text{sub}} &= F_{\text{sub}}\Big(\mathbf{q}_{\text{sub}},\{\mathbf{V}^m_i\}_{m\in\{\text{EEG}\}\cup\mathcal{A},\,i}\Big),\\ \mathbf{C}_\tau &= E_{\text{task}}(\tau).\end{aligned} \tag{11}$$

Here, F_{mod} is permutation-invariant set pooling (Zaheer et al., 2017) over recording-level features tagged with learned modality embeddings $\mathbf{e}^m_{\text{mod}}$; F_{sub} is attention pooling (Vaswani et al., 2017) of all sample-level features with a learned, task-independent query $\mathbf{q}_{\text{sub}}$; and E_{task} is a learned embedding of the supported task. All are part of ξ; details and invariance properties are given in Appendix B.

3.4 Adapter Generator Training

Intuition. Fitted reference adapters reveal how heterogeneous blocks can be jointly configured to work with the frozen EFM. Their shared architecture and fitting objective suggest that these configurations contain recurring structure, making it unnecessary to model every degree of freedom in the raw weight space. Learning a compact, block-aligned representation can capture this structure while retaining the distinct roles and dependencies of adapter components. Conditional generation

can then focus on variations within this learned representation, while functional supervision keeps both the decoded and the generated adapters in predictive agreement with their references.

Operationalization. We implement this design in two stages: function-preserving representation learning followed by function-constrained conditional generation. Both use the functional discrepancy δ_{fun} introduced in C3, which we evaluate on a source subset $\mathcal{R}^{\mathrm{fun}}$ held out from reference-adapter fitting:

$$\delta_{\mathrm{fun}}(\omega, \omega') = \frac{1}{|\mathcal{R}^{\mathrm{fun}}|} \sum_{\mathbf{r}_i \in \mathcal{R}^{\mathrm{fun}}} D_{\mathrm{KL}}\big(\sigma(\mathbf{L}_i^{\mathcal{A}}(\omega)) \,\big\|\, \sigma(\mathbf{L}_i^{\mathcal{A}}(\omega'))\big)\,, \tag{12}$$

where $\mathbf{L}_i^{\mathcal{A}}(\omega)$ denotes the logits of sample $\mathbf{r}_i$ produced with adapter parameters ω, and σ denotes softmax. This label-free measure compares two adapters solely through the predictions they induce in the same frozen EFM.

3.4.1 Stage 1: Function-Preserving Representation Learning

Block-Wise Latent Encoding. Let $\omega^* = [\omega_b^*]_{b=1}^B$ denote a fitted reference adapter partitioned into the functional blocks of Sec. 3.2. A VAE encoder maps each block to a latent token $\mathbf{U}_{b,\mathrm{pre}}$, which is combined with a learned block embedding $\mathbf{E}_b$ and contextualized by a cross-block transformer (CBT):

$$\mathbf{U} = \mathrm{CBT}\big([\mathbf{U}_{b,\mathrm{pre}} + \mathbf{E}_b]_{b=1}^B\big)\,, \tag{13}$$

whose b-th token is denoted $\mathbf{U}_b$. We denote the resulting stochastic encoder by $e(\cdot;\eta)$.

MoE Decoding. To share capacity across heterogeneous block types while respecting their distinct output forms, each token is routed to shared experts $\{r_j\}$ according to its content and block identity, followed by a block-type-specific output head:

$$\begin{aligned} \boldsymbol{\pi}_b &= \mathrm{softmax}(\mathrm{route}([\mathbf{U}_b; \mathbf{E}_b]; \psi_r))\,, \\ \widetilde{\omega}_b &= \mathrm{head}_{\mathrm{type}(b)}\left(\sum_{j \in \mathrm{TopK}(\boldsymbol{\pi}_b)} \pi_{b,j} r_j(\mathbf{U}_b)\right), \end{aligned} \tag{14}$$

where $\mathrm{type}(b) \in \{\mathrm{temp}, \mathrm{spec}, \mathrm{proj}, \mathrm{shared}\}$. Collectively, the decoder gives $\widetilde{\omega} = d(\mathbf{U}; \psi)$.

Function-Preserving Reconstruction. Since parameter reconstruction alone does not guarantee that adapter behavior is preserved, Stage 1 combines parameter-level and function-level objectives:

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{blk}} + \lambda_{\mathrm{cos}}\mathcal{L}_{\mathrm{cos}} + \lambda_{\mathrm{fun}}\delta_{\mathrm{fun}}(\omega^*, \widetilde{\omega}) + \beta\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{blk}}$ and $\mathcal{L}_{\mathrm{cos}}$ reconstruct adapter parameters, δ_{fun} preserves predictive behavior, $\mathcal{L}_{\mathrm{KL}}$ regularizes the latent posterior, and $\mathcal{L}_{\mathrm{bal}}$ balances expert utilization. After training, the encoder and decoder are frozen, and $\mathbf{U}_0 = \overline{e}(\omega^*; \eta)$, obtained by contextualizing posterior means, serves as the deterministic target latent for Stage 2.

3.4.2 Stage 2: Function-Constrained Conditional Generation

Conditional Latent Diffusion. For each source triple $(\mathcal{R}, \tau, \omega^*)$, a condition $\mathbf{C} = E_{\mathrm{cond}}(\mathcal{R}^{\mathrm{cond}}, \tau; \xi)$ is constructed from an unlabeled subset $\mathcal{R}^{\mathrm{cond}} \subseteq \mathcal{R}^{\mathrm{fit}}$, where $\mathcal{R}^{\mathrm{fit}}$ is the subset used to fit ω^*. In the frozen Stage 1 latent space, a DiT denoiser (Peebles & Xie, 2023) $\boldsymbol{\epsilon}_\theta(\mathbf{U}_t, t, \mathbf{C})$ predicts the noise added to $\mathbf{U}_0$. Its prediction yields the clean-latent estimate $\widehat{\mathbf{U}}_0 = (\mathbf{U}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta)/\sqrt{\bar{\alpha}_t}$, which the frozen decoder maps to $\widehat{\omega} = d(\widehat{\mathbf{U}}_0; \psi)$.

Function-Constrained Generation. The condition encoder and denoiser are jointly optimized by

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{lat}}\mathcal{L}_{\mathrm{lat}} + \lambda_{\mathrm{w}}\mathcal{L}_{\mathrm{w}} + \lambda_{\mathrm{gf}}\delta_{\mathrm{fun}}(\omega^*, \widehat{\omega}), \tag{16}$$

where $\mathcal{L}_{\mathrm{diff}}$ supervises denoising, $\mathcal{L}_{\mathrm{lat}}$ and $\mathcal{L}_{\mathrm{w}}$ align $\widehat{\mathbf{U}}_0$ and $\widehat{\omega}$ with their references, and δ_{fun} aligns predictive behavior, with gradients passing through the frozen decoder. Because the division by $\sqrt{\bar{\alpha}_t}$ amplifies noise-prediction errors at high noise levels, the latent, weight, and functional terms are applied only at low-noise timesteps (Appendix H.1), whereas $\mathcal{L}_{\mathrm{diff}}$ is applied at all timesteps.

After training, E_{cond}, $\boldsymbol{\epsilon}_\theta$, and d are fixed and realize the label-free mapping of C2 (Eq. (6)).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Tasks. We evaluate ZeroMAG on a multi-dataset benchmark spanning sleep staging, emotion recognition, cognitive load, motor imagery, and freezing-of-gait detection. The recordings cover EOG, EMG, ECG, eye movement, acceleration, and skin conductance, including settings in which modality combinations or modality types are excluded from generator training. Table 1 reports results on six target datasets that are disjoint from the source training datasets; FoG is also a task-holdout target with no source adapter for its task. Task coverage, the remaining targets, and further analyses are in Appendix C.

Backbones and Baselines. We use three frozen EEG foundation models: CBraMod (Wang et al., 2025), CSBrain (Zhou et al., 2025), and CodeBrain (Ma et al., 2026). Each backbone has a separately trained generator. ZeroMAG and the target-label-free baselines use the same frozen EEG encoder and prediction head. These baselines include EEG-only inference, mean and nearest source weights, and Direct MLP. A separate supervised multimodal fine-tuning experiment uses target training labels to measure the gap to supervised adaptation on the same query split; it is not part of the ZeroMAG pipeline. PhysioOmni (Fu et al., 2025), SleepFM (Thapa et al., 2024), and Brant-X (Zhang et al., 2024) serve as cross-paradigm references.

Protocol. Evaluation uses a dataset-disjoint source–target split: all target datasets are excluded from EEG encoder/head training, source adapter fitting, both generator-training stages, and model selection. For each target subject, calibration and query subsets are disjoint. ZeroMAG uses only unlabeled calibration signals and the supported task to generate an adapter, then predicts on query inputs with all model parameters fixed. Query labels are used only to compute evaluation metrics. Dataset lists and split details are given in Appendix H.3.

Metrics and Implementation. We report balanced accuracy (B-ACC) averaged over five matched seeds. Both stages use AdamW (learning rate 10^{-4}, batch size 64); the CBT and DiT have 2 and 4 layers of width 256, the decoder uses 4 experts with top-2 routing, and latent tokens have width 128. Loss weights are $\lambda_{\text{cos}}{=}\lambda_{\text{lat}}{=}\lambda_{\text{w}}{=}0.1$, $\lambda_{\text{fun}}{=}\lambda_{\text{gf}}{=}1$, $\beta{=}10^{-4}$, and $\lambda_{\text{bal}}{=}0.01$, with the low-noise gate $\bar{\alpha}_t{>}0.5$. The source bank holds 311 adapters per backbone, and each target subject provides 64 unlabeled calibration windows. Full settings are in Appendix H (Table H.1).

4.2 PERFORMANCE COMPARISON WITH BASELINES

Comparison with label-free and supervised adaptation. ZeroMAG ranks first among target-label-free methods in all 18 backbone–dataset evaluations, improving B-ACC on average by 7.22 points over EEG-only inference, 6.78 over mean source weights, 6.00 over nearest source weights, and 4.89 over Direct MLP, the strongest competing generator. Despite using neither target labels nor target-side optimization, it remains only 0.38–0.61 points below full-modality supervised tuning in every setting (0.50 on average), recovering 93.50% of the supervised gain over EEG-only inference.

Backbone and cross-paradigm generalization. The gain over EEG-only inference is stable across CBraMod, CSBrain, and CodeBrain at 7.12, 7.30, and 7.24 points; the range is approximately 0.19 points. With CodeBrain, ZeroMAG exceeds SleepFM on all six displayed targets by 1.85 points on average, trails PhysioOmni by only 0.31 points, and trails Brant-X by 0.80 points on average.

4.3 ABLATION STUDY

We ablate the condition, generator, adapter, and functional supervision while keeping the source adapter bank, source training splits, frozen backbone, prediction heads, and evaluation protocol fixed. Table 2 reports CBraMod results averaged across all held-out evaluation settings.

Condition. Removing the subject factor and retaining only task and modality information lowers B-ACC by 11.09 points. Shuffling subject conditions produces a similarly large 9.24-point drop,

Table 1: Balanced accuracy (%, mean±std over five seeds) on six held-out target datasets. Within each EEG-backbone group, ZeroMAG and the label-free baselines use the same frozen encoder and prediction head, without target labels or target-side optimization. Supervised multimodal fine-tuning is a separate reference using target training labels and all available modalities. Bold entries indicate the best target-label-free result in each group.

Method	ISRUC-S3	HMC	MASS	SEED-V	EEGMAT	FoG
CBraMod						
EEG-only	75.83±0.76	72.78±0.50	74.29±0.58	40.62±1.24	73.38±2.48	76.61±1.35
Supervised tuning	84.04±0.88	80.81±0.41	82.41±0.68	45.44±1.58	81.46±1.93	85.02±1.21
Mean source weights	74.31±0.82	72.93±0.49	75.91±0.63	39.83±1.13	73.52±0.92	76.73±0.78
Nearest source weights	76.63±0.61	73.18±0.52	76.35±0.51	40.68±1.37	74.63±0.81	77.52±0.73
Direct MLP	76.89±0.73	73.43±0.55	77.48±0.65	40.99±1.05	75.21±1.02	78.03±0.62
ZeroMAG	83.59±0.59	80.29±0.42	82.03±0.52	44.83±1.22	80.99±0.81	84.47±0.72
CSBrain						
EEG-only	75.94±0.93	73.31±0.48	75.89±0.54	41.38±1.32	75.38±2.67	75.96±1.52
Supervised tuning	84.10±1.43	81.39±0.39	84.15±0.62	46.32±1.17	83.57±1.43	85.13±1.31
Mean source weights	75.42±0.62	73.51±0.45	77.72±0.58	41.75±0.62	75.91±0.71	77.08±0.65
Nearest source weights	76.74±0.55	73.82±0.52	77.98±0.63	42.28±0.55	75.83±0.66	78.73±0.59
Direct MLP	79.18±0.71	74.21±0.48	80.45±0.72	42.69±0.48	78.12±0.58	80.02±0.52
ZeroMAG	83.67±0.61	80.83±0.52	83.74±0.83	45.74±0.44	83.08±0.71	84.60±0.65
CodeBrain						
EEG-only	76.57±0.73	72.95±0.49	77.24±0.51	41.52±1.46	76.62±2.83	76.77±1.26
Supervised tuning	84.92±1.61	80.93±0.38	85.67±0.58	46.39±1.34	85.02±1.63	85.25±1.33
Mean source weights	77.02±0.68	73.28±0.42	79.12±0.72	41.88±0.58	77.22±0.66	77.85±0.71
Nearest source weights	77.22±0.62	73.65±0.48	80.21±0.68	42.35±0.52	77.98±0.59	79.22±0.65
Direct MLP	79.92±0.55	73.92±0.51	81.65±0.61	42.87±0.45	79.28±0.52	80.55±0.58
ZeroMAG	84.44±0.77	80.43±0.48	85.25±0.72	45.84±0.55	84.51±0.63	84.65±0.58
Multimodal physiological models						
PhysioOmni	84.62±0.63	81.05±0.42	85.50±0.65	46.04±1.32	84.81±2.13	84.93±0.53
SleepFM	83.27±0.61	80.01±0.47	81.61±0.52	44.48±0.58	80.54±0.72	84.09±0.68
Brant-X	85.19±0.54	81.51±0.46	86.07±0.59	46.49±0.63	85.31±0.61	85.37±0.72

although the representations retain the same marginal distribution and dimensionality. The condition must therefore encode subject-specific information aligned with the target recording.

Generator. Direct weight regression decreases B-ACC by 11.07 points. The modular latent space raises B-ACC from 62.34 to 65.23, DiT to 68.34, and the MoE decoder to 73.41.

Adapter. Sharing the modality encoder reduces B-ACC by 4.16 points, removing the spectral branch by 8.12 points, and replacing learned coordination with mean fusion by 6.58 points, supporting modality-specific encoding, temporal–spectral processing, and learned fusion.

Functional supervision. Removing functional supervision from Stage 1, Stage 2, or both lowers B-ACC by 1.15, 1.58, and 2.87 points. Without the Stage 1 term, reconstruction error falls by 6% while prediction KL to the reference adapter rises by 31%: closer parameter reconstruction need not yield closer predictions (Appendix G.3).

4.4 Generalization and Efficiency

Unseen modalities. On unseen combinations of known modalities and modality types in $\mathcal{M}$ excluded from generator training, ZeroMAG recovers 89.7% and 82.6% of the gain achieved by supervised multimodal adaptation, respectively. It outperforms mean source transfer, nearest-source retrieval, and direct weight regression in both settings.

Table 2: Ablation results with CBraMod, averaged over all nine held-out datasets (% B-ACC, mean±std over five seeds). Ablated results are shaded blue, and Δ denotes the decrease from Full ZeroMAG in the same run. Ablations use a separate run under the same protocol, so Full differs from the Table 1 average.

Component	**Variant**	**B-ACC** $\uparrow$	$\Delta \downarrow$
Condition	Task + modality only	$62.32_{\pm 0.41}$	$11.09_{\pm 0.72}$
	Shuffled subject condition	$64.17_{\pm 0.53}$	$9.24_{\pm 0.98}$
Generator	Direct weight MLP	$62.34_{\pm 0.67}$	$11.07_{\pm 0.88}$
	Modular latent MLP	$65.23_{\pm 0.62}$	$8.18_{\pm 0.83}$
	Modular latent DiT (w/o MoE)	$68.34_{\pm 0.21}$	$5.07_{\pm 0.71}$
Adapter	Shared modality encoder	$69.25_{\pm 0.33}$	$4.16_{\pm 0.81}$
	Temporal branch only	$65.29_{\pm 0.72}$	$8.12_{\pm 1.03}$
	Mean fusion	$66.83_{\pm 0.32}$	$6.58_{\pm 0.71}$
Functional	w/o Stage 1 functional	$72.26_{\pm 0.76}$	1.15
	w/o Stage 2 functional	$71.83_{\pm 0.79}$	1.58
	w/o both	$70.54_{\pm 0.86}$	2.87
Full ZeroMAG		$73.41_{\pm 0.71}$	—

Data and target-side efficiency. Performance saturates at approximately 64 unlabeled calibration windows. Moreover, using only 50% of the source adapter bank retains 97.2% of the full multimodal gain. ZeroMAG constructs the condition and generates an adapter in 2 minutes, compared with 15 minutes for supervised target adaptation under the same hardware and protocol ($7.5\times$ faster); offline training cost is amortized across targets (Appendix F).

5 Related Work

EEG foundation and multimodal physiological models. EEG foundation models provide transferable backbones across downstream tasks (Kuruppu et al., 2025; Jiang et al., 2024; Wang et al., 2025; 2024a; Yang et al., 2023; Zhang et al., 2023), but largely retain EEG-centered interfaces. Prior multimodal physiological work integrates signals through multimodal foundation-model pretraining or EEG-to-physiology alignment (Fu et al., 2025; Thapa et al., 2024; Zhang et al., 2024). ZeroMAG generates a plug-in adapter for an existing EFM while keeping its encoder and prediction head frozen.

Adaptation and weight generation. Parameter-efficient adaptation methods (Houlsby et al., 2019; Hu et al., 2022; Lester et al., 2021; Han et al., 2024; Suzumura et al., 2024) still require target-side optimization, whereas hypernetworks generate parameters from task descriptors or unlabeled prompts (Ha et al., 2017; Karimi Mahabadi et al., 2021; Charakorn et al., 2025; Liang et al., 2025). ZeroMAG instead generates a modular physiological adapter from unlabeled target information in a structured latent space. Unlike test-time adaptation (Liang et al., 2023; Wang et al., 2021; Liang et al., 2020), deployment requires no pseudo-labels, gradients, or iterative optimization (Appendix J.1).

6 Conclusion

We introduced ZeroMAG, a zero-shot framework that extends frozen EFMs to heterogeneous multimodal recordings through generated plug-in adapters. Combining configuration-invariant adaptation, target-derived conditioning, and function-constrained latent generation, it consistently improves over target-label-free baselines and approaches supervised multimodal adaptation, requiring no target labels or target-side optimization for the added adapter and no repeated multimodal pretraining. ZeroMAG relies on a supplied fine-tuned EFM and adequate source-bank coverage, and cannot incorporate modality types outside $\mathcal{M}$ without retraining.

AI USE STATEMENT

We used ChatGPT, Codex, and Claude to draft, edit, and polish manuscript text; retrieve and verify references; prepare and refine figures; organize existing experimental outputs into tables and summaries; discuss research framing and related-work positioning; review and optimize auxiliary code to reduce repetitive workflow tasks; and check the manuscript for internal consistency. The authors made the final research and methodological decisions. AI assistance with code was limited to auxiliary workflow tasks; AI tools were not used to design or implement the ZeroMAG method or baselines, design the experimental protocol, run experiments, generate or alter experimental measurements, or interpret the findings. The authors checked AI-assisted text, figures, code-related changes, and output summaries against the manuscript and original experiment files, and are responsible for the accuracy of all claims, citations, figures, code, and results.

REPRODUCIBILITY STATEMENT

Implementation details and hyperparameters are listed in Table H.1; datasets are listed in Appendix C, and split details are given in Appendix H.3. An initial implementation snapshot is available at https://anonymous.4open.science/r/ZeroMAG-795E/.

REFERENCES

Isabela Albuquerque, Abhishek Tiwari, Mark Parent, Raymundo Cassani, Jean-François Gagnon, Daniel Lafond, Sébastien Tremblay, and Tiago H. Falk. WAUC: A multi-modal database for mental workload assessment under physical activity. *Frontiers in Neuroscience*, 14:549524, 2020. doi: 10.3389/fnins.2020.549524.

Diego Alvarez-Estevez and Roselyne M. Rijsman. Inter-database validation of a deep learning approach for automatic sleep scoring. *PLOS ONE*, 16(8):e0256111, 2021. doi: 10.1371/journal.pone.0256111.

Marc Bächlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Tröster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, 2010. doi: 10.1109/TITB.2009.2036165.

Kay H. Brodersen, Cheng Soon Ong, Klaas E. Stephan, and Joachim M. Buhmann. The Balanced Accuracy and Its Posterior Distribution. In *2010 20th International Conference on Pattern Recognition*, 2010. doi: 10.1109/ICPR.2010.764.

Clemens Brunner, Robert Leeb, Gernot R. Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. BCI Competition 2008: Graz data set A (data set 2a). BCI Competition IV dataset description, 2008. URL https://www.bbci.de/competition/iv/desc_2a.pdf.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-LoRA: Instant Transformer Adaption. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 7485–7514. PMLR, 2025. URL https://proceedings.mlr.press/v267/charakorn25a.html.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*, 16(3):031001, 2019. doi: 10.1088/1741-2552/ab0ab5.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://www.jmlr.org/papers/v23/21-0998.html.

Xi Fu, Wei-Bang Jiang, Yi Ding, and Cuntai Guan. Towards Robust Multimodal Physiological Foundation Models: Handling Arbitrary Missing Modalities. *arXiv preprint arXiv:2504.19596*, 2025. URL https://arxiv.org/abs/2504.19596.

David Ha, Andrew Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *arXiv preprint arXiv:2403.14608*, 2024. URL https://arxiv.org/abs/2403.14608.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large Brain Model for Learning Generic Representations with Tremendous EEG Data in BCI. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/47393e8594c82ce8fd83adc672cf9872-Paper-Conference.pdf.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, 2021. doi: 10.18653/v1/2021.acl-long.47. URL https://aclanthology.org/2021.acl-long.47/.

Stamos Katsigiannis and Naeem Ramzan. DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. *IEEE Journal of Biomedical and Health Informatics*, 22(1):98–107, 2018. doi: 10.1109/JBHI.2017.2688239.

Bob Kemp, Aeilko H. Zwinderman, Bert Tuk, Hilbert A. C. Kamphuisen, and Josefien J. L. Oberyé. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000. doi: 10.1109/10.867928.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 124:180–192, 2016. doi: 10.1016/j.cmpb.2015.10.013.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Sander Koelstra, Christian Mühl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. DEAP: A database for emotion analysis using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, 2012. doi: 10.1109/T-AFFC.2011.15.

Gayal Kuruppu, Neeraj Wagh, Vaclav Kremen, Sandipan Pati, Gregory Worrell, and Yogatheesan Varatharajah. EEG Foundation Models: A Critical Review of Current Progress and Future Directions. *arXiv preprint arXiv:2507.11783*, 2025. URL https://arxiv.org/abs/2507.11783.

Hyung-Tak Lee, Miseon Shim, Xianghong Liu, Hye-Ran Cheon, Sang-Gyu Kim, Chang-Hee Han, and Han-Jeong Hwang. A review of hybrid EEG-based multimodal human–computer interfaces using deep learning: applications, advances, and challenges. *Biomedical Engineering Letters*, 15 (4):587–618, 2025. doi: 10.1007/s13534-025-00469-5.

Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. Eeg dataset and openbmi toolbox for three bci paradigms: An investigation into bci illiteracy. *GigaScience*, 8(5):giz002, 2019. doi: 10.1093/gigascience/giz002.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021. URL https://aclanthology.org/2021.emnlp-main.243/.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6028–6039, 2020. URL https://proceedings.mlr.press/v119/liang20a.html.

Jian Liang, Ran He, and Tieniu Tan. A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts. *arXiv preprint arXiv:2303.15361*, 2023. URL https://arxiv.org/abs/2303.15361.

Zhiyuan Liang, Dongwen Tang, Yuhao Zhou, Xuanlei Zhao, Mingjia Shi, Wangbo Zhao, Zekai Li, Peihao Wang, Konstantin Schürholt, Damian Borth, Michael M. Bronstein, Yang You, Zhangyang Wang, and Kai Wang. Drag-and-Drop LLMs: Zero-Shot Prompt-to-Weights. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.16406.

Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Comparing Recognition Performance and Robustness of Multimodal Deep Learning Models for Multimodal Emotion Recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(2):715–729, 2022. doi: 10.1109/tcds.2021.3071170.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. CodeBrain: Bridging Decoupled Tokenizer and Multi-Scale Architecture for EEG Foundation Model. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.09110.

Tsendsuren Munkhdalai and Hong Yu. Meta Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2554–2563, 2017. URL https://proceedings.mlr.press/v70/munkhdalai17a.html.

Christian O'Reilly, Nadia Gosselin, Julie Carrier, and Tore Nielsen. Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of Sleep Research*, 23(6):628–635, 2014. doi: 10.1111/jsr.12169.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Stuart F. Quan, Barbara V. Howard, Conrad Iber, James P. Kiley, F. Javier Nieto, Gregory T. O'Connor, David M. Rapoport, Susan Redline, John Robbins, Jonathan M. Samet, and Patricia W. Wahl. The Sleep Heart Health Study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997. doi: 10.1093/sleep/20.12.1077.

Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1701.06538.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. URL https://arxiv.org/abs/2010.02502.

Ramanathan Subramanian, Julia Wache, Mojtaba Khomami Abadi, Radu L. Vieriu, Stefan Winkler, and Nicu Sebe. ASCERTAIN: Emotion and personality recognition using commercial sensors. *IEEE Transactions on Affective Computing*, 9(2):147–160, 2016. doi: 10.1109/TAFFC.2016.2625250.

Toyotaro Suzumura, Hiroki Kanezashi, and Shotaro Akahori. Graph Adapter of EEG Foundation Models for Parameter Efficient Fine Tuning. *arXiv preprint arXiv:2411.16155*, 2024. URL https://arxiv.org/abs/2411.16155.

Mario Giovanni Terzano, Liborio Parrino, A. Sherieri, Robert Chervin, Sudhansu Chokroverty, Christian Guilleminault, Max Hirshkowitz, Mark Mahowald, Harvey Moldofsky, Agostinho Rosa, Robert Thomas, and Arthur Walters. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep Medicine*, 2(6):537–553, 2001. doi: 10.1016/S1389-9457(01)00149-6.

Rahul Thapa, Bryan He, Magnus Ruud Kjær, Hyatt Moore, IV, Gauri Ganjoo, Emmanuel Mignot, and James Zou. SleepFM: Multi-modal Representation Learning for Sleep Across Brain Activity, ECG and Respiratory Signals. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 48019–48037. PMLR, 2024. URL https://proceedings.mlr.press/v235/thapa24a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Guangyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. EEGPT: Pretrained Transformer for Universal and Reliable Representation of EEG Signals. In *Advances in Neural Information Processing Systems*, volume 37, pp. 39249–39280, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4540d267eeec4e5dbd9dae9448f0b739-Paper-Conference.pdf.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. CBraMod: A Criss-Cross Brain Foundation Model for EEG Decoding. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.07236.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural Network Diffusion. *arXiv preprint arXiv:2402.13144*, 2024b. URL https://arxiv.org/abs/2402.13144.

Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002. doi: 10.1016/s1388-2457(02)00057-3.

Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. BIOT: Biosignal Transformer for Cross-data Learning in the Wild. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f6b30f3e2dd9cb53bbf2024402d02295-Paper-Conference.pdf.

Michael K. Yeung and Vivian W. Chu. Viewing neurovascular coupling through the lens of combined EEG–fNIRS: A systematic review of current methods. *Psychophysiology*, 59(6):e14054, 2022. doi: 10.1111/psyp.14054.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation Model for Intracranial Neural Signal. In *Advances in Neural Information Processing Systems*, volume 36, pp. 26304–26321, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/535915d26859036410b0533804cee788-Paper-Conference.pdf.

Daoze Zhang, Zhizhang Yuan, Junru Chen, Kerui Chen, and Yang Yang. Brant-X: A Unified Physiological Signal Alignment Framework. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4155–4166, 2024. doi: 10.1145/3637528.3671953.

Wei Zhang, Zhuokun Yang, Hantao Li, Debin Huang, Lipeng Wang, Yanzhao Wei, Lei Zhang, Lin Ma, Huanhuan Feng, Jing Pan, Yuzhu Guo, and Piu Chan. Multimodal Data for the Detection of Freezing of Gait in Parkinson's Disease. *Scientific Data*, 9(1):606, 2022. doi: 10.1038/s41597-022-01713-8.

Wei-Long Zheng and Bao-Liang Lu. A multimodal approach to estimating vigilance using EEG and forehead EOG. *Journal of Neural Engineering*, 14(2):026017, 2017. doi: 10.1088/1741-2552/aa5a98.

Yuchen Zhou, Jiamin Wu, Zichen Ren, Zhouheng Yao, Weiheng Lu, Kunyu Peng, Qihao Zheng, Chunfeng Song, Wanli Ouyang, and Chao Gou. CSBrain: A Cross-scale Spatiotemporal Brain Foundation Model for EEG Decoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7e199ad8ae40eb19b2980f61f659cb07-Abstract-Conference.html.

Igor Zyma, Sergii Tukaev, Ivan Seleznov, Ken Kiyono, Anton Popov, Mariia Chernykh, and Oleksii Shpenkov. Electroencephalograms during Mental Arithmetic Task Performance. *Data*, 4(1):14, 2019. doi: 10.3390/data4010014.

Appendix Contents

A. Notation and Model Scope 17
B. Adapter and Condition 19
C. Datasets and Additional Results 21
D. Training and Deployment 25
E. Objective Analysis 28
F. Efficiency 31
G. Component Ablations 32
H. Implementation and Evaluation Protocol 34
I. Generalization and Data Efficiency 37
J. Comparison with Related Methods 39

A Notation and Model Scope

A.1 EEG representations

EEG representation learning has progressed from task-specific supervised networks (Craik et al., 2019) to foundation models trained across recordings (Kuruppu et al., 2025). These models differ in tokenization, objectives, and recording domains. LaBraM uses neural signal tokenization (Jiang et al., 2024); EEGPT learns EEG representations through self-supervised transformer pretraining (Wang et al., 2024a). BIOT addresses heterogeneous biosignal inputs (Yang et al., 2023), whereas Brant focuses on intracranial neural recordings (Zhang et al., 2023). These differences matter when defining the adapter's input and output interface. Our experiments use the three backbone instantiations described below.

A.2 Frozen models and separate generators

For each backbone, we directly load the EFM encoder checkpoint and pair it with a compatible prediction head trained on source data for the supported task. The head has a shared feature layer and task-specific linear output layers, and it is kept frozen throughout; no target labels are used. The Daphnet Freezing of Gait dataset (**?**), introduced in the associated study (Bächlin et al., 2010), provides source supervision for the freezing-of-gait output; it is used to train the compatible head, not to fit reference adapters or train the adapter generator. EEG-only inference sets the auxiliary residual to zero; multimodal inference adds the adapter-generated residual before the shared head layer. Other source labels are used offline to fit reference adapters.

For a fused representation $\mathbf{z}^{\mathrm{fused}} = \mathbf{z}^{\mathrm{EEG}} + \Delta\mathbf{z}^{\mathrm{aux}}$, the head is

$$\mathbf{u} = \mathrm{GELU}\big(W_s\,\mathrm{LN}(\mathbf{z}^{\mathrm{fused}}) + \mathbf{b}_s\big), \qquad \mathbf{u} \in \mathbb{R}^{128}, \tag{A.1}$$

$$\boldsymbol{\ell}_t = W_t\mathbf{u} + \mathbf{b}_t \in \mathbb{R}^{K_t}, \qquad \mathbf{p}_t = \mathrm{softmax}(\boldsymbol{\ell}_t). \tag{A.2}$$

The shared projection $(W_s, \mathbf{b}_s)$ is specific to a backbone, as are the task output layers $(W_t, \mathbf{b}_t)$. For a given task t, the same output layer is used across its datasets. The adapter and the frozen head therefore share the same prediction path in EEG-only and multimodal evaluation.

We train the generators for CBraMod (Wang et al., 2025), CSBrain (Zhou et al., 2025), and CodeBrain (Ma et al., 2026) separately. The results across these backbones evaluate separate instantiations of the same framework. They do not represent the transfer of one shared generator checkpoint from one backbone architecture to another. Within each instantiation, the adapter layout, decoder output, frozen EEG representation, and prediction head form the applicable model interface. The block schema is shared within an interface, but the number of blocks and total parameter count depend on the active modalities.

Throughout the appendix, $f, h, e, d, E_{\mathrm{cond}}$, and s_θ refer to the selected backbone instantiation. The conditioning input remains the recording and task; the selected generator determines the backbone interface.

A.3 Signals and adapter parameters

Tables A.1 and A.2 collect the notation used below. As in the main text, K denotes the number of classes and C the fused condition.

Table A.1: Signal and adapter notation. Dimensions are symbolic here; numerical settings are listed in Table H.1.

Symbol	Meaning
X^{EEG}	EEG input to the frozen encoder.
$\mathcal{M}$	Complete set of supported companion modalities.
$A \subseteq \mathcal{M}$	Companion modalities available in a particular recording.
X^A	Collection $\{X^m : m \in A\}$ of available companion inputs.
$R = \{r_i\}$	Recording samples, with $r_i = (X_i^{\mathrm{EEG}}, X_i^A)$.

Symbol	Meaning
R^{fit}	Source samples whose labels are used to fit a reference adapter.
R^{cond}	Unlabeled samples used to construct the condition.
R^{fun}	Source samples used for functional supervision, disjoint from fitting and conditioning.
R^{qry}	Target query samples, disjoint from target conditioning.
τ	Supported prediction task associated with the frozen head.
f, h	Frozen EEG encoder and prediction head.
$Z = f(X^{\mathrm{EEG}})$	EEG representation before the adapter.
$a(Z, X^A; \omega)$	Multimodal adapter, with parameter vector $\omega \in \mathbb{R}^{P(A)}$.
$Z_A, L^A(\omega)$	Adapted representation and logits $h(Z_A)$.
$\omega^*, \widetilde{\omega}, \widehat{\omega}$	Fitted reference, reconstructed reference, and generated adapter.
ω^{sh}	Parameters of the shared coordinator.
$\omega^m_{\mathrm{temp}}, \omega^m_{\mathrm{spec}}, \omega^m_{\mathrm{proj}}$	Temporal, spectral, and projection parameters for modality m.
$B, p_b, P(A)$	Active block count $B = 3\lvert A\rvert + 1$, block size, and configuration-specific parameter count.

A.4 Latents and conditioning

The pre-contextual adapter token is denoted $U_{b,\mathrm{pre}}$, following the main text. It is distinct from the signal features V_i^m and V^m used for conditioning. The deterministic latent U_0 is obtained by substituting posterior means before the cross-block transformer; it is not a new encoder architecture.

Table A.2: Latent, condition, and optimization notation.

Symbol	Meaning
$e(\cdot; \eta)$	Stochastic block encoder, including the cross-block transformer.
$\bar{e}(\cdot; \eta)$	Deterministic encoder path using posterior means.
μ_b, ℓ_b	Mean and log variance of the Gaussian for block b; ℓ_b is introduced here for the posterior expansion.
$U_{b,\mathrm{pre}}$	Sampled block token before contextualization.
E_b, CBT	Block identity embedding and cross-block transformer.
U, U_0, U_t	Contextualized sample, canonical latent, and noise-corrupted canonical latent.
$\widehat{U}_0$	Clean-latent estimate computed from a noisy latent and predicted noise.
$d(\cdot; \psi)$	MoE decoder mapping latent tokens to adapter weights.
$\pi_{b,j}, r_j$	Routing weight and reconstruction expert for expert j.
V_i^m, V^m	Sample-level and recording-level condition features for modality m.
$e^m_{\mathrm{mod}}, q_{\mathrm{sub}}$	Modality embedding and task-independent attention query.
$F_{\mathrm{mod}}, F_{\mathrm{sub}}, E_{\mathrm{task}}$	Set pooling, attention pooling, and task embedding.
$C_{\mathrm{mod}}, C_{\mathrm{sub}}, C_\tau$	Modality, subject, and task factors.
$C, E_{\mathrm{cond}}(\cdot; \xi), F_{\mathrm{cond}}$	Fused condition, complete condition encoder, and factor-fusion function.
$\epsilon, \epsilon_\theta, s_\theta$	Gaussian noise, conditional noise predictor, and 50-step deterministic DDIM sampler using ϵ_θ.
$t, \bar{\alpha}_t$	Diffusion timestep and cumulative signal coefficient.
δ_{fun}	Reference-to-student predictive KL averaged on R^{fun}.
$\mathcal{L}_{\mathrm{AE}}, \mathcal{L}_{\mathrm{gen}}$	Stage 1 and Stage 2 objectives.

A.5 Parameter ownership

The learning stages update different parameter sets. Table A.3 separates parameter updates from differentiable use. A frozen decoder remains part of the computational graph for a loss on its output, even though the optimizer does not change its parameters.

Table A.3: Parameter updates in the ZeroMAG pipeline. "Fixed" describes parameter values, not a zero input Jacobian.

Component	Reference fitting	Stage 1	Stage 2	Deployment
EEG encoder and head	Fixed	Fixed	Fixed	Fixed
Reference adapter ω^*	Updated	Fixed	Fixed	Not required
Encoder η	Not used	Updated	Fixed	Not required
Decoder ψ	Not used	Updated	Fixed	Fixed
Condition encoder ξ	Not used	Not used	Updated	Fixed
Denoiser θ	Not used	Not used	Updated	Fixed

Each generated $\widehat{\omega}$ is an output of the trained system, rather than an optimizer state fitted to the target. Generation can involve repeated denoiser evaluations without any target-side gradient update. This distinction separates iterative sampling from iterative supervised adaptation.

B Adapter and Condition

B.1 Variable-length block layout

Within a backbone instantiation, only the active companion set $A \subseteq \mathcal{M}$ contributes modality-specific parameters:

$$\omega = \left[\omega^{\mathrm{sh}}; \{\omega^m_{\mathrm{temp}}, \omega^m_{\mathrm{spec}}, \omega^m_{\mathrm{proj}}\}_{m\in A}\right] \in \mathbb{R}^{P(A)}. \tag{B.1}$$

Each active modality contributes three blocks, with one additional shared block:

$$M = |A|, \qquad B = 3M + 1, \qquad U \in \mathbb{R}^{B\times 128}. \tag{B.2}$$

Thus $M = 1, 2, 3$ gives $B = 4, 7, 10$, respectively; ten is not a universal block count. Missing modalities create neither parameter blocks nor placeholder latent tokens. Block-type and modality conditioning, together with attention masks, support variable-length sequences while preserving the same functional block schema.

With block size p_b, the parameter count is

$$P(A) = p_{\mathrm{sh}} + \sum_{m\in A}(p^m_{\mathrm{temp}} + p^m_{\mathrm{spec}} + p^m_{\mathrm{proj}}) = \sum_{b=1}^{B} p_b. \tag{B.3}$$

For the active sequence, let $o_b = \sum_{r<b} p_r$ and $\omega_b = \omega[o_b : o_b + p_b]$. Offsets are defined within that configuration; semantic block identity is determined by type and modality, not by a universal vector position.

B.2 Temporal and spectral branches

An active companion modality $m \in A$ contributes two feature streams:

$$H^m_{\mathrm{temp}} = T^m(X^m; \omega^m_{\mathrm{temp}}), \tag{B.4}$$

$$H^m_{\mathrm{spec}} = S^m(\mathrm{FFT}(X^m); \omega^m_{\mathrm{spec}}), \tag{B.5}$$

$$F^m = P^m([H^m_{\mathrm{temp}}; H^m_{\mathrm{spec}}]; \omega^m_{\mathrm{proj}}). \tag{B.6}$$

The temporal stream represents variation in the input sequence, while the spectral stream provides frequency-domain information. Projection combines the two streams into the interface consumed by the shared coordinator. The distinct parameter groups allow weight generation to preserve the functional role of each block instead of treating the entire adapter as an unstructured vector.

The companion projection is distinct from the EFM encoder. It changes how additional signals enter the frozen representation, while the original EEG mapping f remains unchanged. The generated weights therefore specify a multimodal extension of an existing representation rather than a replacement EEG backbone.

B.3 Shared coordination and masking

The coordinator processes the EEG representation and available companion features to produce an auxiliary residual:

$$\Delta Z_{\text{aux}} = K(Z, \{F^m : m \in A\}; \omega^{\text{sh}}), \qquad Z_A = Z + \Delta Z_{\text{aux}}. \tag{B.7}$$

Only available branches contribute. For an empty companion set, the adapter returns the EEG representation directly:

$$a(Z, \varnothing; \omega) = Z, \qquad L_t^{\varnothing}(\omega) = h_t(f(X^{\text{EEG}})). \tag{B.8}$$

For EEG-only inference, the auxiliary residual is zero and the frozen EEG path is unchanged. Unavailable modalities have no parameter blocks or latent tokens, and therefore make no contribution to the prediction path.

B.4 Three condition factors

The condition encoder uses unlabeled calibration signals and the supported task:

$$C_{\text{mod}} = F_{\text{mod}}(\{V^m + e^m_{\text{mod}}\}_{m \in \{\text{EEG}\} \cup A}), \tag{B.9}$$

$$C_{\text{sub}} = F_{\text{sub}}(q_{\text{sub}}, \{V_i^m\}_{m \in \{\text{EEG}\} \cup A, i}), \tag{B.10}$$

$$C_\tau = E_{\text{task}}(\tau; \xi_\tau), \tag{B.11}$$

$$C = E_{\text{cond}}(R^{\text{cond}}, \tau; \xi) = F_{\text{cond}}([C_{\text{mod}}; C_{\text{sub}}; C_\tau]). \tag{B.12}$$

Here V_i^m is a sample-level feature and V^m aggregates a recording's features for modality m. The embeddings e^m_{mod} identify signal types. The subject query q_{sub} is task-independent; the task enters separately through C_τ and the fusion function F_{cond}.

The subject factor is inferred from the observed recording, not retrieved from a target-subject identifier. Its role is to represent recording-specific physiology that is absent from modality availability alone. Shuffling this factor across recordings preserves a collection of feature vectors but breaks their correspondence to the recording for which an adapter is generated, as examined in Appendix G.

B.5 Order and identity

Set pooling follows the permutation-invariant viewpoint of Deep Sets (Zaheer et al., 2017). Reordering signal-feature/identity pairs leaves the modality factor unchanged. This invariance does not apply to exchanging signal identities while holding the feature values fixed: assigning an ECG identity to an eye-movement feature changes the meaning of the condition.

For attention pooling (Vaswani et al., 2017) with a fixed query and consistently permuted key/value pairs, a permutation only reorders the terms in the weighted sum. To see this, write $c = \sum_i a_i v_i$, where $a_i = \exp(s_i)/\sum_j \exp(s_j)$. A permutation ρ gives

$$\sum_i \frac{\exp(s_{\rho(i)})}{\sum_j \exp(s_{\rho(j)})} v_{\rho(i)} = \sum_i a_i v_i. \tag{B.13}$$

This identity explains the distinction between order invariance and semantic identity. It assumes that scores and values are permuted together, as required for pooling a set of features.

B.6 Support of the condition

The task factor identifies a supported prediction task rather than a dataset identifier. Dataset holdout and task holdout are distinct: a new dataset can share a task with source data, whereas a task-holdout target has no source adapter for its task. FoG is the task-holdout target in our benchmark; its task factor C_τ is a learned unspecified-task embedding. Similarly, a new combination of supported signals differs from a signal type outside $\mathcal{M}$. The supported block schema defines the modality interface; it does not by itself create a new encoder for an arbitrary signal type.

C Datasets and Additional Results

C.1 Evaluation targets

The displayed benchmark covers six targets and four task families: ISRUC-S3 (Khalighi et al., 2016), HMC (Alvarez-Estevez & Rijsman, 2021), and MASS (O'Reilly et al., 2014) for sleep staging; SEED-V (Liu et al., 2022) for emotion recognition; EEGMAT (Zyma et al., 2019) for cognitive-load estimation; and FoG (Zhang et al., 2022) for freezing-of-gait detection. This section analyzes the six held-out target datasets in main Table 1. The full evaluation additionally includes CAP, DREAMER, and BCI-IV. The CBraMod ablation average in main Table 2 and Appendix G covers all nine target datasets; results on CAP, DREAMER, and BCI-IV are reported in Table C.1.

Table C.1: B-ACC (%, mean±std over five seeds) on CAP, DREAMER, and BCI-IV. Protocol and layout as in main Table 1.

Method	CAP	DREAMER	BCI-IV
CBraMod			
EEG-only	53.86±0.83	43.91±1.04	37.82±1.31
Supervised tuning	68.16±0.64	60.20±0.98	51.11±0.89
Mean source weights	53.68±0.97	43.45±1.35	37.54±1.70
Nearest source weights	54.78±0.55	45.08±1.23	39.26±0.91
Direct MLP	55.93±0.65	46.05±0.74	41.00±1.61
ZeroMAG	66.11±0.69	58.07±0.98	49.69±1.72
CSBrain			
EEG-only	54.23±0.79	44.38±0.98	38.57±1.26
Supervised tuning	66.30±0.97	59.77±1.05	52.62±1.06
Mean source weights	54.67±1.02	43.71±1.07	38.41±1.21
Nearest source weights	55.59±1.06	45.92±0.94	39.54±1.50
Direct MLP	56.91±1.07	46.85±1.34	40.29±1.77
ZeroMAG	64.49±0.56	58.10±1.05	51.24±1.11
CodeBrain			
EEG-only	53.12±0.91	43.46±1.11	37.24±1.38
Supervised tuning	67.50±0.59	55.46±0.96	50.23±1.27
Mean source weights	52.50±0.70	43.40±0.97	37.02±0.93
Nearest source weights	53.77±0.55	44.37±0.78	38.48±1.33
Direct MLP	54.19±1.03	45.14±0.95	38.87±1.76
ZeroMAG	65.60±0.74	54.51±1.14	48.09±1.71

Table C.2: Task coverage and holdout type of the target datasets.

Target	Task	Holdout
ISRUC-S3	Sleep staging	Dataset
HMC	Sleep staging	Dataset
MASS	Sleep staging	Dataset
SEED-V	Emotion recognition	Dataset
EEGMAT	Cognitive-load estimation	Dataset
FoG	Freezing-of-gait detection	Task
CAP	Sleep staging	Dataset
DREAMER	Emotion recognition	Dataset
BCI-IV-2a (BCI-IV)	Motor imagery	Dataset

C.2 Controlled comparisons

Within each backbone group, the EEG-only model provides the original frozen prediction path. Mean source weights and nearest source weights reuse fitted source adapters; Direct MLP predicts

weights from the condition. These methods and ZeroMAG use the same frozen EEG encoder and head. The supervised reference is a separate multimodal fine-tuning experiment using the labeled target training split and all available modalities. Its purpose is to measure the gap between ZeroMAG and supervised adaptation; it is not a step in ZeroMAG training or inference and is not a mathematical upper bound.

The cross-paradigm comparisons use PhysioOmni (Fu et al., 2025), the original ICML 2024 SleepFM (Thapa et al., 2024), and Brant-X (Zhang et al., 2024). These models have native backbones and different multimodal learning procedures. Their scores are the evaluations reported in this paper, rather than scores taken from the cited publications. In particular, the SleepFM reference here is the original `sleepfm-codebase` model, not the later clinical model.

C.3 Results by task family

Tables C.3 and C.4 collect all means and standard deviations from main Table 1 in two task-family views. Both tables use the same method order.

Table C.3: Sleep staging: balanced accuracy (%, mean ± reported SD over five seeds). Bold identifies ZeroMAG in each backbone group.

Method	ISRUC-S3	HMC	MASS
CBraMod			
EEG-only	75.83 ± 0.76	72.78 ± 0.50	74.29 ± 0.58
Supervised reference	84.04 ± 0.88	80.81 ± 0.41	82.41 ± 0.68
Mean source weights	74.31 ± 0.82	72.93 ± 0.49	75.91 ± 0.63
Nearest source weights	76.63 ± 0.61	73.18 ± 0.52	76.35 ± 0.51
Direct MLP	76.89 ± 0.73	73.43 ± 0.55	77.48 ± 0.65
ZeroMAG	$\mathbf{83.59 \pm 0.59}$	$\mathbf{80.29 \pm 0.42}$	$\mathbf{82.03 \pm 0.52}$
CSBrain			
EEG-only	75.94 ± 0.93	73.31 ± 0.48	75.89 ± 0.54
Supervised reference	84.10 ± 1.43	81.39 ± 0.39	84.15 ± 0.62
Mean source weights	75.42 ± 0.62	73.51 ± 0.45	77.72 ± 0.58
Nearest source weights	76.74 ± 0.55	73.82 ± 0.52	77.98 ± 0.63
Direct MLP	79.18 ± 0.71	74.21 ± 0.48	80.45 ± 0.72
ZeroMAG	$\mathbf{83.67 \pm 0.61}$	$\mathbf{80.83 \pm 0.52}$	$\mathbf{83.74 \pm 0.83}$
CodeBrain			
EEG-only	76.57 ± 0.73	72.95 ± 0.49	77.24 ± 0.51
Supervised reference	84.92 ± 1.61	80.93 ± 0.38	85.67 ± 0.58
Mean source weights	77.02 ± 0.68	73.28 ± 0.42	79.12 ± 0.72
Nearest source weights	77.22 ± 0.62	73.65 ± 0.48	80.21 ± 0.68
Direct MLP	79.92 ± 0.55	73.92 ± 0.51	81.65 ± 0.61
ZeroMAG	$\mathbf{84.44 \pm 0.77}$	$\mathbf{80.43 \pm 0.48}$	$\mathbf{85.25 \pm 0.72}$
Multimodal physiological models			
PhysioOmni	84.62 ± 0.63	81.05 ± 0.42	85.50 ± 0.65
SleepFM	83.27 ± 0.61	80.01 ± 0.47	81.61 ± 0.52
Brant-X	85.19 ± 0.54	81.51 ± 0.46	86.07 ± 0.59

Sleep staging. ZeroMAG improves over EEG-only by 7.48–7.87 points on ISRUC-S3 and HMC, and by 7.74–8.01 points on MASS. Its smallest supervised-reference gap in the displayed benchmark is 0.38 points for CBraMod on MASS. Improvements occur for all three separately trained backbone instantiations, although their absolute scores differ.

Emotion recognition. SEED-V has smaller absolute improvements over EEG-only, at 4.21, 4.36, and 4.32 points for CBraMod, CSBrain, and CodeBrain. Its recovery of supervised gain is also lower than that of the other displayed targets, ranging from 87.34% to 88.71%. Thus, the lower gain is not explained solely by a smaller supervised-minus-EEG baseline interval. The table does not identify the source of this task-dependent gap.

Table C.4: Emotion, cognitive load, and gait: balanced accuracy (%, mean ± reported SD over five seeds). Bold identifies ZeroMAG in each backbone group.

Method	SEED-V	EEGMAT	FoG
CBraMod			
EEG-only	40.62 ± 1.24	73.38 ± 2.48	76.61 ± 1.35
Supervised reference	45.44 ± 1.58	81.46 ± 1.93	85.02 ± 1.21
Mean source weights	39.83 ± 1.13	73.52 ± 0.92	76.73 ± 0.78
Nearest source weights	40.68 ± 1.37	74.63 ± 0.81	77.52 ± 0.73
Direct MLP	40.99 ± 1.05	75.21 ± 1.02	78.03 ± 0.62
ZeroMAG	$\mathbf{44.83 \pm 1.22}$	$\mathbf{80.99 \pm 0.81}$	$\mathbf{84.47 \pm 0.72}$
CSBrain			
EEG-only	41.38 ± 1.32	75.38 ± 2.67	75.96 ± 1.52
Supervised reference	46.32 ± 1.17	83.57 ± 1.43	85.13 ± 1.31
Mean source weights	41.75 ± 0.62	75.91 ± 0.71	77.08 ± 0.65
Nearest source weights	42.28 ± 0.55	75.83 ± 0.66	78.73 ± 0.59
Direct MLP	42.69 ± 0.48	78.12 ± 0.58	80.02 ± 0.52
ZeroMAG	$\mathbf{45.74 \pm 0.44}$	$\mathbf{83.08 \pm 0.71}$	$\mathbf{84.60 \pm 0.65}$
CodeBrain			
EEG-only	41.52 ± 1.46	76.62 ± 2.83	76.77 ± 1.26
Supervised reference	46.39 ± 1.34	85.02 ± 1.63	85.25 ± 1.33
Mean source weights	41.88 ± 0.58	77.22 ± 0.66	77.85 ± 0.71
Nearest source weights	42.35 ± 0.52	77.98 ± 0.59	79.22 ± 0.65
Direct MLP	42.87 ± 0.45	79.28 ± 0.52	80.55 ± 0.58
ZeroMAG	$\mathbf{45.84 \pm 0.55}$	$\mathbf{84.51 \pm 0.63}$	$\mathbf{84.65 \pm 0.58}$
Multimodal physiological models			
PhysioOmni	46.04 ± 1.32	84.81 ± 2.13	84.93 ± 0.53
SleepFM	44.48 ± 0.58	80.54 ± 0.72	84.09 ± 0.68
Brant-X	46.49 ± 0.63	85.31 ± 0.61	85.37 ± 0.72

Cognitive load and gait. On EEGMAT, the improvements over EEG-only are 7.61, 7.70, and 7.89 points. The corresponding FoG improvements are 7.86, 8.64, and 7.88 points. CSBrain on FoG has the largest displayed gain over EEG-only, whereas CodeBrain has the highest absolute ZeroMAG score on both targets. The best absolute score and the largest incremental gain therefore need not identify the same backbone.

C.4 Gains across backbones

Fig. C.1 compares the improvement over EEG-only and Direct MLP with a common color scale. All 18 backbone–dataset settings have positive mean differences for both comparisons. The largest advantage over Direct MLP occurs on HMC with CBraMod (6.86 points); the smallest occurs on SEED-V with CodeBrain (2.97 points).

C.5 Aggregate comparisons

Let e_j, z_j, s_j, m_j be EEG-only, ZeroMAG, supervised, and Direct MLP balanced-accuracy means for setting j. Values are in percentage points. Define

$$\Delta_j^{\mathrm{EEG}} = z_j - e_j, \qquad \Delta_j^{\mathrm{MLP}} = z_j - m_j, \qquad G_j^{\mathrm{sup}} = s_j - z_j. \tag{C.1}$$

Table C.5 lists differences between reported means. Paired uncertainty requires the seed-wise covariance.

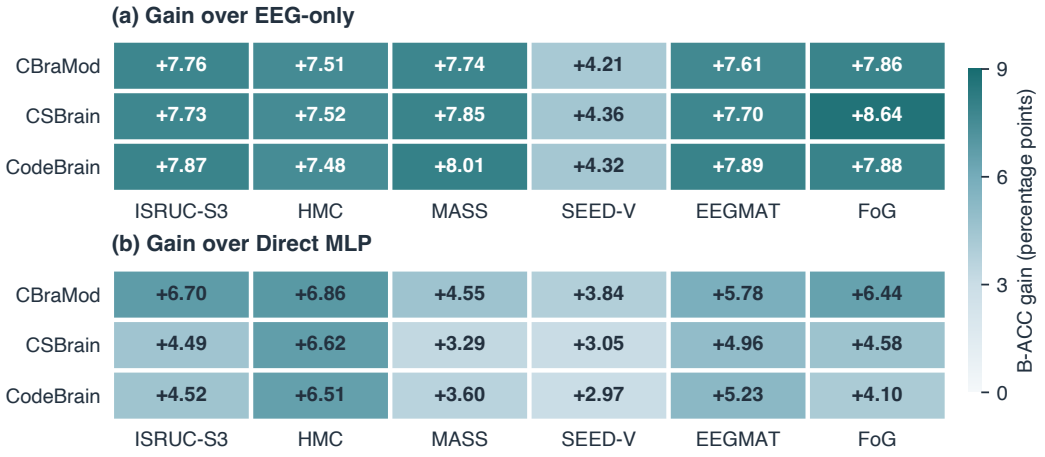

Figure C.1: Gains in balanced accuracy over EEG-only and Direct MLP, in percentage points. Both panels use the same scale. Values are differences of means from main Table 1; no paired uncertainty or significance is inferred.

Table C.5: Arithmetic derived from main Table 1; no additional experimental runs. Recovery is $100(z-e)/(s-e)$ for each row.

Backbone	Target	Δ^{EEG}	Δ^{MLP}	G^{sup}	Recovery (%)
CBraMod	ISRUC-S3	7.76	6.70	0.45	94.52
CBraMod	HMC	7.51	6.86	0.52	93.52
CBraMod	MASS	7.74	4.55	0.38	95.32
CBraMod	SEED-V	4.21	3.84	0.61	87.34
CBraMod	EEGMAT	7.61	5.78	0.47	94.18
CBraMod	FoG	7.86	6.44	0.55	93.46
CSBrain	ISRUC-S3	7.73	4.49	0.43	94.73
CSBrain	HMC	7.52	6.62	0.56	93.07
CSBrain	MASS	7.85	3.29	0.41	95.04
CSBrain	SEED-V	4.36	3.05	0.58	88.26
CSBrain	EEGMAT	7.70	4.96	0.49	94.02
CSBrain	FoG	8.64	4.58	0.53	94.22
CodeBrain	ISRUC-S3	7.87	4.52	0.48	94.25
CodeBrain	HMC	7.48	6.51	0.50	93.73
CodeBrain	MASS	8.01	3.60	0.42	95.02
CodeBrain	SEED-V	4.32	2.97	0.55	88.71
CodeBrain	EEGMAT	7.89	5.23	0.51	93.93
CodeBrain	FoG	7.88	4.10	0.60	92.92

C.5.1 Aggregate gain recovery

Averaging the 18 displayed means with equal weight yields 69.6133 for EEG-only, 76.8322 for ZeroMAG, 77.3344 for the supervised reference, and 71.9383 for Direct MLP. The corresponding gains are 7.22 and 4.89 points, with a 0.50-point supervised gap. The corresponding advantages over mean and nearest source weights are 6.7772 and 5.9989 points, rounding to 6.78 and 6.00.

Recovery can be aggregated in two ways:

$$R_{\text{ratio}} = 100\frac{\sum_j (z_j - e_j)}{\sum_j (s_j - e_j)}, \tag{C.2}$$

$$R_{\text{mean}} = \frac{100}{J}\sum_j \frac{z_j - e_j}{s_j - e_j}. \tag{C.3}$$

The ratio of aggregate gains is 93.50%; averaging row-wise recovery gives 93.12%. We use the ratio of aggregate gains for the 93.50% summary. The row-wise ratios are listed separately in Table C.5. All displayed supervised gains are positive.

The fraction of the Direct MLP gap closed by ZeroMAG is

$$100\frac{\sum_j (z_j - m_j)}{\sum_j (s_j - m_j)} = 90.6929\%, \tag{C.4}$$

This 90.69% value measures the fraction of the supervised-minus-MLP performance gap closed by ZeroMAG.

C.5.2 Backbone comparison

The mean improvement over EEG-only is 7.1150 points for CBraMod, 7.3000 for CSBrain, and 7.2417 for CodeBrain. Their rounded values are 7.12, 7.30, and 7.24. Their range is 0.185 points (approximately 0.19 points), calculated before rounding the backbone means.

The backbone-wise ratios of aggregate gains are 93.47%, 93.59%, and 93.42%. Across the 18 rows, the supervised-minus-ZeroMAG margin ranges from 0.38 to 0.61 points. ZeroMAG's displayed mean is higher than the EEG-only, source-transfer, and Direct MLP means in every setting. Significance is not inferred from these means.

C.6 Reported variation and the supervised gap

Fig. C.2 preserves the reported standard deviations for EEG-only, Direct MLP, ZeroMAG, and supervised fitting. These summarize seed-to-seed variation. They are not confidence intervals for pairwise differences, and overlapping whiskers do not establish either significance or equivalence.

The supervised-reference gap ranges from 0.38 to 0.61 points across the 18 settings, with mean 0.5022 points. This is a small residual difference in the displayed averages, but it is not a formal equivalence result. The corresponding gain-recovery ratios describe the position of ZeroMAG between EEG-only and supervised fitting, rather than the probability of recovering the fitted adapter parameters.

C.7 Cross-paradigm comparisons

With CodeBrain, ZeroMAG exceeds the original SleepFM on all six targets by an average of 1.8533 points. The average gaps to PhysioOmni and Brant-X are 0.3050 and 0.8033 points, respectively. These comparisons place the generated-adapter framework alongside native multimodal models; they do not isolate backbone architecture, pretraining exposure, or target adaptation as a single controlled factor.

PhysioOmni explicitly supports missing modalities, while Brant-X studies physiological-signal alignment and SleepFM learns multimodal sleep representations. The distinguishing question here is how a separately trained generator equips an existing frozen EEG model with an adapter. Missing-modality support alone does not separate these approaches.

D Training and Deployment

D.1 Two training stages

ZeroMAG has two learned stages followed by deployment. Stage 1 learns the adapter representation and decoder; Stage 2 learns the conditional diffusion model and condition encoder while retaining

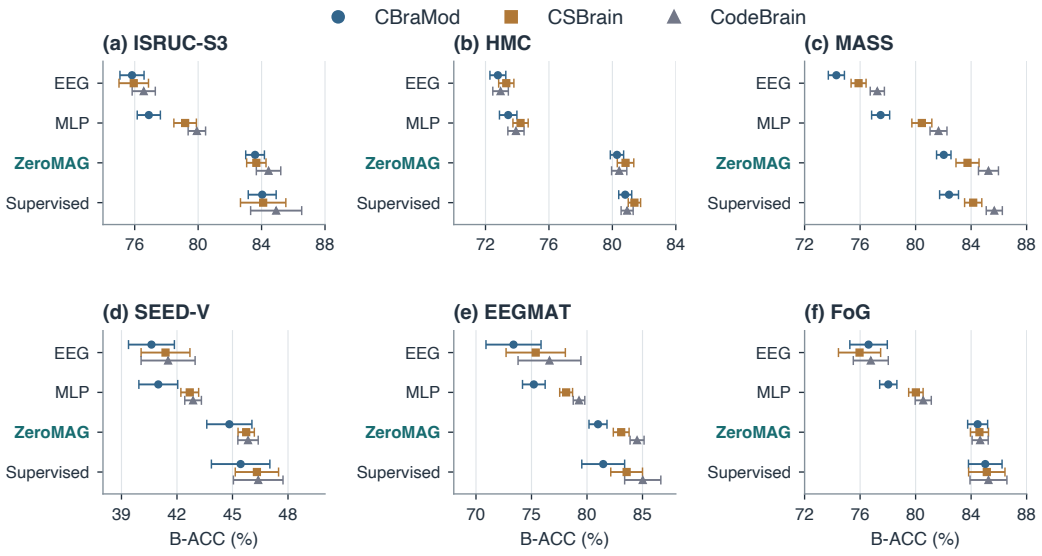

Figure C.2: Means and reported ± 1 SD for four methods within each target. Marker shapes identify backbones. Horizontal limits vary by target to show within-target differences; all values are balanced accuracy (%).

the Stage 1 representation. Fig. D.1 shows their inputs, outputs, and update boundaries. The target deployment path is an inference procedure, not a third training stage.

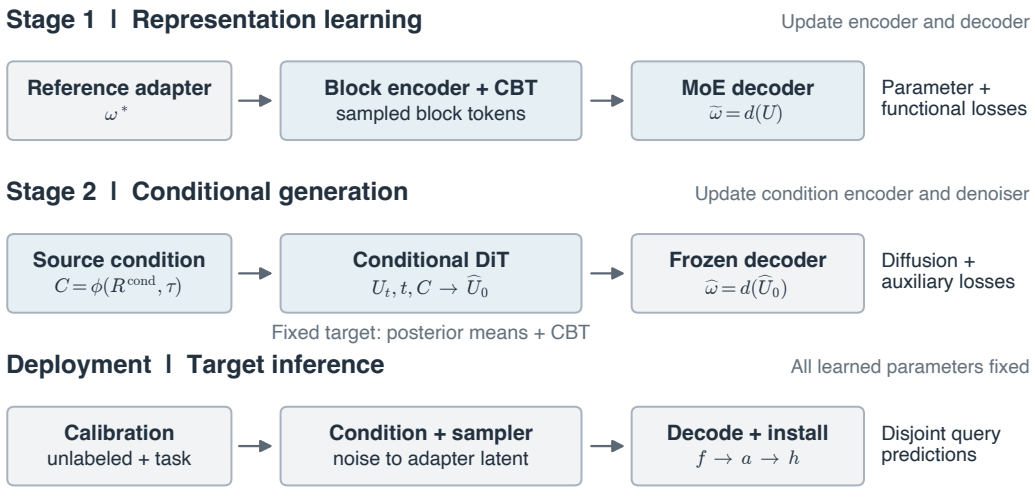

Figure D.1: Training and deployment within one separately trained backbone instantiation. Stage 1 produces the encoder and decoder. Stage 2 uses a fixed canonical latent and propagates auxiliary-loss gradients through the frozen decoder. Deployment uses unlabeled calibration data and fixed model parameters.

D.2 Reference adapters

For a source recording, the reference adapter solves the supervised fitting problem with the EFM encoder and head fixed:

$$\omega^* = \arg\min_{\omega \in \mathbb{R}^{P(A)}} \sum_{r_i \in R^{\mathrm{fit}}} \ell(L_i^A(\omega), Y_i). \tag{D.1}$$

Its weights provide the parameter target for representation learning and conditional generation. Its predictions on the disjoint source subset R^{fun} provide the functional target. Thus, reference fitting and functional supervision use different source samples, and the latter still belongs to offline training rather than to target evaluation.

D.3 Stage 1: representation learning

For each block, the encoder predicts a Gaussian mean and variance, samples a pre-contextual token, and applies the cross-block transformer:

$$U_{b,\mathrm{pre}} = \mu_b + \exp(\ell_b/2) \odot \epsilon_b, \qquad \epsilon_b \sim \mathcal{N}(0, I), \tag{D.2}$$

$$U = \mathrm{CBT}([U_{b,\mathrm{pre}} + E_b]_{b=1}^B), \tag{D.3}$$

$$\widetilde{\omega} = d(U; \psi). \tag{D.4}$$

The representation follows variational encoding (Kingma & Welling, 2014); the shared block schema and functional objective specify how it is applied to adapter weights. Stage 1 combines block reconstruction, direction alignment, functional agreement, posterior regularization, and expert balancing (Fedus et al., 2022):

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{blk}} + 0.1\mathcal{L}_{\cos} + 10^{-4}\mathcal{L}_{\mathrm{KL}} + 1.0\delta_{\mathrm{fun}}(\omega^*, \widetilde{\omega}) + 0.01\mathcal{L}_{\mathrm{bal}}. \tag{D.5}$$

The reconstruction term acts on block parameters, the cosine term on direction, the KL term on the Gaussian posterior, and the balancing term on expert use. Functional supervision constrains the decoded adapter's predictions through the frozen EFM. The terms address different properties and do not turn the whole objective into a pure likelihood bound.

Algorithm D.1 Stage 1: adapter representation learning

Require: Source references ω^* and associated R^{fun}; fixed f, h.
Ensure: Learned encoder η, decoder ψ, and canonical targets U_0.

```
while Stage 1 training continues do
    Sample source reference adapters and associated functional inputs.
    Encode blocks into (μ_b, ℓ_b) and sample U_{b,pre}.
    U ← CBT([U_{b,pre} + E_b]_{b=1}^B).
    ω̃ ← d(U; ψ).
    Evaluate L_AE in Eq. (D.5).
    Update η, ψ; keep f, h, ω* fixed.
end while
Freeze η, ψ.
for each source reference ω* do
    U_0 ← ē(ω*; η) = CBT([μ_b + E_b]_{b=1}^B).
end for
```

D.4 Stage 2: conditional generation

The canonical target fixes a deterministic coordinate for each reference adapter:

$$U_0 = \bar{e}(\omega^*; \eta) = \mathrm{CBT}([\mu_b + E_b]_{b=1}^B). \tag{D.6}$$

The condition $C = E_{\mathrm{cond}}(R^{\mathrm{cond}}, \tau; \xi)$ guides the diffusion-transformer noise predictor (Peebles & Xie, 2023). The noise-prediction, latent, weight, and functional terms jointly train θ and ξ. The latter three are active only when $\bar{\alpha}_t > 0.5$:

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{diff}} + \mathbf{1}[\bar{\alpha}_t > 0.5]\,(0.1\mathcal{L}_{\mathrm{lat}} + 0.1\mathcal{L}_{\mathrm{w}} + 1.0\delta_{\mathrm{fun}}(\omega^*, \widehat{\omega}))\,. \tag{D.7}$$

The diffusion term is active at every timestep. Expert balancing is used only in Stage 1; neither stage uses a diversity loss.

Algorithm D.2 Stage 2: conditional adapter generation

Require: Source triples (R, τ, ω^*); fixed η, ψ, f, h and U_0.
Ensure: Learned denoiser θ and condition encoder ξ.

while Stage 2 training continues **do**
  Sample a source triple and unlabeled $R^{\mathrm{cond}} \subseteq R^{\mathrm{fit}}$.
  $C \leftarrow E_{\mathrm{cond}}(R^{\mathrm{cond}}, \tau; \xi)$.
  Sample a training timestep t and $\epsilon \sim \mathcal{N}(0, I)$.
  $U_t \leftarrow \sqrt{\bar{\alpha}_t} U_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.
  Predict $\epsilon_\theta(U_t, t, C)$ and evaluate $\mathcal{L}_{\mathrm{diff}}$.
  if $\bar{\alpha}_t > 0.5$ **then**
    Estimate $\widehat{U}_0$ using Eq. (E.9).
    $\widehat{\omega} \leftarrow d(\widehat{U}_0; \psi)$.
    Add weighted latent, weight, and functional losses.
  end if
  Update θ, ξ, propagating auxiliary gradients through fixed d.
end while

D.5 Deployment

For an unseen recording, generation requires only the unlabeled calibration subset, the supported task, and the selected backbone's trained generator and decoder. No reference adapter is required. Query samples are excluded from condition construction.

Algorithm D.3 Target inference with fixed parameters

Require: Unlabeled R^{cond}, task τ, query inputs R^{qry}; fixed model components.
Ensure: Query predictions with a generated adapter.

$C \leftarrow E_{\mathrm{cond}}(R^{\mathrm{cond}}, \tau; \xi)$.
Draw $\epsilon \sim \mathcal{N}(0, I)$.
$\widehat{\omega} \leftarrow d(s_\theta(\epsilon, C); \psi)$.
for each query sample $(X_i^{\mathrm{EEG}}, X_i^A)$ **do**
  $Z_i \leftarrow f(X_i^{\mathrm{EEG}})$.
  $\widehat{Y}_i \leftarrow \arg\max h(a(Z_i, X_i^A; \widehat{\omega}))$.
end for

The sampled initial noise makes adapter generation stochastic even when the remaining inference operations are fixed. This differs from posterior sampling during Stage 1: deployment samples a conditional adapter latent, whereas Stage 1 samples a representation of a known reference vector. Neither process requires target labels.

E Objective Analysis

E.1 Canonical latent coordinates

The block posterior is defined before cross-block contextualization. Let F_η denote adding block embeddings and applying CBT. For pre-contextual tokens U_{pre}, the stochastic output is $F_\eta(U_{\mathrm{pre}})$ and the canonical output is $F_\eta(\mu)$. In general,

$$F_\eta(\mathbb{E}[U_{\mathrm{pre}}]) \neq \mathbb{E}[F_\eta(U_{\mathrm{pre}})]. \tag{E.1}$$

For example, a scalar nonlinear map $F(u) = u^2$ gives $F(\mathbb{E}[u]) = \mu^2$ but $\mathbb{E}[F(u)] = \mu^2 + \sigma^2$. The same distinction applies to nonlinear contextualization. The Stage 2 target is the explicit means-before-CBT path in Eq. (D.6), not an estimate of the exact mean after a nonlinear transformer.

E.2 Posterior regularization

For a diagonal Gaussian posterior $q_b = \mathcal{N}(\mu_b, \mathrm{diag}(\exp \ell_b))$, its KL to a standard normal has the closed form

$$D_{\mathrm{KL}}(q_b \| \mathcal{N}(0, I)) = \frac{1}{2} \sum_{r=1}^{d} (\mu_{b,r}^2 + \exp \ell_{b,r} - 1 - \ell_{b,r}). \tag{E.2}$$

Here d is token width and r indexes a token coordinate. This identity concerns the pre-contextual posterior. It does not imply that the distribution after CBT is diagonal Gaussian. It also does not specify whether a training implementation sums or averages over blocks; a change of reduction changes the scale of the corresponding loss coefficient.

For one coordinate, the partial derivatives are $\partial D_{\mathrm{KL}}/\partial \mu = \mu$ and $\partial D_{\mathrm{KL}}/\partial \ell = (\exp \ell - 1)/2$. The minimum occurs at zero mean and unit variance when this term is considered alone. Reconstruction and functional objectives compete with that regularizer, so a trained posterior is not expected to minimize the KL term independently of the other losses.

E.3 Block reconstruction and direction

Parameter reconstruction and cosine alignment provide distinct constraints. A positive rescaling of a nonzero reference block, $\widetilde{\omega}_b = c\omega_b^*$ with $c > 0$, has cosine similarity one even when $c \neq 1$. Its squared reconstruction error is $(c-1)^2 \|\omega_b^*\|_2^2$. Conversely, a small error in a large block can dominate a coordinate-wise reconstruction sum while a smaller block has a larger relative discrepancy.

Thus, block structure carries information that a whole-vector direction score alone does not recover. The reconstruction and direction objectives complement predictive supervision through the frozen EFM. Neither imposes a direct bound on target accuracy.

E.4 Functional discrepancy and its derivative

On a fixed source functional subset, write the reference probability vector as $p_i = \mathrm{softmax}(L_i^A(\omega^*))$ and the student vector as $q_i = \mathrm{softmax}(L_i^A(\omega'))$. The main discrepancy is

$$\delta_{\mathrm{fun}}(\omega^*, \omega') = \frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{K} p_{i,c} \log \frac{p_{i,c}}{q_{i,c}}. \tag{E.3}$$

Reference probabilities are constants for the student update. For a student logit $z_{i,c}$, the derivative is

$$\frac{\partial \delta_{\mathrm{fun}}}{\partial z_{i,c}} = \frac{1}{n}(q_{i,c} - p_{i,c}). \tag{E.4}$$

This follows by expressing the cross-entropy term as $-\sum_c p_c z_c + \log \sum_c \exp z_c$ and using $\sum_c p_c = 1$. It shows that the objective aligns distributions, not just the most likely class.

No labels are required to compute this discrepancy. Nonetheless, its reference predictions inherit the behavior of an adapter that was fitted using source labels. Functional matching is therefore not an unsupervised proof of correctness: it transfers the source reference's useful behavior and its errors. A confident but wrong teacher prediction is still a target of the KL term.

The direction of KL also matters. The main text uses reference-to-student KL; reversing its arguments produces a different objective. The expression uses the untempered softmax in the main formulation. Stable log-softmax evaluation preserves this objective without taking logarithms of rounded probabilities.

E.5 Prediction agreement

A small parameter distance alone does not ensure a small predictive error. Locally, for a differentiable logit map L,

$$L(\omega + \Delta\omega) - L(\omega) = J_L(\omega)\Delta\omega + o(\|\Delta\omega\|_2). \tag{E.5}$$

Sensitivity depends on the Jacobian and on the direction of the error. Two equally sized perturbations can have different functional effects. Conversely, multiple parameter vectors can implement the same predictions, so exact weight recovery is not necessary for functional agreement.

Prediction agreement admits a simple margin condition. Suppose p and q are two class-probability vectors, the reference winning class is c^*, and its margin is

$$m = p_{c^*} - \max_{c \neq c^*} p_c > 0. \tag{E.6}$$

If $\|p - q\|_\infty < m/2$, the student has the same unique winning class. Indeed, for any competitor c,

$$q_{c^*} - q_c \geq p_{c^*} - p_c - 2\|p - q\|_\infty > 0. \tag{E.7}$$

This statement concerns agreement with the reference, not agreement with the true label. It also does not convert a mean training-subset KL into a uniform target-query bound. A new dataset can shift both margins and discrepancies. Functional consistency is an objective evaluated through predictive behavior, rather than a universal guarantee of target generalization.

E.6 Forward diffusion and clean-latent estimation

Following the Gaussian diffusion formulation (Ho et al., 2020), for schedule coefficients $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{r=1}^{t} \alpha_r$, the forward marginal is

$$U_t = \sqrt{\bar{\alpha}_t} U_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{E.8}$$

The conditional denoiser predicts $\epsilon_\theta(U_t, t, C)$. Solving the forward relation for U_0 after substituting this prediction gives

$$\widehat{U}_0 = \frac{U_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(U_t, t, C)}{\sqrt{\bar{\alpha}_t}}. \tag{E.9}$$

No new assumption is needed for this algebra beyond $\bar{\alpha}_t > 0$. The estimator's quality does depend on learned noise prediction and numerical stability.

E.7 Low-noise supervision

Subtract the true U_0 from its estimator to obtain the exact identity

$$\widehat{U}_0 - U_0 = \sqrt{\frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t}}\big(\epsilon - \epsilon_\theta(U_t, t, C)\big). \tag{E.10}$$

Thus squared clean-latent error multiplies squared noise-prediction error by $(1 - \bar{\alpha}_t)/\bar{\alpha}_t$. This factor increases as the signal coefficient decreases. Low-noise gating in the main text is therefore consistent with avoiding strongly amplified reconstruction and functional gradients from heavily corrupted latents.

Writing $\mathrm{SNR}_t = \bar{\alpha}_t/(1 - \bar{\alpha}_t)$, the multiplier is $1/\mathrm{SNR}_t$. For a fixed noise-prediction residual, a lower signal-to-noise ratio therefore yields a larger clean-latent error. This algebra motivates applying the auxiliary losses at low-noise timesteps; the implementation uses $\bar{\alpha}_t > 0.5$ (SNR > 1) with a cosine noise schedule, as specified in Appendix H.1.

E.8 MoE decoding

Sparse mixture-of-experts models select a subset of experts for each input (Shazeer et al., 2017). In ZeroMAG, routing uses the contextualized token and its block identity:

$$\pi_b = \mathrm{softmax}(\mathrm{route}([U_b; E_b]; \psi_r)), \tag{E.11}$$

$$\widetilde{\omega}_b = \mathrm{head}_{\mathrm{type}(b)}\left(\sum_{j \in \mathrm{TopK}(\pi_b)} \pi_{b,j} r_j(U_b)\right). \tag{E.12}$$

The block type is temporal, spectral, projection, or shared. Experts act on latent tokens; the output heads map their mixtures back to the appropriate parameter blocks. The mixture uses the softmax routing weights $\pi_{b,j}$ for the selected experts, as in the main formulation.

Top-k selection restricts the active expert set. Away from routing-boundary changes, the selected computation is differentiable through expert outputs and routing weights. At a boundary, the selected set can change discontinuously. The presence of an expert-balancing objective does not establish uniform expert utilization; the main formulation uses it as a training regularizer.

E.9 Gradients through a frozen decoder and EFM

Write the student logit path as $z = h(a(f(X), X^A; d_\psi(\widehat{U}_0)))$. For an auxiliary loss $\mathcal{J}(z)$, the chain rule includes

$$\frac{\partial \mathcal{J}}{\partial \theta} = \frac{\partial \mathcal{J}}{\partial z}\frac{\partial z}{\partial \widehat{\omega}}\frac{\partial d_\psi(\widehat{U}_0)}{\partial \widehat{U}_0}\frac{\partial \widehat{U}_0}{\partial \theta}. \tag{E.13}$$

Freezing ψ means not updating its stored values. It does not set its input Jacobian to zero. Similarly, the head can remain frozen while its derivative transmits a functional error to generated adapter weights. Reference predictions may be detached; generated student predictions must preserve the relevant graph.

More explicitly, the Stage 2 parameter update satisfies $\psi^{(k+1)} = \psi^{(k)}$, whereas θ and ξ receive gradients. This fixed-parameter condition does not require the mathematical derivative $\partial \mathcal{J}/\partial \psi$ to vanish. The relevant derivative for generator learning is the decoder input Jacobian, which remains in the chain rule.

For the condition encoder, the same path additionally contains $\partial \widehat{U}_0/\partial C$ and $\partial E_{\text{cond}}/\partial \xi$. Thus the functional objective can shape the recording-derived condition even though the teacher, decoder, and EFM are fixed. Detaching the generated weights would eliminate this auxiliary supervision rather than merely freezing the decoder.

E.10 Latent error and functional error

Let $\widehat{U}_0 = U_0 + \Delta U$ and assume the decoder is differentiable at U_0. Locally,

$$d(U_0 + \Delta U; \psi) - d(U_0; \psi) = J_d(U_0)\Delta U + o(\|\Delta U\|). \tag{E.14}$$

Combining this with the logit Jacobian gives a first-order predictive perturbation $J_L J_d \Delta U$. Its size depends on both decoder sensitivity and the frozen EFM's response to the decoded adapter. Equal latent reconstruction errors can therefore yield different logit errors, motivating an objective that observes the final prediction path.

There are also two distinct discrepancies: reconstruction compares $d(\bar{e}(\omega^*))$ with ω^*, while generation compares $d(\widehat{U}_0)$ with ω^*. A generator that exactly recovers U_0 still inherits any reconstruction error in the learned representation. Conversely, a high-quality autoencoder does not ensure that the conditional diffusion model recovers the appropriate latent for every recording.

F Efficiency

F.1 Measured target-side time

Target-side wall-clock time was measured programmatically with timers in the code, under the same hardware and protocol for the two compared procedures. Condition construction and adapter generation took approximately 2 minutes, compared with approximately 15 minutes for supervised target adaptation. These measurements give

$$\text{speedup} = \frac{15}{2} = 7.5, \qquad \text{time reduction} = 100\left(1 - \frac{2}{15}\right) \approx 86.7\%. \tag{F.1}$$

The difference is approximately 13 minutes per measured adaptation procedure. These are elapsed-time measurements of the reported procedures, rather than isolated GPU kernel timings. The timed procedures exclude source-bank construction and offline generator training.

F.2 Cost decomposition

The target workflow consists of conditioning, iterative denoising, decoding, and installing the generated weights. A symbolic decomposition is

$$T_{\text{gen}} = T_{\text{cond}} + N_{\text{eval}} T_{\text{denoise}} + T_{\text{decode}} + T_{\text{install}}. \tag{F.2}$$

Here N_{eval} is the number of denoiser evaluations and each T denotes the corresponding elapsed cost under a fixed execution convention. This expression describes the operations; it does not assign

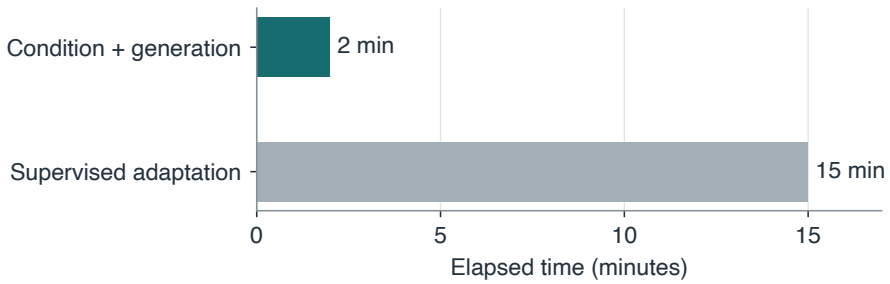

Figure F.1: Target-side wall-clock time measured programmatically: condition construction plus adapter generation versus supervised adaptation, under the same hardware and protocol. The 7.5× ratio uses the reported 2- and 15-minute measurements. No timing variance or component-wise timing breakdown is inferred.

portions of the measured time to individual components. Query prediction then adds the cost of the fixed EEG model and active adapter branches.

No target-side optimization is required, but generation still performs computation. The 7.5× measurement compares adaptation procedures; it does not establish a 7.5× speedup for every subsequent query prediction or for the entire offline-plus-online pipeline.

F.3 Amortization across targets

Let T_{off} denote source-bank construction and both training stages for one separately trained backbone instantiation. For N deployments, the generated-adapter route has cost $T_{\text{off}} + NT_{\text{gen}}$, excluding shared query prediction. The supervised route has incremental adaptation cost NT_{sup}. Under equal per-target costs, the generated route becomes cheaper when

$$N > \frac{T_{\text{off}}}{T_{\text{sup}} - T_{\text{gen}}}. \tag{F.3}$$

Using the measured target-side difference gives $N > T_{\text{off}}/(13\ \text{minutes})$. This is a symbolic break-even condition, not a measured deployment count: no numerical offline training cost is assigned here. Offline training is associated with a fixed source configuration and backbone; separate backbone instantiations have separate offline costs.

F.4 Latent and output complexity

For B tokens of width d, dense latent self-attention stores an attention map with B^2 entries per head. Token projections and feed-forward transformations also depend on B and d. Reducing latent width can reduce the denoiser computation, while the decoder still produces P scalar adapter coordinates. Compact latent generation does not remove the output storage and installation cost of the adapter itself.

Sparse expert selection limits the active reconstruction functions in the stated decoder. The runtime benefit depends on how expert computation is executed; the selection rule alone does not provide a measured speedup. Functional supervision adds an adapter-and-head prediction path during offline training, but this loss evaluation is absent from target deployment.

G Component Ablations

G.1 Component ablations

Main Table 2 reports Full ZeroMAG at 73.41 balanced accuracy and eight component variants averaged across all nine held-out target datasets. All variants come from one ablation run under the main protocol, separate from the runs in main Table 1 and Table C.1, where the CBraMod nine-target average of ZeroMAG is 70.01; each decrease is measured against Full ZeroMAG within the

ablation run; its three functional-supervision variants are analyzed in Appendix G.3. The arithmetic drops from 73.41 are 11.09 for task+modality only, 9.24 for shuffled subject condition, 11.07 for direct weight MLP, 8.18 for modular latent MLP, 5.07 for modular latent DiT without MoE, 4.16 for a shared modality encoder, 8.12 for temporal-only encoding, and 6.58 for mean fusion. All differences use the same full-model reference.

The reported task+modality reduction of 11.09 points corresponds to $100 \times 11.09/73.41 = 15.1069\%$ of the full score. This is a relative score reduction, not a 15.1 percentage-point change and not a loss of 15.1% of the supervised gain.

Table G.1: Component ablations from main Table 2. B-ACC and decrease are reported in percentage points as mean ± standard deviation over five seeds. The aggregation uses CBraMod over all nine target datasets, as in main Table 2.

Component	Variant	B-ACC	Decrease
Condition	Task + modality only	62.32 ± 0.41	11.09 ± 0.72
	Shuffled subject	64.17 ± 0.53	9.24 ± 0.98
Generator	Direct weight MLP	62.34 ± 0.67	11.07 ± 0.88
	Modular latent MLP	65.23 ± 0.62	8.18 ± 0.83
	Latent DiT, no MoE	68.34 ± 0.21	5.07 ± 0.71
Adapter	Shared encoder	69.25 ± 0.33	4.16 ± 0.81
	Temporal only	65.29 ± 0.72	8.12 ± 1.03
	Mean fusion	66.83 ± 0.32	6.58 ± 0.71
Full	ZeroMAG	73.41 ± 0.71	—

Ablation settings. The source bank, source training splits, frozen EEG model, prediction head, and evaluation protocol are held fixed as stated in the main text. The condition variants test whether recording-specific information and its alignment are useful. The generator variants compare direct parameter regression, latent regression, diffusion, and MoE decoding. The adapter variants alter modality-specific encoding, spectral information, and shared coordination.

Results analysis. The modular latent MLP exceeds direct weight MLP by 2.89 points; replacing that latent regressor with DiT contributes another 3.11 points in the displayed sequence, and the full configuration exceeds the no-MoE variant by 5.07 points. These are incremental differences between the listed variants, not an additive causal decomposition independent of training configuration. Removing subject conditioning and shuffling it cause substantial declines, consistent with recording-aligned information contributing beyond task and modality identity.

G.2 Interpreting the interventions

The shuffled-subject comparison changes alignment between a condition and its recording while retaining the set of subject-condition vectors. Its 9.24-point decrease is consistent with the importance of recording-aligned information. The task-plus-modality variant removes that factor and decreases accuracy by 11.09 points. The two interventions answer related but distinct questions: availability of subject information and correctness of its assignment.

The temporal-only variant has an 8.12-point decrease, compared with 4.16 points for a shared modality encoder and 6.58 points for mean fusion. The larger temporal-only decrease is consistent with a useful contribution from the spectral branch in this experiment. It does not quantify an independent spectral contribution that can be added to the other decreases, because the variants are separate trained systems.

The reported decreases are paired with the standard deviations given in main Table 2. They are computed from seed-wise paired differences rather than from the two marginal score standard deviations. The functional-loss ablation below isolates the supervision applied in the two learning stages.

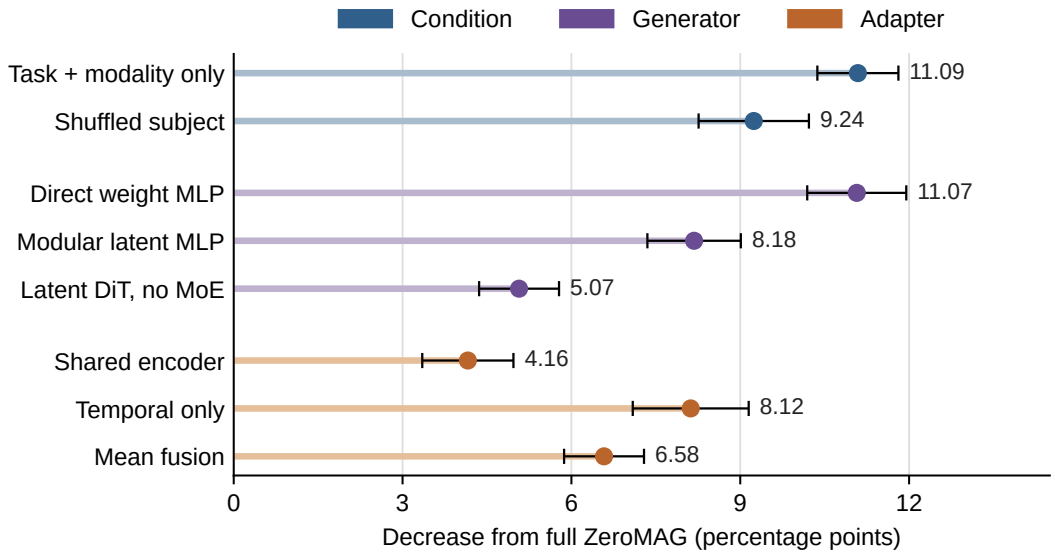

Figure G.1: **Performance decrease under component ablation.** Markers show the decreases reported in main Table 2 relative to full ZeroMAG (B-ACC 73.41 ± 0.71). Whiskers reproduce the table's reported standard deviations of the decreases; they are not newly calculated confidence intervals. Variants are grouped by condition, generator, and adapter. These are separate interventions, not additive contributions. The aggregation covers all nine target datasets, as in main Table 2.

G.3 Functional supervision

We evaluate four functional-loss settings using CBraMod on the same set of nine held-out target datasets as the main ablation study. The variants use five matched random seeds, the same data splits and reference-adapter banks, and matched training budgets, initialization rules, and evaluation protocols. B-ACC is reported as mean $\pm$ standard deviation over the five seeds.

Evaluation. Reconstruction and generated-weight errors use the same parameter-distance definition, comparing $\widetilde{\omega}$ and $\widehat{\omega}$, respectively, with the supervised reference ω^*. The prediction discrepancy is $D_{\mathrm{KL}}(p_{\omega^*} \| p_{\widehat{\omega}})$, evaluated on a fixed independent evaluation subset and averaged over samples, targets, and seeds. Full ZeroMAG is the comparison baseline in Table G.2: each error is divided by the corresponding error of Full ZeroMAG. Its normalized values are therefore 1.00 by definition. The normalization baseline is distinct from the reference adapter ω^* used in the underlying error definitions; the three normalized metrics need not have equal absolute values.

Results. Removing Stage 1 functional supervision reduces reconstruction error by 6%, yet increases final generated-weight error by 5% and prediction KL by 31%; B-ACC falls by 1.15 percentage points. Thus, better reconstruction of reference parameters does not necessarily yield better generated-adapter predictions. Removing Stage 2 functional supervision leaves Stage 1 reconstruction error unchanged, while generated-weight error increases by 8%, prediction KL by 48%, and B-ACC decreases by 1.58 points. Removing both losses gives the largest observed B-ACC decrease, 2.87 points, and raises prediction KL by 82%. These results support functional supervision during adapter representation learning and conditional generation.

H Implementation and Evaluation Protocol

H.1 Implementation Details

Table H.1 gives the training and inference settings. Batch size counts reference-adapter samples per optimizer step, not subjects or signal windows. Each sample has $B = 3M + 1$ active blocks and

Table G.2: Functional-loss ablation. Parameter-error and KL columns are normalized to their respective Full values; lower is better. Reconstruction error evaluates the Stage 1 output against the reference adapter. Generated-weight error evaluates the final generated adapter against the same reference. KL evaluates the final generated adapter.

Setting	B-ACC (%)	Stage 1 recon. error	Generated weight error	Prediction KL
Full	73.41 ± 0.71	1.00	1.00	1.00
w/o Stage 1 functional	72.26 ± 0.76	0.94	1.05	1.31
w/o Stage 2 functional	71.83 ± 0.79	1.00	1.08	1.48
w/o both functional	70.54 ± 0.86	0.93	1.14	1.82

a $B \times 128$ latent sequence. Block identity conditioning and attention masks handle configuration-dependent sequence lengths without allocating missing-modality blocks.

H.2 Separate backbone instantiations

Each of CBraMod, CSBrain, and CodeBrain has a separately trained generator. The EFM encoder is loaded from its checkpoint, and the compatible source-trained prediction head is fixed throughout the ZeroMAG pipeline (Appendix A.2). Reference-adapter fitting updates only adapter parameters. Functional losses use the same frozen head, whose parameters are not part of the adapter bank or generator output. At deployment, the known task selects its output layer; no target labels or target-side updates are used.

Within a selected instantiation, generated vectors use the block layout expected by its decoder and adapter. This scope is essential for the meaning of $\widehat{\omega} = g(R, \tau)$: the symbol g identifies that instantiation's generator, rather than one universal weight generator shared by all architectures. The reported multi-backbone evaluation tests repeated applicability of the framework.

H.3 Dataset holdout and recording splits

The adapter-bank and generator source datasets are ISRUC-S1 (Khalighi et al., 2016), Sleep-EDF Expanded (Kemp et al., 2000), SHHS (Quan et al., 1997), DEAP (Koelstra et al., 2012), ASCERTAIN (Subramanian et al., 2016), SEED-VIG (Zheng & Lu, 2017), WAUC (Albuquerque et al., 2020), and OpenBMI (Lee et al., 2019). The target datasets are ISRUC-S3 (Khalighi et al., 2016), HMC (Alvarez-Estevez & Rijsman, 2021), MASS-SS3 (reported as MASS) (O'Reilly et al., 2014), SEED-V (Liu et al., 2022), EEGMAT (Zyma et al., 2019), Multimodal FoG (Zhang et al., 2022), CAP (Terzano et al., 2001), DREAMER (Katsigiannis & Ramzan, 2018), and BCI-IV-2a (reported as BCI-IV) (Brunner et al., 2008). Let $\mathcal{D}_{\mathrm{src}}$ and $\mathcal{D}_{\mathrm{tgt}}$ denote these two disjoint sets:

$$\mathcal{D}_{\mathrm{src}} \cap \mathcal{D}_{\mathrm{tgt}} = \varnothing, \qquad |\mathcal{D}_{\mathrm{tgt}}| = 9. \tag{H.1}$$

All target datasets are excluded from encoder/head training, source reference-adapter fitting, both generator-training stages, and model selection in the ZeroMAG pipeline. Task supervision comes from source training data; target recordings are used only for unlabeled conditioning and evaluation.

For a source recording, $R^{\mathrm{cond}} \subseteq R^{\mathrm{fit}}$ is read without its labels. The functional subset is separate:

$$R^{\mathrm{fun}} \cap R^{\mathrm{fit}} = \varnothing, \qquad R^{\mathrm{fun}} \cap R^{\mathrm{cond}} = \varnothing. \tag{H.2}$$

The labels associated with R^{fit} train the reference adapter. Functional supervision compares predictions on R^{fun} without reading its labels. These distinct roles prevent the condition encoder's input from being confused with the reference adapter's supervised fitting data.

For each target subject, calibration and query subsets are disjoint:

$$R^{\mathrm{cond}}_{\mathrm{tgt}} \cap R^{\mathrm{qry}}_{\mathrm{tgt}} = \varnothing. \tag{H.3}$$

The condition is constructed from calibration signals and the supported task. The generated adapter is evaluated on query inputs without target-side gradient updates. Query labels serve evaluation rather than adapter fitting or condition construction.

Table H.1: Implementation settings. M is the number of active companion modalities. Auxiliary Stage 2 losses share the same low-noise gate; diffusion loss is always active.

Group	Setting	Value
Representation	Active parameter blocks / latent shape	$3M+1$ / $(3M+1)\times 128$
	CBT layers / hidden width / heads	2 / 256 / 8
Decoder	Experts / routing	4 / learned soft top-2
	Expert MLP widths	$128 \to 256 \to 128$
Both stages	Optimizer / learning rate / weight decay	AdamW / 10^{-4} / 10^{-4}
	Batch size	64 reference-adapter samples
Stage 1	Training duration	300 epochs
	Block / cosine / KL coefficients	1.0 / 0.1 / 10^{-4}
	Functional / balance coefficients	1.0 / 0.01
Stage 2	DiT layers / hidden width / heads	4 / 256 / 8
	Training diffusion horizon / schedule / prediction	1000 / cosine / ϵ
	Training duration	10,000 optimizer updates
	Diffusion / latent / weight coefficients	1.0 / 0.1 / 0.1
	Functional coefficient	1.0
	Auxiliary reconstruction/function losses	$\bar{\alpha}_t > 0.5$ (SNR > 1)
Training	Gradient clipping threshold / EMA decay	1.0 / 0.9999
Source bank	Size / eligibility	311 per backbone / all eligible
	Sampling	Dataset-balanced
Inference	DDIM (Song et al., 2020) steps / η_{DDIM}	50 / 0
	Calibration per subject	64 unlabeled windows
	Sleep window duration	30 s
Evaluation	Matched seeds	5

H.4 Source adapter bank

For backbone b, the bank contains all eligible fitted reference adapters from the fixed source datasets:

$$\mathcal{A}_{\mathrm{src}}^{(b)} = \bigcup_{D_i \in \mathcal{D}_{\mathrm{src}}} \mathcal{A}_i^{(b)}, \qquad N_{\mathrm{src}}^{(b)} = |\mathcal{A}_{\mathrm{src}}^{(b)}|. \tag{H.4}$$

Here eligibility includes excluding source records from any subject present in the target datasets. This exclusion is applied across related cohorts, including ISRUC-S1/S3, before reference-adapter fitting. No target dataset contributes adapters to the bank. There is no fixed bank-size cap or per-dataset quota. After eligibility filtering, the bank contains 311 reference adapters per backbone. Training uses dataset-balanced sampling so that large source datasets do not dominate the adapter batches.

H.5 Baseline interfaces

The EEG-only baseline evaluates $h(f(X^{\mathrm{EEG}}))$. Mean-source transfer averages compatible fitted source vectors,

$$\widehat{\omega}_{\mathrm{mean}} = \frac{1}{n}\sum_{j=1}^{n} \omega_j^*. \tag{H.5}$$

Nearest-source transfer selects a source adapter for reuse, whereas Direct MLP directly maps the condition to adapter weights. The controlled adapter comparisons are grouped by frozen backbone; native multimodal models remain separate cross-paradigm references.

H.6 Reported metrics

Let N_{ab} count predictions with true class a and predicted class b, with $n_a = \sum_b N_{ab}$, $\widehat{n}_b = \sum_a N_{ab}$, and $N = \sum_a n_a$. For K classes with positive support, balanced accuracy (Brodersen et al., 2010) is

$$\text{B-ACC} = \frac{1}{K}\sum_{c=1}^{K} \frac{N_{cc}}{n_c}. \tag{H.6}$$

Equal class weighting distinguishes balanced accuracy from ordinary accuracy, which weights the class recalls by their observed prevalences. The displayed result tables express B-ACC as a percentage; differences between those values are percentage points.

Weighted F1 uses the class-support weights:

$$\mathrm{WF1} = \sum_{c=1}^{K} \frac{n_c}{N} \mathrm{F1}_c, \qquad \mathrm{F1}_c = \frac{2N_{cc}}{n_c + \widehat{n}_c}, \tag{H.7}$$

for nonzero denominators. Cohen's kappa (Cohen, 1960) adjusts observed agreement for agreement induced by the marginal label frequencies:

$$p_o = \frac{\sum_c N_{cc}}{N}, \quad p_e = \sum_c \frac{n_c \widehat{n}_c}{N^2}, \quad \kappa = \frac{p_o - p_e}{1 - p_e}. \tag{H.8}$$

These metrics are not determined by B-ACC alone. Consequently, no kappa or weighted-F1 values are inferred from the B-ACC tables in this appendix. Likewise, the binary-task AUROC and AUPRC require score-level predictions; class-balanced accuracy cannot reconstruct either curve. Precision–recall analysis is particularly useful when class prevalence is imbalanced (Saito & Rehmsmeier, 2015).

H.7 Seed variation and pairwise comparisons

Main Tables 1 and 2 report means and standard deviations over five matched seeds. The plotted whiskers reproduce these reported SDs. A standard deviation describes seed-to-seed variation and is distinct from a standard error or confidence interval.

For two matched seed scores x_s, y_s, the difference is $d_s = x_s - y_s$. Its variance satisfies

$$\mathrm{Var}(d) = \mathrm{Var}(x) + \mathrm{Var}(y) - 2\,\mathrm{Cov}(x, y). \tag{H.9}$$

Marginal SDs do not determine the covariance. Therefore the derived mean gains in Appendix C are shown without newly inferred paired error bars. The reported SDs of ablation decreases remain those given in main Table 2.

H.8 Aggregation

The six-target comparisons average the 18 displayed backbone–dataset means with equal weight. This gives equal weight to each displayed setting, not to each participant or window. Within a target, the backbone groups reuse the target dataset and are not independent participant cohorts. The CBraMod ablation summary in main Table 2 averages results over all nine target datasets, including CAP, DREAMER, and BCI-IV, yielding 73.41 for Full ZeroMAG. It therefore has a different evaluation scope from the six-target comparison.

Recovery ratios and score differences answer different questions. A percentage-point difference measures absolute movement in B-ACC; a recovery percentage divides that movement by a supervised-reference gain. The ratio-of-aggregate-gains convention in Appendix C makes the 93.50% summary explicit and separates it from the average of row-wise percentages.

I Generalization and Data Efficiency

I.1 Modality holdout

The generalization experiments summarized in the main text consider unseen combinations of known modalities and modality types in $\mathcal{M}$ that are excluded from generator training. ZeroMAG recovers 89.7% and 82.6% of the supervised multimodal gain, respectively. Fig. I.1 compares the two recovery rates.

The type-holdout summary is 7.1 percentage points lower in recovery rate than the combination-holdout summary. This difference is in recovery percentage, not in balanced-accuracy percentage points. Both refer to the supported modality interface. They do not establish compatibility with an arbitrary signal type for which the model defines no encoder, identity embedding, or adapter block.

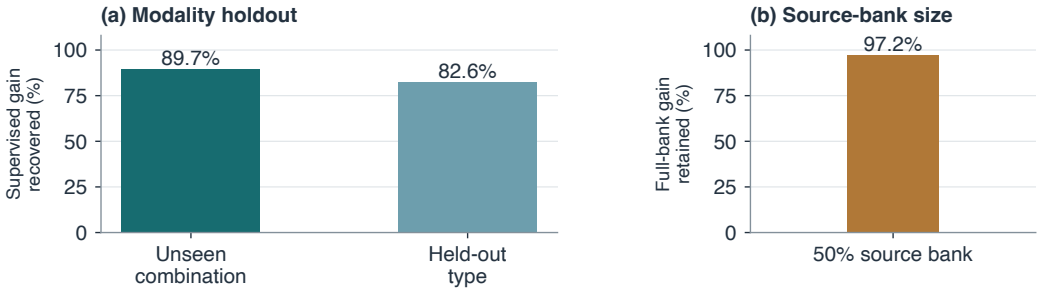

Figure I.1: Reported generalization and source-bank summaries from the main text. Left: recovery of supervised gain for unseen combinations and held-out modality types. Right: retention of the full-bank multimodal gain when using half the source bank. The panels have different reference denominators; no seed-level intervals are inferred.

I.2 Calibration data

Performance saturates at approximately 64 unlabeled calibration windows in the experiment summarized in the main text. This concerns the amount of data used to form the condition, rather than a target optimization batch size: there is no target-side fitting step. Once constructed, the condition guides generation of the adapter used on the disjoint query subset.

A window count describes the number of calibration units rather than a universal duration in seconds. Window duration and signal sampling rates differ across physiological tasks. The reported saturation point therefore characterizes the evaluated calibration setting and does not imply that all recordings or tasks require exactly the same observation duration.

I.3 Source-bank size

Using 50% of the source adapter bank retains 97.2% of the full-bank multimodal gain in the reported data-efficiency experiment. This is a 2.8% reduction in the reference gain, not a 2.8-point reduction in B-ACC. The denominator is the full-bank multimodal gain.

Together, the calibration and bank-size summaries concern two different resource requirements. Calibration windows are collected from a new target recording at deployment. Source adapters are fitted and organized offline. Reducing one does not directly reduce the other, and the reported summaries do not provide a joint optimum over calibration budget and bank size.

J Comparison with Related Methods

J.1 Extended related work

Multimodal physiological models. SleepFM and PhysioOmni learn multimodal physiological representations through pretraining (Thapa et al., 2024; Fu et al., 2025), whereas Brant-X uses a pretrained EEG foundation model to align EEG and other physiological signals and trains an EXG encoder (Zhang et al., 2024). ZeroMAG generates a plug-in adapter for a selected existing EFM, preserving its frozen encoder and prediction head at target deployment.

Parameter-efficient and test-time adaptation. Adapters, LoRA, and prompt tuning (Houlsby et al., 2019; Hu et al., 2022; Lester et al., 2021; Han et al., 2024), including EEG-specific adapter designs (Suzumura et al., 2024), reduce the number of trainable parameters but still optimize them on labeled target data. Test-time and unsupervised domain adaptation (Liang et al., 2023; Wang et al., 2021; Liang et al., 2020) avoid target labels but update parameters on target data, where hyperparameters and stopping criteria are difficult to validate. ZeroMAG amortizes adaptation into an offline-trained generator and performs no target-side gradient step.

Parameter generation. Hypernetworks (Ha et al., 2017) and meta networks (Munkhdalai & Yu, 2017) generate weights from task representations, shared hypernetworks generate adapters for multi-task fine-tuning (Karimi Mahabadi et al., 2021), and neural network diffusion models parameters with latent diffusion (Wang et al., 2024b). Recent work applies parameter generation to zero-shot adaptation of large language models: Text-to-LoRA generates LoRA adapters from natural-language task descriptions (Charakorn et al., 2025), and Drag-and-Drop LLMs maps batches of unlabeled task prompts to LoRA weights (Liang et al., 2025). ZeroMAG shares the goal of replacing per-target optimization with generation, but differs in three respects: its condition is computed from unlabeled physiological recordings rather than text; the generated adapter must accommodate varying modality configurations within a shared block schema; and generation combines parameter-level targets with functional agreement through the frozen EFM.

J.2 Adaptation mechanisms

Table J.1 distinguishes methods by the object they adapt and the information used for adaptation. It is a methodological comparison; the methods listed here are not additional experimental baselines.

Table J.1: Adaptation mechanisms in related work. Target-time optimization and conditional parameter generation are separate design choices.

Method family	Adapted object	Adaptation mechanism
Adapters (Houlsby et al., 2019)	Inserted modules	Optimize a small set of task-specific parameters.
LoRA (Hu et al., 2022)	Low-rank weight updates	Learn an update while freezing the original weights.
Prompt tuning (Lester et al., 2021)	Continuous input prompts	Optimize prompt embeddings for the task.
Tent (Wang et al., 2021)	Normalization parameters	Minimize prediction entropy on test inputs.
SHOT (Liang et al., 2020)	Target feature extractor	Adapt with a fixed source classifier and unlabeled target data.
Hypernetworks (Ha et al., 2017; Karimi Mahabadi et al., 2021)	Generated network or adapter weights	Predict parameters from a learned conditioning representation.
Meta Networks (Munkhdalai & Yu, 2017)	Fast weights	Use meta-learning to generate task-dependent parameters.
Neural network diffusion (Wang et al., 2024b)	Network parameters	Generate parameters through a learned latent diffusion model.
ZeroMAG	Multimodal adapter	Generate weights from unlabeled calibration signals and task context; keep the EEG model fixed.

The closest connection is to adapter generation and diffusion over network parameters. ZeroMAG uses a variable-length semantic block sequence to represent companion-signal adapters, and evaluates decoded weights through the frozen EEG prediction path. Its target condition comes from the recording and supported task. The generators for the three EEG backbones are trained separately.

The method depends on the supplied frozen EFM and compatible prediction head, the quality of source reference adapters, and the coverage of the calibration data. The reported results concern these evaluated interfaces. They do not establish transfer of one generator to an arbitrary new architecture or a task outside the supported vocabulary.