Neuro-KE: Knowledge-Guided Interfaces for Semantically Grounded EEG Foundation Models

Hanrui Chen[1,2,*] **Xiang Chen**[1,2,*] **Haotian Deng**[1,*] **Kexin Lou**[1,2]
Shinan Wang[1,2] **Chen Wei**[1,2] **Quanying Liu**[1,2,3,†]
[1]Southern University of Science and Technology, Shenzhen, China
[2]Omni-Intelligence, Shenzhen, China
[3]Shenzhen Loop Area Institute, Shenzhen, China
[*]Equal contribution. [†]Supervising author.

Abstract

Foundation Models for Electroencephalography (EEG) are trained with masked reconstruction, contrastive alignment, and language-model interfaces, yet it is unclear which EEG features their representations retain: raw EEG segments do not have predefined meanings, and models may preserve waveform detail while discarding physiologically meaningful structure. We introduce Neuro-KE (Neuro-Knowledge Engine), a framework that computes classical EEG features, such as band power and Hjorth parameters, without additional human annotation and uses them as supervision for different training settings. Rather than using handcrafted features as standalone classifiers, Neuro-KE converts the same domain knowledge into numeric prediction targets, text anchors describing feature states (e.g., high alpha power), and feature-grounded instruction answers. Performance improves overall, although not on every dataset: it improves balanced accuracy (BAcc) by 2.60 points over reconstruction-only masked pretraining across 12 datasets and three backbones, improves six-dataset EEG–text validation BAcc by 2.23 points over label-only anchors, and improves EEG-MLLM feature-grounded generation to ROUGE-L F1 0.607 and BERTScore F1 0.929. These results support semantic grounding: retaining named EEG properties that can be recovered from representations or described by the model.

1 Introduction

The advent of Foundation Models has transformed representation learning across Computer Vision (CV), Natural Language Processing (NLP), and, more recently, neuroscience. This shift has given rise to EEG Foundation Models that learn transferable representations from large-scale brain-signal archives [1, 2, 3, 4].

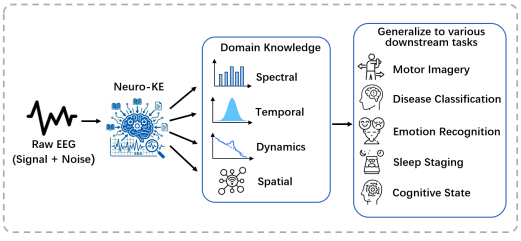

Figure 1. Neuro-KE as a knowledge-guided interface.

While masked modeling remains dominant for raw time-series reconstruction [5, 6], contrastive learning (e.g., EEG-CLIP) [7] and EEG-Large Language Models (e.g., NeuroLM) [8] are emerging as complementary paradigms.

Despite these rapid advancements, a fundamental challenge persists: EEG foundation models are powerful, but it is difficult to check which EEG features their representations retain.

Unlike images or text, where models can exploit visual structure or token-level semantics [9, 10], raw EEG has no known universal syntax and is dominated by non-stationary noise [11]. Purely

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

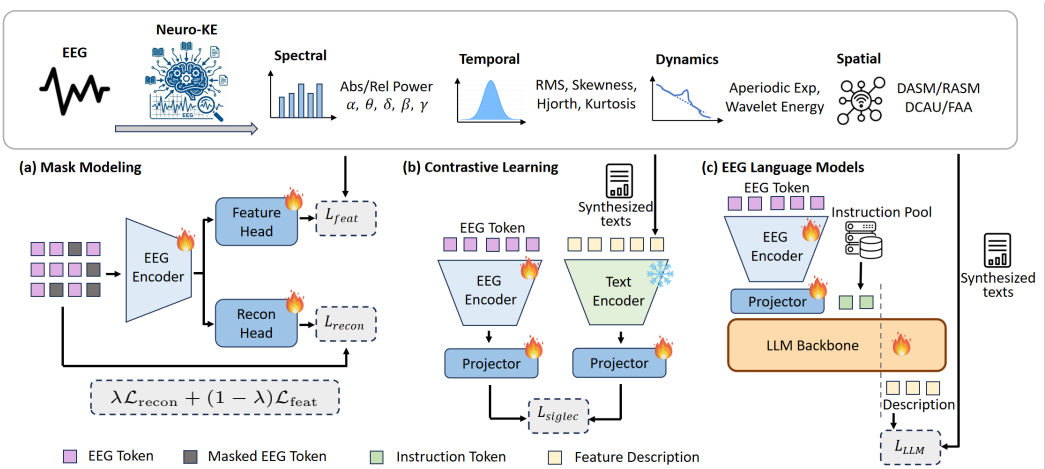

Figure 2. **The Neuro-KE Knowledge Interfaces with three learning paradigms.** Neuro-KE converts raw EEG into a reusable EEG knowledge representation and renders it as numeric targets, feature-state text, and instruction answers across multiple learning paradigms.

data-driven reconstruction objectives may therefore allocate capacity to stochastic artifacts rather than neural dynamics [6, 1]. This semantic gap limits pre-training efficiency, hinders multimodal alignment when high-quality EEG–text pairs are scarce, and makes it hard to decipher whether a model has retained physiologically meaningful structure.

To bridge this gap, we propose **Neuro-KE (Neuro-Knowledge Engine)** (Figure 1), a framework that computes classical EEG features without human annotation and uses them as supervision for foundation-model learning. By revisiting decades of signal processing research [12, 11, 13], Neuro-KE converts time-domain, spectral-power, spectral-structure, complexity, and cross-frequency descriptors into a *reusable knowledge representation*. This representation is not a single universal feature vector; instead, each learning paradigm uses the format it needs, such as numeric descriptor targets, feature-state text, or instruction-answer supervision. We call this use of EEG features as supervision a knowledge interface. It supports three settings: (a) feature prediction during masked modeling, (b) feature-state text anchors for contrastive EEG–text learning, and (c) feature-grounded generation for EEG–Large Language Models.

Our contributions are three-fold:

(1) **Knowledge Interface**: We introduce Neuro-KE, an EEG knowledge interface that extracts physiologically meaningful signal properties without human annotation and represents them in numeric, channel/segment, aggregate, and text-state forms.

(2) **Cross-Paradigm Grounding Probes**: We instantiate Neuro-KE across Masked Modeling, Contrastive EEG–text Learning, and EEG–LLM instruction tuning, using each paradigm as a measurable probe of semantic grounding.

(3) **Feature-Level Grounding Behaviors**: Beyond downstream accuracy, Neuro-KE enables feature-aware model behaviors: contrastive EEG–text pretraining can retrieve held-out feature-state descriptions from EEG segments, and EEG–LLM tuning supports conversational responses that identify and explain concrete EEG features rather than only predicting task labels.

2 Related Work

EEG Foundation Models. Recent EEG foundation models use masked modeling, contrastive pretraining, generative objectives, and scalable Transformer backbones to reduce dependence on dense labels [14, 5, 6, 15, 16, 17, 4]. These models improve transfer, but many objectives still treat EEG as a generic time series and do not directly test whether learned embeddings preserve physiologically meaningful signal structure. OmniEEG-Bench provides standardized evaluation protocols across EEG datasets and task families [18].

EEG Tokenization and Semantic Representations. Tokenization and discretization methods attempt to give continuous neural signals more structured computational units, including VQ-style neural tokenization and general time-series tokenization [19, 20, 21]. Neuro-KE is complementary to these approaches: instead of learning only data-driven tokens, it exposes interpretable time-frequency states that can be inspected, verbalized, and rendered across learning paradigms.

Knowledge-Guided EEG Learning. Classical EEG analysis has long used spectral bands, time-domain statistics, and non-linear complexity measures to interpret neural dynamics [12, 13, 22]. Recent neuro-grounded pretraining methods such as DeeperBrain [3] show that predicting statistics can improve EEG representation learning. FAME addresses low-frequency bias in masked reconstruction through band-wise target standardization and equally weighted band-specific losses [23]. Our novelty is not merely predicting EEG statistics; Neuro-KE formalizes the descriptor engine as a reusable cross-paradigm knowledge interface.

EEG–Language and MLLM Interfaces. EEG–language systems and EEG-LLM models such as NeuroLM, EEG-CLIP, and UniMind explore cross-modal alignment and instruction-based brain decoding [7, 8, 24]. Neuro-KE differs by making the language supervision explicitly feature-grounded: the text anchors and instruction answers are generated from objective EEG descriptor states rather than only labels or free-form clinical narratives.

Neuro-KE addresses this by formalizing a knowledge-guided framework that turns explicit time-frequency priors into learning signals for modern architectures. A detailed review of these developments is provided in Section A.

3 Neuro-KE: Knowledge Representation and Learning Interfaces

Neuro-KE consists of two components: a descriptor-based EEG knowledge representation and a set of deterministic rendering functions that convert this representation into supervision signals for different learning paradigms. We first define the descriptor catalog and its implementation-specific views, and then describe how the same descriptor state is used in masked modeling, EEG–text contrastive learning, and EEG-MLLM instruction tuning.

3.1 Descriptor-Based Knowledge Representation

The core of Neuro-KE is an EEG *knowledge representation*: a family of physiologically meaningful signal descriptors that can be rendered into the supervision format required by each learning paradigm. For each EEG segment, Neuro-KE computes a segment-level descriptor $s \in \mathbb{R}^{87}$, together with implementation-specific views such as standardized numeric targets, discretized feature states, and verbalized text anchors. Neuro-KE is therefore not defined as a single fixed-dimensional feature vector. Instead, it separates the underlying descriptor catalog from the paradigm-specific views used by different learning objectives.

Neuro-KE uses a lightweight descriptor catalog drawn from established neuroscience and clinical signal-analysis practice. The catalog is not intended as a new handcrafted feature set. Rather, it provides objective and interpretable signal states that can be reused across supervision views. It covers spectral properties such as band power, relative power, differential entropy, spectral shape, and composite frequency-domain indices. It also covers temporal properties such as amplitude, variability, zero-crossing, and Hjorth-style morphology, spatial properties such as inter-hemispheric asymmetry and anterior–posterior relations, and dynamics properties such as nonlinear complexity and microstate duration, occurrence, coverage, correlation, and explained-variance patterns. Detailed descriptor definitions and paradigm-specific preprocessing are provided in Sections C and D.

As illustrated in Figure 2, Neuro-KE renders the same descriptor-based representation into three learning interfaces: auxiliary numeric targets for masked modeling, feature-state text anchors for EEG–text contrastive learning, and instruction-answer supervision for EEG-MLLM tuning. These interfaces test whether objective descriptor states can regularize representation learning, improve text alignment, and support feature-grounded generation. Additional construction details are provided in Section D.

3.2 Latent Projection for Masked Modeling

Masked reconstruction is a widely used paradigm for EEG pretraining. But low reconstruction error does not necessarily imply that the learned representation preserves physiologically meaningful properties, such as spectral power, hemispheric asymmetry, or nonlinear dynamics. As shown in Figure 2(a), we introduce Neuro-KE descriptors as targets for masked modeling.

Given an EEG segment, the masked encoder produces contextual spatial–temporal representations, which are pooled into a segment-level latent representation $\mathbf{z}$. A lightweight feature head g_ϕ then maps $\mathbf{z}$ to a descriptor prediction $\hat{\mathbf{y}} = g_\phi(\mathbf{z})$, where $\mathbf{y}$ denotes the z-scored Neuro-KE descriptor target extracted from the same raw segment. The descriptor loss is the MSE between predicted and target descriptor values:

$$\mathcal{L}_{feat} = \frac{1}{N} \sum_{i=1}^{N} ||g_\phi(\mathbf{z}_i) - \mathbf{y}_i||^2 \tag{1}$$

The overall masked-modeling loss is defined as a weighted combination of reconstruction and feature-prediction objectives:

$$\mathcal{L}_{MM} = \lambda \mathcal{L}_{recon} + (1 - \lambda)\mathcal{L}_{feat}, \tag{2}$$

where $\lambda \in [0, 1]$ weights reconstruction and $1 - \lambda$ weights descriptor prediction. Balanced joint pretraining uses $\lambda = 0.5$.

We also build and compare concrete injection variants, including prefix-style and parallel-head designs.(Section G)

3.3 EEG-Text Pairs for Contrastive Learning

Because high-quality EEG–text pairs are scarce, and label-only text provide only coarse semantics. Following the idea of language-supervised EEG representation learning [25, 7, 26], we render Neuro-KE descriptor states into structured feature-state text anchors and test whether they can support finer-grained EEG–language alignment (See Figure 2 b).

We construct two anchor families: (1) a *label-only* anchor contains only the benchmark class name; (2) a *Label + Neuro-KE Text* anchor appends feature-state sentences generated by a grammar mapping $T = \mathrm{Template}(\mathbf{f})$, such as "The alpha relative power is high, suggesting a strong alpha rhythm." For retrieval-only analysis, we additionally construct standalone feature-state anchors that describe one descriptor and one percentile state. All text anchors are embedded offline with Qwen3-Embedding-8B [27]; detailed templates are provided in Section D.3.

Given EEG embeddings E_{eeg} and text embeddings E_{text}, separate MLP adapters h_e and h_t project both modalities into a shared latent space:

$$\mathbf{v}_e = \frac{h_e(E_{\mathrm{eeg}})}{\|h_e(E_{\mathrm{eeg}})\|_2}, \qquad \mathbf{v}_t = \frac{h_t(E_{\mathrm{text}})}{\|h_t(E_{\mathrm{text}})\|_2}. \tag{3}$$

We optimize a SigLIP-style sigmoid contrastive loss using an explicit positive-pair mask T_{ij}:

$$\mathcal{L}_{\mathrm{siglip}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N_t} m_j \log \sigma((2T_{ij} - 1)s_{ij}), \qquad s_{ij} = \alpha\, \mathbf{v}_{e,i}^{\top} \mathbf{v}_{t,j} + \beta. \tag{4}$$

Here N is the EEG batch size, N_t counts text anchors across all text parts, and m_j excludes padded anchors. The sigmoid function is σ, and α and β are learnable scale and bias parameters. For label anchors, $T_{ij} = 1$ when the EEG and text have the same class within a dataset. For each descriptor-text part, $T_{ij} = 1$ only for the EEG sample from which that text was generated; all other valid pairs have $T_{ij} = 0$. Thus, label anchors may have multiple positives, while matching feature states alone do not create additional descriptor-text positives. This interface evaluates whether Neuro-KE-derived feature-state text improves EEG–language alignment beyond label-only anchors.

3.4 EEG-QA for Feature-Grounded EEG–LLM

We further ask whether the same descriptor states can support EEG-conditioned language generation. As shown in Figure 2(c), we construct EEG-question-answer triplets $(x_{\text{eeg}}, x_{\text{prompt}}, A)$, where the answer A is derived from Neuro-KE signal properties and, when applicable, task metadata. Detailed construction of the EEG-question-answer data is provided in Section D.4.

Training and Interface. We use a two-stage EEG-MLLM curriculum. Stage 1 aligns EEG segments with description text generated from Neuro-KE feature descriptions and available task context. Stage 2 performs task-oriented instruction tuning over five EEG task types: single-feature state explanation, three-feature description, label-support explanation, multiple-choice feature selection, and direct label prediction. The Stage 2 model is initialized from Stage 1 and jointly updates the EEG encoder, MLP adapter, and LLM LoRA parameters [28]. Because the EEG inputs are at most 10 seconds, the projected EEG sequence is typically below 200 tokens. We therefore use a lightweight interface in which a LaBraM encoder produces patch tokens, a three-layer MLP projects them into the LLM embedding space, and the projected EEG tokens are concatenated with instruction tokens before causal language modeling.

$$\mathcal{L}_{\text{MLLM}} = -\sum_{t \in \mathcal{A}} \log P(A_t | x_{\text{eeg}}, x_{\text{prompt}}, A_{<t}; \Theta), \tag{5}$$

where $\mathcal{A}$ denotes assistant-answer tokens, so EEG and prompt tokens are masked from the loss. Full model and training details are provided in Sections D and E.

4 Experiments

We evaluate Neuro-KE through three research questions aligned with the interfaces in Section 3: (1) whether descriptor-guided masked modeling improves representation quality and feature retention; (2) whether Neuro-KE-derived feature-state text improves EEG–text alignment beyond label-only anchors; (3) whether EEG–LLM training produces feature-grounded outputs tied to objective signal properties. Details about evaluation metrics are provided in Section F

4.1 Neuro-KE Improves Masked-Model Representations and Feature Retention

We evaluate the masked-modeling interface in Figure 2(a) by comparing Neuro-KE-guided joint pretraining with reconstruction-only masked modeling. We test this comparison across three EEG backbones: CBraMod [17], CSBrain [29], and EEGMamba [30]. Detailed architectures, hyperparameters and datasets are provided in Sections B and E.

Downstream transfer. We evaluate joint masked modeling against reconstruction-only pretraining on 12 downstream datasets: SEED, BCIC2A, MDD, AD65, SHU, HMC, SleepEDF, SEEDV, TUEP, ISRUC, TUAB, and TUEV. Figure 3 shows the full-finetuning results for CBraMod, CSBrain, and EEGMamba. Joint pretraining improves 34 of 36 backbone–dataset comparisons. Across all evaluated backbones and datasets, joint pretraining improves mean balanced accuracy by 2.60 percentage points under full fine-tuning (Section G). This establishes the main transfer result: descriptor-supervised masked modeling yields pretrained representations that adapt better to downstream EEG tasks than reconstruction-only pretraining.

Ablations analyses. We use CBraMod as a controlled setting to conduct a series of analyses and ablations that identify which design choices contribute to the gains.To reduce computational cost, all ablation experiments are evaluated using linear probing with a frozen pretrained encoder. We ablate three factors: (1) the descriptor-family composition, (2) the injection strategy, and (3) the reconstruction–feature loss balance. Details of the masked-modeling setup and corresponding results are provided in Section G and Figures 7 to 10. We also evaluate whether the learned representation preserves the physiological descriptors used for Neuro-KE supervision by predicting all 87 EEG features from the validation set. As shown in Figure 4, the model predicts many descriptors accurately, with an average R^2 of 0.637 and an average PCC of 0.758 across all features. This analysis suggests that the joint objective does not only improve downstream transfer, but also retains interpretable spectral, temporal, spatial, and dynamical EEG information in the masked-model representation.

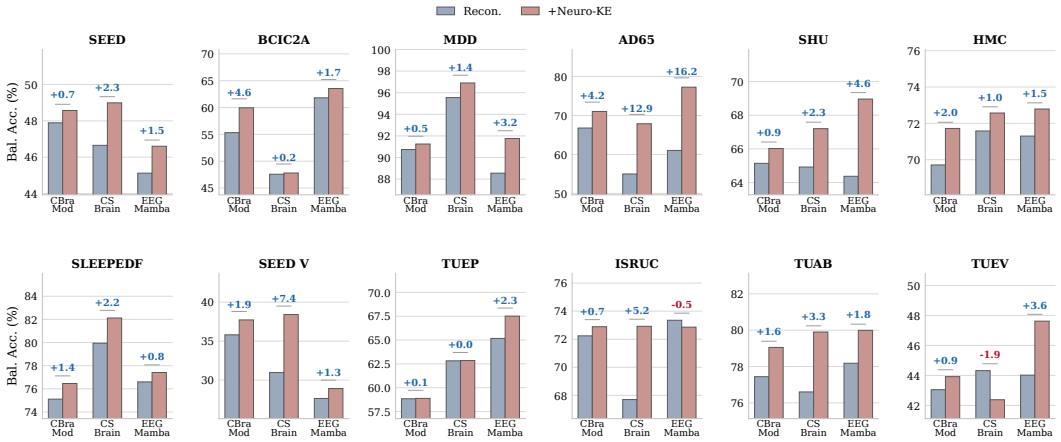

Figure 3. **Joint-versus-reconstruction full-finetuning results.** Mean balanced-accuracy gains for Neuro-KE joint pretraining over reconstruction-only pretraining on the selected 12 downstream datasets under full fine-tuning.

Descriptor-family composition. We ablate the descriptor families with leave-one-group-out targets over spectral, temporal, spatial, and dynamics descriptors. The full joint target obtains the best selected-12 macro BAcc (0.5624), and removing any group lowers macro performance. Spectral and spatial removals cause the largest macro drops, while temporal and dynamics removals reveal some large-dataset exceptions in weighted BAcc. This supports using the full descriptor set as the default global target, while leaving room for task-specific feature-family tuning.

Injection strategy. We compare how descriptor supervision is introduced into masked modeling: a one-CLS-token prefix, an eight-CLS-token prefix with feature-subspace tokens, and a DeeperBrain-like parallel feature-prediction head. Reconstruction-only and feature-only objectives are included as baselines. On the 12 selected datasets with complete runs, the eight-token prefix joint variant obtains the best mean test BAcc (56.24%) and wins 7/12 datasets, outperforming the one-token prefix variant (51.30%) and the parallel-head joint variant (50.39%). This indicates that descriptor supervision is most effective when introduced through a structured prefix-style interface rather than a single coarse token or a simple auxiliary head.

Reconstruction–feature loss balance. We sweep the loss balance by adjusting the value of λ,where $\lambda = 0$ is pure feature prediction and $\lambda = 1$ is pure reconstruction. The balanced setting $\lambda = 0.50$ gives the highest mean BAcc (0.5624) and most dataset-level wins (9/12), while both endpoints are less competitive. Thus, the benefit is not explained by feature prediction or reconstruction alone; it comes from balancing the two supervisory signals. Detailed selected-suite heatmaps, CBraMod ablation panels, and feature-prediction diagnostics are provided in Section G.

Table 1. **Multi-dataset classification with Neuro-KE text anchors.** Validation BAcc (%, mean ± std across random seeds) is reported; the better mean for each dataset is bolded.

Dataset	Label Only Val	Label + Neuro-KE Text Val	Δ
TUAB	79.52 ± 0.77	**80.02 ± 0.98**	+0.50
TUEP	69.67 ± 5.99	**72.40 ± 2.96**	+4.73
SEED	**49.72 ± 0.27**	49.25 ± 1.13	-0.47
SHU	64.73 ± 9.29	**66.27 ± 8.24**	+1.54
SleepEDF	65.88 ± 1.81	**69.98 ± 1.15**	+4.10
Workload	48.18 ± 5.08	**51.16 ± 6.02**	+2.98
Macro Avg.	62.62	**64.85**	+2.23

4.2 Neuro-KE Text Improves EEG–Text Alignment Beyond Label Text

We evaluate whether Neuro-KE-derived feature-state text provides a richer alignment signal than label text alone. Each EEG segment is paired with deterministic text generated from its Neuro-KE profile and, where applicable, benchmark task labels, and the EEG–text encoders are optimized with the SigLIP-style loss. We evaluate the aligned space in three settings: (1) multidataset classification

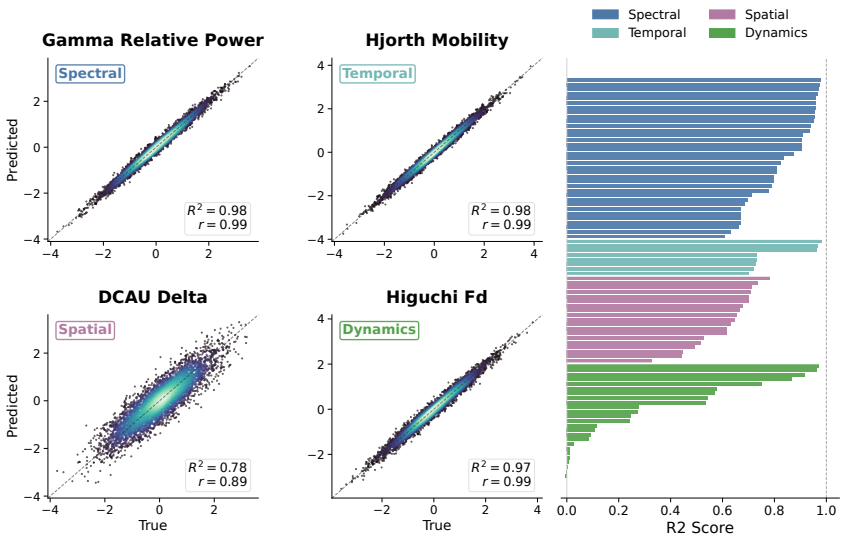

Figure 4. **Feature prediction performance across four EEG feature families.** Representative scatter plots show true versus predicted values for the best feature in each group, while the grouped bar chart reports per-feature R^2 scores for all 87 features. Higher predictability for spectral and temporal features suggests that these physiological descriptors are strongly encoded in the learned representation. Detailed per-feature R^2 and PCC results are compared in Appendix Fig. 12.

on the six contrastive-training datasets, (2) transfer to five unseen downstream datasets, and (3) feature-retrieval from the held-out test splits of EEG–text bank. Templates and construction details are in Sections D and D.3.

Multi-Dataset Classification. Table 1 shows that the contrastive paradigm with Neuro-KE improves five of six datasets and increases BAcc from 62.62% to 64.85% (+2.23 points). The largest gains occur on TUEP (+4.73), SleepEDF (+4.10), and Workload (+2.98), while the only decrease is smallon SEED (-0.47). These results indicate that descriptor-derived text provides a useful alignment signal beyond coarse label names.

Downstream Transfer. Table 2 tests whether the aligned space transfers beyond the six datasets used for contrastive training. In zero-shot classification, L+NKT improves four of five unseen datasets and ties EEGMAT, suggesting that feature-state anchors make the text space more compatible with unseen EEG label names. After supervised adaptation, L+NKT remains better in eight of ten fine-tuning settings, with exceptions only at TUEV 50% and EEGMAT 50%. Thus, Neuro-KE text improves both zero-shot transfer and most data-efficient fine-tuning settings.

Table 2. **Transfer from contrastive EEG–text pretraining.** BAcc (%) is reported for zero-shot classification and downstream full fine-tuning with 50% or 100% of the target training data.

Setting	SEED-IV		TUEV		HMC		PhysioNet-MI		EEGMAT	
	L-only	L+NKT	L-only	L+NKT	L-only	L+NKT	L-only	L+NKT	L-only	L+NKT
zero-shot	25.0	**26.4**	26.5	**29.8**	33.8	**34.8**	22.7	**25.5**	55.6	55.6
50% data	33.2	**35.5**	**59.6**	56.3	58.7	**62.4**	49.6	**52.2**	**77.8**	76.7
100% data	40.3	**42.7**	60.0	**62.0**	72.6	**73.5**	54.5	**55.4**	75.6	**81.1**

Feature-Retrieval Case Study. We use retrieval as a direct probe of feature-level alignment. The retrieval bank is built from held-out test splits of the six contrastive-training datasets, so retrieval evaluates recovery of unseen EEG–feature-state pairs rather than memorization of training examples. Each descriptor is discretized into five percentile states. In EEG-to-text (E2T), an EEG query ranks the five text anchors for the queried descriptor. In text-to-EEG (T2E), a feature-state text query ranks held-out EEG segments, with positives defined by the same feature-state label.

Table 4. **Quantitative feature-retrieval examples.** E2T uses a shared five-way random baseline, while T2E uses dataset-specific chance baselines because the number of positive EEG segments varies across retrieval banks.

EEG-to-Text (E2T)			
Dataset	**R@1**	**R@3**	**MRR**
Chance	20.00	60.00	0.457
SHU	**48.54**	74.60	0.650
SleepEDF	**44.87**	80.87	0.644
SEED	**43.18**	74.42	0.620

Text-to-EEG (T2E)						
	R@1		**R@10**		**mAP**	
Dataset	**Score**	**Chance**	**Score**	**Chance**	**Score**	**Chance**
SHU	21.14	0.163	57.64	1.619	22.39	0.163
SleepEDF	32.91	4.320	77.71	35.702	32.63	4.320
SEED	20.22	0.994	64.84	9.504	20.69	0.994

Each feature is split into five percentile-state labels. E2T ranks the five text anchors for the queried feature; T2E uses a feature-state text query to rank EEG segments from the held-out six-dataset EEG–text bank, with positives sharing the same feature-state label. Table 4 reports SHU, SleepEDF, and SEED examples. Figure 5 then gives qualitative examples of the retrieved feature-state descriptions.

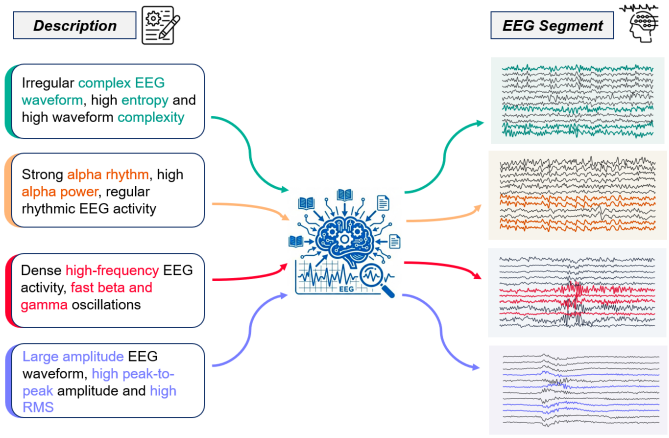

Figure 5. **Feature-retrieval examples with Neuro-KE.** The left panel shows various textual descriptions of EEG characteristics (e.g., complex waveforms, strong alpha rhythms, high-frequency oscillations, and large amplitudes). Our model interprets these descriptions to accurately retrieve and match the corresponding multi-channel EEG segments (right panel).

4.3 EEG-MLLM Preserves Neuro-KE Feature Evidence in Generated Outputs

Label Prediction with Neuro-KE Text. Before evaluating free-form generation, we use direct label prediction as a compact sanity check following the same two-stage EEG-MLLM training procedure. Training-mixture coverage for the evaluated datasets is summarized in Table 8; the comparison is not presented as a zero-shot benchmark on datasets absent from the listed mixtures. Table 3 shows that adding Neuro-KE text improves balanced accuracy on four of six datasets and increases the macro average from 0.693 to 0.729. The largest gains appear on AD65 (+13.61 points), SEED (+10.24), and TUAB (+2.64), while TUEV and SHU-MI drop, suggesting that event and motor-imagery labels may require more task-specific Neuro-KE text templates.

Feature-Grounded Generation. Label balanced accuracy alone is not a sufficient metric for EEG-LLM performance, because most tasks require free-form explanation and feature interpretation

Table 3. **Label prediction with Neuro-KE text supervision.** Balanced accuracy is reported for label-only supervision and label plus Neuro-KE text supervision.

DATASET	LABEL ONLY	LABEL + NEURO-KE TEXT	Δ
AD65	52.84	**66.45**	+13.61
SEED	35.14	**45.38**	+10.24
TUAB	79.67	**82.31**	+2.64
MDD	93.00	**93.39**	+0.39
TUEV	**87.92**	84.94	-2.98
SHU-MI	**67.25**	64.67	-2.58
MACRO AVG.	69.30	**72.86**	+3.55

Table 5. **EEG-MLLM feature-grounded generation evaluation.** Reference-based metrics evaluate generated answers against deterministic Neuro-KE-derived targets; MCQ accuracy evaluates discrete feature-choice questions; LLM-as-Judge reports clinical utility on a 0–5 scale.

METRIC	COMPONENT	LABEL ONLY	LABEL + NEURO-KE TEXT	Δ
ROUGE-L	PRECISION	0.5756	**0.6121**	+0.0365
ROUGE-L	RECALL	0.1741	**0.6227**	+0.4486
ROUGE-L	F1	0.2199	**0.6072**	+0.3873
BERTSCORE	PRECISION	0.6864	**0.9303**	+0.2439
BERTSCORE	RECALL	0.6263	**0.9284**	+0.3021
BERTSCORE	F1	0.6550	**0.9293**	+0.2743
METEOR	SCORE	0.1504	**0.5934**	+0.4430
MCQ ACC.	SCORE	0.0000	**0.8700**	+0.8700
LLM-AS-JUDGE	SCORE	0.2934 / 5	**3.39 / 5**	+3.0966

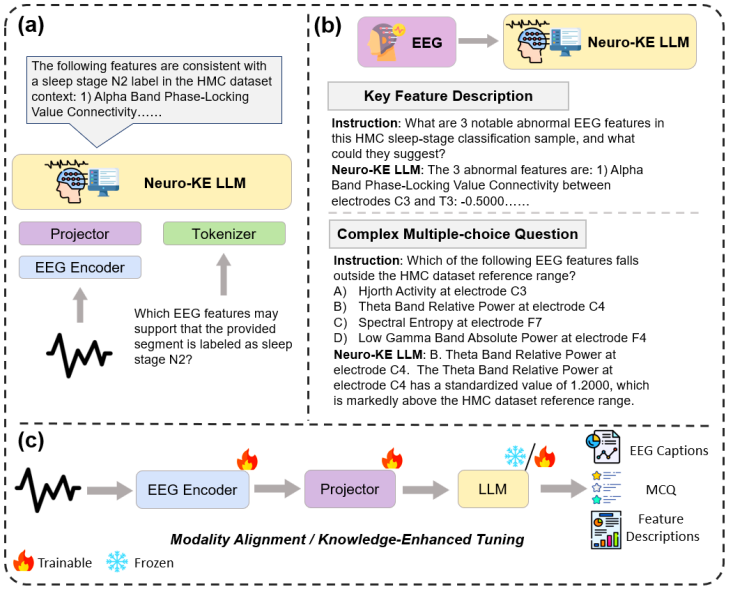

Figure 6. **Neuro-KE EEG-LLM training interface.** The EEG encoder and projector convert short EEG segments into LLM-compatible tokens, while Neuro-KE-derived instructions supervise feature-grounded description, reasoning, and task responses.

rather than only class selection. We compare generated `model_output` against the deterministic reference `answer` using ROUGE-L, METEOR, and BERTScore; compute MCQ accuracy only on `multiple_choice` samples by parsing A/B/C/D options; and use an external LLM judge to score non-classification explanatory answers on a 0–5 clinical-utility scale. Table 5 shows a large gap between label-only supervision and Neuro-KE text supervision. Neuro-KE text supervision improves every text-overlap, semantic-similarity, multiple-choice, and judge-based metric, indicating that generated Neuro-KE text provides the feature evidence needed for grounded instruction following.

5 Discussion and Conclusion

Neuro-KE reframes classical EEG descriptors as a "grey-box" knowledge interface rather than standalone features. By rendering domain knowledge as numeric targets, text anchors, and instruction answers, Neuro-KE improves semantic grounding across masked modeling, contrastive learning, and EEG-MLLMs. The current evidence supports treating descriptors as complementary supervision that encourages physiologically meaningful structure rather than as replacements for learned EEG representations.

Limitations and Future Work. Neuro-KE currently represents EEG knowledge through individual descriptor states, leaving dependencies and hierarchical structure implicit. A natural extension is to

model these relationships using structured latent states or descriptor graphs, and test whether targeted changes in physiological features lead to consistent changes in predictions and explanations. Our EEG-MLLM evaluation focuses on feature-grounded generation rather than multi-step neurophysiological reasoning. Incorporating evidence across EEG segments, detecting conflicting cues, and tracking temporal transitions could enable more coherent interpretations of evolving neural activity. Finally, descriptor contributions differ across families, datasets, and tasks, suggesting that a uniform knowledge interface is not always optimal. Future work can explore adaptive descriptor selection and redundancy-aware supervision to identify task-relevant physiological information and strengthen generalization, grounding, and interpretability across EEG learning paradigms.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (3254100307, 62472206); the Brain Science and Brain-like Intelligence Technology–National Science and Technology Major Project (2021ZD0200500, 2026ZD0220402); the National Key R&D Program of China (2025YFC3410000); the Shenzhen Science and Technology Innovation Committee (RCYX20231211090405003); the Guangdong Basic and Applied Basic Research Foundation (2026B1515020099); the Guangdong S&T Program (2026B0101110003); the Shanghai Municipal Special Program for Basic Research on General AI Foundation Models (2025SHZDZX026D05); the Shenzhen Loop Area Institute (FPF10120250012); and the open research fund of the Guangdong Provincial Key Laboratory of Mathematical and Neural Dynamical Systems, the Center for Computational Science and Engineering at Southern University of Science and Technology, and the Shenzhen Key Laboratory of Smart Healthcare Engineering.

Beyond the funding and institutional affiliations disclosed in this paper, the authors declare no additional competing financial interests.

References

[1] 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.

[2] Fei Ma, Han Lin, Yifan Xie, Hongwei Ren, Xiaoyu Shen, Wenbo Ding, and Qi Tian. E^2-llm: Bridging neural signals and interpretable affective analysis. *arXiv preprint arXiv:2601.07877*, 2026.

[3] Jiquan Wang, Sha Zhao, Yangxuan Zhou, Yiming Kang, Shijian Li, and Gang Pan. Deeperbrain: A neuro-grounded eeg foundation model towards universal bci. *arXiv preprint arXiv:2601.06134*, 2026.

[4] Zhisheng Chen, Yingwei Zhang, Qizhen Lan, Tianyu Liu, Huacan Wang, Yi Ding, Ziyu Jia, Ronghao Chen, Kun Wang, and Xinliang Zhou. Uni-ntfm: A unified foundation model for eeg signal representation learning. *arXiv preprint arXiv:2509.24222*, 2025.

[5] Christopher Wang, Vighnesh Subramaniam, Adam U Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. In *International Conference on Learning Representations*, 2023.

[6] Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. *Advances in Neural Information Processing Systems*, 36:26304–26321, 2023.

[7] Tidiane Camaret Ndir, Robin Tibor Schirrmeister, and Tonio Ball. Eeg-clip : Learning eeg representations from natural language descriptions. *Frontiers in Robotics and AI*, 2025.

[8] Wei-Bang Jiang, Yansen Wang, Bao-Liang Lu, and Dongsheng Li. Neurolm: A universal multi-task foundation model for bridging the gap between language and eeg signals. *arXiv preprint arXiv:2409.00101*, 2024.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[11] Saeid Sanei and Jonathon A. Chambers. *EEG Signal Processing*. John Wiley & Sons, 2007.

[12] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006.

[13] Amjed S. Al-Fahoum and Ausilah A. Al-Fraihat. Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains. *ISRN Neuroscience*, 2014:1–7, 2014.

[14] Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in Human Neuroscience*, 15:653659, 2021.

[15] Chaoqi Yang, M. Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. In *Advances in Neural Information Processing Systems*, volume 36, pages 78240–78260, 2023.

[16] Guagnyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. EEGPT: Pretrained transformer for universal and reliable representation of EEG signals. In *Neural Information Processing Systems*, 2024.

[17] 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.

[18] Ziling Lu, Zongsheng Li, Xinke Shen, Kexin Lou, Yingyue Xin, Xiaoqi Chen, Shinan Wang, Xiang Chen, Jiahao Fan, Chenyu Huang, Xin Xu, Zhoujie Hou, Chen Wei, and Quanying Liu. Omnieeg-bench: A standardized evaluation benchmark for eeg foundation models. *arXiv preprint arXiv:2606.00815*, 2026.

[19] Haokun Gui, Xiucheng Li, and Xinyang Chen. Vector quantization pretraining for EEG time series with random projection and phase alignment. In *International Conference on Machine Learning*, 2024.

[20] Konstantinos Barmpas, Na Lee, Dimitrios Chalatsis, William Raftery, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos Laskaris, Alexandros Koliousis, Dario Farina, and Stefanos Zafeiriou. Neurorvq: Multi-scale biosignal tokenization for generative foundation models. *arXiv preprint arXiv:2510.13068*, 2025.

[21] Sabera Talukder, Yisong Yue, and Georgia Gkioxari. Totem: Tokenized time series embeddings for general time series analysis. *Transactions on Machine Learning Research*, 2024.

[22] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American journal of physiology-heart and circulatory physiology*, 278(6):H2039–H2049, 2000.

[23] Junjie Yu, Zihan Deng, Jianyu Zhang, Junrong Mu, Jiahui An, Wenxiao Ma, Ziling Lu, Yue Wang, Yan Zhu, Kexin Lou, and Quanying Liu. Understanding and correcting low-frequency bias in EEG foundation model. *arXiv preprint arXiv:2608.01898*, 2026.

[24] Weiheng Lu, Zhouheng Yao, Jiamin Wu, Pengyu Zhu, Yuchen Zhou, Weijian Mai, Qihao Zheng, Wanli Ouyang, and Chunfeng Song. Unimind: Unleashing the power of llms for unified multi-task brain decoding. *arXiv preprint arXiv:2506.18962*, 2025.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[26] Sam Gijsen and Kerstin Ritter. Eeg-language pretraining for highly label-efficient clinical phenotyping. *arXiv preprint arXiv:2409.07480*, 2024.

[27] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[28] 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, 2021.

[29] 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. *arXiv preprint arXiv:2506.23075*, 2025.

[30] Yiyu Gui, MingZhi Chen, Yuqi Su, Guibo Luo, and Yuchao Yang. Eegmamba: Bidirectional state space model with mixture of experts for eeg multi-task classification. *arXiv preprint arXiv:2407.20254*, 2024.

[31] Li Hu and Zhiguo Zhang. *EEG Signal Processing and Feature Extraction*. Springer, 2019.

[32] Paul L Nunez and Ramesh Srinivasan. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, 2006.

[33] Maria Trigka, Elias Dritsas, and Christos Fidas. A survey on signal processing methods for eeg-based brain computer interface systems. In *Pan-Hellenic Conference on Informatics*, 2022.

[34] Anupreet Kaur Singh and Sridhar Krishnan. Trends in eeg signal feature extraction applications. *Frontiers in Artificial Intelligence*, 5, 2023.

[35] Ahmad Chaddad, Yihang Wu, Reem Kateb, and Ahmed Bouridane. Electroencephalography signal processing: A comprehensive review and analysis of methods and techniques. *Sensors*, 23(14):6434, 2023.

[36] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye. 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.

[37] 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.

[38] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015.

[39] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[40] Robin Tibor Schirrmeister, Lukas Gemein, Katharina Eggensperger, Frank Hutter, and Tonio Ball. Deep learning with convolutional neural networks for decoding and visualization of eeg pathology. In *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–7, 2017.

[41] Yonghao Song, Qingqing Zheng, Bingchuan Liu, and Xiaorong Gao. Eeg conformer: Convolutional transformer for eeg decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719, 2023.

[42] Zhijiang Wan, Manyu Li, Shichang Liu, Jiajin Huang, Hai Tan, and Wenfeng Duan. Eegformer: A transformer–based brain activity classification method using eeg signal. *Frontiers in Neuroscience*, 17:1148855, 2023.

[43] Y Song, X Jia, L Yang, and L Xie. Transformer-based spatial-temporal feature learning for eeg decoding. *arXiv preprint arXiv:2106.11170*, 2021.

[44] Sylvain Chevallier, Igor Carrara, Bruno Aristimunha, Pierre Guetschel, Sara Sedlar, Bruna Lopes, Sebastien Velut, Salim Khazem, and Thomas Moreau. The largest eeg-based bci reproducibility study for open science: the moabb benchmark. *arXiv preprint arXiv:2404.15319*, 2024.

[45] Jonathan Dan, Una Pale, Alireza Amirshahi, William Cappelletti, Thorir Mar Ingolfsson, Xiaying Wang, Andrea Cossettini, Adriano Bernini, Luca Benini, Sandor Beniczky, David Atienza, and Philippe Ryvlin. Szcore: Seizure community open-source research evaluation framework for the validation of electroencephalography-based automated seizure detection algorithms. *Epilepsia*, 66(S3):14–24, 2025.

[46] M. J. Crockett and Lisa Messeri. AI Surrogates and illusions of generalizability in cognitive science. *Trends in Cognitive Sciences*, 30(3):203–215, 2026.

[47] Geeling Chau, Yujin An, Ahamed Raffey Iqbal, Soon-Jo Chung, Yisong Yue, and Sabera Talukder. Generalizability under sensor failure: Tokenization + transformers enable more robust latent spaces. *arXiv preprint arXiv:2402.18546*, 2024.

[48] Tongtian Yue, Xuange Gao, Shuning Xue, Yepeng Tang, Longteng Guo, Jie Jiang, and Jing Liu. Braingpt: Unleashing the potential of eeg generalist foundation model by autoregressive pre-training. *arXiv preprint arXiv:2410.19779*, 2024.

[49] Navid Mohammadi Foumani, Geoffrey Mackellar, Soheila Ghane, Saad Irtza, Nam Nguyen, and Mahsa Salehi. Eeg2rep: Enhancing self-supervised eeg representation through informative masked inputs. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5544–5555, 2024.

[50] Xinxu Wei, Kanhao Zhao, Yong Jiao, Hua Xie, Lifang He, and Yu Zhang. Pre-training graph contrastive masked autoencoders are strong distillers for eeg. *arXiv preprint arXiv:2411.19230*, 2024.

[51] Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

[52] 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.

[53] Dongyang Li, Haoyang Qin, Mingyang Wu, Chen Wei, and Quanying Liu. Brainflora: Uncovering brain concept representation via multimodal neural embeddings. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pages 5577–5586, New York, NY, USA, 2025. Association for Computing Machinery.

[54] Yizi Zhang, Linyang He, Chaofei Fan, Tingkai Liu, Han Yu, Trung Le, Jingyuan Li, Scott Linderman, Lea Duncker, Francis R. Willett, Nima Mesgarani, and Liam Paninski. Decoding inner speech with an end-to-end brain-to-text neural interface. *arXiv preprint arXiv:2511.21740*, 2025.

[55] Hongqi Li, Yitong Chen, Yujuan Wang, Weihang Ni, and Haodong Zhang. Foundation models for cross-domain eeg analysis application: A survey. *arXiv preprint arXiv:2508.15716*, 2025.

[56] Xinliang Zhou, Chenyu Liu, Zhisheng Chen, Kun Wang, Yi Ding, Ziyu Jia, and Qingsong Wen. Brain foundation models: A survey on advancements in neural signal processing and brain discovery. *IEEE Signal Processing Magazine*, 42(5):22–35, 2025.

[57] Junhong Lai, Jiyu Wei, Lin Yao, and Yueming Wang. A simple review of eeg foundation models: Datasets, advancements and future perspectives. *arXiv preprint arXiv:2504.20069*, 2025.

[58] Konstantinos Barmpas, Na Lee, Yannis Panagakis, Dimitrios A Adamos, Nikolaos Laskaris, and Stefanos Zafeiriou. Advancing brainwave modeling with a codebook-based foundation model. *arXiv preprint arXiv:2505.16724*, 2025.

[59] Zijian Dong, Ruilin Li, Joanna Su Xian Chong, Niousha Dehestani, Yinghui Teng, Yi Lin, Zhizhou Li, Yichi Zhang, Yapei Xie, Leon Qi Rong Ooi, B. T. Thomas Yeo, and Juan Helen Zhou. Brain harmony: A multimodal foundation model unifying morphology and function into 1d tokens. *arXiv preprint arXiv:2509.24693*, 2025.

[60] Jiazhen Hong, Geoffrey Mackellar, and Soheila Ghane. Samba: Toward a long-context eeg foundation model via spatial embedding and differential mamba. *arXiv preprint arXiv:2511.18571*, 2025.

[61] Rukuang Huang, Sungjun Cho, Chetan Gohil, Oiwi Parker Jones, and Mark Woolrich. Meggpt: A transformer-based foundation model for magnetoencephalography data. *arXiv preprint arXiv:2510.18080*, 2025.

[62] Qinfan Xiao, Ziyun Cui, Chi Zhang, Siqi Chen, Wen Wu, Andrew Thwaites, Alexandra Woolgar, Bowen Zhou, and Chao Zhang. Brainomni: A brain foundation model for unified eeg and meg signals. *arXiv preprint arXiv:2505.18185*, 2025.

[63] Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in neural information processing systems*, volume 33, pages 21271–21284, 2020.

[64] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736, 2022.

[65] Francisco Varela, Jean-Philippe Lachaux, Eugenio Rodriguez, and Jacques Martinerie. The brainweb: phase synchronization and large-scale integration. *Nature reviews neuroscience*, 2(4):229–239, 2001.

[66] Ryan T Canolty, Erik Edwards, Sarang S Dalal, Maryam Soltani, Srikantan S Nagarajan, Heidi E Kirsch, Mitchel S Berger, Nicholas M Barbaro, and Robert T Knight. High gamma power is phase-locked to theta oscillations in human neocortex. *Science*, 313(5793):1626–1628, 2006.

[67] Jathurshan Pradeepkumar, Xihao Piao, Zheng Chen, and Jimeng Sun. Tokenizing single-channel eeg with time-frequency motif learning. *arXiv preprint arXiv:2502.16060*, 2025.

[68] Christopher Sandino, Sayeri Lala, Geeling Chau, Melika Ayoughi, Behrooz Mahasseni, Ellen Zippi, Ali Moin, Erdrin Azemi, and Hanlin Goh. Learning the relative composition of eeg signals using pairwise relative shift pretraining. *arXiv preprint arXiv:2511.11940*, 2025.

[69] Sha Zhao, Mingyi Peng, Haiteng Jiang, Tao Li, Shijian Li, and Gang Pan. EEGAgent: A unified framework for automated EEG analysis using large language models. *arXiv preprint arXiv:2511.09947*, 2025.

[70] Kang Yin and Hye-Bin Shin. Neurolex: A lightweight domain language model for eeg report understanding and generation, 2025.

[71] 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 *The Fourteenth International Conference on Learning Representations*, 2026.

[72] Jingying Ma, Feng Wu, Yucheng Xing, Qika Lin, Tianyu Liu, Chenyu Liu, Ziyu Jia, and Mengling Feng. SCOPE: Structured prototype-guided adaptation for EEG foundation models with limited labels. *arXiv preprint arXiv:2602.17251*, 2026.

[73] Iyad Obeid and Joseph Picone. The temple university hospital eeg data corpus. *Frontiers in neuroscience*, 10:196, 2016.

[74] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J Miller, Gernot R Müller-Putz, et al. Review of the bci competition iv. *Frontiers in neuroscience*, 6:55, 2012.

[75] Thomas Donoghue, Matar Haller, Erik J. Peterson, Paroma Varma, Priyadarshini Sebastian, Richard Gao, Torben Noto, Antonio H. Lara, Joni D. Wallis, Robert T. Knight, Avgusta Shestyuk, and Bradley Voytek. Parameterizing neural power spectra into periodic and aperiodic components. *Nature Neuroscience*, 23:1655–1665, 2020.

[76] Wolfgang Klimesch. Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2-3):169–195, 1999.

A Extended Related Work

A.1 Traditional EEG Analysis and Deep Learning

Early EEG research relied heavily on manual feature engineering to interpret neural dynamics. Researchers extracted domain-specific descriptors such as spectral power bands [12, 13], time-domain statistics [11, 31], and non-linear complexity measures like entropy [22, 32]. Despite the rise of deep learning, these handcrafted features remain the dominant approach in practical clinical and BCI applications due to their exceptional robustness [33, 34]. Unlike end-to-end models that can be brittle to distribution shifts, explicit features exhibit remarkable stability across varying subjects, recording devices, and electrode montages [35]. This invariance makes them indispensable for real-world scenarios where data heterogeneity is the norm rather than the exception. These features served as robust inputs for classical classifiers in applications ranging from sleep staging [36, 37] to emotion recognition [38]. With the advent of deep learning, the field shifted toward end-to-end architectures that learn directly from raw signals. Convolutional Neural Networks (CNNs) like EEGNet [39] and ShallowConvNet [40] became standard baselines for decoding oscillatory patterns. Subsequently, Transformer-based models such as EEG-Conformer [41], EEGFormer [42], and ST-Transformer [43] were introduced to capture long-range temporal dependencies. While these data-driven approaches achieved state-of-the-art performance on specific benchmarks [44, 45], they often require massive labeled datasets, which are scarce in the medical domain. Moreover, due to their limited generalization capabilities under distribution shifts, the practical performance of these end-to-end models is often significantly compromised in diverse clinical settings [46, 47].

A.2 EEG Foundation Models

To address the label scarcity bottleneck, the community has rapidly adopted diverse self-supervised learning paradigms to train Foundation Models. Masked Modeling remains the most dominant approach, adapting BERT-style objectives to reconstruct masked signal patches [5, 6, 14, 16]. Beyond simple masking, researchers have explored a rich variety of technical routes to capture temporal and spatial dependencies. Generative approaches leverage autoregressive next-token prediction to model the sequential nature of brain waves, as seen in BrainGPT [48] and NeuroLM [8]. Vector Quantization (VQ) techniques have also gained traction, discretizing continuous signals into codebook indices to facilitate token-based learning [19, 20, 21]. Parallel efforts explore contrastive learning strategies to align EEG representations with other modalities or augmentations [49, 50, 51]. Recent works have scaled these paradigms to larger datasets and more complex architectures, including BrainBERT [5], Brant [6], and BIOT [15]. More recently, researchers have investigated integrating EEG with LLMs to enable language-grounded EEG understanding [48, 8, 2, 3]. Innovations such as NeuroLM [8], BrainGPT [48], and LaBraM [52] demonstrate the potential of cross-modal alignment [7, 53, 54]. Comprehensive surveys [1, 55, 56, 57] highlight the rapid proliferation of these models, listing over dozens of variants proposed in the last two years [58, 59, 60, 61, 24, 62]. Fundamentally, EEG Foundation Models shift the paradigm from task-specific supervision to task-agnostic universality, enabling the learning of general-purpose representations from massive unlabeled corpora. This transition unlocks superior generalization capabilities, allowing models to adapt to unseen subjects and diverse hardware configurations with minimal fine-tuning [1, 56].

A.3 Knowledge Integration in Learning

Despite the success of data-driven Foundation Models, there is a growing recognition of the value of integrating domain knowledge into the learning loop. In Computer Vision and NLP, structured priors have been shown to improve sample efficiency and robustness [9, 63, 25, 64]. In the EEG domain, however, most Foundation Models continue to treat the signal as a generic time-series, ignoring the rich biophysical constraints established by traditional neuroscience [65, 66, 34]. A notable exception is DeeperBrain [3], which introduces a "Neurodynamics Statistics Prediction" objective. This approach explicitly trains the model to predict macroscopic order parameters, including Relative Spectral Power [12], Phase-Locking Values (PLV) [65], Cross-Frequency Coupling (CFC) [66], and Sample Entropy [22]. Such "Neuro-Grounded" designs demonstrate that incorporating biophysical inductive biases can significantly enhance representation robustness. While DeeperBrain focuses on specific statistics as reconstruction targets, few works have turned EEG signal-processing descriptors into a cross-paradigm knowledge interface. Neuro-KE fills this gap by formalizing a "Knowledge

Engine" that injects explicit time-frequency priors [67, 68] into modern architectures. This approach bridges the divide between classical feature engineering and modern representation learning, offering a unified framework for knowledge-guided foundation models [69, 70].

Recent EEG foundation models and adaptation. CodeBrain combines a decoupled time–frequency tokenizer with a multi-scale architecture for EEG pretraining [71]. SCOPE uses confidence-aware prototype guidance and lightweight adapters for adapting EEG foundation models with limited labeled subjects [72].

B Dataset Details

B.1 Dataset Summary

We utilize the TUEG corpus for self-supervised masked-modeling pretraining and a core set of downstream datasets for masked-modeling evaluation. Table 6 summarizes the key statistics and task definitions for this core set; additional datasets used in contrastive and EEG-MLLM analyses are described below.

Table 6. Summary of the 14 downstream EEG benchmarks.

Domain	Task	Dataset	Rate	Ch.	Window	Samples	Subjects	Label
Emotion	Emotion recognition	SEED	200	62	10 s	14,895	15	3-class
Emotion	Emotion recognition	SEED-V	1,000	62	1 s	117,744	16	5-class
Motor	Motor imagery	BCIC2A	250	22	4 s	5,184	9	4-class
Motor	Motor imagery	SHU-MI	250	32	4 s	7,571	25	2-class
Motor	Motor imagery	PhysioNet-MI	160	64	4 s	9,837	109	4-class
Cognitive	Mental arithmetic	EEGMAT	500	19	5 s	1,515	36	2-class
Sleep	Sleep staging	SleepEDF	100	2	30 s	414,961	78	5-class
Sleep	Sleep staging	ISRUC	200	6	30 s	50,196	57	5-class
Sleep	Sleep staging	HMC	256	4	30 s	111,890	124	5-class
Pathology	Abnormal EEG detection	TUAB	mixed	16	10 s	409,223	2,329	2-class
Pathology	Epilepsy detection	TUEP	mixed	16	10 s	226,837	200	2-class
Pathology	EEG event classification	TUEV	250	16	5 s	113,353	370	6-class
Pathology	Depression detection	MDD	256	19	5 s	7,143	63	2-class
Pathology	Alzheimer's detection	AD65	500	19	10 s	5,292	65	2-class

Note. Rate denotes the original sampling rate in Hz; all downstream windows are resampled to 200 Hz before evaluation. Mixed-rate TUH datasets contain recordings sampled at multiple original rates.

B.2 Dataset Descriptions

TUEG (Temple University EEG Corpus) [73]. TUEG serves as our large-scale unlabeled pretraining corpus for masked modeling. It contains over 30,000 clinical EEG recordings from diverse patients. Since the pretraining objective is self-supervised, we do not use diagnostic labels from TUEG.

SEED (Emotion Recognition) [38]. SEED is an emotion-recognition dataset in which subjects watched movie clips designed to elicit positive, neutral, or negative affective states. We use the three-class emotion-recognition protocol.

SEED-V (Emotion Recognition). SEED-V extends the SEED affective EEG setting to five emotion categories. We use 1-second windows from 62-channel recordings and evaluate five-class emotion recognition.

BCIC2A (Motor Imagery) [74]. BCIC2A is a classic motor-imagery BCI benchmark. Subjects imagine movements of the left hand, right hand, both feet, or tongue, forming a four-class motor-imagery task.

SHU-MI (Motor Imagery). SHU-MI is a binary motor-imagery dataset collected from 25 subjects. We use the benchmark protocol with 4-second EEG windows and classify the two motor-imagery conditions.

PhysioNet-MI (Motor Imagery). PhysioNet-MI is a large motor-imagery EEG benchmark with 109 subjects. We use the four-class motor-imagery protocol with 4-second windows.

EEGMAT (Mental Arithmetic). EEGMAT is a cognitive-state dataset for mental-arithmetic classification. We use the 19-channel protocol with 5-second windows and classify task-related cognitive state in a binary setting.

SleepEDF (Sleep Staging). SleepEDF is a sleep-staging benchmark built from overnight PSG recordings. We use 30-second windows and evaluate five-class sleep staging.

ISRUC (Sleep Staging). ISRUC is a sleep-staging dataset. In the current benchmark protocol, ISRUC is represented with 6 EEG channels and 30-second windows for five-class sleep-stage classification.

HMC (Sleep Staging). HMC is a sleep-staging benchmark with 30-second windows. We use the 4-channel protocol and evaluate five-class sleep staging.

TUAB (Abnormal EEG) [73]. TUAB is a subset of the Temple University Hospital EEG corpus labeled for clinical abnormality. The task is binary abnormal EEG detection, distinguishing normal from abnormal recordings.

TUEP (Epilepsy) [73]. TUEP focuses on epilepsy detection. The task is to distinguish patients with epilepsy from those without epilepsy, regardless of whether a specific segment contains an ongoing seizure.

TUEV (EEG Events) [73]. TUEV contains event-level annotations for clinically relevant EEG events. We use the six-class event-classification protocol with 5-second windows.

MDD (Depression Detection). MDD is a depression-detection EEG dataset. We use the 19-channel protocol with 5-second windows and evaluate binary classification between major depressive disorder and control subjects.

AD65 (Alzheimer's Disease Detection). AD65 is an Alzheimer's disease EEG dataset. We use the binary control-versus-Alzheimer's protocol, excluding the original FTD class and treating Alzheimer's disease as the positive class.

Additional datasets used in contrastive and EEG-MLLM analyses. Several sections use broader benchmark coverage than the 14 benchmarks listed in Table 6. HMC and SleepEDF/SleepEDF-full are sleep-staging datasets; ISRUC-S1 is a sleep-staging subset; SHU and SHU-MI are motor-imagery datasets; Workload is a cognitive-load dataset; FACED, SEED-IV, SEED-V, and SEED-VII are affective EEG datasets; PhysioNet-MI is a motor-imagery dataset; MDD is a depression-detection dataset; TUAR is an artifact-recognition subset of TUH; and TUSL is a slowing/event-related TUH subset used in the EEG-MLLM pipeline.

B.3 Experimental Setup

For all downstream tasks, we strictly adhere to a Cross-Subject evaluation protocol to ensure that the model is tested on subjects it has never seen during training, evaluating its ability to learn universal, subject-invariant EEG representations. We employ a consistent subject-wise splitting strategy across all datasets, allocating 80% of subjects for training, 10% for validation (hyperparameter tuning), and holding out the remaining 10% for final testing. For datasets with pre-defined splits (e.g., BCIC2A and SEED), we follow the standard benchmarks to ensure comparability.

B.4 Data Leakage Control

We use TUEG as the common unlabeled pretraining corpus for masked modeling, following the standard EEG foundation-model practice of pretraining on large TUH/TUEG-style recordings and evaluating on downstream subsets or separate benchmarks. No diagnostic labels from TUEG are used during masked-modeling pretraining. For TUH-derived downstream tasks such as TUAB, TUEP, and TUEV, downstream training, validation, and testing are still conducted with subject-disjoint splits, so recordings from the same subject are not shared across evaluation partitions.

For contrastive learning and EEG-MLLM experiments, all EEG segments, labels, descriptor states, text anchors, cached text embeddings, retrieval banks, and instruction-answer records are constructed within the corresponding split. The held-out retrieval banks used for EEG-to-text and text-to-EEG analysis are built only from test splits. Likewise, EEG-MLLM instruction tuning follows cross-subject splits for the EEG datasets used in Stage 1 and Stage 2, and evaluation examples are drawn from

held-out subjects. Metadata fields such as subject identifiers, recording identifiers, file paths, and segment indices are retained only for split control, indexing, and auditability; they are not included as supervision text or model inputs. These controls are intended to prevent segment-level leakage, subject-level leakage, and train/test reuse of descriptor-derived text.

B.5 Preprocessing

We apply a standardized preprocessing pipeline to all datasets. First, a bandpass filter from 0.1 Hz to 75 Hz is used to remove DC drift and high-frequency noise, accompanied by a notch filter at 60 Hz (or 50 Hz depending on dataset region) to eliminate power line interference. Subsequently, we perform per-channel z-score normalization, computed independently for each input segment according to:

$$\tilde{x}_{c,t} = \frac{x_{c,t} - \mu_{c,\text{seg}}}{\sigma_{c,\text{seg}} + \epsilon} \tag{6}$$

where $\mu_{c,\text{seg}}$ and $\sigma_{c,\text{seg}}$ denote the mean and standard deviation of channel c within the current segment. Finally, we standardize all recordings to the International 10-20 system , where missing channels are interpolated or zero-padded as necessary.

C Neuro-KE Feature Definitions

This appendix provides the precise mathematical formulations and taxonomy for one numeric Neuro-KE descriptor catalog used in the masked-modeling analyses. Let $x(t)$ denote the discrete EEG time-series of length T, sampled at frequency f_s. Let $P(f)$ denote the PSD of $x(t)$, computed via Welch's method or FFT.

The current TUEG descriptor catalog contains 87 exported EEG features designed to capture complementary aspects of neural activity. Following the feature names stored in the external TUEG feature H5 files, we group these descriptors into four families: `spectral`, `temporal`, `spatial`, and `dynamics`. The main text only summarizes these groups; this appendix records the concrete descriptor definitions and feature list.

Spectral descriptors include absolute and relative band powers, total power, spectral shape descriptors, band-wise differential entropy (DE), and composite spectral indices. Band power is computed by integrating the PSD over a frequency interval,

$$P_{\text{band}} = \int_{f_{\text{low}}}^{f_{\text{high}}} P(f)\, df, \tag{7}$$

and relative power normalizes each band by the total spectral power. This group also includes the aperiodic exponent, spectral entropy, spectral centroid, individual alpha frequency, and ratios such as theta/beta and delta/theta [12, 75, 22, 76].

Temporal descriptors summarize waveform morphology directly from $x(t)$, including mean absolute amplitude, channel standard deviation, peak-to-peak amplitude, RMS, zero-crossing rate, and Hjorth activity/mobility/complexity [13, 11, 31]. These features capture amplitude, variability, roughness, and time-domain proxies for spectral behavior.

Spatial descriptors represent relational structure across scalp regions. Because the current 87-feature export does not contain explicit coherence, PLV, or graph-topology features, this group is formed by inter-hemispheric asymmetry and anterior-posterior relational descriptors: DASM, RASM, DCAU, and frontal alpha asymmetry (FAA). For band b, DASM uses left-right differences, RASM uses left-right ratios, and DCAU uses anterior-posterior differences.

Dynamics descriptors combine nonlinear complexity features with EEG microstate dynamics. The nonlinear part includes wavelet energy entropy and fractal/long-memory descriptors such as Hurst exponent, Higuchi fractal dimension, Katz fractal dimension, and Petrosian fractal dimension. The microstate part records mean duration, occurrence, time coverage, mean correlation, and global explained variance (GEV) for each of four microstate classes.

C.1 Feature List Table

Table 7. Complete list of the 87-feature Neuro-KE catalog used for masked-modeling feature prediction.

Index	Group	Family	Feature Name	Description
0	spectral	Power	delta_power	Absolute delta-band power (1–4 Hz)
1	spectral	Power	theta_power	Absolute theta-band power (4–8 Hz)
2	spectral	Power	alpha_power	Absolute alpha-band power (8–13 Hz)
3	spectral	Power	beta_power	Absolute beta-band power (13–30 Hz)
4	spectral	Power	gamma_power	Absolute gamma-band power
5	spectral	Power	low_gamma_power	Absolute low-gamma power
6	spectral	Power	high_gamma_power	Absolute high-gamma power
7	spectral	Relative power	delta_relative_power	Delta power normalized by total power
8	spectral	Relative power	theta_relative_power	Theta power normalized by total power
9	spectral	Relative power	alpha_relative_power	Alpha power normalized by total power
10	spectral	Relative power	beta_relative_power	Beta power normalized by total power
11	spectral	Relative power	gamma_relative_power	Gamma power normalized by total power
12	spectral	Relative power	low_gamma_relative_power	Low-gamma power normalized by total power
13	spectral	Relative power	high_gamma_relative_power	High-gamma power normalized by total power
14	spectral	Structure	peak_frequency	Frequency with maximum PSD power
15	spectral	Structure	spectral_entropy	Entropy of normalized spectral power
16	spectral	Structure	spectral_centroid	Center of mass of the power spectrum
17	spectral	Structure	individual_alpha_frequency	Peak frequency within the alpha band
18	spectral	Ratio	theta_beta_ratio	Theta-to-beta power ratio
19	spectral	Ratio	delta_theta_ratio	Delta-to-theta power ratio
20	spectral	Ratio	low_high_power_ratio	Low-frequency to high-frequency power ratio
21	spectral	Aperiodic	aperiodic_exponent	Slope of the aperiodic 1/f spectral background
22	spectral	Total power	mean_total_power	Total integrated spectral power
23	spectral	DE	de_delta	Differential entropy in the delta band
24	spectral	DE	de_theta	Differential entropy in the theta band
25	spectral	DE	de_alpha	Differential entropy in the alpha band
26	spectral	DE	de_beta	Differential entropy in the beta band
27	spectral	DE	de_gamma	Differential entropy in the gamma band
28	spectral	DE	de_low_gamma	Differential entropy in the low-gamma band
29	spectral	DE	de_high_gamma	Differential entropy in the high-gamma band
30	spectral	Ratio	theta_alpha_ratio	Theta-to-alpha power ratio
31	spectral	Ratio	frontal_beta_ratio	Frontal beta-band ratio index
32	spectral	Composite	cognitive_load_estimate	Composite spectral estimate of cognitive load
33	spectral	Composite	alertness_estimate	Composite spectral estimate of alertness
34	spectral	Composite	relaxation_index	Composite spectral relaxation index
35	temporal	Amplitude	mean_abs_amplitude	Mean absolute signal amplitude
36	temporal	Variability	mean_channel_std	Mean channel-wise amplitude standard deviation
37	temporal	Amplitude	mean_peak_to_peak	Mean peak-to-peak amplitude
38	temporal	Amplitude	mean_rms	Root mean square amplitude
39	temporal	Morphology	mean_zero_crossing_rate	Mean zero-crossing rate
40	temporal	Hjorth	hjorth_activity	Hjorth activity; signal variance
41	temporal	Hjorth	hjorth_mobility	Hjorth mobility; frequency proxy
42	temporal	Hjorth	hjorth_complexity	Hjorth complexity; bandwidth/change proxy
43	spatial	DASM	dasm_delta	Delta-band differential asymmetry
44	spatial	DASM	dasm_theta	Theta-band differential asymmetry
45	spatial	DASM	dasm_alpha	Alpha-band differential asymmetry
46	spatial	DASM	dasm_beta	Beta-band differential asymmetry
47	spatial	DASM	dasm_gamma	Gamma-band differential asymmetry
48	spatial	RASM	rasm_delta	Delta-band rational asymmetry
49	spatial	RASM	rasm_theta	Theta-band rational asymmetry
50	spatial	RASM	rasm_alpha	Alpha-band rational asymmetry
51	spatial	RASM	rasm_beta	Beta-band rational asymmetry
52	spatial	RASM	rasm_gamma	Gamma-band rational asymmetry
53	spatial	DCAU	dcau_delta	Delta-band anterior-posterior differential caudality
54	spatial	DCAU	dcau_theta	Theta-band anterior-posterior differential caudality
55	spatial	DCAU	dcau_alpha	Alpha-band anterior-posterior differential caudality
56	spatial	DCAU	dcau_beta	Beta-band anterior-posterior differential caudality
57	spatial	DCAU	dcau_gamma	Gamma-band anterior-posterior differential caudality
58	spatial	FAA	faa_f3f4	Frontal alpha asymmetry for F3/F4
59	spatial	FAA	faa_f7f8	Frontal alpha asymmetry for F7/F8
60	spatial	FAA	faa_fp1fp2	Frontal alpha asymmetry for FP1/FP2
61	spatial	FAA	faa_mean	Mean frontal alpha asymmetry

Continued on next page

Index	Group	Family	Feature Name	Description
62	dynamics	Entropy	wavelet_energy_entropy	Entropy of normalized wavelet energy
63	dynamics	Fractal	hurst_exponent	Long-range dependence / Hurst exponent
64	dynamics	Fractal	higuchi_fd	Higuchi fractal dimension
65	dynamics	Fractal	katz_fd	Katz fractal dimension
66	dynamics	Fractal	petrosian_fd	Petrosian fractal dimension
67	dynamics	Microstate 0	Microstate_0_meandurs	Mean duration of microstate 0
68	dynamics	Microstate 0	Microstate_0_occurrence	Occurrence rate of microstate 0
69	dynamics	Microstate 0	Microstate_0_timecov	Time coverage of microstate 0
70	dynamics	Microstate 0	Microstate_0_mean_corr	Mean template correlation of microstate 0
71	dynamics	Microstate 0	Microstate_0_gev	Global explained variance of microstate 0
72	dynamics	Microstate 1	Microstate_1_meandurs	Mean duration of microstate 1
73	dynamics	Microstate 1	Microstate_1_occurrence	Occurrence rate of microstate 1
74	dynamics	Microstate 1	Microstate_1_timecov	Time coverage of microstate 1
75	dynamics	Microstate 1	Microstate_1_mean_corr	Mean template correlation of microstate 1
76	dynamics	Microstate 1	Microstate_1_gev	Global explained variance of microstate 1
77	dynamics	Microstate 2	Microstate_2_meandurs	Mean duration of microstate 2
78	dynamics	Microstate 2	Microstate_2_occurrence	Occurrence rate of microstate 2
79	dynamics	Microstate 2	Microstate_2_timecov	Time coverage of microstate 2
80	dynamics	Microstate 2	Microstate_2_mean_corr	Mean template correlation of microstate 2
81	dynamics	Microstate 2	Microstate_2_gev	Global explained variance of microstate 2
82	dynamics	Microstate 3	Microstate_3_meandurs	Mean duration of microstate 3
83	dynamics	Microstate 3	Microstate_3_occurrence	Occurrence rate of microstate 3
84	dynamics	Microstate 3	Microstate_3_timecov	Time coverage of microstate 3
85	dynamics	Microstate 3	Microstate_3_mean_corr	Mean template correlation of microstate 3
86	dynamics	Microstate 3	Microstate_3_gev	Global explained variance of microstate 3

D Feature-Guided Data Construction

D.1 Knowledge-Guided Masked Modeling: Feature Preprocessing

Features for this paradigm are extracted from the TUEG pretraining corpus. To ensure stable training convergence and mitigate the impact of non-Gaussian distributions (e.g., extreme skewness and kurtosis commonly found in spectral power), we implement a hybrid adaptive preprocessing pipeline. This pipeline processes each feature dimension independently through three sequential steps. First, in the distribution quality check, we evaluate the normality of each feature using Skewness (S) and Kurtosis (K), classifying a feature as "Normal" if $|S| \leq 2.0$ and $K \leq 7.0$, and "Skewed" otherwise. Second, we apply adaptive transformation based on this classification: for Normal features, we retain the raw values; for Skewed features, we first apply a 6-Sigma Truncation to clamp values within $[\mu - 6\sigma, \mu + 6\sigma]$ to remove extreme outliers, and subsequently apply a Log1p transformation ($y = \log(x + 1)$) if the values are non-negative to correct right-skewed heavy tails. Finally, regardless of the previous steps, all features undergo global standardization via Z-Score Normalization ($z = \frac{x-\mu}{\sigma}$) to unify the feature space into a standard scale ($\mu = 0, \sigma = 1$). The final pretraining dataset comprises approximately 8872 hours of recordings.

D.2 Reproducibility Details for Descriptor Supervision

The descriptor pipeline is deterministic once the raw segment, channel montage, sampling rate, and preprocessing settings are fixed. For masked modeling, each descriptor dimension is processed independently using the distribution-quality rule above: features with $|S| \leq 2.0$ and $K \leq 7.0$ are treated as approximately normal, while the remaining features are treated as skewed. Skewed dimensions are clipped with the 6-sigma rule and log-transformed when non-negative before global z-score normalization. The normalized descriptor targets, loss weight λ, random seed, split seed, backbone checkpoint, and fold identifier are recorded for each run.

For contrastive learning and EEG-MLLM data construction, descriptor values are converted into five percentile states: < 10th, 10–25th, 25–75th, 75–90th, and > 90th percentile. The same threshold table is used for label-plus-Neuro-KE anchors, standalone retrieval anchors, and instruction-answer generation. All generated text is derived from fixed templates or from prompts that require the model to use only supplied feature values and states. The judge prompt for EEG-MLLM evaluation asks for agreement with the provided deterministic reference and the cited feature states, not for independent diagnosis or clinical recommendation.

D.3 Knowledge-Driven Contrastive Learning: Structured EEG-Text Pairs

To address the scarcity of paired EEG-text data, we construct deterministic EEG-text pairs from Neuro-KE descriptor profiles and existing benchmark labels. The contrastive experiments use six training datasets: TUAB, TUEP, SHU, SEED, SleepEDF, and Workload. Metadata such as dataset, split, subject/session when available, file identifier, and segment identifier is retained as indexing information for joining EEG segments with labels, feature states, cached embeddings, and retrieval-bank entries; it is not treated as supervision text.

We compare two text-anchor settings. **Label only** uses only the benchmark class-label text as the text anchor. **Label + Neuro-KE text** keeps the label anchor and adds deterministic Neuro-KE feature-state text generated from the segment descriptor profile. For each feature, the value is mapped to a percentile state, and the corresponding text records the feature name, raw value, z-score, percentile, state phrase, and a concise physiological/clinical interpretation. The descriptor-derived feature-state sentences are therefore generated automatically and require no new human annotation; when benchmark labels are included, we describe the setting as label-conditioned structured supervision rather than fully self-supervised training.

All text anchors are embedded offline with Qwen3-Embedding-8B [27]. The resulting 4096-dimensional vectors are cached and indexed back to EEG segments using the metadata keys above. The text side is divided into a label part and descriptor-text parts. In the label part, EEG segments with the same dataset class are positive for the corresponding class-label anchors, including off-diagonal pairs. In each descriptor-text part, only the source EEG sample is positive for its own text; other samples are negative even if they share a feature-state label. These explicit rules define T_{ij} in Eq. 4, and padded text columns are excluded by m_j. The loss sums valid text-pair terms for each EEG sample and averages over the EEG batch. Multiple label positives arise from the target mask, not from the sigmoid function alone; shared-state descriptor pairs can still be false negatives under the sample-specific rule.

Template family A: label-only anchors. The label-only setting is the control condition used to test whether adding Neuro-KE text improves EEG–text alignment beyond ordinary class names. For a segment with task label c, the text anchor is kept deliberately short:

> This EEG segment is associated with the class label: {label name}.

For zero-shot downstream transfer, the class prototype text uses the same label-name style without Neuro-KE descriptors:

> An EEG segment associated with {target label name}.

Template family B: label plus segment-feature anchors. The main Neuro-KE contrastive setting appends deterministic segment-feature descriptions to the label anchor. Each selected descriptor is verbalized with the same grammar used in the text-generation resource: feature name, raw value, z-score, percentile, discrete state phrase, and task-aware physiological context. For each EEG segment, the template is:

> This EEG segment is associated with the class label: {label name}. In this EEG segment, the {feature name 1} shows a raw value of {value 1}, a Z-score of {z-score 1}, and is at the {percentile 1} percentile, indicating it is {state phrase 1}. As physiological context, {task-relevant interpretation 1}. In this EEG segment, the {feature name 2} shows a raw value of {value 2}, a Z-score of {z-score 2}, and is at the {percentile 2} percentile, indicating it is {state phrase 2}. As physiological context, {task-relevant interpretation 2}.

A concrete instance is:

> This EEG segment is associated with the class label: abnormal EEG. In this EEG segment, the delta relative power shows a raw value of 0.55, a Z-score of 1.65, and is at the 95th percentile, indicating it is markedly increased (>95th %ile). As physiological context, focal delta waves can appear during interictal periods of epilepsy, and increased frontal delta has been reported in some depressed patients while awake. In this EEG segment, the alpha relative power shows a raw value of 0.10, a Z-score of -1.45, and is at the 7th percentile, indicating it is suppressed (<10th %ile). As physiological context, alpha decreases can

> occur during contralateral sensory-motor ERD and can also reflect altered relaxation or arousal state.

The exact number of descriptor sentences can be adjusted by the feature-selection heuristic, but every sentence is generated deterministically from descriptor values, percentile states, and a fixed feature-interpretation dictionary rather than manually written per sample.

Template family C: standalone feature-state anchors for retrieval. For feature-level retrieval, we remove the class label and use one feature-state sentence as the text anchor. This isolates whether the learned EEG–text space can recover a held-out physiological state rather than only task labels.

> In this EEG segment, the {feature name} shows a raw value of {value}, a Z-score of {z-score}, and is at the {percentile} percentile, indicating it is {state phrase}. As physiological context, {task-relevant interpretation}.

Example feature-state anchors include:

> In this EEG segment, the mean abs amplitude shows a raw value of 3.4521, a Z-score of 0.92, and is at the 82nd percentile, indicating it is moderately prominent (75–90th %ile). As physiological context, amplitude bursts can occur during epileptic seizures, and generalized amplitude changes may be associated with high emotional arousal.
>
> In this EEG segment, the hjorth mobility shows a raw value of 0.1823, a Z-score of -0.85, and is at the 15th percentile, indicating it is attenuated (10–25th %ile). As physiological context, Hjorth mobility can fluctuate with fast/slow wave dominance and may shift over contralateral cortex during motor imagery.
>
> In this EEG segment, the wavelet energy entropy shows a raw value of 2.1, a Z-score of 0.40, and is at the 65th percentile, indicating it is within the reference range (25–75th %ile). As physiological context, wavelet energy entropy can decrease when energy concentrates in narrow rhythms and can increase with more transient signal complexity.

For feature-level retrieval, each feature is discretized into five percentile-state labels: < 10th, 10–25th, 25–75th, 75–90th, and > 90th percentile. These state labels define reusable feature-state anchors shared across samples and datasets. For EEG-to-text retrieval, an EEG segment and a feature name query the five possible state anchors for that feature. For text-to-EEG retrieval, a feature-state anchor queries all EEG segments in the held-out retrieval bank, and positives are segments with the same feature-state label.

D.4 Knowledge-Enhanced EEG-Large Language Models: Two-Stage Instruction Data

We construct EEG-MLLM instruction data in the format of (x_{eeg}, x_{prompt}, A), where x_{eeg} is a short EEG segment, x_{prompt} is the user instruction, and A is the supervised assistant answer. The current data pipeline uses two stages rather than a three-stage classification/text curriculum.

Stage 1: EEG Description Alignment. Stage 1 uses EEG description text that combines Neuro-KE feature descriptions, metadata, labels, and other relevant context. The goal is to align EEG tokens with the language space by training the model to generate rich descriptions of the signal rather than only a class label. The Stage 1 index covers seven EEG datasets: HMC, TUAB, TUAR, TUEP, TUEV, TUSL, and TUSZ.

Stage 2: Task-Oriented EEG Instruction Tuning. Stage 2 uses four benchmark-style EEG datasets: HMC, TUEV, SHU-MI, and SEED. It contains five task types:

1. **Single-feature state explanation:** explain the numerical state of a specified feature and its possible physiological meaning.
2. **Three-feature description:** select three important features, describe them, and analyze their physiological implications.
3. **Label-support explanation:** select one to four important features and explain how these features support the label.
4. **Multiple-choice feature selection:** choose the most reasonable option among four feature-related choices (one correct, three distractors) and explain the decision.
5. **Direct label prediction:** predict the task label from the EEG segment.

Stage 2 also mixes text-only instruction data to preserve general language ability while adapting the LoRA parameters to EEG tasks.

Training coverage and label-prediction evaluation. The label-prediction comparison in Table 3 follows the two-stage training procedure above, without an additional third training stage. Its evaluation datasets are not identical to the datasets listed in the instruction mixtures. Table 8 makes this distinction explicit: AD65 and MDD are not listed in either mixture, while TUAB is listed in Stage 1 only. These memberships alone do not establish whether a dataset is unseen during the corresponding evaluation run. Dataset-specific training configurations, checkpoint identifiers, class-name mappings, and evaluation split manifests are not fully specified here; we plan to provide them with the training code on GitHub.

Table 8. Documented instruction-mixture coverage of the label-prediction evaluation datasets. "Not listed" refers to the mixtures described above and does not establish a zero-shot evaluation protocol.

Dataset	Stage 1 mixture	Stage 2 EEG mixture
AD65	Not listed	Not listed
SEED	Not listed	Listed
TUAB	Listed	Not listed
MDD	Not listed	Not listed
TUEV	Listed	Listed
SHU-MI	Not listed	Listed

D.5 Fixed-Template Generation of EEG Segment Descriptions

We generated textual descriptions of EEG segments from structured, standardized feature representations. Each EEG segment was represented by a set of numerical features extracted from time-domain, frequency-domain, differential entropy, complexity, and microstate measurements. Metadata fields, such as subject identifiers, segment identifiers, labels, timestamps, and file paths, were excluded from the feature selection process and retained only for traceability.

For each segment, a small subset of valid numerical EEG features was selected. Features with non-finite values were excluded. Feature names originally stored as machine-readable column identifiers were converted into human-readable descriptions. For example, channel-specific frequency-domain features were expressed as band-specific power at a given electrode, and differential entropy features were described by frequency band and electrode.

Each standardized feature value was mapped to a qualitative state relative to the dataset reference distribution. Values close to the dataset reference range were described as being within the reference range, whereas larger positive or negative deviations were described as above, below, markedly above, or markedly below the reference range. This mapping enabled the generated text to combine numerical precision with interpretable qualitative summaries.

The final description for each segment was produced using a fixed natural-language template. The instruction specified the selected EEG features to be analyzed:

```
Analyze the EEG segment using {selected_feature_names}.
```

The corresponding description summarized the selected features, their qualitative states, and their standardized values:

```
In this EEG segment, {feature_name} is {feature_state} with a standardized
feature value of {value}.
```

When multiple features were selected, the same clause structure was repeated and joined into a single sentence. This fixed-template strategy produced deterministic, grounded descriptions while avoiding unsupported clinical interpretation.

D.6 LLM Prompt Design for EEG QA Generation

To generate more diverse question-answer data, we constructed prompts for an external large language model using the same standardized EEG feature representations. The model was instructed to generate

concise English QA examples grounded only in the provided feature values. The prompt design consisted of a system prompt, a shared user-prompt header, a list of candidate EEG features, and a task-specific instruction.

D.6.1 System Prompt

```
You are an EEG feature QA generation assistant. Generate concise, accurate,
grounded QA data from structured EEG feature values.
Do not invent measurements, channels, labels, files, or diagnoses.
```

D.6.2 Shared User-Prompt Header

Each user prompt began with the dataset name and a brief task context, followed by grounding instructions:

```
Dataset: {dataset}
Dataset task context: {label_context}

Use only the EEG features provided below. Feature format is:
Full EEG feature name: standardized value, state.
All generated QA text must be in English, concise, and grounded in the
provided feature values.
```

Candidate features were then provided in the following format:

```
- {full_feature_name}: {standardized_value}, {feature_state}
```

D.6.3 Single-Feature State QA

This prompt asked the model to generate one QA pair about a single EEG feature, including its numerical state and possible physiological interpretation:

```
Specified EEG feature:
{feature_lines}

Task:
Generate one concise QA pair asking about the numerical state and possible
physiological meaning of this feature.

Rules:
- The answer must mention the standardized value and state.
- Explain a plausible physiological signal meaning in 1-2 sentences.
- Do not claim a definitive clinical diagnosis.
- Output only valid JSON with keys: qa_type, question, answer,
  selected_features.
```

D.6.4 Three-Feature Description QA

This prompt provided multiple candidate features and asked the model to select exactly three informative features:

```
Random candidate EEG features:
{feature_lines}

Task:
Choose exactly 3 features that you judge to be most informative.
Generate one concise QA pair describing the 3 features and their
possible physiological meaning.

Rules:
- Choose only from the candidate features above.
- Mention each selected feature's full name, value, and state.
- Keep the answer short and do not overstate certainty.
- Output only valid JSON with keys: qa_type, question, answer,
  selected_features.
```

D.6.5 Label-Support QA

This prompt incorporated the segment label and asked the model to identify features that may support that label:

```
Provided segment label: {label}

Random candidate EEG features:
{feature_lines}

Task:
Choose 1 to 4 features that are most consistent with the provided label.
Generate one concise QA pair explaining how those features may support the
label.

Rules:
- Choose only from the candidate features above.
- Use cautious wording such as "consistent with" or "may support".
- Do not invent symptoms, EEG waveforms, or pathology not present in the
  provided data.
- Output only valid JSON with keys: qa_type, question, answer,
  selected_features.
```

D.6.6 Multiple-Choice QA

This prompt asked the model to construct a four-option multiple-choice question:

```
Random candidate EEG features:
{feature_lines}

Task:
Generate one concise multiple-choice QA item about these features. You must
write the question, four options (A-D), the correct answer, and a brief
explanation.

Rules:
- Create exactly 4 options: 1 true and 3 false.
- The true option must be directly supported by the provided feature values
  or states.
- False options must be plausible but contradicted by or unsupported by the
  provided features.
- Keep the explanation to 1-2 sentences.
- Output only valid JSON with keys: qa_type, question, options,
  correct_option, answer, explanation, selected_features.
```

Overall, the LLM prompts were designed to preserve factual grounding, require explicit use of the provided feature values, and discourage unsupported diagnostic claims. The output format was constrained to JSON to facilitate automated parsing and downstream dataset construction.

Table 9. EEG-MLLM instruction data summary.

STAGE	DATA SOURCE	TRAINING EXAMPLES
STAGE 1	HMC, TUAB, TUAR, TUEP, TUEV, TUSL, TUSZ	1,550,537
STAGE 2 EEG	HMC, TUEV, SHU-MI, SEED	316,325
STAGE 2 TEXT	ULTRACHAT 200K	200,000

E Model Architecture & Hyperparameters

E.1 Masked Modeling Architecture

E.1.1 Model Architecture

The CBraMod masked-modeling backbone is a 12-layer Transformer designed to process high-dimensional EEG signals efficiently. The detailed specifications are listed in Table 10.

We employ a factorized attention mechanism and specialized embeddings. The **Criss-Cross Attention** module factorizes attention into **Spatial Attention** (across $C = 19$ channels) and **Temporal Attention** (across $N = 60$ time patches), reducing complexity from quadratic to linear with respect to sequence length. The embedding layer consists of three components: a **Temporal** embedding using a 3-layer 1D Convolution stack, a **Spectral** embedding using FFT-based frequency encoding, and a **Positional** embedding implemented via depthwise separable convolution.

Table 10. Encoder Specifications.

COMPONENT	SPECIFICATION	DESCRIPTION
BACKBONE	TRANSFORMER	HIERARCHICAL TRANSFORMER WITH CRISS-CROSS ATTENTION
MODEL DIMENSION	$d_{model} = 200$	HIDDEN DIMENSION OF THE ENCODER
FEED-FORWARD DIM	$d_{ff} = 800$	DIMENSION OF THE FEED-FORWARD NETWORK
ATTENTION HEADS	$H = 8$	NUMBER OF PARALLEL ATTENTION HEADS
LAYERS	$L = 12$	NUMBER OF TRANSFORMER LAYERS
PATCH SIZE	$P = 200$	1 SECOND AT 200 HZ
SEQUENCE LENGTH	$N = 10$	10 PATCHES (10 SECONDS)

E.1.2 Training Strategy

We adopt patch-level Bernoulli masking with probability $p = 0.5$. Consistent with Eq. (2), $\mathcal{L}_{MM} = \lambda\mathcal{L}_{\text{recon}} + (1-\lambda)\mathcal{L}_{\text{feat}}$, with $\lambda = 0.5$ for joint pretraining. Reconstruction uses masked signal MSE, while feature prediction uses MSE on z-scored Neuro-KE descriptors. The pre-training hyperparameters are summarized in Table 11.

Table 11. Pre-training hyperparameters. Loss weights use the normalized coefficients in Eq. (2).

HYPERPARAMETER	VALUE
OPTIMIZER	ADAMW
LEARNING RATE	3×10^{-4}
WEIGHT DECAY	0.05
BATCH SIZE	512 (4×128)
EPOCHS	30
SCHEDULER	COSINE ANNEALING ($\eta_{min} = 10^{-6}$)
WARMUP	NONE
GRADIENT CLIPPING	1.0
RECONSTRUCTION WEIGHT (λ)	0.5
FEATURE WEIGHT ($1-\lambda$)	0.5

E.2 Contrastive Learning Architecture

The contrastive learning framework aligns EEG representations with cached text-anchor embeddings using a dual-encoder architecture.

E.2.1 Model Architecture

The EEG encoder adopts the `CBraMod` backbone, consisting of a patch embedding layer and a Transformer encoder. The patch embedding module processes raw EEG signals using a sequence of 2D convolutions (kernels 1×49, 1×3, 1×3) combined with spectral features extracted via FFT.

The Transformer encoder comprises 12 layers with 8 attention heads, a model dimension (d_{model}) of 200, and a feedforward dimension of 800. The text side uses pre-computed Qwen3-Embedding-8B text embeddings with a dimension of 4096. The text encoder itself is frozen because all label and feature-state texts are embedded offline; only the EEG encoder and modality adapters are trained during contrastive alignment. Both modalities are projected into a shared 1024-dimensional latent space for alignment. The detailed architecture of the adapters and dimension transformations is presented in Table 12.

Table 12. Contrastive Adapter Architecture and Dimension Alignment.

PIPELINE	INPUT SOURCE	SOURCE DIM	ADAPTER STRUCTURE	OUTPUT DIM
EEG STREAM	CBRAMOD TOKENS	$C \times N \times 200$	FLATTEN + MLP WITH 512 AND 256 HIDDEN UNITS	1024
TEXT STREAM	QWEN3 TEXT EMBEDDING	4096	LINEAR PROJECTION	1024

E.2.2 Training Strategy

The model is trained using the Sigmoid Loss for Language Image Pre-training (SigLIP) on TUAB, TUEP, SHU, SEED, SleepEDF, and Workload. For the Label-only setting, the text side contains only label anchors. For the Label + Neuro-KE text setting, the text side contains the label anchor and the Neuro-KE feature-state anchors associated with the segment. The main supervised evaluations in Table 1 use the same downstream classification protocol for both settings, so improvements reflect the effect of adding Neuro-KE text during contrastive pretraining. Key hyperparameters are detailed in Table 13.

Table 13. Paradigm 2: Contrastive Learning Hyperparameters.

HYPERPARAMETER	VALUE
OPTIMIZER	ADAMW
LEARNING RATE	1×10^{-4}
WEIGHT DECAY	0.01
BATCH SIZE	1024
EPOCHS	UP TO 100, DATASET-LEVEL EARLY STOPPING
EARLY-STOP PATIENCE	6 EPOCHS WITHOUT AVERAGE BACC IMPROVEMENT
SCHEDULER	COSINE ANNEALING
LOSS FUNCTION	SIGLIP
TEXT EMBEDDING MODEL	QWEN3-EMBEDDING-8B (4096-DIM, FROZEN CACHE)
PROJECTION DIM	1024
DROPOUT	0.1

E.3 EEG Language Model Architecture

E.3.1 Model Architecture

The EEG-MLLM system integrates a LaBraM EEG encoder with a Large Language Model through a lightweight MLP adapter, as shown in Table 14. All EEG segments in the current setup are no longer than 10 seconds, so the projected EEG sequence is typically shorter than 200 tokens. We directly concatenate the projected EEG tokens with the instruction tokens before feeding them into the LLM.

Table 14. EEG-LLM System Configuration.

MODULE	COMPONENT	DETAILS
EEG ENCODER	LABRAM BASE	INITIALIZED FROM PRETRAINED `patch200_200` WEIGHTS AND FINE-TUNED
EEG LENGTH	$\leq$10 SECONDS	PROJECTED EEG SEQUENCE USUALLY $<$200 TOKENS
PROJECTOR	3-LAYER MLP ADAPTER	LINEAR-RELU-LINEAR-RELU-LINEAR TO d_{llm}
TOKEN FUSION	CONCATENATION	PROJECTED EEG TOKENS + INSTRUCTION TOKENS
LLM BACKBONE	QWEN3-4B-BASE	LOADED IN BFLOAT16
LORA CONFIG	LOW-RANK ADAPTATION	$r = 32$, $\alpha = 64$, DROPOUT=0.1
TARGET MODULES	ATTENTION/MLP LINEAR LAYERS	`q,k,v,o,gate,up,down_proj`

E.3.2 Training Strategy

The training proceeds in two stages. In **Stage 1 (EEG Description Alignment)**, the LLM is frozen while the EEG encoder and MLP adapter are trained on EEG description text that contains feature descriptions, metadata, labels, and other relevant context. **Stage 2 (Task-Oriented Instruction Tuning)** initializes from Stage 1, injects LoRA into the LLM, and jointly trains the EEG encoder, MLP adapter, and LoRA parameters on five EEG task types plus mixed text-only instruction data. The hyperparameters for the instruction tuning phase are provided in Table 15.

Table 15. Instruction Tuning Hyperparameters.

HYPERPARAMETER	STAGE 1	STAGE 2
OPTIMIZER	ADAMW	ADAMW
BATCH SIZE	8	6
EPOCHS	2	2
TOTAL STEPS	60,000	20,000
EEG ENCODER LR	1×10^{-4}	1×10^{-5}
MLP ADAPTER LR	1×10^{-4}	1×10^{-4}
LLM LORA LR	–	1×10^{-4}
WARMUP	5% OF TOTAL STEPS	5% OF TOTAL STEPS
SCHEDULER	COSINE WITH WARMUP	COSINE WITH WARMUP
PRECISION	BF16	BF16
EEG/TEXT BATCH RATIO	EEG ONLY	0.7 / 0.3

F Evaluation Metrics

To assess model performance across EEG tasks, we use task-specific metrics for masked modeling, contrastive learning, and EEG-MLLM generation. For EEG-MLLM, balanced accuracy is reported for direct label prediction (Table 3), while MCQ accuracy is the fraction of correct multiple-choice answers (Table 5). Free-form generation is evaluated using ROUGE-L, METEOR, BERTScore, and LLM-as-Judge (Table 5).

F.1 Balanced Accuracy (B-Acc)

For classification-style EEG benchmarks, including EEG-MLLM label prediction, balanced accuracy is defined as the arithmetic mean of per-class recall:

$$\text{B-Acc} = \frac{1}{K} \sum_{k=1}^{K} \frac{TP_k}{TP_k + FN_k}, \tag{8}$$

where K denotes the number of classes. This metric ensures that minority and majority classes contribute equally to the final score.

F.2 Contrastive Classification and Downstream Transfer

For the six contrastive-training datasets in Table 1, we evaluate whether the EEG–text aligned representation supports the same downstream classification protocol under two pretraining text settings: Label only and Label + Neuro-KE text. Both settings use the same EEG encoder family, adapter architecture, classifier/training recipe, and dataset splits; the only changed factor is whether Neuro-KE text is included during contrastive pretraining.

For downstream transfer in Table 2, we load the pretrained contrastive EEG weights and evaluate five target datasets not used for the main six-dataset comparison. The zero-shot setting does not use target-dataset training examples. It embeds only the target label names as class prototypes and assigns each EEG segment to the label with the highest EEG–label similarity in the shared contrastive space. The 50% and 100% settings then use the corresponding fraction of labeled target training data for downstream full fine-tuning and report test-set BAcc. No Neuro-KE text is provided as an input during downstream testing; the comparison measures whether pretraining weights learned with Neuro-KE text transfer better when evaluated from EEG and labels.

F.3 Feature-State Retrieval Metrics

We use feature-state retrieval as an alignment case study rather than as the main downstream benchmark. Each Neuro-KE feature is converted into a five-way percentile-state label: < 10th, 10–25th, 25–75th, 75–90th, and > 90th percentile. The retrieval bank concatenates the test splits of the six contrastive-training datasets: TUAB, TUEP, SHU, SEED, SleepEDF, and Workload.

For EEG-to-text (E2T) retrieval, an EEG segment and a selected feature define the query, and the system ranks the five text anchors for that feature. We report recall at k and mean reciprocal rank:

$$\mathrm{R@}k = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{1}[\mathrm{rank}_q(y_q) \le k], \qquad \mathrm{MRR} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{\mathrm{rank}_q}, \tag{9}$$

where y_q is the correct percentile-state label for query q and Q is the number of queries. Because E2T is a five-way ranking problem, the random chance values are R@1=20.00%, R@3=60.00%, and MRR=$\frac{1}{5}(1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}) = 0.457$.

For text-to-EEG (T2E) retrieval, a feature-state text anchor queries EEG segments in the six-dataset test bank. An EEG segment is counted as positive if it has the same feature and percentile-state label. We report R@1, R@10, and mean average precision (mAP). For a query with P_q positives among M EEG candidates, random-ranking chance is approximately R@k=$1 - \frac{\binom{M-P_q}{k}}{\binom{M}{k}}$ for top-k hit rate, while random mAP depends on the full positive count and candidate order distribution. These quantities depend on the positive-set size for each query. Table 4 reports dataset-specific chance baselines for the displayed retrieval banks.

F.4 ROUGE-L

For EEG-MLLM generative tasks, ROUGE-L measures the longest-common-subsequence overlap between the model output and the ground-truth answer. It is useful for checking whether the generated explanation covers key reference expressions while preserving a similar ordering.

F.5 METEOR

METEOR complements ROUGE-L by accounting for token-level matching beyond exact surface overlap. We use it to evaluate whether generated feature explanations and physiological interpretations are close to the reference answers in natural-language form.

F.6 BERTScore

BERTScore computes semantic similarity between model outputs and ground-truth answers using contextual language-model embeddings. It is better suited than surface metrics when correct EEG explanations use different wording from the reference.

F.7 LLM-as-Judge

The LLM judge assesses relevance and correctness against deterministic reference answers, and feature-state faithfulness: citing supplied feature names, values, and states without unsupported claims. Aggregate scores appear in Table 5.

G Masked-Modeling Supplementary Results

This appendix reports masked-modeling transfer, CBraMod ablations, and descriptor-prediction diagnostics. The downstream suite contains SEED, BCIC2A, MDD, AD65, SHU, HMC, SleepEDF, SEEDV, TUEP, ISRUC, TUAB, and TUEV. The ablation overview is followed by descriptor-family, injection-strategy, and loss-balance results; the final subsection examines descriptor predictions and frozen representations.

G.1 Selected-12 Joint-Versus-Reconstruction Results

Table 16. **Full-finetuning summary on the selected 12-dataset suite.** Balanced accuracy (%) is averaged over valid backbone–dataset comparisons. +NKE denotes Neuro-KE joint pretraining, and Δ reports +NKE minus reconstruction-only pretraining in percentage points.

Scope	Comp.	+NKE wins	Recon. wins	Ties	+NKE	Recon.	Δ
Overall	36	34	2	0	65.35	62.75	+2.60
CBraMod	12	12	0	0	64.79	63.17	+1.62
CSBrain	12	11	1	0	64.99	61.97	+3.03
EEGMamba	12	11	1	0	66.27	63.10	+3.17

Figure 3 provides the compact view behind the main-text transfer claim. It reports gains for Neuro-KE joint pretraining relative to reconstruction-only pretraining across the selected-12 suite. The latest selected-suite summaries show that joint pretraining improves full-finetuning performance in 34 of 36 backbone–dataset comparisons, with a mean balanced-accuracy gain of 2.60 percentage points.

G.2 Overview of the CBraMod Ablations

Figure 7 summarizes the three ablation factors.All CBraMod ablation experiments in Sections G.2–G.5 are evaluated using linear probing (LP), with the pretrained encoder frozen and only a linear classifier trained; Section G.1 separately reports the selected-12 transfer results under full fine-tuning.Panel (a) compares the selected full-joint setting with the strongest alternative within each factor, while panels (b)–(d) show injection strategies, loss weights, and descriptor-family gaps. The following subsections explain these comparisons individually.

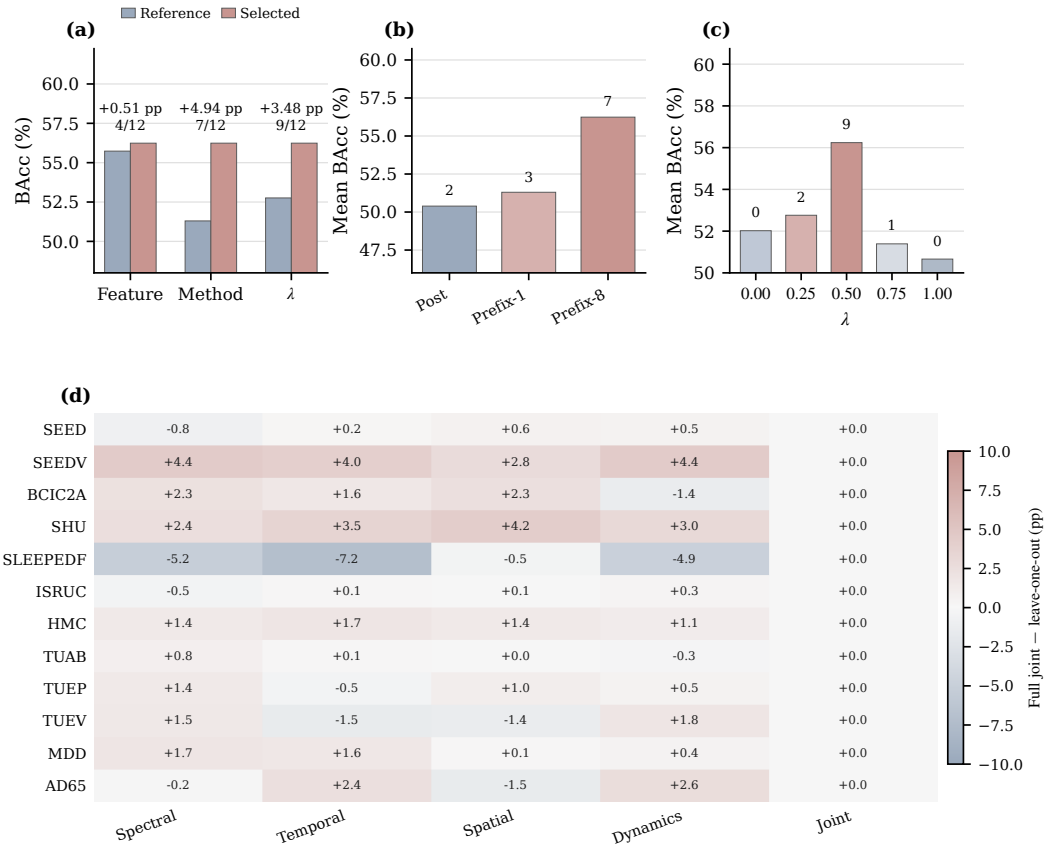

Figure 7. **Overview of CBraMod ablations on the selected 12 datasets.** (a) Macro BAcc of the selected setting and strongest alternative for each factor; annotations give the gain in percentage points and dataset win count. (b) Injection strategies: parallel head (Post), one-token prefix (Prefix-1), and eight-token prefix (Prefix-8). (c) Reconstruction weight λ in Eq. (2). Numbers above bars in (b,c) count dataset-wise wins. (d) Full joint minus leave-one-family-out BAcc; Spectral, Temporal, Spatial, and Dynamics name the omitted family. The Joint column is the zero reference.

G.3 Descriptor-Family Composition

We use CBraMod as a controlled backbone to test which parts of the descriptor catalog are most useful. The descriptor-family ablation removes spectral, temporal, spatial, or dynamics targets from the full joint descriptor set. The full joint target gives the best macro BAcc (0.5624) and the most dataset wins (4/12). However, the weighted BAcc is highest when the temporal group is removed (0.6032), mainly because some large-sample datasets prefer that leave-one-group-out setting. This is why we describe the descriptor set as a robust default rather than a uniformly monotonic improvement over every task and feature family. The bars in Figure 8 and the gaps in Figure 9 show the same comparisons at the dataset level.

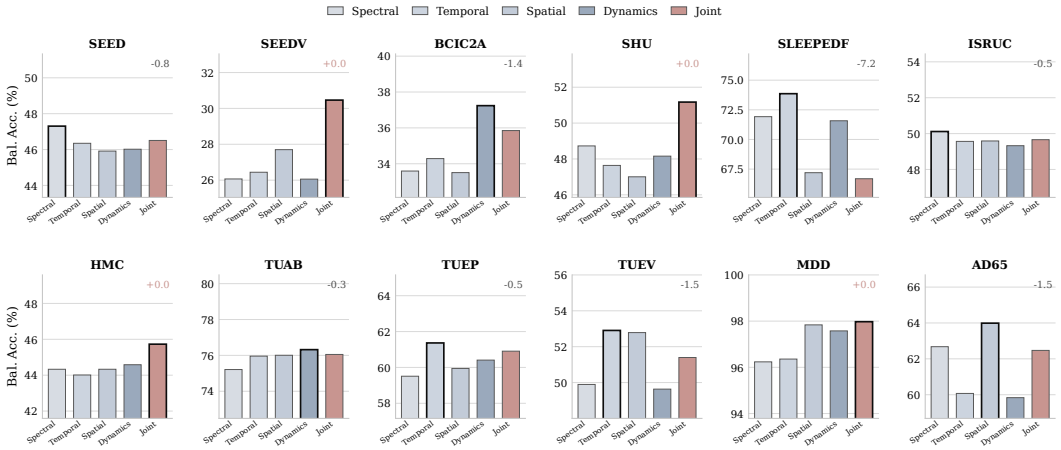

Figure 8. **Descriptor-family ablations by dataset.** Joint uses all four families; the other bar labels name the family omitted from supervision. Annotations give Joint minus the best leave-one-family-out BAcc, in percentage points. A negative annotation indicates that an omission performs better on that dataset.



Figure 9. **Full-joint versus leave-one-family-out gaps.** Rows are datasets and columns name the omitted family; Joint is the zero reference. Each cell is full-joint minus the corresponding alternative BAcc, in percentage points. Positive values favor the full descriptor set.

G.4 Injection Strategy

The injection-method comparison asks how descriptor supervision should enter masked modeling. We compare a post/parallel-head prediction design, a one-token prefix, and an eight-token feature-group prefix. The eight-token feature-group prefix obtains the strongest overall result on the selected-12 suite, with mean Test BAcc 56.24%, weighted Test BAcc 59.91%, and 7/12 dataset wins. The one-token prefix reaches 51.30% mean Test BAcc and wins 3/12 datasets, while the parallel-head design reaches 50.39% and wins 2/12 datasets. This supports the main text's choice of a structured prefix-style interface, while also showing that one-token and parallel-head designs retain local advantages on specific datasets. Figure 7(b) displays these three strategies and their dataset win counts.

G.5 Reconstruction–Feature Loss Balance

The loss-balance sweep varies λ in $\mathcal{L}_{MM} = \lambda\mathcal{L}_{recon} + (1-\lambda)\mathcal{L}_{feat}$. The balanced setting $\lambda = 0.50$ achieves the highest mean BAcc (0.5624), median BAcc (0.5129), weighted BAcc (0.5991), and the most dataset-level wins (9/12). The next strongest setting is $\lambda = 0.25$, which wins ISRUC and TUEV, while the pure endpoints $\lambda = 0.00$ and $\lambda = 1.00$ are weaker as global defaults. Here λ multiplies reconstruction and $1-\lambda$ multiplies descriptor prediction: $\lambda = 0$ is feature-only, and $\lambda = 1$ is reconstruction-only. Figure 10 reports the five tested settings; its panel (a) also appears in Figure 7(c).

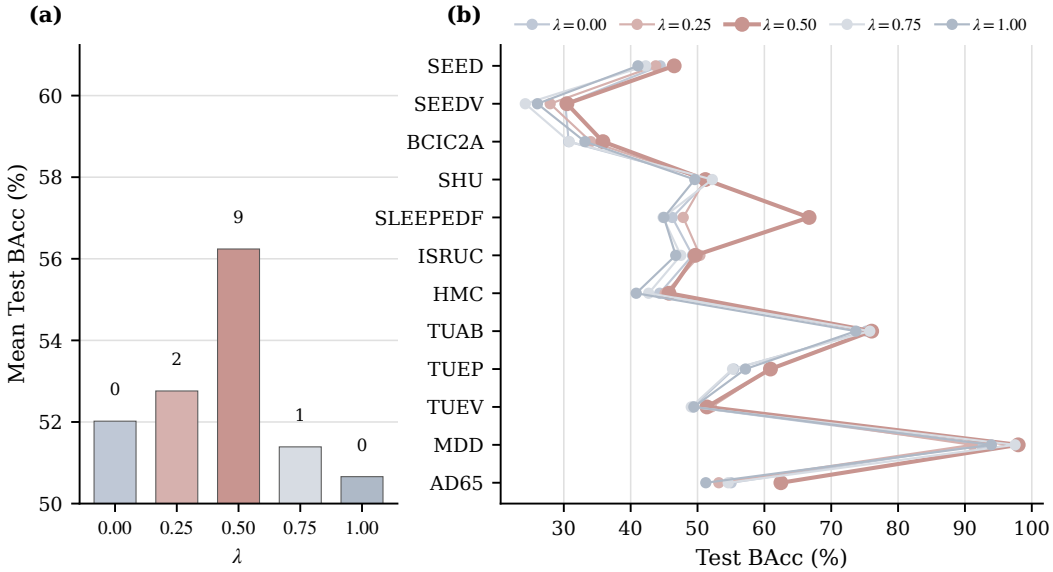

Figure 10. **Reconstruction–feature loss balance for CBraMod.** (a) Macro test BAcc for $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$; numbers count datasets on which each setting has the best score. (b) Per-dataset test BAcc for the same settings, with one curve per λ. The balanced coefficients are $(\lambda, 1-\lambda) = (0.5, 0.5)$. All gains and win counts refer to the selected 12-dataset suite.

G.6 Feature-Prediction and Representation Diagnostics

Figure 11 shows all 87 descriptor-prediction panels, enlarged and continued across four pages. Figure 12 compares R^2 and Pearson correlation for the same features, grouped by descriptor family. These metrics describe recovery of descriptor targets; they are distinct from downstream classification BAcc. The frozen-embedding t-SNE visualization is shown in Figure 13.

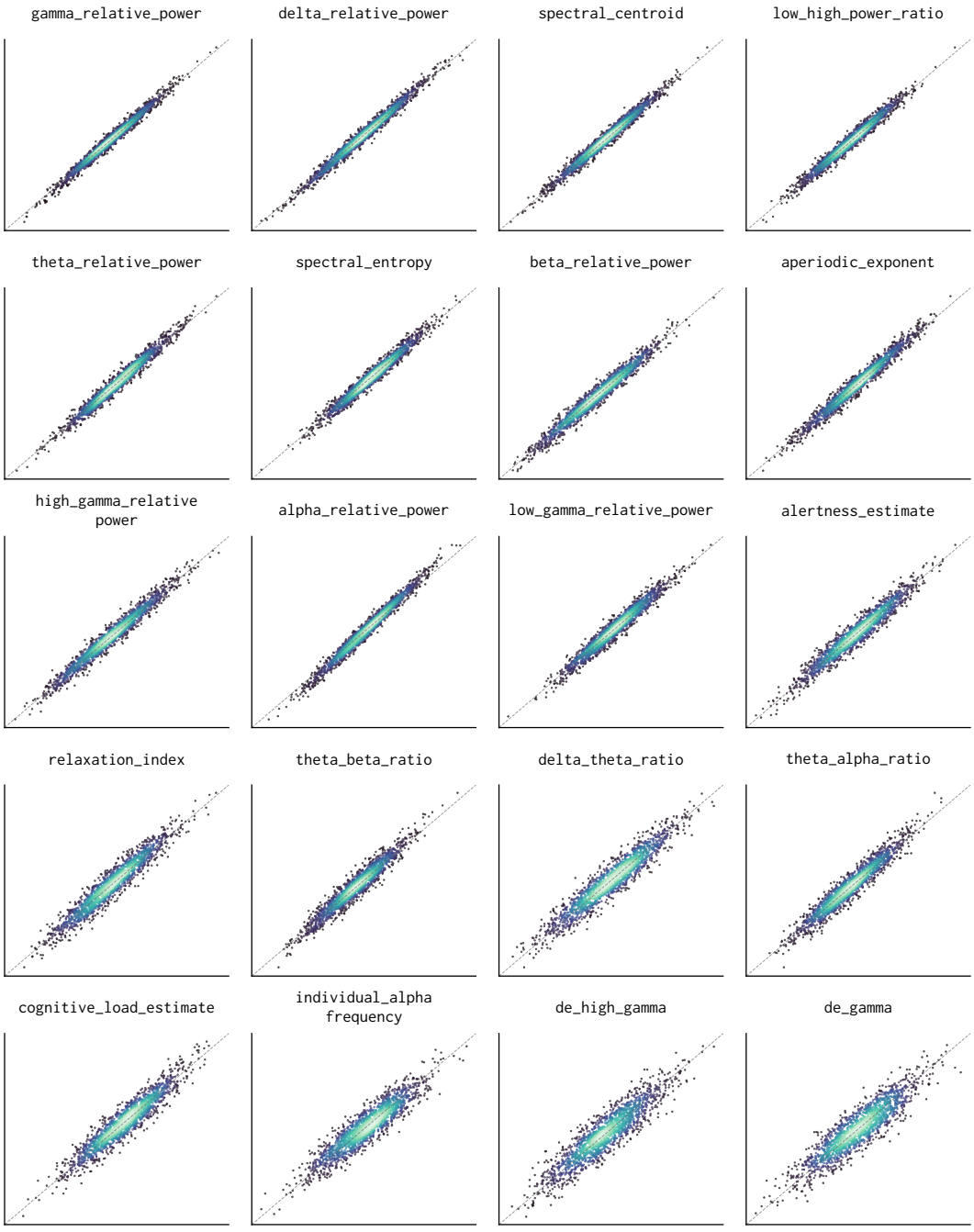

Figure 11. **Descriptor predictions (1/4).** Each panel compares target and predicted values for one descriptor; the dotted diagonal marks equality. Feature names are preserved from the original panels. All 87 features are shown across this figure and its continuations.

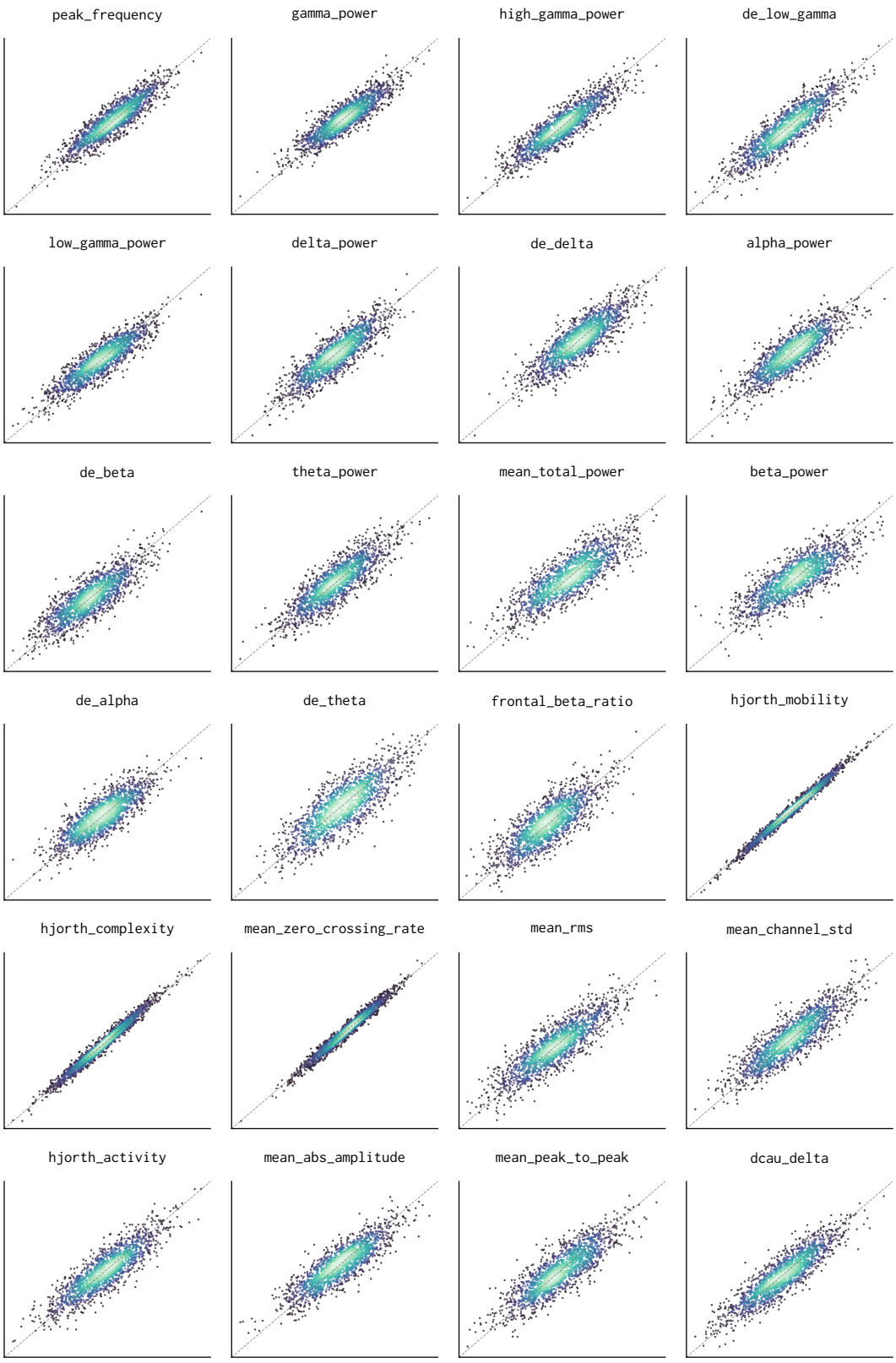

Figure 11. **Descriptor predictions (2/4), continued.** Same comparison as the preceding panels; feature names identify the displayed descriptors.

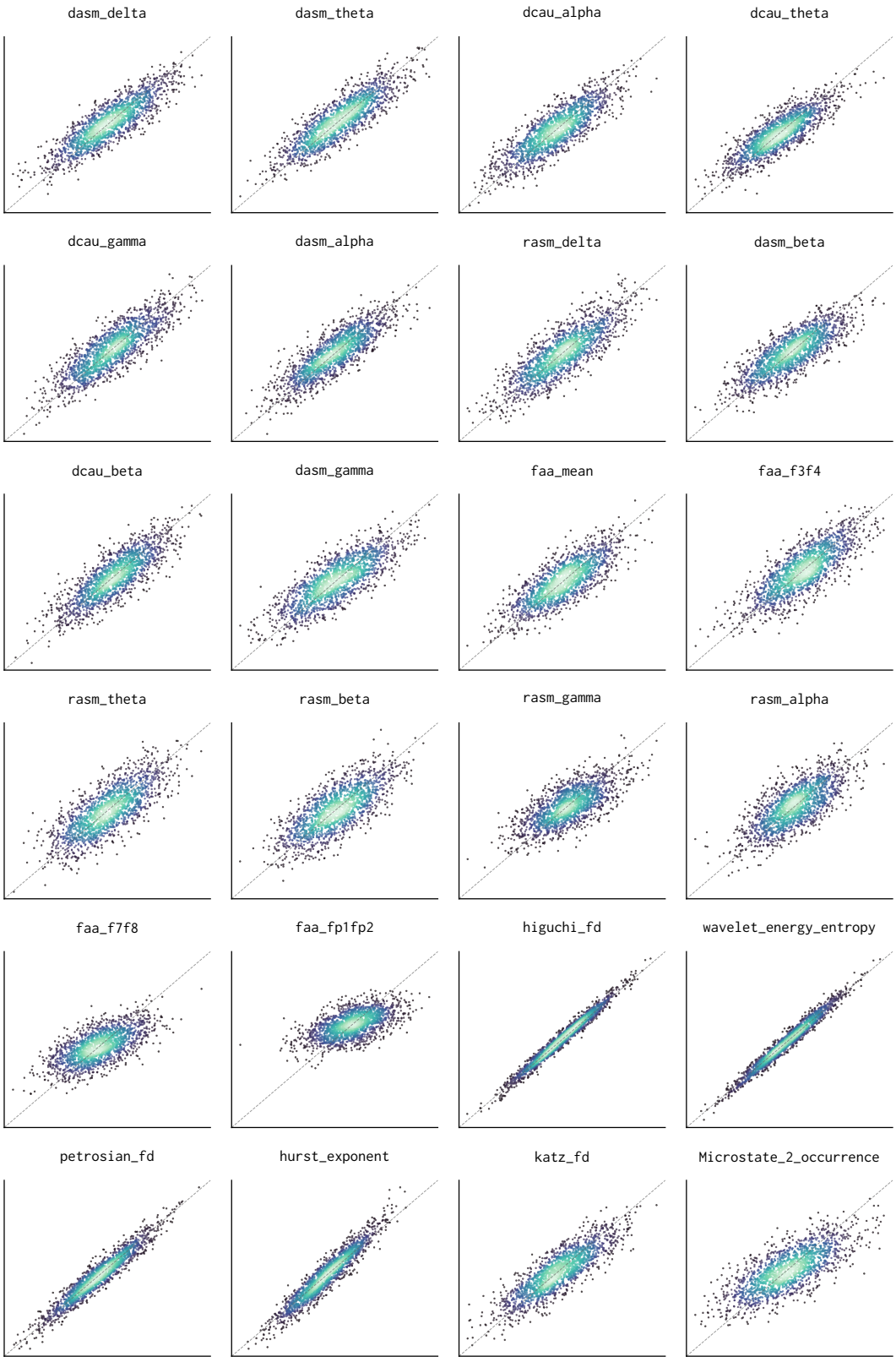

Figure 11. **Descriptor predictions (3/4), continued.** Same comparison as the preceding panels; feature names identify the displayed descriptors.

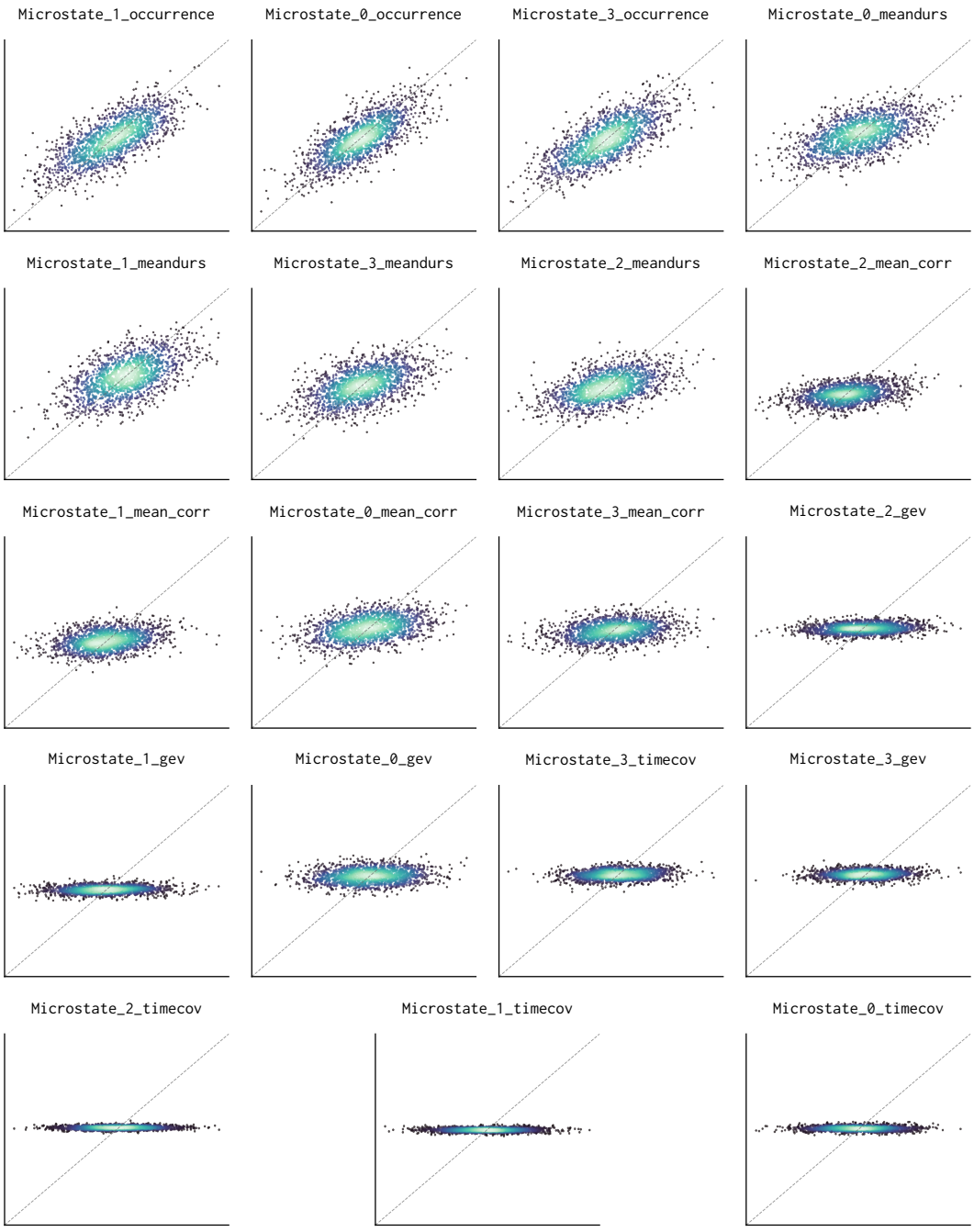

Figure 11. **Descriptor predictions (4/4), continued.** Same comparison as the preceding panels; feature names identify the displayed descriptors.

Descriptor	R^2	PCC
Spectral (35 features)		
gamma_relative_power	0.976	0.989
delta_relative_power	0.975	0.988
spectral_centroid	0.970	0.986
low_high_power_ratio	0.965	0.983
theta_relative_power	0.961	0.983
spectral_entropy	0.960	0.981
beta_relative_power	0.959	0.980
aperiodic_exponent	0.955	0.979
high_gamma_relative_power	0.953	0.978
alpha_relative_power	0.952	0.986
low_gamma_relative_power	0.941	0.973
alertness_estimate	0.935	0.969
relaxation_index	0.908	0.958
theta_beta_ratio	0.906	0.960
delta_theta_ratio	0.906	0.953
theta_alpha_ratio	0.904	0.957
cognitive_load_estimate	0.876	0.941
individual_alpha_frequency	0.836	0.918
de_high_gamma	0.826	0.913
de_gamma	0.810	0.905
peak_frequency	0.808	0.901
gamma_power	0.796	0.892
high_gamma_power	0.795	0.892
de_low_gamma	0.788	0.894
low_gamma_power	0.778	0.882
delta_power	0.714	0.858
de_delta	0.695	0.844
alpha_power	0.685	0.833
de_beta	0.669	0.824
theta_power	0.669	0.835
mean_total_power	0.668	0.822
beta_power	0.668	0.819
de_alpha	0.664	0.824
de_theta	0.630	0.816
frontal_beta_ratio	0.607	0.801
Temporal (8 features)		
hjorth_mobility	0.980	0.991
hjorth_complexity	0.965	0.984
mean_zero_crossing_rate	0.961	0.981
mean_rms	0.732	0.859
mean_channel_std	0.730	0.858
hjorth_activity	0.729	0.857
mean_abs_amplitude	0.721	0.852
mean_peak_to_peak	0.700	0.843

Figure 12. **Per-feature R^2 and Pearson correlation (1/2): spectral and temporal descriptors.** Paired bars compare both metrics for each feature; the pale track spans 0–1. Values retain the original three-decimal precision. Features are ordered by R^2 within each family.

Descriptor	R^2	PCC
Spatial (19 features)		
dcau_delta	0.781	0.888
dasm_delta	0.734	0.857
dasm_theta	0.712	0.846
dcau_alpha	0.710	0.846
dcau_theta	0.702	0.838
dcau_gamma	0.700	0.842
dasm_alpha	0.679	0.830
rasm_delta	0.665	0.826
dasm_beta	0.655	0.810
dcau_beta	0.646	0.813
dasm_gamma	0.632	0.798
faa_mean	0.618	0.790
faa_f3f4	0.615	0.796
rasm_theta	0.526	0.749
rasm_beta	0.514	0.740
rasm_gamma	0.492	0.709
rasm_alpha	0.448	0.705
faa_f7f8	0.444	0.667
faa_fp1fp2	0.328	0.574
Dynamics (25 features)		
higuchi_fd	0.969	0.986
wavelet_energy_entropy	0.961	0.982
petrosian_fd	0.916	0.958
hurst_exponent	0.867	0.949
katz_fd	0.752	0.868
Microstate_2_occurrence	0.578	0.760
Microstate_1_occurrence	0.569	0.754
Microstate_0_occurrence	0.541	0.740
Microstate_3_occurrence	0.536	0.733
Microstate_0_meandurs	0.277	0.530
Microstate_1_meandurs	0.273	0.544
Microstate_3_meandurs	0.246	0.505
Microstate_2_meandurs	0.242	0.493
Microstate_2_mean_corr	0.114	0.338
Microstate_1_mean_corr	0.109	0.332
Microstate_0_mean_corr	0.093	0.320
Microstate_3_mean_corr	0.083	0.290
Microstate_2_gev	0.026	0.162
Microstate_1_gev	0.012	0.111
Microstate_0_gev	0.011	0.148
Microstate_3_timecov	0.010	0.122
Microstate_3_gev	0.005	0.097
Microstate_2_timecov	0.002	0.045
Microstate_1_timecov	-0.003	0.009
Microstate_0_timecov	-0.007	0.000

Figure 12. **Per-feature R^2 and Pearson correlation (2/2), continued: spatial and dynamics descriptors.** Same scales and ordering as the preceding display. Negative R^2 extends to the left of zero.

G.6.1 Frozen-Representation Visualization

Figure 13 compares reconstruction-only, feature-only, and joint-pretrained representations using TUAB normal/abnormal labels. The displayed silhouette and Calinski–Harabasz (CH) scores are embedding diagnostics; the projection is a qualitative visualization, not a downstream classification result.

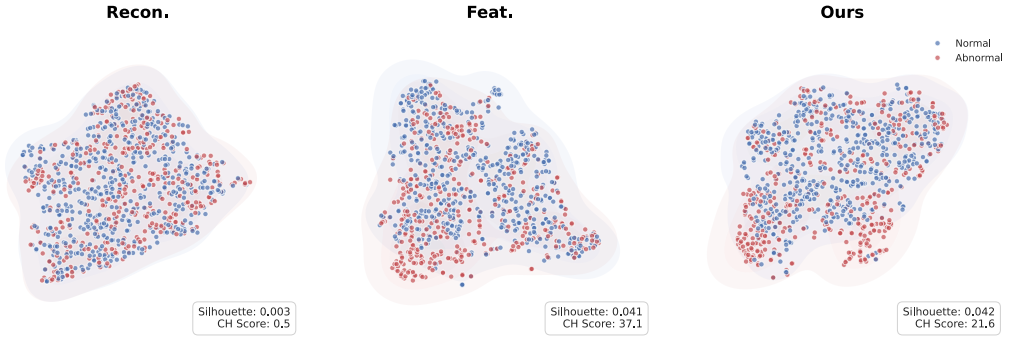

Figure 13. **t-SNE visualization of frozen EEG representations.** From left to right: reconstruction-only (Recon.), feature-only (Feat.), and joint pretraining (Ours). Colors indicate TUAB normal and abnormal labels. Silhouette and CH scores are reproduced from the supplied visualization.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction reflect the paper's contributions, Neuro-KE.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discuss the limitations of Neuro-KE in discussion part.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: No theoretical result in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The method and experimental setup are described in Sections 3–4 and Appendices B–F, but the code and complete run-level configurations and evaluation manifests are not yet available. We plan to provide these materials with the code release on GitHub.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code is not yet publicly available. We plan to release it on GitHub together with training configurations and reproducibility documentation once the repository is prepared.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Model and training hyperparameters are provided, but dataset-specific checkpoint and evaluation-split details for the EEG-MLLM label-prediction comparison are not fully specified. We plan to document these details with the training code on GitHub.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Some results include means and standard deviations, but repeat counts and uncertainty estimates are not fully reported for all experiments supporting the main claims. We plan to share available run configurations, random seeds, and per-run results with the code release on GitHub.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We report model and training hyperparameters, but do not provide a complete account of hardware, memory requirements, or execution times for all experiments. We plan to document available hardware and runtime information with the code release on GitHub.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No ethics issues with this paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: No societal impact of this work.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No such risks with this paper.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper includes source references for existing datasets and models, but does not provide a complete listing of asset licenses and terms of use. We plan to document the relevant licenses and access conditions in the GitHub repository accompanying the code release.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: See Sections 3.4, 4.3, D, and E for EEG–MLLMs and LLM evaluation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.