vNEURO: FOUNDATION FOR EEG VISION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

EEG foundation models are typically pretrained from scratch on large corpora of neural recordings. Yet experienced clinical neurophysiologists interpret EEG largely by visual inspection, recognizing spikes, rhythmic bursts, and time–frequency patterns by eye. This raises the question of whether representations learned from natural visual data can be reused for EEG. We introduce vNeuro, a framework that adapts a frozen, video-pretrained V-JEPA 2.1 encoder to EEG through a lightweight trainable interface. vNeuro converts multichannel time–frequency representations into visual tokens that preserve electrode geometry, and trains this interface on unlabeled EEG through latent forecasting with distribution regularization, while the visual backbone remains fixed. During downstream training, only the adapter and classification head are optimized. With a ViT-B backbone, vNeuro achieves balanced accuracies of 81.40% on TUAB abnormality detection and 68.35% on TUEV event classification (68.73% with ViT-G), competitive with EEG foundation models pretrained on tens of thousands of hours of EEG. Ablation and representation analyses show that both the EEG-to-visual token mapping and the regularized self-supervised adaptation contribute to performance. These results suggest that pretrained visual encoders can serve as effective EEG backbones without pretraining a large EEG-specific model.

1 INTRODUCTION

Foundation models are reshaping electroencephalogram (EEG) analysis (Pradeepkumar et al., 2026). Rather than training a separate network for each task, recent models pretrain a general-purpose encoder on large, heterogeneous corpora and adapt it to downstream tasks ranging from abnormality detection to event classification (Jiang et al., 2024; Wang et al., 2025; El Ouahidi et al., 2025). These models follow what we call a "signal-native" paradigm: the tokenizer and the encoder are learned from EEG alone, so the model must acquire both generic local structure, such as transients, oscillations, and time–frequency textures from the same data. As a result, progress has been driven largely by scaling EEG pretraining, with corpora growing to tens of thousands of hours (Ma et al., 2026). Yet EEG remains limited in volume, noisy, and heterogeneous across devices and montages compared with the image and video collections behind modern visual encoders, and expert annotation is costly. We therefore ask: *must an EEG encoder learn local signal structure from signal perspective?*

Clinical practice suggests that it need not. Doctors interpret EEGs through **visual inspection**, identifying spikes, rhythmic bursts, and broadband transients and tracking their evolution across electrodes and time (Kotoge et al., 2025). For continuous recordings, quantitative time–frequency displays further compress hours of activity into visible patterns of frequency, power, and rhythmicity (Chen et al., 2022). This practice does not imply that EEG is an image; rather, it points to a shared computational structure: local patterns acquire meaning through their multiscale context. Seen this way, the "signal-native" paradigm learns both EEG-specific patterns and how to organize them into contextual representations from EEG alone. This makes even general structural learning dependent on the scale and diversity of EEG corpora. Visual encoders pretrained on large images have already learned such organization, and visual pretraining has been shown to transfer to one-dimensional time series (Chen et al., 2025). This suggests that a visual encoder serves as a structural prior, while only the EEG-specific interface and temporal dynamics are learned from EEGs.

In this paper, we ask: *can a vision model become an EEG modeling foundation?* We call this class of models *EEG vision models*, which learn EEG through visual encoders rather than from EEG alone. We realized this is non-trivial and raises three challenges. **1) What should a visual encoder**

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

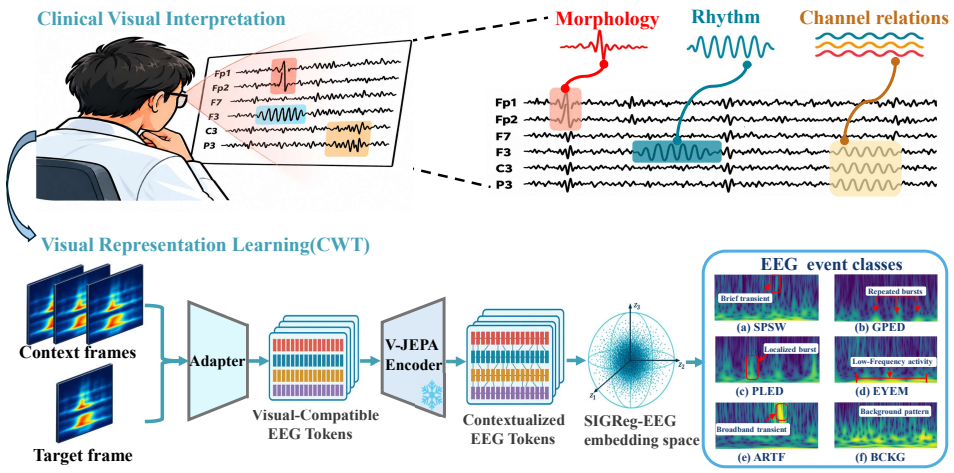


Figure 1: **Clinical motivation and conceptual overview of vNeuro. Figure 1: Clinical motivation and overview of vNeuro. Top:** Clinical EEG interpretation integrates waveform morphology, rhythmic activity, and inter-channel relationships. **Bottom:** vNeuro converts multichannel EEG into continuous wavelet transform (CWT) frames, maps them to visual-compatible tokens through a trainable adapter, and processes them with a frozen, video-pretrained V-JEPA 2.1 encoder. Self-supervised adaptation performs latent forecasting from context to target frames, while SIGReg regularizes the EEG embedding space. The resulting representations support downstream event classification. The right panel shows representative CWT patterns from the six TUEV event classes; highlighted regions illustrate characteristic time–frequency patterns rather than model attribution.

see? EEG has no natural visual form. Its informative structure lies in time-frequency patterns and their organization across electrodes, and an interface must present this structure as the kind of local structure a visual encoder can organize. A poor interface hides this structure, however strong the encoder. **2) What must be learned from EEG?** A visual encoder brings general machinery for organizing local patterns, but no prior knowledge of how EEG activity evolves across windows. This EEG-specific knowledge must be acquired from the recordings themselves and, because labels are scarce, without relying on annotations. **3) How can adaptation remain informative?** When learning is confined to a small EEG-specific part of the model, it can find shortcuts that satisfy the training objective while discarding what distinguishes one recording from another. Adaptation must therefore keep representations diverse and informative, not merely predictable.

In this paper, we therefore propose `vNeuro`, an EEG foundation model that opens a new paradigm of EEG vision models. Rather than pretraining an EEG encoder, we keep a pretrained visual encoder frozen and learn only what is specific to EEG, answering the three questions above through three designs. **1) EEG-to-vision tokenizer.** The adapter transforms multichannel wavelet representations into local time–frequency tokens on the encoder's native grid and injects bipolar electrode geometry through topology-conditioned gating. **2) Latent temporal forecasting.** A position-conditioned predictor trains the interface without labels by forecasting the latent state of a future window from preceding windows, capturing cross-window dynamics without reconstructing raw signals. **3) Representation geometry.** A distribution regularizer encourages the distributions of randomly projected EEG states to match a standard Gaussian reference across samples at each frame position, discouraging representation collapse during latent forecasting. Overall, our contributions are as follows:

- **Formulating New EEG Foundation Modeling.** We first frame this problem as how to repurpose a visual encoder as a structural prior for EEG, so that only the EEG-specific interface and temporal dynamics are learned from recordings. This leads to a new *EEG vision models* paradigm.
- **Proposing novel `vNeuro`.** To our knowledge, `vNeuro` is the first EEG vision model. It keeps a video-pretrained V-JEPA 2.1 encoder frozen and trains only a lightweight EEG-to-vision tokenizer through label-free latent temporal forecasting, with SIGReg preventing representation collapse.
- **Broad Evaluation and Analysis.** We evaluate `vNeuro` against 17 signal-processing. Experiments on TUEV and TUAB, `vNeuro` achieves balanced accuracies of 68.73% on TUEV and 81.40% on TUAB, that improvements of 11.77% and 1.30% over the baselines, respectively.

2 Related Work

EEG Foundation Models. EEG foundation models aim to learn representations that transfer across tasks and recording configurations. Early decoding architectures established useful inductive biases: EEGNet uses compact temporal and spatial convolutions (Lawhern et al., 2018), while EEG-Conformer augments convolutional features with self-attention (Song et al., 2023). More recent models extend representation learning to unlabeled and heterogeneous recordings. BENDR introduces masked contrastive prediction (Kostas et al., 2021); BIOT supports varying channel sets and recording lengths through channel-wise spectral tokens and channel–temporal embeddings (Yang et al., 2023a); okenization-based approaches include TFM-Tokenizer, which learns a discrete vocabulary of single-channel time–frequency motifs (Pradeepkumar et al., 2026) and REVE addresses acquisition variability using electrode–time position encoding and large-scale masked autoencoding (El Ouahidi et al., 2025). Their pretraining objectives range from predicting discrete spectral codes in LaBraM (Jiang et al., 2024) and reconstructing masked waveforms in CBraMod (Wang et al., 2025), to jointly aligning and reconstructing signals in EEGPT (Wang et al., 2024) and predicting contextual latent targets in EEG-JEPA (Li et al., 2026). While these approaches derive their representation backbones from biosignal recordings, we investigate visual pretraining as a complementary source of representations for EEG decoding.

Visual Priors for Time Series and Signals. VisionTS renders time series as images and uses an ImageNet-pretrained masked autoencoder for zero-shot forecasting (Chen et al., 2025), showing that visual pretraining can transfer to temporal signals. For EEG, this raises the question of how to present electrode relationships and time–frequency structure at the encoder's input. JEPA-style models learn visual representations by predicting the latent embeddings of masked image and video regions (Assran et al., 2023; Bardes et al., 2024), and V-JEPA 2.1 extends this to dense spatiotemporal features (Mur-Labadia et al., 2026). EEG-VJEPA applies the V-JEPA framework to EEG, learning the encoder from EEG recordings (Hojjati et al., 2025). vNeuro instead transfers the pretrained visual weights: it keeps V-JEPA 2.1 frozen and learns only an EEG-to-token adapter that encodes time–frequency structure and electrode geometry.

3 Methodology

3.1 Framework

We propose vNeuro, which adapts a frozen, video-pretrained V-JEPA 2.1 (Mur-Labadia et al., 2026) encoder to EEG classification through token adaptation and regularized latent forecasting.

- **EEG-to-Visual Token Adaptation.** The EEG-CWT-Adapter converts multichannel wavelet representations into tokens compatible with the visual Transformer. It combines local time–frequency encoding, topology-conditioned token gating, and temporal mixing, then projects the resulting embeddings to the backbone width.
- **Frozen Visual Encoding and Latent Forecasting.** The frozen encoder processes the adapted tokens to obtain contextualized EEG representations. We encode a four-frame context and its target frame, pool the tokens within each frame, and project them into normalized states. A position-conditioned Transformer predictor uses the context states to forecast the target state.
- **Forecasting and Distribution-Regularized Learning.** We jointly optimize the adapter and auxiliary projection, position-encoding, and prediction modules using future-state mean-squared error and SIGReg (Balestriero & LeCun, 2025). The forecasting term encourages the states to retain information predictive of subsequent EEG activity, while SIGReg constrains the distributions of encoded states across samples at each frame position.

3.2 EEG-to-Visual Token Adaptation

EEG differs from video in both channel semantics and spatial organization: its channels are bipolar derivations, whereas its local structure is expressed over time and frequency. The EEG-CWT-Adapter therefore constructs the visual token interface through three operations. Given five frames of per-lead continuous wavelet transform (CWT) maps, we first separate the four-frame context from the fifth-frame target so that all subsequent temporal operations remain within the supplied clip.

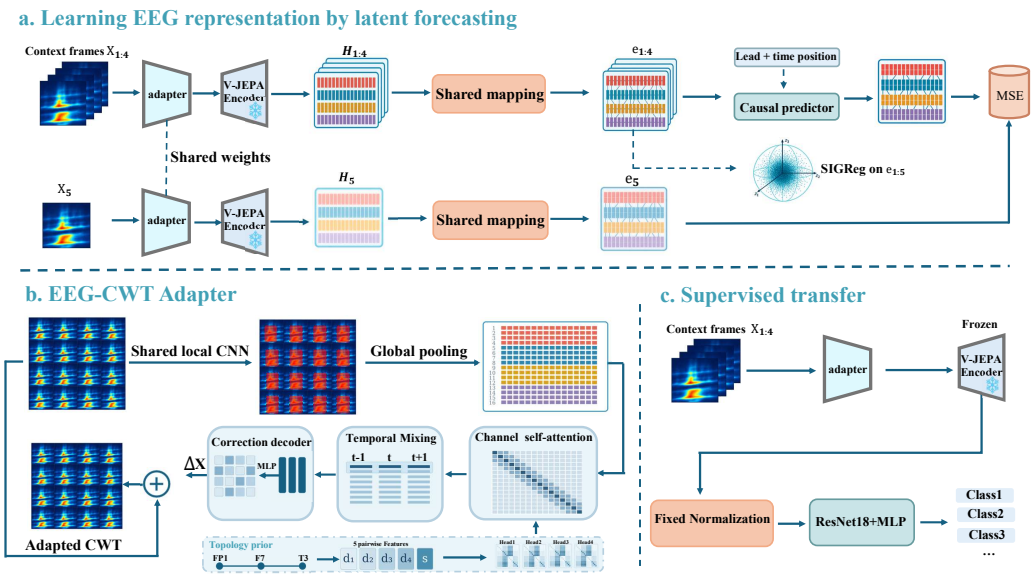


Figure 2: **Overview of vNeuro. a,** Context and target clips are encoded separately through a shared adapter and frozen V-JEPA 2.1 encoder. A position-conditioned predictor forecasts the target state with MSE supervision, while SIGReg regularizes both context and target states. **b,** The EEG-CWT adapter combines local time–frequency encoding, topology-conditioned gating and temporal mixing to produce visual-compatible tokens. **c,** Supervised transfer applies fixed training-set normalization and a ResNet18–MLP classification head to the context token grids.

Local time–frequency tokens. A shared CNN processes each frame with the 16 bipolar leads as input channels. Its initial 1×1 convolution mixes lead values at each time–frequency location, and subsequent convolutions extract local patterns while reducing the map to $L_t \in \mathbb{R}^{64\times128}$ on a 4×16 grid. Grid rows index frequency and columns index within-frame time, so each token represents a local time–frequency region after joint lead processing rather than an individual electrode. The same CNN is applied to every context and target frame, placing all frames in a common token coordinate system before temporal mixing and visual encoding.

Topology-conditioned token gating. Channel mixing removes an explicit lead axis from L_t. A parallel branch retains this structure by pooling each lead map to 4×8 and mapping it to a 64-dimension. For bipolar leads i and j, geometry-aware self-attention adds the head-specific bias:

$$b_{ij}^{(h)} = (w^{(h)})^\top [d_{ij}^{++}, d_{ij}^{+-}, d_{ij}^{-+}, d_{ij}^{--}, s_{ij}]^\top, \tag{1}$$

where the four distances connect their ordered positive and negative electrode endpoints, and s_{ij} indicates whether distinct leads share an electrode. The ordered distances distinguish bipolar derivations that a single lead midpoint cannot, while s_{ij} explicitly marks intersecting derivations. Local tokens then query the updated lead descriptors through cross-attention to generate gate features G_t:

$$\widetilde{L}_t = L_t \odot (1 + 0.1 \tanh G_t). \tag{2}$$

Geometry therefore leads self-attention before cross-attention transfers it to the time–frequency grid. This two-step interaction conditions local signal features without assigning one visual token to each electrode. The final gate projection is initialized to zero, giving an identity mapping at the start of training, while the bounded multiplicative factor restricts feature scaling to $[0.9, 1.1]$.

Temporal mixing and visual interface. At each grid position, a residual depthwise temporal convolution with kernel size three mixes neighboring frames independently within each feature channel. Depthwise mixing preserves the identity of each grid position and separates temporal aggregation from feature mixing. Because context and target are separated before adaptation, this operation cannot cross their boundary. Layer normalization and a linear projection then match the backbone width E. The resulting adapter A_ϕ supplies a $(T', 4, 16)$ token grid, where $T' = 4$ for context and $T' = 1$

for target, preserving the time–frequency layout while exposing a visual temporal token sequence to the frozen backbone. Appendix B.2 gives the layer specifications and complete attention operations.

3.3 Frozen Visual Encoding and Latent Forecasting

The shared adapter and frozen visual Transformer F_Θ encode context and target separately, bypassing native video patch embedding. The four context frames interact jointly within the backbone. For each output grid $H_t \in \mathbb{R}^{64\times E}$, a shared projector $\mathcal{P}_\eta$ maps the mean token feature to $s_t \in \mathbb{R}^d$. Spherical normalization and a position-conditioned predictor then define:

$$e_t = \sqrt{d}\,\frac{s_t}{\|s_t\|_2}, \qquad \widehat{e}_5 = \mathcal{Q}_\rho([\mathcal{T}_\psi(e_{1:4} + \tau_{1:4}; p_{1:4})]_4)\,. \tag{3}$$

Here, we set d to 192 dimensions and τ_t is a learned temporal embedding, and p_t encodes lead midpoints and frame order. The LeWorldModel-based causal predictor (Maes et al., 2026) uses the final context output to forecast the target state; the prediction projector $\mathcal{Q}_\rho$ leaves its output unnormalized. The predictor's causal mask acts on states already informed by all four context frames. Encoding and conditioning details appear in Appendices B.3 and B.4.

3.4 Forecasting and Distribution-Regularized Learning

Self-supervised adaptation. Future-state prediction encourages temporal information retention, but constant states can minimize prediction error even with fixed norms. We therefore combine forecasting with SIGReg (Balestriero & LeCun, 2025):

$$\mathcal{L}_{\text{Stage1}} = \underbrace{\frac{1}{Nd}\sum_{n=1}^{N}\|\widehat{e}_{n,5} - e_{n,5}\|_2^2}_{\mathcal{L}_{\text{pred}}} + 0.09\,\mathcal{L}_{\text{SIG}}. \tag{4}$$

SIGReg compares random one-dimensional projections of $e_{1:5}$ with a Gaussian reference, separately at each frame position across the N samples in a local microbatch. It regularizes encoded states, including the target; the states remain on a fixed-radius sphere. Forecasts receive the prediction loss. Both branches backpropagate through the shared adapter and projector, without a stop-gradient or momentum target encoder. We jointly train the adapter, projectors, position encoder, and predictor while keeping Θ fixed. Appendix B.5 gives the full regularizer.

Supervised classification. Stage 2 retains the adapter and frozen backbone, discarding the auxiliary forecasting modules. The first four frames' full token grids undergo fixed training-set normalization, shared per-frame ResNet18 readout, and concatenation for MLP classification. Class-weighted, label-smoothed cross-entropy updates the adapter and classification head (Appendix B.6).

4 Experiments and Results

Section 1 hypothesized that a frozen visual encoder can supply the machinery for organizing local structure, while EEG-specific learning is confined to an adapter trained via cross-window prediction. In our experiments, we test and study in depth this hypothesis through three research questions. **RQ1:** Can a frozen visual encoder serve as a competitive EEG backbone without large-scale pretraining on EEG?, **RQ2:** How should multichannel EEG be tokenized for a frozen visual encoder, and does this require structure-aware adaptation?, **RQ3:** Does cross-window prediction with distribution regularization suffice to train this interface without labels, and is the regularizer essential? and **RQ4:** Does spatiotemporal position conditioning contributes to learning?

We address the above questions in the main experiments. Appendix A.1 discusses EEG tokenization and electrode geometry, while Appendix B.1 and Appendix B.2 detail EEG preprocessing and the EEG-to-visual adapter, respectively. Additional TUEV studies compare EEG-to-image representations (Appendix C.1), visualize encoder attention (Appendix C.2), and examine backbone choice (Appendix C.3), visual pretraining (Appendix C.4), input resolution (Appendix C.5), channel arrangement (Appendix C.6), and time–frequency normalization (Appendix C.7).

Table 1: Performance on TUAB and TUEV (mean ± standard deviation). Higher is better for all metrics. The best result in each column is bold and highlighted in orange; the best baseline within each category is highlighted in blue.

Category	Method	Learnable params.	TUAB			TUEV		
			BA	AUC-PR	AUROC	BA	Cohen's kappa	Weighted-F1
Task-specific models	EEGNet	0.003M	0.7642 ±0.0036	0.8299 ±0.0043	0.8412 ±0.0031	0.3876 ±0.0143	0.3577 ±0.0155	0.6539 ±0.0120
	EEGConformer	0.55M	0.7758 ±0.0049	0.8427 ±0.0054	0.8445 ±0.0038	0.4074 ±0.0164	0.3967 ±0.0195	0.6983 ±0.0152
	SPaRCNet	0.79M	0.7896 ±0.0018	0.8414 ±0.0018	0.8676 ±0.0012	0.4161 ±0.0262	0.4233 ±0.0181	0.7024 ±0.0104
	ContraWR	1.6M	0.7746 ±0.0041	0.8421 ±0.0104	0.8456 ±0.0074	0.4384 ±0.0349	0.3912 ±0.0237	0.6893 ±0.0136
	CNN-Transformer	3.2M	0.7777 ±0.0022	0.8433 ±0.0039	0.8461 ±0.0013	0.4087 ±0.0161	0.3815 ±0.0134	0.6854 ±0.0293
	FFCL	2.4M	0.7848 ±0.0038	0.8448 ±0.0065	0.8569 ±0.0051	0.3979 ±0.0104	0.3732 ±0.0188	0.6783 ±0.0120
	ST-Transformer	3.5M	0.7966 ±0.0023	0.8521 ±0.0026	0.8707 ±0.0019	0.3984 ±0.0228	0.3765 ±0.0306	0.6823 ±0.0190
EEG foundation models	BIOT	3.2M	0.7959 ±0.0057	0.8792 ±0.0023	0.8815 ±0.0043	0.5281 ±0.0225	0.5273 ±0.0249	0.7492 ±0.0082
	LaBraM	5.8M	0.7735 ±0.0030	0.8531 ±0.0028	0.8557 ±0.0027	0.5550 ±0.0403	0.5175 ±0.0339	0.7450 ±0.0194
	CBraMod	4.0M	0.5000 ±0.0000	0.4938 ±0.0443	0.5281 ±0.0409	0.5696 ±0.0221	0.5588 ±0.0273	0.7702 ±0.0137
	WaveBrain	3.0M	0.7861 ±0.0071	0.8827 ±0.0071	0.8742 ±0.0028	0.5685 ±0.0126	0.5690 ±0.0298	0.7716 ±0.0144
	REVE-Base	12.0M	0.8010 ±0.0050	0.8890 ±0.0040	0.8790 ±0.0040	0.4540 ±0.0160	0.5590 ±0.0380	0.7710 ±0.0180
Ours	**vNeuro-B (ViT-B)**	12.64M	**0.8140** ±0.0023	**0.8961** ±0.0011	**0.8926** ±0.0025	0.6835 ±0.0119	0.6498 ±0.0277	0.8110 ±0.0135
	vNeuro-L (ViT-L)	12.82M	0.8062 ±0.0026	0.8878 ±0.0033	0.8853 ±0.0035	0.6743 ±0.0121	0.6406 ±0.0044	0.8062 ±0.0017
	vNeuro-G (ViT-G)	13.27M	0.8023 ±0.0035	0.8724 ±0.0073	0.8793 ±0.0041	**0.6873** ±0.0116	**0.6995** ±0.0126	**0.8238** ±0.0119

Table 2: Comparison of adapter architectures on TUAB and TUEV (mean ± standard deviation). Higher is better for all metrics. Dashes indicate unavailable values.

Adapter Architecture	TUAB			TUEV		
	BA	AUC-PR	AUROC	BA	Cohen's κ	Weighted F1
3D CNN V-JEPA	0.7880 ± 0.0020	0.8409 ± 0.0037	0.8442 ± 0.0015	0.5880 ± 0.0160	0.7064 ± 0.0114	0.8426 ± 0.0061
3D CNN Adapter	0.7911 ± 0.0025	0.8501 ± 0.0027	0.8736 ± 0.0036	0.5840 ± 0.0181	0.5518 ± 0.0212	0.7743 ± 0.0126
EEG-CWT Adapter	**0.8140** ± 0.0023	**0.8961** ± 0.0011	**0.8926** ± 0.0025	**0.6835** ± 0.0119	**0.7198** ± 0.0277	**0.8510** ± 0.0135

Datasets. We evaluate our method on two datasets from the Temple University EEG Corpus. (1) **TUEV** (Harati et al., 2015) consists of six-class EEG event classification, including spike-and-sharp-wave (SPSW), generalized periodic epileptiform discharges (GPED), periodic lateralized epileptiform discharges (PLED), eye movement (EYEM), artifact (ARTF), and background (BCKG). Each 5-second sample is partitioned into five 1-second frames. (2) **TUAB** (Obeid & Picone, 2016) targets binary abnormal EEG classification. Each 10-second is partitioned into five 2-second frames.

Baselines. We compare against two categories of models. *(1) Task-specific models* that are trained from scratch on the target dataset including EEGNet (Lawhern et al., 2018), EEGConformer (Song et al., 2023), SPaRCNet (Jing et al., 2023), ContraWR (Yang et al., 2023b), CNN-Transformer (Peh et al., 2022), FFCL (Li et al., 2022), and ST-Transformer (Song et al., 2021). *(2) EEG foundation models* that are pretrained on large-scale EEG corpora and subsequently fine-tuned on the downstream task, including BIOT (Yang et al., 2023a), LaBraM (Jiang et al., 2024), CBraMod (Wang et al., 2025), WaveBrain, and REVE-Base (El Ouahidi et al., 2025). REVE was pretrained on more than 60,000 hours of EEG from 25,000 subjects, whereas LaBraM and CBraMod were pretrained on corpora comprising several thousand hours of EEG. Unless otherwise noted, baseline results are taken from the corresponding original publications.

Metrics. We report balanced accuracy (BA), Cohen's κ and weighted F1 for TUEV and balanced accuracy, AUROC, and AUC-PR for binary classification for TUAB. Additionally we report mean ± standard deviation over five different random seeds.

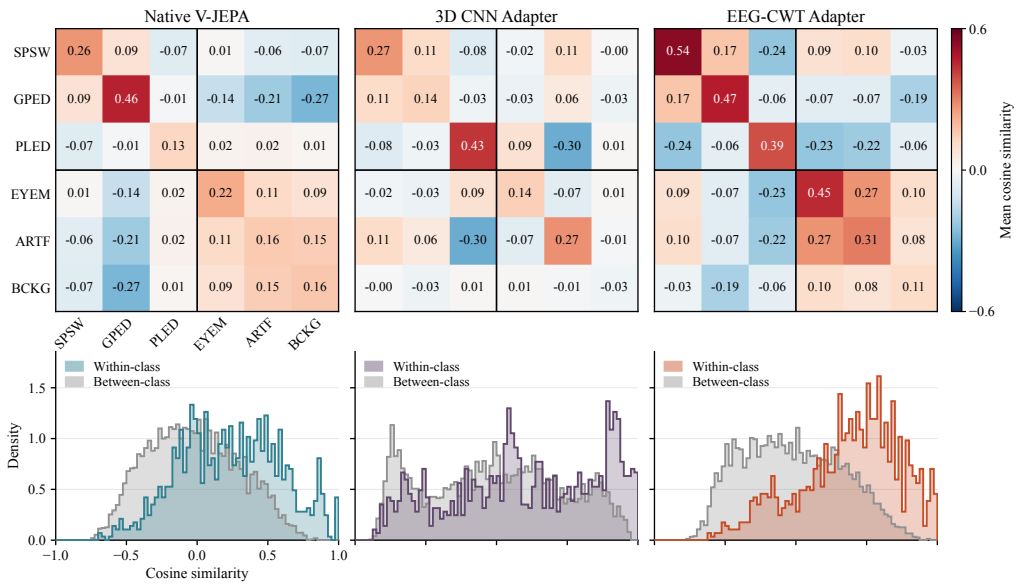


Figure 3: Class-mean cosine similarity (top) and within- vs. between-class pairwise similarity (bottom) of features for three interfaces under identical supervision. EEG-CWT adapter yields compact within-class and well-separated between-class structure.

4.1 How Does a Frozen Visual Encoder Compare to EEG Foundation Models?

RQ1 examines whether a frozen visual encoder can support competitive EEG classification without EEG pretraining of the backbone. Table 1 compares vNeuro with baselines on TUAB and TUEV. The visual backbone remains frozen throughout both stages: Stage 1 trains the EEG adapter and auxiliary forecasting modules, while Stage 2 fine-tunes the adapter and classification head.

On TUAB, vNeuro-B achieves 81.40% balanced accuracy, exceeding REVE-Base (80.10%) by 1.30 percentage points and ST-Transformer (79.66%) by 1.74 points. On TUEV, vNeuro-G reaches 68.73%, outperforming CBraMod (56.96%) by 11.77 points among the reported configurations, while vNeuro-B reaches 68.35%. However, Gigantic improves over Base by only 0.38 points on TUEV, while Base performs best on TUAB. These limited scaling gains may be related to the compact 4×16 token grid (64 tokens per frame), a hypothesis requiring further ablation. Together, these results support a frozen visual encoder as a competitive EEG backbone when combined with EEG-specific adaptation, offering a complementary route to large-scale EEG backbone pretraining.

4.2 How Should Multichannel EEG Be Tokenized for a Visual Encoder?

RQ2 examines how to tokenize multichannel EEG for a frozen visual encoder and whether this demands structure-aware adaptation. We compare three interfaces under the same frozen encoder and supervision (Table 2), and examine the class structure of their representations (Figure 3) to locate where the accuracy differences arise.

Figure 3 shows pairwise cosine similarities between features extracted from the same class-balanced set of samples. Native V-JEPA features exhibited substantial overlap between the two distributions, and the 3D CNN adapter provided only partial separation. In contrast, the EEG-CWT adapter shifted within-class similarities toward larger values while keeping between-class similarities lower. It notably increased the within-class similarity of SPSW from (0.26) to (0.54), PLED from (0.13) to (0.39), and EYEM from (0.22) to (0.45). It also reduced the similarity between easily confused SPSW and PLED events from (-0.07) to (-0.24), yielding more class-discriminative representations.

What the encoder attends to. Given adapted tokens, attention over the CWT map (Figure 4) follows the events present in the signal rather than being uniform. The frozen encoder raises attention at slow deflections, after-going slow waves, and rhythmic activity in the frequency rows carrying their

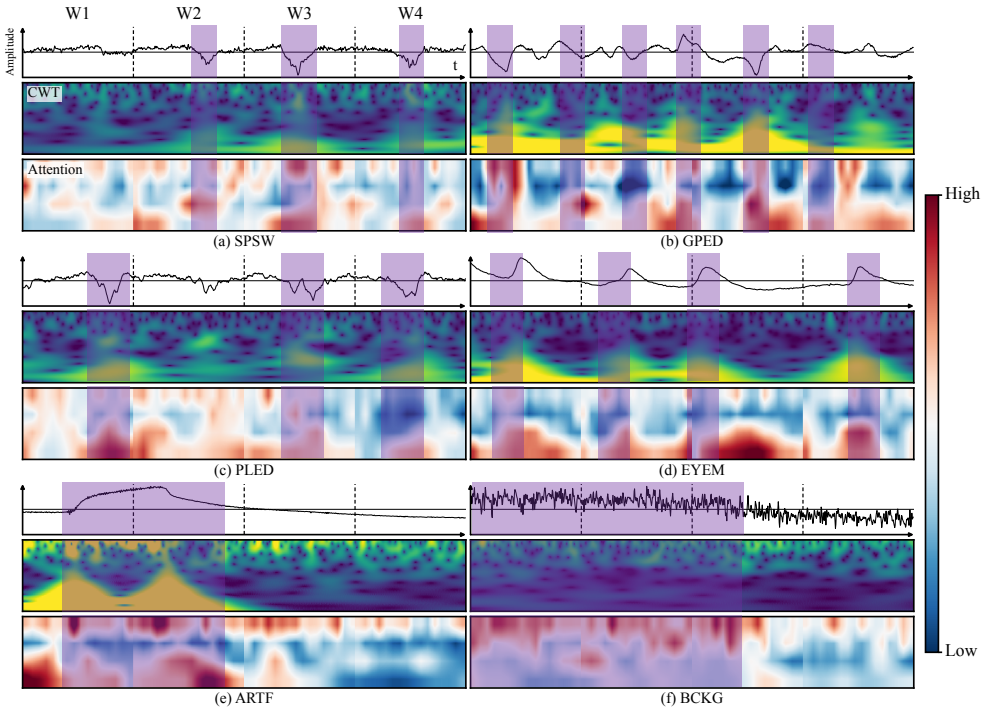


Figure 4: Attention maps of the frozen encoder across six TUEV event classes. From top to bottom, each panel shows the waveform of the displayed lead, its CWT map, and the attention assigned by the encoder relative to a uniform distribution. Red and blue indicate above- and below-uniform attention, respectively, revealing class-dependent emphasis on transient discharges, rhythmic bursts, and sustained low-frequency activity. Shaded regions highlight salient waveform events for visual alignment with the attention maps.

energy, without any EEG-specific pretraining of its own, while the sharp component of discharges receives below-uniform attention.

In summary, our answer to RQ2 is that EEG should be tokenized with a representation that exposes local time–frequency structure, together with a front-end that preserves this structure on the encoder's token grid. Such structure-aware adaptation is necessary but not sufficient on its own, since performance also depends on how the interface is trained, which we examine in Sec. 4.3.

4.3 Is Distribution Regularization Necessary?

RQ3 asks whether cross-window forecasting with distribution regularization suffices to train the interface without labels, and whether the regularizer is necessary. As hypothesized in Section 3, naive forecasting without regularization can admit a degenerate solution where representations collapse into trivial states. We validate this hypothesis in a strictly controlled setting: holding the backbone, adapter architecture, and readout fixed while varying only the Stage 1 pre-training objective.

Is the regularizer necessary? The pretrained V-JEPA encoder yields well-spread representations (Figure 5, left), but this property is not preserved when the EEG interface is trained with the latent prediction objective alone. Since the interface is trainable, it can reduce the prediction loss by mapping inputs into a narrow region where targets become trivially predictable: the representations shrink into a near-degenerate cluster (middle), downstream training fluctuates across epochs (right), and TUEV BA drops to 0.6585. Adding SIGReg keeps the representations spread out, stabilizes convergence, and improves BA on both TUEV (0.6585→0.6835) and TUAB (0.8069→0.8140), with the larger gain on the fine-grained six-class TUEV task (Table 3).

Table 3: Effect of distribution regularization on downstream EEG classification. All variants retain the same latent prediction objective and differ only in the distribution regularizer. Results are reported as mean ± standard deviation.

Regularization	TUAB			TUEV		
	BA	AUC-PR	AUROC	BA	Cohen's κ	Weighted-F1
Prediction Only	0.8069 ± 0.0031	0.8803 ± 0.0039	0.8882 ± 0.0017	0.6585 ± 0.0148	0.6431 ± 0.0394	0.8066 ± 0.0242
Prediction + SIGReg	**0.8140** ± 0.0023	**0.8961** ± 0.0011	**0.8926** ± 0.0025	**0.6835** ± 0.0119	**0.6498** ± 0.0277	**0.8110** ± 0.0135

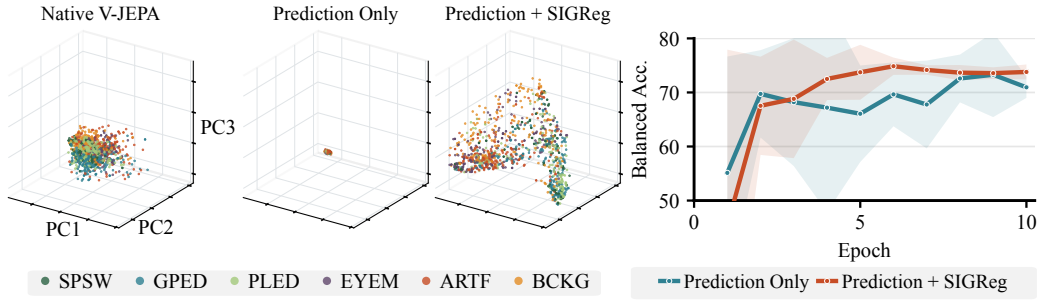


Figure 5: PCA visualization of the encoder representations under different distribution regularization strategies. Colors indicate the ground-truth EEG event labels.

Table 4: Ablation study on spatiotemporal conditioning. Results are reported as mean ± standard deviation. Higher is better for all metrics.

Architecture	TUAB			TUEV		
	BA	AUC-PR	AUROC	BA	Cohen's κ	Weighted-F1
w/o Condition	0.8034 ± 0.0024	0.8771 ± 0.0069	0.8787 ± 0.0015	0.6725 ± 0.0183	0.6483 ± 0.0199	0.8099 ± 0.0100
w/ 4D Condition	**0.8140** ± 0.0023	**0.8961** ± 0.0011	**0.8926** ± 0.0025	**0.6835** ± 0.0119	**0.6498** ± 0.0277	**0.8110** ± 0.0135

4.3.1 Does Position Conditioning Contribute to Learning?

Conditioning the predictor on the 4D lead–window coordinates of Sec. 3 adds a modest, consistent gain (0.8034 → 0.8140 and 0.6725 → 0.6835, Table 4); its design target, varying electrode configurations, is not exercised by the single-montage benchmarks used here. In summary, the results revealed a clear division of labor within the self-supervised objective. Latent forecasting encouraged the interface to preserve information about future EEG dynamics, while distribution regularization prevented this predictive task from being satisfied through a collapsed representation. Their combination therefore provided a stable label-free mechanism for adapting visual representations to EEG.

5 Conclusion

We presented vNeuro, an EEG vision model that transferred representations learned from natural video to EEG without pretraining a large EEG-specific backbone. `vNeuro` transformed multichannel time–frequency representations into topology-aware visual tokens through a lightweight interface and adapted them on unlabeled EEG using latent forecasting and distribution regularization, while keeping the visual encoder fixed. We evaluated 17 input formulations spanning raw signals, global spectra, windowed and continuous time–frequency transforms, wavelets, phase-based methods, and time-series imaging. CWT produced the strongest representations On TUAB and TUEV, `vNeuro` achieved balanced accuracies of (81.40%) and (68.35%), respectively, reaching (68.73%) on TUEV with the ViT-G backbone. These results were competitive with EEG foundation models pretrained on substantially larger neural-recording corpora. Ablation and representation analyses further showed that the EEG-to-visual token mapping and regularized self-supervised adaptation both contributed to downstream performance. Together, the findings demonstrated that visual pretraining provided effective structural priors for EEG and established EEG vision models as a complementary direction to signal-native pretraining. Broader evaluation across datasets, acquisition settings, and electrode configurations remained necessary to establish the generality of this transfer.

AI USE STATEMENT

We used OpenAI CHATGPT to check references, LaTeX formatting, and consistency between the mathematical descriptions and the code implementation. Its outputs served as suggestions for verification. The authors remain responsible for the accuracy of the manuscript, references, mathematical statements, experimental claims, and related research materials.

ETHICS STATEMENT

This study uses existing, de-identified clinical EEG recordings from TUAB and TUEV, both derived from the Temple University Hospital EEG Corpus. Access to and use of each dataset remain subject to its respective license and data-use requirements. Our experiments evaluate EEG classification on these benchmarks. Performance on these datasets does not establish clinical utility or generalization to other patient populations, hospitals, or acquisition settings; these require separate evaluation.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the preprocessing, EEG-to-token adapter, frozen visual encoder, position-conditioned predictor, learning objectives, and supervised readout. These descriptions distinguish the four-frame context and one-frame target used in Stage 1 from the first-four-frame input used for classification in Stage 2. Feature-normalization statistics are estimated on the training set and held fixed, and checkpoints are selected by validation balanced accuracy. The documented TUAB multi-seed runs repeat Stage 2 from a shared Stage 1 checkpoint and normalization statistics; they do not represent independent repetitions of the complete two-stage pipeline. Original EEG data must be obtained through the provider's access procedure.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5DSj3MfWrB.

Zheng Chen, Lingwei Zhu, Ziwei Yang, and Renyuan Zhang. Multi-tier platform for cognizing massive electroencephalogram. In *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 2464–2470, 2022.

Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. Reve: A foundation model for eeg - adapting to any setup with large-scale pretraining on 25,000 subjects. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 22541–22577. Curran Associates, Inc., 2025. doi: 10.52202/085713-0760.

A. Harati, M. Golmohammadi, S. Lopez, I. Obeid, and J. Picone. Improved eeg event classification using differential energy. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–4, 2015. doi: 10.1109/SPMB.2015.7405421.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Amir Hojjati, Lu Li, Ibrahim Hameed, Anis Yazidi, Pedro G. Lind, and Rabindra Khadka. From video to EEG: Adapting joint embedding predictive architecture to uncover visual concepts in brain signal analysis. arXiv preprint arXiv:2507.03633, 2025. URL https://arxiv.org/abs/2507.03633v1.

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

Jin Jing, Wendong Ge, Shenda Hong, Marta Bento Fernandes, Zhen Lin, Chaoqi Yang, Sungtae An, Aaron F. Struck, Aline Herlopian, Ioannis Karakis, Jonathan J. Halford, Marcus C. Ng, Emily L. Johnson, Brian L. Appavu, Rani A. Sarkis, Gamaleldin Osman, Peter W. Kaplan, Monica B. Dhakar, Lakshman Arcot Jayagopal, Zubeda Sheikh, Olga Taraschenko, Sarah Schmitt, Hiba A. Haider, Jennifer A. Kim, Christa B. Swisher, Nicolas Gaspard, Mackenzie C. Cervenka, Andres A. Rodriguez Ruiz, Jong Woo Lee, Mohammad Tabaeizadeh, Emily J. Gilmore, Kristy Nordstrom, Ji Yeoun Yoo, Manisha G. Holmes, Susan T. Herman, Jennifer A. Williams, Jay Pathmanathan, Fábio A. Nascimento, Ziwei Fan, Samaneh Nasiri, Mouhsin M. Shafi, Sydney S. Cash, Daniel B. Hoch, Andrew J. Cole, Eric S. Rosenthal, Sahar F. Zafar, Jimeng Sun, and M. Brandon Westover. Development of expert-level classification of seizures and rhythmic and periodic patterns during eeg interpretation. *Neurology*, 100(17):e1750–e1762, 2023. doi: 10.1212/WNL.0000000000207127. URL https://www.neurology.org/doi/abs/10.1212/WNL.0000000000207127.

Demetres Kostas, Stéphane 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*, Volume 15 - 2021, 2021.

Rikuto Kotoge, Zheng Chen, Tasuku Kimura, Yasuko Matsubara, Takufumi Yanagisawa, Haruhiko Kishima, and Yasushi Sakurai. Evobrain: Dynamic multi-channel eeg graph modeling for time-evolving brain network. In *Thirty-Ninth Conference on Neural Information Processing Systems*, 2025.

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.

Hongli Li, Man Ding, Ronghua Zhang, and Chunbo Xiu. Motor imagery eeg classification algorithm based on cnn-lstm feature fusion network. *Biomedical Signal Processing and Control*, 72:103342, 2022. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2021.103342. URL https://www.sciencedirect.com/science/article/pii/S1746809421009393.

Jinhao Li, Zhiyuan Ma, Xueqiao Han, Zhongye Xia, Xinche Zhang, Shanghong Xie, Yixuan Liu, Yongjian Li, Runmin Gan, Tianlin Huo, et al. Eeg-jepa: Structured latent prediction for eeg foundation models. *arXiv preprint arXiv:2608.00114*, 2026.

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.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. arXiv preprint arXiv:2603.14482, 2026. URL https://arxiv.org/abs/2603.14482.

Iyad Obeid and Joseph Picone. The temple university hospital eeg data corpus. *Frontiers in Neuroscience*, Volume 10 - 2016, 2016. ISSN 1662-453X. doi: 10.3389/fnins.2016.00196. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2016.00196.

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

Wei Yan Peh, Yuanyuan Yao, and Justin Dauwels. Transformer convolutional neural networks for automated artifact detection in scalp eeg. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3599–3602, 2022. doi: 10.1109/EMBC48229.2022.9871916.

Jathurshan Pradeepkumar, Xihao Piao, Zheng Chen, and Jimeng Sun. Tokenizing single-channel EEG with time-frequency motif learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yonghao Song, Xueyu Jia, Lie Yang, and Longhan Xie. Transformer-based spatial-temporal feature learning for eeg decoding. *arXiv preprint arXiv:2106.11170*, 2021.

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. doi: 10.1109/TNSRE.2022.3230250.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 39249–39280. Curran Associates, Inc., 2024. doi: 10.52202/079017-1239.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 75310–75346, 2025.

Chaoqi Yang, M Brandon Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=c2LZyTyddi.

Chaoqi Yang, Danica Xiao, M Brandon Westover, and Jimeng Sun. Self-supervised eeg representation learning for automatic sleep staging. *JMIR AI*, 2023b.

APPENDIX

A Extended Related Work **15**

A.1 EEG Tokenization and Electrode Geometry 15

A.2 Predictive Learning and Distribution Regularization 15

B Implementation Details **15**

B.1 EEG preprocessing and frame construction 15

B.2 EEG-to-visual adapter 16

B.3 Frozen visual encoding and frame-level states 17

B.4 Position conditioning and latent forecasting 18

B.5 Self-supervised objectives and SIGReg implementation 19

B.6 Supervised readout and evaluation 20

C Additional Results on TUEV **21**

C.1 EEG-to-image representations 21

C.2 What the frozen encoder attends to 21

C.3 Backbone and input formulation 22

C.4 Role of visual pretraining 22

C.5 Input resolution 23

C.6 Channel arrangement 23

C.7 Normalization of the time–frequency map 23

A Extended Related Work

A.1 EEG Tokenization and Electrode Geometry

EEG tokenization determines what constitutes a local unit of representation. TFM-Tokenizer (Pradeepkumar et al., 2026) learns a discrete vocabulary of time–frequency motifs from individual channels, whereas REVE (El Ouahidi et al., 2025) associates signal patches with electrode–time coordinates. These approaches highlight two distinct forms of structure: locality within a signal and relationships between recording locations. The distinction becomes particularly relevant for bipolar EEG, where each lead is defined by two ordered electrode endpoints rather than a single location. A lead midpoint alone cannot retain this endpoint structure. In vNeuro, local tokens represent time–frequency regions after joint lead processing, while a separate branch introduces bipolar geometry through token gating. This separates the spatial organization expected by the visual backbone from the electrode relationships used to condition its inputs.

A.2 Predictive Learning and Distribution Regularization

LeWorldModel (Maes et al., 2026) combines next-state latent prediction with distribution regularization for learning action-conditioned visual dynamics. Its formulation motivates predictive learning without reconstructing observations. Our setting instead concerns observational EEG sequences: the predictor is conditioned on lead and temporal information, and forecasting provides a learning signal for the interface to a frozen visual backbone.

The regularization mechanism follows LeJEPA (Balestriero & LeCun, 2025), which introduces SIGReg to compare randomly projected embeddings with a Gaussian reference. We adopt this existing mechanism to discourage collapse during EEG adaptation. Freezing the backbone does not by itself eliminate this risk, because the trainable adapter and state projector can still produce uninformative states. Our contribution concerns the integration of these learning objectives with an EEG-specific token interface and transferred visual representations.

B Implementation Details

This appendix expands the main-text computation from the complete data flow to individual modules and optimization rules. Stage 1 transforms five consecutive EEG frames into a four-frame context and a one-frame target, encodes them separately, and learns by forecasting the target state with distribution regularization. Stage 2 retains the learned adapter and frozen visual Transformer, and trains a classifier on the full token grids of the first four frames.

For one sample, X_t denotes the multilead CWT representation of frame t, L_t its local tokens, and H_t its contextualized backbone tokens. The pooled state e_t is used only for the self-supervised objectives. We omit the sample index until defining the losses; all frame indices refer to their order within the original five-frame sample.

B.1 EEG preprocessing and frame construction

Signals and frame duration. Inputs contain $C = 16$ bipolar leads sampled at 256 Hz. TUAB input units follow the loader configuration, with conversion from volts to microvolts when required. Each TUAB whole10 sample spans ten seconds and is divided into five non-overlapping two-second frames. Each TUEV sample spans five seconds and is divided into five one-second frames. The shared montage uses the ordered bipolar chains FP1–F7–T3–T5–O1, FP2–F8–T4–T6–O2, FP1–F3–C3–P3–O1, and FP2–F4–C4–P4–O2, yielding four leads per chain. This order is preserved in both signal processing and lead geometry.

Time–frequency representation. For each lead and frame, we compute a continuous wavelet transform using PyWavelets and the complex Morlet wavelet `cmor1.5-1.0`. The transform uses 64 linearly spaced frequencies from 45 to 1 Hz in descending order. TUAB whole10 uses all 512 samples of each two-second frame before resizing its time axis; TUEV uses the 256 samples of each one-second frame.

Let $W_{t,c}$ be the complex CWT coefficients and $A_{t,c} = \log(1 + |W_{t,c}|)$ the compressed magnitudes. Denote by $q_{1,t,c}$ and $q_{99,t,c}$ the first and ninety-ninth percentiles of $A_{t,c}$, computed separately within each lead and frame. We form

$$B_{t,c} = \mathrm{clip}\left(\frac{A_{t,c} - q_{1,t,c}}{q_{99,t,c} - q_{1,t,c} + \epsilon_{t,c}}, 0, 1\right), \qquad \epsilon_{t,c} = 10^{-6} \max(|q_{1,t,c}|, |q_{99,t,c}|, 10^{-300}). \tag{5}$$

The normalized values are multiplied by 255 and converted to unsigned eight-bit integers. Bilinear resizing, when required, produces a 64×256 map, which is divided by 255 to obtain the adapter input:

$$X \in [0, 1]^{T \times C \times 64 \times 256}, \qquad T = 5, \quad C = 16. \tag{6}$$

The last two axes index frequency and within-frame time. At this point, each lead retains its own map.

Context–target separation. Before any adapter operation, we split X into $X^c = X_{1:4}$ and $X^y = X_5$. Thus the temporal adapter and visual Transformer can exchange information among the four context frames, but cannot access the target frame when producing context features. The target is processed in a separate one-frame call with shared parameters. This isolation applies to the model's frame-level inputs; the upstream signal pipeline includes offline filtering and is not claimed to be strictly causal. Stage 2 uses only $X_{1:4}$: the first eight seconds of TUAB whole10 or the first four seconds of TUEV.

B.2 EEG-TO-VISUAL ADAPTER

The adapter A_ϕ combines a local CNN, a parallel lead-geometry branch, temporal mixing, and projection to backbone width. Its local and topology operations are applied independently to each supplied frame; only the temporal convolution mixes frames within the supplied clip.

Local time–frequency tokens. The shared CNN treats the 16 leads as input channels. Its first 1×1 convolution mixes lead values at each time–frequency location; subsequent layers extract local patterns while reducing the grid resolution. The layers are listed in Table 5. The output is reshaped from $128 \times 4 \times 16$ to

$$L_t = \mathrm{CNN}_\phi(X_t) \in \mathbb{R}^{P \times 128}, \qquad P = 4 \times 16 = 64. \tag{7}$$

Here the four rows index frequency and the 16 columns index within-frame time. Each token summarizes a time–frequency region after processing all leads. The grid therefore contains 64 signal-feature locations; its columns do not represent electrodes. The CNN weights are shared across frames and across the context and target calls.

Table 5: Local CNN configuration. Kernel size, stride, and padding apply to both time–frequency axes. Shapes exclude sample and frame axes.

Layer	Channels	Kernel	Stride	Padding	Activation	Output grid
Conv 1	$16 \to 32$	1	1	0	GELU	64×256
Conv 2	$32 \to 64$	3	2	1	GELU	32×128
Conv 3	$64 \to 128$	3	2	1	GELU	16×64
Conv 4	$128 \to 128$	4	4	0	None	4×16

Lead descriptors and geometry. A parallel branch retains a lead axis to condition the mixed local features on electrode geometry. Each original CWT map is reduced by adaptive average pooling to 4×8, flattened, and mapped to a 64-dimensional descriptor:

$$v_{t,c} = \mathrm{LN}(\mathrm{GELU}(W_s \, \mathrm{vec}(\mathrm{Pool}_{4\times 8}(X_{t,c})) + b_s)) \,. \tag{8}$$

The linear map has input width 32 and output width 64. Stacking the descriptors yields $V_t \in \mathbb{R}^{C \times 64}$.

Each bipolar lead i has ordered positive and negative electrode positions r_i^+ and r_i^-, taken from the endpoint-coordinate table in `electrode_topology.py`. For $\sigma, \tau \in \{+, -\}$, define $d_{ij}^{\sigma\tau} =$

$\|r_i^\sigma - r_j^\tau\|_2$ in meters. Let s_{ij} equal one when distinct leads i and j share a physical electrode and zero otherwise, including $s_{ii} = 0$. These features produce a learned additive attention bias for head h:

$$b_{ij}^{(h)} = (w^{(h)})^\top [d_{ij}^{++}, d_{ij}^{+-}, d_{ij}^{-+}, d_{ij}^{--}, s_{ij}]^\top, \qquad w^{(h)} \in \mathbb{R}^5. \tag{9}$$

Four-head self-attention adds this bias to the scaled query–key logits before softmax. Each head has width $d_h = 16$; writing its query, key, and value projections as $Q^{(h)}, K^{(h)}, V^{(h)}$ gives

$$O_t^{(h)} = \mathrm{softmax}_j \left(\frac{Q_t^{(h)} (K_t^{(h)})^\top}{\sqrt{d_h}} + b^{(h)} \right) V_t^{(h)}. \tag{10}$$

Concatenating the heads and applying the attention output projection forms $\mathrm{MHA}_{\mathrm{lead}}(V_t; b)$. A residual connection and LayerNorm then give

$$\widetilde{V}_t = \mathrm{LN}(V_t + \mathrm{MHA}_{\mathrm{lead}}(V_t; b)). \tag{11}$$

The four endpoint distances retain the ordering of the bipolar pairs, while the shared-electrode feature explicitly identifies intersecting derivations.

Transfer to local tokens. Each local token queries the updated lead descriptors through four-head cross-attention. The query map q consists of LayerNorm at width 128 followed by a $128 \to 64$ linear projection. The lead descriptors supply both keys and values:

$$\begin{aligned} G_t &= \mathrm{Linear}_{64\to128}\Big(\mathrm{MHA}_{\mathrm{cross}}(q(L_t), \widetilde{V}_t, \widetilde{V}_t)\Big), \\ \widetilde{L}_t &= L_t \odot (1 + \alpha \tanh G_t), \qquad \alpha = 0.1. \end{aligned} \tag{12}$$

The geometry bias is used in lead self-attention; cross-attention reads its updated descriptors without a separate topology mask. The final $64 \to 128$ projection has zero-initialized weights and bias, so the gate initially leaves L_t unchanged. Its multiplicative factor is bounded between 0.9 and 1.1, permitting feature-wise adjustments while retaining the original local token. Gating is enabled in the main method and uses only the CWT maps of the same frame.

Temporal mixing and backbone projection. At every grid position, a depthwise one-dimensional convolution processes the frame axis independently within each of the 128 feature channels. It uses 128 groups, kernel size three, and padding one. Residual addition, LayerNorm, and a biased linear projection produce

$$\begin{aligned} U &= \mathrm{LN}\Big(\widetilde{L} + \mathrm{DWConv}_t(\widetilde{L})\Big), \\ Z &= UW_E + b_E \in \mathbb{R}^{T' \times P \times E}, \qquad T' \in \{4, 1\}. \end{aligned} \tag{13}$$

The convolution is noncausal within the supplied clip. Because the four-frame context and one-frame target were already separated, its receptive field cannot cross their boundary. The grid positions are preserved throughout this operation.

Backbone widths are $E = 768$, 1024, and 1664 for the Base, Large, and Gigantic variants. Flattening Z in chronological, frequency-row, and time-column order gives $T'P$ embeddings. The positional grid metadata $(T', 4, 16)$ is passed alongside them to the visual Transformer. All local, geometry-conditioned, temporal, and projection parameters belong to the trainable adapter ϕ.

B.3 Frozen visual encoding and frame-level states

Backbone interface. The shared adapter and frozen V-JEPA 2.1 Transformer (**?**) process the two clips in separate forward calls:

$$\begin{aligned} H^c &= F_\Theta(A_\phi(X^c)) \in \mathbb{R}^{4 \times P \times E}, \\ H^y &= F_\Theta(A_\phi(X^y)) \in \mathbb{R}^{1 \times P \times E}. \end{aligned} \tag{14}$$

Adapted embeddings enter the pretrained Transformer blocks directly, bypassing the native patch and video modality embeddings. The backbone applies its rotary positional mechanism using the

supplied grid and returns contextualized tokens through its final normalization. Its parameters Θ remain fixed.

The context call jointly encodes all four frames, allowing attention across their 256 tokens. The target call contains only the fifth frame's 64 tokens. We write $H_t \in \mathbb{R}^{P\times E}$ for each resulting frame grid. Shared parameters do not imply that these calls exchange input tokens.

State construction. Forecasting uses one compact state per frame. We average its $P = 64$ tokens, apply a shared trainable projector, and normalize the resulting vector:

$$\begin{aligned} h_t &= \frac{1}{P}\sum_{p=1}^{P} H_{t,p}, \qquad s_t = \mathcal{P}_\eta(h_t) \in \mathbb{R}^d, \\ e_t &= \sqrt{d}\,\frac{s_t}{\|s_t\|_2}, \qquad d = 192. \end{aligned} \tag{15}$$

The projector consists of Linear$(E, 2048)$, LayerNorm, GELU, and Linear$(2048, 192)$. The same projector and normalization are used for context and target, placing all backbone variants in the same 192-dimensional prediction space.

For nonzero s_t, the state has norm $\sqrt{d}$ and a learnable direction. If a is uniform on the unit sphere, then $\mathbb{E}_a[(a^\top e_t)^2] = \|e_t\|_2^2/d = 1$. This scaling matches the second-moment scale of the standard Gaussian projection reference used by SIGReg. It does not force the distribution of states across samples to be Gaussian or prevent every sample from sharing the same direction.

The states change as the adapter and projector are updated, despite the fixed backbone parameters. The full grids H_t remain the features used for supervised classification; the pooled states e_t are auxiliary representations for Stage 1.

B.4 Position conditioning and latent forecasting

Lead and frame coordinates. After pooling the token grid, we supply the predictor with an explicit condition describing lead configuration and frame order. Following the four-dimensional position encoding of REVE (El Ouahidi et al., 2025), let $z_{t,c} = [r_c; t-1] \in \mathbb{R}^4$, where r_c is the bipolar lead midpoint in meters. Midpoints are fixed and read from `tuev_positions.json`. They form a separate resource from the endpoint coordinates used by the topology gate.

Fourier and learned branches. Let $\nu_1, \ldots, \nu_{96}$ be the first 96 tuples of $\{0,1,2,3\}^4$, ordered with the time component varying fastest. For the Fourier branch, define $\widetilde{z}_{t,c} = [r_c; 0.1(t-1)] + 0.4\mathbf{1}_4$. The encoding is

$$\begin{aligned} a_{t,c,m} &= \frac{2\pi}{1 + 2\times 0.4}\nu_m^\top \widetilde{z}_{t,c}, \qquad m = 1, \ldots, 96, \\ \mathcal{F}(z_{t,c}) &= [\cos a_{t,c,1}, \ldots, \cos a_{t,c,96}, \sin a_{t,c,1}, \ldots, \sin a_{t,c,96}]. \end{aligned} \tag{16}$$

The learned branch g_ξ applies a bias-free $4 \to 192$ linear map, GELU, and LayerNorm to the original coordinates $z_{t,c}$. We sum the two branches, normalize each lead encoding, and then average across leads:

$$p_t = \frac{1}{C}\sum_{c=1}^{C} \mathrm{LN}(\mathcal{F}(z_{t,c}) + g_\xi(z_{t,c})) \in \mathbb{R}^d. \tag{17}$$

Averaging occurs after the nonlinear coordinate transformation. The temporal coordinate is a frame index; its physical duration is two seconds per step for TUAB whole10 and one second per step for TUEV.

Conditioned Transformer predictor. The predictor follows LeWorldModel (Maes et al., 2026), with action conditioning replaced by p_t. It contains six causal Transformer layers at state width 192. Each layer uses 16 attention heads of width 64, for internal attention width 1024, with learned projections to and from the state space. The feedforward hidden width is 2048 and dropout is 0.1. Learned temporal embeddings τ_t are added to input states, while p_t conditions every block through AdaLN-Zero (Peebles & Xie, 2023).

A prediction projector $\mathcal{Q}_\rho$ maps the final context output to the forecast:

$$\widehat{e}_5 = \mathcal{Q}_\rho\big([\mathcal{T}_\psi(e_{1:4} + \tau_{1:4}; p_{1:4})]_4\big) \in \mathbb{R}^d. \tag{18}$$

Its layers are Linear$(192, 2048)$, LayerNorm, GELU, and Linear$(2048, 192)$. The output is left unnormalized, and only this final-position forecast is supervised against e_5. We do not train intermediate predictor outputs to forecast intermediate frames.

The predictor mask is causal over its input positions, but those states already incorporate the jointly encoded context. The complete forecast therefore uses all four context frames. The three position mechanisms serve different interfaces: electrode-endpoint bias conditions the adapter's lead attention, the rotary grid locates backbone tokens, and p_t conditions the frame-level predictor.

B.5 Self-supervised objectives and SIGReg implementation

Prediction objective and gradient paths. For a device-local microbatch of N recordings, forecasting minimizes the sample- and dimension-averaged error

$$\mathcal{L}_{\text{pred}} = \frac{1}{Nd} \sum_{n=1}^{N} \|\widehat{e}_{n,5} - e_{n,5}\|_2^2 . \tag{19}$$

The fifth-frame state is computed from the same recording as the context. Both the forecast and target branches receive gradients; there is no stop-gradient or momentum target encoder. The shared adapter and state projector consequently learn from both branches.

Prediction alone permits a constant solution: every frame can map to the same state on the sphere and the predictor can output that state, obtaining zero error without preserving recording-specific information. Fixed state norms do not rule out this solution. Distribution regularization penalizes such concentration across samples.

Projected distribution estimates. Sketched Isotropic Gaussian Regularization (SIGReg; Balestriero & LeCun, 2025) compares one-dimensional projections of the encoded states with a standard Gaussian reference. Each forward pass samples $K = 1024$ independent vectors from $\mathcal{N}(0, I_d)$ and normalizes them into unit directions a_k. At frame position t, the empirical characteristic function is

$$\widehat{\varphi}_{t,k}(\omega) = \frac{1}{N} \sum_{n=1}^{N} \exp\big(\mathrm{i}\omega a_k^\top e_{n,t}\big), \qquad \mathrm{i}^2 = -1. \tag{20}$$

The average is over recordings at a fixed frame position, producing five separate distribution estimates. The Gaussian reference has characteristic function $\varphi_0(\omega) = \exp(-\omega^2/2)$. SIGReg evaluates their discrepancy at $J = 17$ quadrature points:

$$\mathcal{L}_{\text{SIG}} = \frac{N}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{j=0}^{J-1} w_j \left|\widehat{\varphi}_{t,k}(\omega_j) - \exp(-\omega_j^2/2)\right|^2, \qquad T = 5. \tag{21}$$

The penalty acts on the encoded states $e_{1:5}$, including the target, not on the forecast $\widehat{e}_5$. A constant encoding becomes a point mass in every projection and is penalized by this comparison. The Gaussian is a one-dimensional reference; the encoded vectors remain on a fixed-radius sphere, and this formulation does not assert an exact multivariate Gaussian state distribution.

Quadrature and statistical scope. The integration points and Gaussian-windowed weights are

$$\begin{aligned} \omega_j &= \frac{3j}{16}, \qquad j = 0, \ldots, 16, \\ w_j &= \frac{3}{16} \exp(-\omega_j^2/2) \begin{cases} 1, & j \in \{0, 16\}, \\ 2, & \text{otherwise.} \end{cases} \end{aligned} \tag{22}$$

These weights implement twice the Gaussian-windowed trapezoidal integral over $[0, 3]$, equivalent to integration over $[-3, 3]$ by symmetry. Equation equation 21 additionally retains the factor N for the local microbatch size.

Features are not gathered across devices or combined across accumulation steps. Distributed gradient averaging leaves each characteristic-function estimate local to its forward microbatch; it does not replace the five estimates by a single distribution of $5N$ states. Directions are resampled at every forward pass, including validation.

Joint optimization. The full Stage 1 objective is

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{pred}} + \lambda_{\text{SIG}}\mathcal{L}_{\text{SIG}}, \qquad \lambda_{\text{SIG}} = 0.09. \tag{23}$$

This coefficient accompanies the batch-size factor and quadrature normalization above. Optimization updates the adapter ϕ, state projector η, learned position branch ξ, predictor ψ including temporal embeddings, and prediction projector ρ. The visual parameters Θ remain fixed, while gradients propagate through their operations to the adapter. Prediction links each context to its subsequent target, and SIGReg constrains how encoded states vary across recordings at each frame position.

B.6 Supervised readout and evaluation

Transfer from Stage 1. Stage 2 initializes the adapter from the final Stage 1 checkpoint and retains the frozen visual Transformer. The auxiliary state projector, position encoder, predictor, and prediction projector are discarded; the backbone's rotary positional mechanism remains part of the encoder. Classification jointly encodes the first four frames and retains all 4×64 output tokens. It therefore reads the original backbone width E, rather than the 192-dimensional forecasting states.

Fixed training-set standardization. Using the Stage 1 encoder, we estimate the population mean μ_a and standard deviation σ_a at each coordinate a of flattened training-set token features. A coordinate identifies a particular frame, grid position, and feature dimension. Normalization is

$$\overline{H}_a = \frac{H_a - \mu_a}{s_a}, \qquad s_a = \begin{cases} \sigma_a, & \sigma_a > 10^{-6}, \\ 1, & \text{otherwise.} \end{cases} \tag{24}$$

The statistics remain fixed while the adapter is trained and are reused for validation and testing. Each normalized frame is restored to its $E \times 4 \times 16$ map for spatial readout.

Frame readout and classification. A shared ResNet18 (He et al., 2016) processes each frame map. Its residual body uses ImageNet-pretrained weights. The input convolution is replaced with a trainable 3×3, stride-one, $E \to 64$ convolution; the initial max-pooling layer and final classifier are removed. The network returns a 512-dimensional feature for each frame. These four vectors are concatenated in temporal order and classified:

$$f_t = R_\gamma(\overline{H}_t) \in \mathbb{R}^{512}, \qquad \widehat{y} = \operatorname{softmax}(C_\zeta([f_1; f_2; f_3; f_4])). \tag{25}$$

The MLP takes 2048 input features, uses hidden width 256, ReLU, dropout 0.2, and M output logits. Here $M = 2$ for TUAB and $M = 6$ for TUEV. The adapter and the complete ResNet18–MLP head are trainable; the visual Transformer remains frozen.

Weighted and smoothed classification loss. Let π_m be the training-set proportion of class m and y_n the integer label of sample n. The class weights and smoothed targets are

$$w_m^{\text{cls}} = \frac{\pi_m^{-1/2}}{\sum_{c=1}^{M} \pi_c^{1/2}}, \qquad q_{n,m} = (1-\varepsilon)\mathbf{1}[m = y_n] + \frac{\varepsilon}{M}, \quad \varepsilon = 0.1. \tag{26}$$

For a device-local minibatch of B examples, mean reduction gives

$$\mathcal{L}_{\text{cls}} = -\frac{\sum_{n=1}^{B}\sum_{m=1}^{M} w_m^{\text{cls}} q_{n,m} \log \widehat{y}_{n,m}}{\sum_{n=1}^{B} w_{y_n}^{\text{cls}}}. \tag{27}$$

The smoothing contribution includes the weights of all classes. Validation and test losses omit class weights but retain label smoothing. The loss updates the adapter and classification head jointly.

EMA and model selection. Evaluation uses exponential moving averages of the adapter and head floating-point states, including batch-normalization running statistics. At optimization update k, the decay is

$$\beta_k = \min\left(0.995, \frac{1+k}{10+k}\right). \tag{28}$$

Fixed feature-normalization statistics, topology features, and integer buffers are copied directly. Validation, testing, and saved inference checkpoints use the EMA states. Checkpoint selection uses validation balanced accuracy, and test evaluation uses the selected checkpoint.

C Additional Results on TUEV

We evaluate six-class classification on TUEV. All metrics are percentages (↑); BA and F1 denote balanced accuracy and macro-F1, respectively, and ± is the standard deviation over seeds. Unlike the main results, which train an adapter end to end (Section 3), every experiment in this appendix is a frozen-feature probe: each channel's EEG-to-image map is encoded by a frozen pretrained backbone, and only a classification head (one hidden layer of 256 units, dropout 0.2) is trained on the channel-averaged embedding. The head is early-stopped on validation BA, and the test split is evaluated once at the selected checkpoint. These numbers are therefore not directly comparable to Table 1. DINOv3 and V-JEPA2.1 use five seeds unless stated otherwise; CLIP uses one. DINOv3 and CLIP use 224-pixel inputs and V-JEPA2.1 uses 384-pixel inputs, unless stated otherwise.

C.1 EEG-to-image representations

CWT yields the highest BA for both backbones, while PAC ranks last (Table 6). The gap is 31.5 percentage points for DINOv3-B and 30.2 for CLIP, highlighting sensitivity to the EEG-to-image representation. The two continuous time–frequency transforms, CWT and the S-transform, rank first and second on DINOv3-B.

Table 6: Representation comparison on TUEV. Bold marks the highest reported value per metric within each backbone.

	DINOv3-B (86M)				**CLIP ViT-B/32 (88M)**			
Representation	Acc.	BA	F1	AUROC	Acc.	BA	F1	AUROC
CWT	85.1	**67.5** ± 2.2	**65.6**	**95.9**	85.7	**65.8**	62.1	94.6
S-transform	84.8	66.1 ± 1.9	62.6	95.0	**87.5**	62.7	62.3	**95.5**
Recurrence	83.3	60.8 ± 0.8	59.7	89.7	84.8	62.0	**63.5**	89.1
STFT	83.1	60.2 ± 2.4	57.9	94.0	83.9	49.2	49.2	87.3
Raw	82.5	59.9 ± 1.6	58.7	92.4	82.2	58.1	58.3	94.1
SWT	**86.3**	58.7 ± 1.2	59.0	93.0	84.4	52.3	53.5	89.6
HHT	83.6	56.4 ± 2.7	55.4	91.5	86.2	51.6	54.1	90.2
Multitaper	81.7	56.3 ± 1.7	56.4	92.7	79.9	53.5	51.5	92.4
DWT	85.2	56.2 ± 1.5	55.6	91.3	84.4	55.6	55.6	90.3
GASF	82.1	55.8 ± 1.8	56.9	88.7	79.5	55.1	54.6	89.4
MTF	82.2	53.7 ± 1.9	53.2	88.5	80.5	56.5	52.8	90.0
GADF	79.2	53.2 ± 1.4	55.0	87.8	82.6	58.0	60.5	89.8
Wavelet packet	80.1	51.4 ± 2.5	51.3	90.8	80.0	41.2	41.0	89.0
Bicoherence	79.9	45.7 ± 1.0	45.0	88.4	80.1	41.4	40.2	86.8
FFT	77.1	44.0 ± 1.8	42.9	87.0	74.8	46.3	41.9	89.4
Welch	70.6	42.8 ± 1.7	40.4	89.2	71.6	39.2	37.6	85.8
PAC	73.1	36.0 ± 0.7	35.6	80.6	75.9	35.6	35.6	80.9

C.2 What the frozen encoder attends to

Figure 6 shows the final-layer CLS attention of frozen DINOv3-B for one spike-and-slow-wave (SPSW) event and one generalized periodic discharge (GPED) event under six representations spanning the ranking of Table 6. In CWT and the S-transform, the SPSW spike appears as a localized broadband transient that draws most of the attention, and the GPED discharges appear as repeated

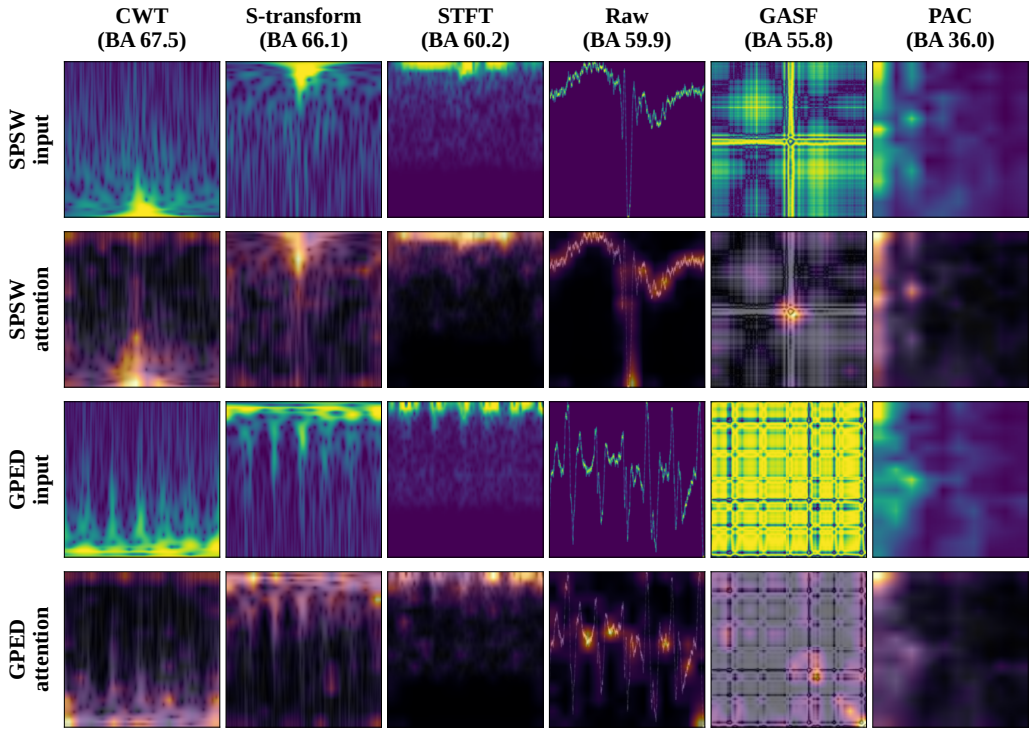


Figure 6: Model input (odd rows) and final-layer CLS attention of frozen DINOv3-B at 224 pixels (even rows) for one SPSW and one GPED event, one channel each. Columns are ordered by DINOv3-B BA from Table 6.

transients. STFT concentrates both energy and attention in one band along the edge of the map, GASF attention collapses onto single crossing points, and PAC attention is diffuse. These maps are qualitative illustrations of one event and one channel per class, not a quantitative analysis.

C.3 Backbone and input formulation

With CWT, DINOv3-B averages independently encoded channel features, whereas V-JEPA2.1 treats the 16 channels as 16 input frames. DINOv3-L reduces BA by 6.4 points relative to DINOv3-B. V-JEPA2.1-B underperforms DINOv3-B by 8.7 points, while V-JEPA2.1-L achieves a similar BA with greater reported uncertainty (Table 7). CLIP results are preliminary due to the single-seed evaluation.

Table 7: CWT results across backbones on TUEV. Bold marks the highest reported value per metric.

Backbone	Params.	Seeds	Acc.	BA	F1	AUROC
DINOv3-B	86M	5	85.1	67.5 ± 2.2	65.6	**95.9**
DINOv3-L	300M	5	84.4	61.1 ± 3.5	59.6	95.0
V-JEPA2.1-B	80M	5	84.6	58.8 ± 1.6	57.7	95.6
V-JEPA2.1-L	300M	5	85.0	$\mathbf{67.6} \pm 4.5$	**66.6**	95.3

C.4 Role of visual pretraining

Freezing a randomly initialized DINOv3-B collapses BA to chance (16.7%, i.e., one in six classes; Table 8). Training the same architecture end to end on TUEV, with all 86M parameters updated, reaches 65.3% BA on one seed, 2.2 points below the frozen pretrained encoder with only a classi-

fication head. The frozen encoder's performance therefore comes from the representations learned during visual pretraining rather than from the ViT architecture alone.

Table 8: Effect of visual pretraining for DINOv3-B with CWT on TUEV.

Encoder	Seeds	Acc.	BA	F1	AUROC
Frozen, pretrained	5	**85.1**	$\mathbf{67.5} \pm 2.2$	**65.6**	**95.9**
Frozen, random initialization	5	66.8	16.7 ± 0.0	13.3	53.7
Trained end to end from scratch	1	83.7	65.3	63.7	91.7

C.5 Input resolution

CWT and the S-transform remain the two best representations at every tested resolution (Table 9). At 64 pixels the S-transform slightly exceeds CWT (68.4 ± 1.8 versus 66.9 ± 3.8), and neither representation gains from 512-pixel inputs, including a finer 256-frequency CWT grid or a log-spaced frequency axis. For DINOv3-L, 64-pixel inputs give 63.7 ± 1.0 BA versus 61.1 ± 3.5 at 224 pixels. Differences across resolutions are within about one seed standard deviation, and the 512-pixel settings use one seed.

Table 9: Input resolution for DINOv3-B on TUEV. The CWT and S-transform use 64 frequencies on a linear axis unless stated otherwise.

		CWT		**S-transform**	
Input	Seeds	BA	AUROC	BA	AUROC
64 px	5	66.9 ± 3.8	95.4	$\mathbf{68.4} \pm 1.8$	**95.9**
224 px	5	$\mathbf{67.5} \pm 2.2$	95.9	66.1 ± 1.9	95.0
512 px	1	64.2	**96.3**	65.3	95.6
512 px, 256 frequencies	1	67.4	**96.3**	66.0	95.6
512 px, log-spaced frequencies	1	64.5	95.9	63.0	94.9

C.6 Channel arrangement

Encoding each channel independently and averaging the embeddings outperforms every single-image mosaic that lets DINOv3-B attend across channels in one forward pass, by 2.9 to 6.6 BA points (Table 10). For V-JEPA2.1-B, treating channels as frames makes BA depend on an arbitrary channel order: reversing the order raises BA by 5.4 points. Packing several channels into each frame, or using consecutive time windows as frames (57.8 for V-JEPA2.1-B and 57.6 for V-JEPA2.1-L, one seed each), does not close this gap. Both observations suggest that a frozen encoder does not by itself exploit cross-channel structure through its spatial or temporal attention, which motivates learning channel fusion in the adapter (Section 3.2).

C.7 Normalization of the time–frequency map

By default, each channel's CWT magnitude is clipped at its own 1st and 99th percentiles before resizing and scaled to $[0, 1]$. Every alternative lowers BA on the same seed (Table 11): clipping after resizing by 2.0 points, sharing one clipping range across the 16 channels of a sample by 4.2, using one dataset-level range estimated on the training split by 5.5, and clipping the raw EEG amplitude before the transform by 9.6. Clipping each channel separately keeps a single high-amplitude artifact channel from compressing the dynamic range of the others. Each variant uses one seed, so gaps below the default's seed standard deviation (2.2) should be read with caution.

Table 10: Channel arrangement with CWT on TUEV. Mosaics tile the 16 channel maps into one image of rows × columns cells (224 pixels each); the padded mosaic leaves 16-pixel gaps between cells. For V-JEPA2.1-B, ch./frame is the number of channels tiled into each input frame.

Backbone	Arrangement	Seeds	Acc.	BA	F1	AUROC
DINOv3-B	Per channel, mean-fused	5	**85.1**	**67.5** $\pm$ 2.2	**65.6**	**95.9**
	Mosaic 4×4	5	83.7	63.1 $\pm$ 1.2	60.2	95.7
	Mosaic 4×4, padded	5	81.7	64.3 $\pm$ 1.8	61.0	95.8
	Mosaic 2×8	5	84.4	61.6 $\pm$ 1.9	59.6	95.7
	Mosaic 8×2	5	83.6	63.6 $\pm$ 0.9	61.1	94.9
	Mosaic 1×16	5	83.5	60.9 $\pm$ 1.3	57.6	94.2
	Mosaic 16×1	5	82.2	64.6 $\pm$ 1.2	60.1	95.2
V-JEPA2.1-B	1 ch./frame (16 frames)	5	84.6	58.8 $\pm$ 1.6	57.7	**95.6**
	1 ch./frame, reversed order	5	**87.1**	**64.2** $\pm$ 2.9	**63.5**	95.1
	2 ch./frame (8 frames)	5	82.4	56.5 $\pm$ 2.4	57.5	95.0
	4 ch./frame (4 frames)	5	82.8	60.9 $\pm$ 3.0	59.2	94.4
	8 ch./frame (2 frames)	5	82.0	60.7 $\pm$ 1.5	58.0	94.4

Table 11: CWT normalization for DINOv3-B on TUEV (seed 0 for all rows).

Clipping range	Acc.	BA	F1	AUROC
Per channel, before resizing (default)	85.9	**68.5**	67.5	96.0
Per channel, after resizing	**87.6**	66.5	**68.0**	**96.1**
Per sample, shared across channels	86.4	64.3	65.2	95.8
Global, from the training split	84.6	63.0	61.0	94.9
Raw EEG amplitude, before the transform	86.1	58.9	58.6	95.8