STIMEM: STIMULUS-CONDITIONED MASKED MODELING FOR EYE-MOVEMENT REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Eye movements provide rich temporal information about human behavior and cognitive function, offering a precise, non-invasive means of assessing cognitive and neurological disorders. However, existing approaches typically rely on handcrafted metrics or task-specific supervised models, whose performance and generalizability are constrained by limited training data and model capacity. Inspired by the success of large language models and foundation models across diverse modalities, we introduce StimEM, an eye-movement foundation model based on Stimulus-Conditioned Eye Movement Masked Modeling. StimEM first learns an ocular tokenizer using finite scalar quantization, jointly optimizing waveform reconstruction and the prediction of informative behavioral features, preserving both local signal structure and trial-level information relevant to cognition. We then pretrain a novel stimulus-conditioned transformer to predict masked tokens. Its structured attention architecture captures dependencies within ocular responses and their relationships with external stimuli. A multi-scale masking strategy further encourages learning of local dynamics and longer-range temporal structure. Pretrained on approximately 500 hours of stimulus-driven eye-tracking recordings spanning multiple cognitive tasks, StimEM achieves state-of-the-art performance on six downstream tasks related to cognitive assessment, demonstrating strong representation learning capabilities and generalizability. Our work demonstrates the value of self-supervised pretraining for eye-movement data and provides a foundation for further research and broader applications in cognitive science and healthcare.

1 INTRODUCTION

Eye-movement data comprise time-resolved measurements of gaze position and associated ocular signals, such as pupil size and blinks, commonly acquired non-invasively using eye-tracking cameras. These recordings capture the fine-grained dynamics of fixation, saccades, and error correction, providing rich behavioral time series closely linked to attention, working memory, and inhibitory control (Hutton, 2008; Klein et al., 2010). In particular, controlled visual stimuli and task instructions elicit responses that probe complementary cognitive functions, making task-evoked eye movements informative for cognitive assessment (Klein et al., 2010; Crawford et al., 2005; Oyama et al., 2019). Previous studies have used these signals to detect mild cognitive impairment (MCI) (Lagun et al., 2011), and to assess cognitive dysfunction in Alzheimer's disease (AD) and Parkinson's disease (PD) (Crawford et al., 2005; Waldthaler et al., 2022). Eye tracking further offers non-invasive, precise, and objective measurements, while automated scoring enables cognitive assessments with minimal participant burden (Hutton, 2008; Oyama et al., 2019). However, realizing this potential depends on how recordings are represented and modeled.

Conventional eye-movement analyses summarize trials using predefined measures such as latency, accuracy, and movement amplitude. Although interpretable, these descriptors compress the original trajectories and depend on expert-designed features and event-detection rules. Supervised models can learn richer representations directly from recordings, but task-specific labeled datasets are often small, limiting the model capacity they can effectively support. Meanwhile, abundant unlabeled recordings and data collected for other tasks remain underused.

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.

Self-supervised learning derives training targets from the recordings themselves, enabling models to learn from unlabeled data and recordings collected for other tasks. This separates general representation learning from specific clinical classification, reducing the reliance on small labeled cohorts. Such self-supervised pretraining strategies have achieved remarkable success in language and vision, while related approaches have enabled effective representation learning from speech and EEG (Devlin et al., 2019; He et al., 2022; Baevski et al., 2020; Jiang et al., 2024). These advances motivate a similar approach to eye movements: pretraining on unlabeled recordings across tasks to learn generalizable representations, followed by fine-tuning for individual clinical applications.

Different clinical tasks depend on different aspects of stimulus-driven eye movements. Multiple-step saccades have been investigated as a marker of PD, while antisaccade and memory-guided saccade measures have shown value in identifying MCI (Ma et al., 2022; Sha et al., 2026). Informative aspects include the timing and shape of individual movements, trial-level behavioral outcomes, and the relationship between initial responses and subsequent corrections (Ma & Zhang, 2023). Interpreting these responses requires the context provided by the stimuli and the task instructions. These observations motivate pretraining objectives that preserve both local dynamics and trial-level behavioral information while capturing stimulus–response relationships across temporal scales.

We introduce **StimEM**, a foundation model for stimulus-conditioned eye-movement masked modeling. We assemble a large-scale eye-movement dataset from healthy participants and patients with cognitive disorders across four stimulus-driven paradigms. Each variable-length trial contains stimulus information, gaze trajectories, pupil size, and blink signals, supporting pretraining without requiring labels. Pretraining proceeds in two stages. First, we train an ocular tokenizer through joint waveform reconstruction and behavioral feature prediction, using finite scalar quantization (FSQ) to produce discrete codes (Mentzer et al., 2024). We then freeze the tokenizer and train a Transformer encoder to predict its codes from masked continuous signal patches using BERT-style prediction. Stimulus-conditioned attention allows ocular representations to use stimulus and response context while keeping stimulus representations independent of ocular responses. Short isolated masks and independent masking of the two eyes can allow prediction from neighboring samples or the visible counterpart. We therefore mask aligned binocular segments over short and long spans to encourage learning across temporal scales. The subject-level fine-tuning aggregates responses across trials and paradigms for clinical prediction. We evaluate StimEM on six clinical classification tasks and examine how the performance scales with model capacity and pretraining data size.

Our contributions are fourfold:

- We assemble approximately 500 hours of stimulus-driven eye-tracking recordings across multiple cognitive tasks and pretrain StimEM on this corpus. To our knowledge, this is the first eye-movement foundation model pretrained at this scale on stimulus-driven recordings from multiple controlled cognitive paradigms.
- We develop an eye-movement tokenizer that jointly learns waveform fidelity and trial-level behavioral structure, incorporating informative behavioral priors into discrete prediction targets tailored to multimodal ocular recordings.
- We introduce a stimulus-conditioned Transformer with directed attention that enables ocular representations to use both stimulus and response information while keeping stimulus representations independent of ocular responses. Combined with paired dual-scale masked prediction, this design promotes learning of eye-movement representations across complementary temporal scales.
- We evaluate StimEM on six clinical downstream tasks, achieving state-of-the-art performance across all six and demonstrating its effectiveness and generalizability. By varying pre-training data size across different model capacities, we further characterize the scaling behavior of eye-movement representation learning.

2 Related Work

2.1 Eye-Movement Analysis

Task-specific eye-movement measures have been used to study cognitive function, including inhibitory control and its impairment (Klein et al., 2010; Crawford et al., 2005). Building on such

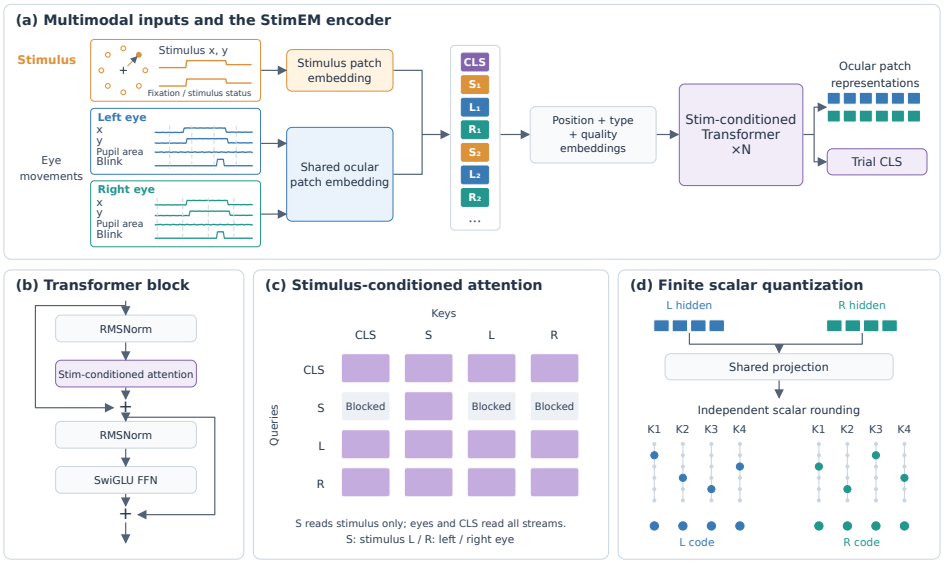


Figure 1: **StimEM architecture.** (a) Aligned stimulus and binocular patches are embedded into an interleaved sequence and processed by a stimulus-conditioned Transformer. (b) A Transformer block. (c) Stimulus queries attend only to stimulus keys; ocular and CLS queries attend to all valid streams. (d) Shared projection and finite scalar quantization produce discrete ocular targets.

behavioral descriptors, Lagun et al. (2011) combined fixation and saccade features with a support vector machine to detect mild cognitive impairment. Other studies have explored learned representations for clinical classification. Zuo et al. (2024) used a multi-layered comparison convolutional neural network to classify Alzheimer's disease from visual-attention heatmaps. Sriram et al. (2023) applied VTNet, combining a CNN for scanpath images with a GRU for eye-tracking time series, to the same clinical task. Uribarri et al. (2026) evaluated InceptionTime and ROCKET on pre-saccadic fixation sequences for Parkinson's disease classification. These studies establish applications of feature-based and learned representations to eye-movement-based clinical assessment. StimEM extends this line of research by learning representations through pretraining across stimulus-driven tasks before adapting them to clinical classification.

2.2 Self-Supervised Learning and Foundation Models

Self-supervised pretraining has been studied across language, vision, speech, and physiological signals. Recent work includes GPT-3 for language modeling and few-shot prediction (Brown et al., 2020), MAE and VideoMAE for visual representation learning (He et al., 2022; Tong et al., 2022), and wav2vec 2.0 for speech representation learning (Baevski et al., 2020). In physiological signal modeling, LaBraM, CBraMod, and NeuroLM have explored pretrained models for EEG analysis (Jiang et al., 2024; Wang et al., 2025; Jiang et al., 2025). Together, these studies provide methodological precedents for learning representations from large collections of signals and adapting them to downstream tasks. Our work applies this pretraining-and-adaptation approach to stimulus-driven eye movements, with objectives and conditioning mechanisms tailored to ocular responses.

3 Methods

StimEM learns discrete ocular targets through tokenizer training, then uses paired dual-scale masked prediction to pretrain a stimulus-conditioned encoder for downstream subject-level classification. Figures 1 and 2 summarize the architecture and training workflow, respectively.

3.1 Model Architecture

Input representation and patch encoding. Each variable-length trial contains synchronized stimulus $S \in \mathbb{R}^{T \times C_S}$ and binocular responses $X^L, X^R \in \mathbb{R}^{T \times C_E}$, where T is the trial length in samples,

and C_S and C_E are the stimulus and ocular channel counts. The ocular channels comprise horizontal and vertical gaze, pupil area, and blink status; the stimulus channels encode fixation status, stimulus status, and horizontal and vertical location. We divide each stream into aligned, non-overlapping patches of P samples, with N temporal positions up to a maximum of $N_{\max}$. Shorter trials are padded. Invalid samples remain flagged and uninterpolated; validity masks distinguish missing observations and padding from measured responses.

As shown in Figure 1(a), convolutional embedders produce d-dimensional vectors e_i^s for stream $s \in \{S, L, R\}$; the eyes share an embedder, while the stimulus uses a separate one. Temporal and type embeddings specify each patch's position and source:

$$h_i^s = e_i^s + t_i + a_s + \eta_i^s. \tag{1}$$

Here, i indexes temporal patches, h_i^s is the input token, and $t_i, a_s \in \mathbb{R}^d$ are learned temporal and type embeddings. During masked pretraining, η_i^s encodes the non-missing sample fraction of an eye patch and is zero for stimulus tokens. A learnable CLS token precedes the interleaved sequence $H^{(0)} = [h_{\mathrm{CLS}}, h_1^S, h_1^L, h_1^R, \ldots, h_N^S, h_N^L, h_N^R]$. We use the CLS state as a fixed-dimensional trial representation that aggregates information across stimulus and ocular tokens (Devlin et al., 2019).

Stim-conditioned Transformer. The stimulus provides an external reference for interpreting ocular responses. We preserve this reference through directed attention, while allowing eye states to integrate stimulus, temporal, and binocular context. The encoder stacks L pre-RMSNorm blocks with stimulus-conditioned attention, SwiGLU feed-forward networks, and residual connections (Figure 1(b)):

$$\begin{aligned} U^{(l)} &= H^{(l-1)} + \mathrm{SCA}(\mathrm{RMSNorm}(H^{(l-1)}); A), \\ H^{(l)} &= U^{(l)} + \mathrm{FFN}(\mathrm{RMSNorm}(U^{(l)})). \end{aligned} \tag{2}$$

Here, $l = 1, \ldots, L$ indexes blocks; $U^{(l)}$ and $H^{(l)}$ are the post-attention and block-output states; SCA denotes stimulus-conditioned attention, and FFN denotes the SwiGLU network. Each head projects normalized states into query, key, and value matrices Q, K, V with head dimension d_h; A is the additive attention mask:

$$\mathrm{Attn}(Q, K, V; A) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + A\right) V. \tag{3}$$

For query and key token indices i, j, let τ_i denote token type and $v_j \in \{0, 1\}$ indicate a valid key. We impose

$$A_{ij} = \begin{cases} 0, & v_j = 1 \text{ and } (\tau_i \neq S \text{ or } \tau_j = S), \\ -\infty, & \text{otherwise.} \end{cases} \tag{4}$$

Stimulus queries therefore read only stimulus keys, whereas eye and CLS queries read every valid stream (Figure 1(c)). Repeating this restriction in each block keeps stimulus states independent of ocular responses, including information passed through CLS. This modification of self-attention (Vaswani et al., 2017) constrains stream interactions without imposing temporal causality. Implementation and attention comparisons are detailed in Appendices B.3, B.4 and C.3.

3.2 Ocular tokenizer training

As illustrated in Figure 2(a), our tokenizer combines the stimulus-conditioned encoder, finite scalar quantization (FSQ) (Mentzer et al., 2024), and an L_D-layer decoder. Waveform reconstruction preserves temporal dynamics, while trial-level behavioral feature prediction supplies complementary task-specific priors. Only eye latents are quantized; the stimulus conditions their encoding.

Discrete ocular codes. Let $p_i^e \in \mathbb{R}^{D_q}$ be the projected latent for patch i of eye $e \in \{L, R\}$, where D_q is the quantization dimension. As shown in Figure 1(d), coordinate $k = 1, \ldots, D_q$ uses ℓ_k odd scalar levels, half-width $a_k = (\ell_k - 1)/2$, and grid $\mathcal{G}_k = \{-a_k, \ldots, a_k\}$:

$$\begin{aligned} u_{i,k}^e &= \tanh(p_{i,k}^e), \qquad r_{i,k}^e = \operatorname*{arg\,min}_{r \in \mathcal{G}_k} |a_k u_{i,k}^e - r|^2, \\ q_{i,k}^e &= r_{i,k}^e / a_k, \qquad z_{i,k}^e = r_{i,k}^e + a_k \in \{0, \ldots, \ell_k - 1\}. \end{aligned} \tag{5}$$

Here, $u_{i,k}^e$ is the bounded latent and $r_{i,k}^e$ its matched grid value. The normalized quantized vector q_i^e feeds the decoder, and the coordinate indices z_i^e form the masked-pretraining target. Rounding implements this scalar matching, with straight-through gradients during training.

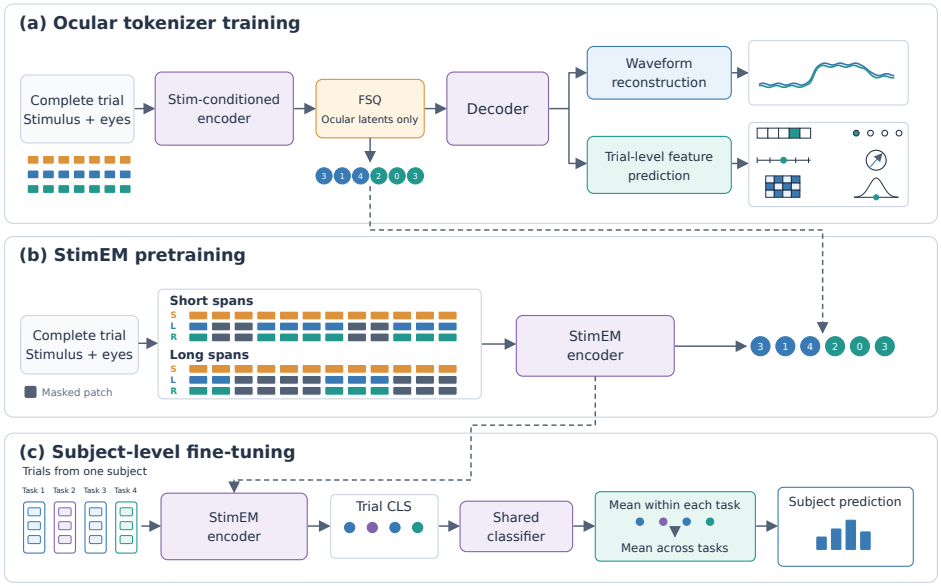


Figure 2: **StimEM training workflow.** (a) The ocular tokenizer learns through waveform reconstruction and trial-level behavioral feature prediction. (b) StimEM encoder predicts frozen-tokenizer targets from short- and long-span masked views, with masks aligned across both eyes. The illustrated code predictions correspond only to masked eye positions. (c) Subject-level fine-tuning uses trial CLS representations and a shared classifier, averaging logits within each task and then across tasks.

Waveform reconstruction. The decoder reconstructs ocular waveforms from its eye states. For a batch $\mathcal{D}$ of trials and ocular channel set $\mathcal{C}$, the reconstruction loss is

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{D}|} \sum_{n \in \mathcal{D}} \sum_{c \in \mathcal{C}} \lambda_c \sum_{e \in \{L,R\}} \sum_{t=1}^{T_n} w^{\text{wave}}_{n,e,t,c} \, \ell_c(\widehat{X}^e_{n,t,c}, X^e_{n,t,c}). \tag{6}$$

Here, n indexes trials, T_n is their length, and t, c index samples and channels; $\widehat{X}$ and X are reconstructed and observed signals.

Trial-level behavioral feature prediction. The decoder's classification state predicts task-specific features derived from each trial, without clinical labels. For available supervised features $\mathcal{F}_n$, the joint objective is

$$\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{rec}} + \frac{\lambda_{\text{feat}}}{|\mathcal{D}|} \sum_{n \in \mathcal{D}} \sum_{j \in \mathcal{F}_n} \alpha_j w^{\text{feat}}_{n,j} \, \ell_j(\widehat{f}_{n,j}, f_{n,j}). \tag{7}$$

Here, $f_{n,j}$ and $\widehat{f}_{n,j}$ are the target and prediction for feature $j \in \mathcal{F}_n$, the set of available feature indices for trial n. The pointwise losses ℓ_c and ℓ_j use smooth-L1 for continuous targets and binary cross-entropy for binary targets. The weights w encode validity and averaging factors; λ_c, λ_{feat}, and α_j balance channels, objectives, and feature types, respectively. Tokenizer details, coefficients and feature definitions appear in Appendices B.4, B.2 and B.5.

3.3 Masked Pretraining

Masked ocular modeling. We freeze the tokenizer to generate discrete targets from unmasked trials, following discrete-target masked modeling (Bao et al., 2022). The StimEM encoder receives continuous patches with selected eye-patch embeddings replaced by a learned mask token. Auxiliary embeddings are retained; stimulus and CLS tokens remain unmasked. The StimEM encoder predicts hidden ocular codes from the visible context using BERT-style prediction (Devlin et al., 2019).

Paired dual-scale masking. Isolated temporal masks may permit local interpolation, while independent eye masking can expose the corresponding response in the other eye. We therefore construct short- and long-span views with aligned binocular masks (Figure 2(b)). For a view v at scale a, masked span lengths satisfy

$$m_{v,b} \in [m_a^-, m_a^+], \qquad a \in \{s, l\}, \qquad m_s^+ < m_l^-. \tag{8}$$

Here, $m_{v,b}$ is the length of masked span b in patches; m_a^- and m_a^+ bound span lengths at the short (s) or long (l) scale. Short spans emphasize local dynamics, and long spans require context across extended intervals. This strategy reduces reliance on local interpolation and cross-eye shortcuts, encouraging representations that capture ocular dynamics across temporal scales.

Training objective. For a masked eye-token state $h_{v,i}^e$ in view v, the k-th categorical head predicts its FSQ coordinate through

$$p_{\theta,k}(c \mid \widetilde{X}_v, S) = \left[\mathrm{softmax}(W_k h_{v,i}^e + b_k)\right]_c, \qquad c \in \{0, \dots, \ell_k - 1\}. \tag{9}$$

Here, $\widetilde{X}_v$ denotes the masked binocular input, θ the trainable encoder and prediction-head parameters, and c a candidate coordinate index. The head parameters are $W_k \in \mathbb{R}^{\ell_k \times d}$ and $b_k \in \mathbb{R}^{\ell_k}$; $[\cdot]_c$ selects probability c. Token indices (i, e) are implicit in $p_{\theta,k}$. Let $\mathcal{V}$ contain batch views with at least one target and $\mathcal{M}_v$ their masked eye-token positions. We minimize

$$\mathcal{L}_{\mathrm{mask}} = -\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{1}{|\mathcal{M}_v|} \sum_{(i,e) \in \mathcal{M}_v} \sum_{k=1}^{D_q} \log p_{\theta,k} \left(z_{i,k}^e \mid \widetilde{X}_v, S \right). \tag{10}$$

Coordinate losses are summed, and valid views receive equal weight regardless of their target counts. Sampling and training details appear in Appendices B.6 and B.

3.4 Subject-level fine-tuning

Clinical prediction combines responses from multiple trials and acquisition tasks (Figure 2(c)). We fine-tune the pretrained encoder and a shared classifier g_ϕ, which maps each trial's CLS representation $h_{u,t,r}^{\mathrm{CLS}}$ to logits for subject u, task t, and trial r. We aggregate these logits hierarchically:

$$o_u = \frac{1}{J} \sum_{t=1}^{J} \frac{1}{|\mathcal{B}_{u,t}|} \sum_{r \in \mathcal{B}_{u,t}} g_\phi\left(h_{u,t,r}^{\mathrm{CLS}}\right), \tag{11}$$

where o_u denotes subject-level logits, ϕ the classifier parameters, J the number of acquisition tasks, and $\mathcal{B}_{u,t}$ the set of trials used for subject u in task t. Averaging within each task and then across tasks gives each task equal weight regardless of its trial count. Sampling, losses, and optimization settings are specified in Appendix B.8.

4 Experiments

We evaluate StimEM through subject-level fine-tuning on six clinical classification tasks and compare it with supervised deep-learning baselines. The main text reports results on MCI and PD, with results for the remaining clinical tasks reported in Appendix D.1. We examine how downstream performance varies with model capacity and pretraining data size. Ablation studies assess the core components, with additional design analyses reported in Appendices D.2–D.7.

4.1 Datasets

Our self-collected corpus contains approximately 500 hours of synchronized stimulus and binocular eye-movement recordings from 4,255 subjects, represented at 1,000 Hz. It covers prosaccade, antisaccade, memory-guided saccade and double-saccade tasks, including recordings without downstream clinical labels. Tokenizer learning and masked pretraining use the same unlabeled corpus.

Downstream fine-tuning evaluates six clinical tasks. MCI, AD, Detox, epilepsy and migraine are binary cohort-versus-control classifications; PD separates controls, parkinson-spectrum and tremor-spectrum groups. For each cohort, we reserve 20% of subjects for testing and use 20% of the

remaining subjects for validation, yielding subject-disjoint training, validation and test partitions in a nominal 64:16:20 ratio. Pretraining and downstream partitions are constructed independently. Split details and eligibility rules appear in Appendix A.3; acquisition tasks and signal-quality procedures are described in Appendices A.1.1 and A.4.

4.2 Experimental setup

Models and training. StimEM-S, M, L and XL contain 7.7M, 14.4M, 24.3M and 56.2M parameters, respectively. All four models are pretrained using the same 12-layer tokenizer. Each patch spans 40 ms, with at most 128 temporal patches per trial. We train our models on four NVIDIA A100 GPUs. Architecture details, method hyperparameters and optimization settings appear in Appendices B.1, B.2 and B.3. StimEM-L is the baseline model for the subsequent ablation studies.

Preprocessing. Quality-aware filtering attenuates high-frequency tracking noise and spurious gaze peaks within stable valid segments. Guard regions preserve detected saccades and neighborhoods of blinks or missing samples, protecting rapid eye movements and recording boundaries from smoothing. Invalid samples remain flagged and uninterpolated. Filter settings and normalization are specified in Appendices A.4 and B.3.

Baselines. We compare against TCN (Bai et al., 2018), TimesNet (Wu et al., 2023), Non-stationary Transformers (NST) (Liu et al., 2022), CNN-Transformer, InceptionTime (Ismail Fawaz et al., 2020) and ConvTran (Foumani et al., 2024). NST is evaluated here for classification, although its original formulation addresses forecasting. CNN-Transformer denotes a convolution–Transformer hybrid; the Transformer component follows the attention architecture of Vaswani et al. (2017). These supervised baselines use the same downstream subject splits, signal normalization, 40-ms patches and trial eligibility rules. Baseline training and pooling settings are detailed in Appendix B.9.

Evaluation. Binary tasks report subject-level AUROC, AUPRC and balanced accuracy (BA, shown as BalAcc in the tables). PD is a multiclass task with three labels: control, parkinson-spectrum and tremor-spectrum. For PD, we report BA, weighted F1 and Kappa. We report mean test performance across runs with seeds 42, 43 and 44 on the designated test split. All reported performance is measured on the test set; the validation set is used only to select the best checkpoint. Evaluation concerns subject-level classification within the recorded clinical cohorts.

4.3 Performance comparison with baselines

Table 1 compares StimEM with supervised baselines on MCI and PD; Appendix D.1 reports the remaining clinical tasks. StimEM-L outperforms every baseline on all reported metrics across these tasks. The advantage is particularly clear on the more challenging multi-class PD task, where even the smaller StimEM models consistently outperform the baselines. These results suggest that StimEM learns transferable representations from unlabeled stimulus-response recordings that support diverse downstream clinical classification tasks. Across the four model sizes, larger models generally achieve better performance. Pretraining learning curves are reported in Appendix D.8.

Table 1: Baseline comparison on MCI and PD. Values are means with variability terms. **Bold** indicates the best.

	MCI			PD		
Model	AUROC	AUPRC	BalAcc	BalAcc	Weighted F1	Kappa
TCN	0.8440 ± 0.00082	0.7572 ± 0.00306	0.7699 ± 0.00592	0.5985 ± 0.01296	0.7118 ± 0.00764	0.5059 ± 0.01228
TimesNet	0.8272 ± 0.00720	0.7115 ± 0.01696	0.7570 ± 0.01184	0.6101 ± 0.00932	0.7257 ± 0.00608	0.5182 ± 0.01024
NST	0.8984 ± 0.00170	0.8392 ± 0.00348	0.7944 ± 0.00804	0.6024 ± 0.01694	0.7136 ± 0.01684	0.5091 ± 0.02706
CNNTransformer	0.8108 ± 0.00136	0.6997 ± 0.00446	0.7369 ± 0.00852	0.6544 ± 0.01004	0.7494 ± 0.00734	0.5674 ± 0.01100
InceptionTime	0.7658 ± 0.00720	0.6153 ± 0.01342	0.7332 ± 0.01056	0.6093 ± 0.00872	0.7272 ± 0.00650	0.5302 ± 0.00784
ConvTran	0.8743 ± 0.00554	0.7928 ± 0.00690	0.8294 ± 0.00890	0.6561 ± 0.00890	0.7546 ± 0.00492	0.5695 ± 0.00920
StimEM-S	0.8855 ± 0.00470	0.8200 ± 0.00584	0.7999 ± 0.00780	0.7278 ± 0.00398	0.8016 ± 0.00382	0.6514 ± 0.00640
StimEM-M	0.8932 ± 0.00076	0.8386 ± 0.00020	0.8192 ± 0.00190	0.7261 ± 0.00252	0.8082 ± 0.00136	0.6637 ± 0.00234
StimEM-L	0.9200 ± 0.00220	0.8774 ± 0.00272	0.8526 ± 0.00258	0.7513 ± 0.00488	**0.8355** ± 0.00360	**0.7107** ± 0.00636
StimEM-XL	**0.9250** ± 0.00252	**0.8798** ± 0.00498	**0.8815** ± 0.00378	**0.7592** ± 0.00482	0.8331 ± 0.00290	0.7059 ± 0.00490

4.4 SCALING PRETRAINING DATA AND MODEL SIZE

We evaluate how downstream performance scales with pretraining data size across four model capacities, using 0%, 12.5%, 25%, 50%, and 100% of the pretraining corpus (Figure 3).

Pretraining improves all four models, with the full corpus giving the best results across the reported metrics. On MCI, metrics gain flatten for StimEM-S and StimEM-M but continue for StimEM-L and StimEM-XL, suggesting that larger models benefit from further expansion of the pretraining corpus. Experimental settings are detailed in Appendix C.1.

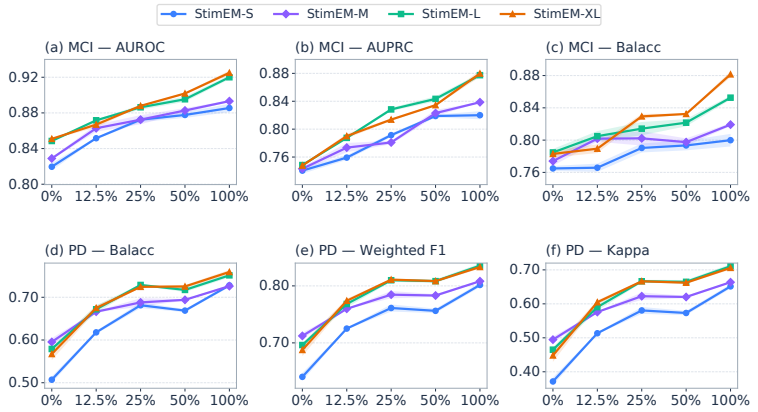
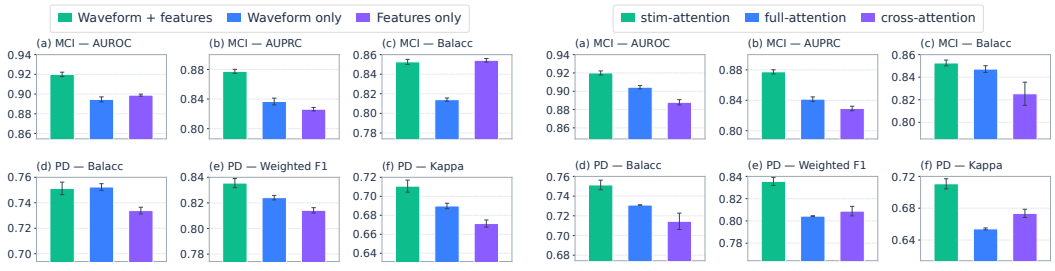


Figure 3: Downstream performance across pretraining data fractions and model sizes. The horizontal axis in each panel shows the fraction of pretraining data used. The 0% condition represents Without pretrain.

4.5 ABLATION STUDY ON TOKENIZER LOSSES

We compare joint waveform and behavioral-feature supervision with either source alone to examine what the tokenizer should preserve (Figure 4(a)). These variants differ in the losses used to train the tokenizer. All three use frozen FSQ targets for masked prediction.

Joint supervision performs best on most metrics. Removing either source weakens the overall result profile. Waveform reconstruction encourages the preservation of time-resolved gaze and pupil dynamics, including trajectory shape and transient fluctuations that trial-level summary features do not fully capture. Behavioral supervision asks the same representation to retain response timing, spatial precision and task outcomes, which waveform fidelity alone does not explicitly prioritize. Their combination therefore gives the tokenizer complementary constraints on how a response unfolds and how it relates to task behavior. Experimental settings are detailed in Appendix C.2.



(a) Tokenizer losses **(b) Attention mechanisms**

Figure 4: Tokenizer-loss and attention ablations on MCI and PD. **(a)** Joint waveform and behavioral-feature supervision compared with either loss alone. **(b)** Stim-conditioned attention compared with full-attention and cross-attention; Stim-attention in the plot denotes stimulus-conditioned attention.

4.6 Attention mechanism comparison

Stim-conditioned attention outperforms full-attention and cross-attention on all reported metrics for MCI and PD (Figure 4(b)). For each variant, we apply the same Transformer attention form in both the tokenizer and the masked-pretraining encoder, aligning stimulus-response processing across the two stages. Full-attention allows all valid stimulus, ocular and CLS tokens to attend to one another. Cross-attention first mixes stimulus and ocular tokens at aligned time positions, then applies temporal attention separately to the stimulus and CLS/eye sequences. Full-attention allows response information to alter stimulus representations, potentially weakening the reference for encoding response timing and trajectory. Cross-attention separates cross-stream interaction from temporal integration, which may make delayed stimulus-response relations harder to capture. Stim-conditioned attention preserves a stimulus-only reference while allowing ocular tokens to integrate stimulus and response information across time. These results favor the stimulus-conditioned design for the complete tokenizer–encoder pipeline. Experimental settings are detailed in Appendix C.3.

4.7 Masking strategy comparison

Paired dual-scale masking outperforms Single, Short, Long and Random masking on all metrics for MCI and PD (Figure 5). This supports learning from complementary short-span and long-span temporal context. The short-span view encourages the encoder to capture local movement dynamics and precise response timing. The long-span view requires it to integrate more distant response context with stimulus information, capturing dependencies that local reconstruction alone may overlook.

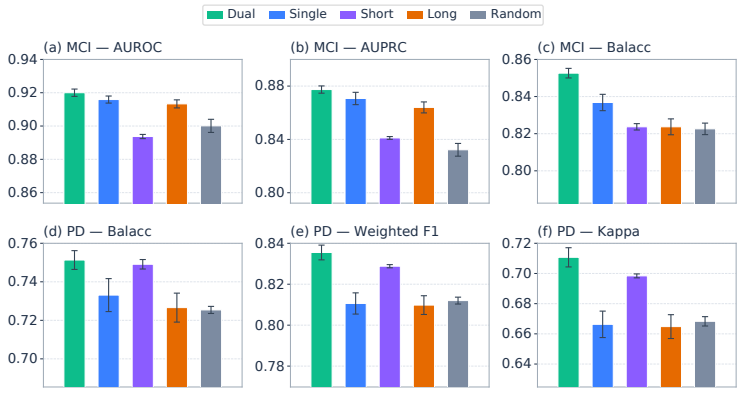


Figure 5: Temporal masking ablation on MCI and PD.

Single samples short and long spans within the same masked trial view, exposing the model to a broad range of prediction lengths at once. Mixing local reconstruction and longer-range inference within one view makes it harder to learn an effective prediction strategy for either scale. Dual assigns these demands to separate views of each trial while sharing the encoder. Short-span prediction provides nearby cues alongside the more demanding long-span task, which may facilitate optimization and help retain both local detail and longer-range response structure. Masking configurations and matched batch sizes are detailed in Appendix C.4.

5 Conclusion

We introduced StimEM, pretrained on a large-scale corpus of stimulus-driven eye movements across multiple paradigms. Its stim-conditioned Transformer integrates binocular responses with a stimulus-only reference, while an ocular tokenizer trained to reconstruct waveforms and predict behavioral features provides discrete targets for paired dual-scale masked prediction. StimEM outperforms SOTA baselines across all downstream tasks, demonstrating strong representation learning and generalization. The observed scaling trends suggest further gains from larger corpora and inform future eye-movement foundation models. Future work will integrate EEG for BCI and broader applications, and align ocular representations with language models to enable more accessible and comprehensive clinical assessment and diagnosis.

AI USE STATEMENT

Generative AI tools assisted in organizing the authors' model description, drafting manuscript text, preparing the LaTeX project, and creating the conceptual illustrations in Figures 1 and 2. The authors reviewed all AI-assisted text and figures for accuracy and consistency with the implemented method. The authors take responsibility for the final content of this work, including all AI-assisted text and figures.

ETHICS STATEMENT

This study was approved by the relevant ethics committees of the participating hospitals. All participants provided written informed consent to participate in the study. All records were de-identified before analysis to protect participant privacy.

REPRODUCIBILITY STATEMENT

Section 3 describes the model architecture, tokenizer objectives, and masked pretraining procedure. Appendix A details data acquisition and subject-level partitions; Appendix B provides preprocessing, model configurations, training hyperparameters, and downstream fine-tuning settings. Ablation protocols and supplementary results are reported in Appendices C and D, respectively. The model implementation is provided in the supplementary material. Model weights and a subset of the test data will be publicly released with the final version of the paper.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. doi: 10.48550/arXiv.1607.06450. arXiv preprint.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12449–12460. Curran Associates, Inc., 2020.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. doi: 10.48550/arXiv.1803.01271. arXiv preprint.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*. OpenReview.net, 2022.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. doi: 10.48550/arXiv.1308.3432. arXiv preprint.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Trevor J. Crawford, Steve Higham, Ted Renvoize, Julie Patel, Mark Dale, Anur Suriya, and Sue Tetley. Inhibitory control of saccadic eye movements and cognitive impairment in Alzheimer's disease. *Biological Psychiatry*, 57(9):1052–1060, 2005. doi: 10.1016/j.biopsych.2005.01.017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I. Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38(1):22–48, 2024. doi: 10.1007/s10618-023-00948-2.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15979–15988. IEEE, 2022. doi: 10.1109/cvpr52688.2022.01553.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

S. B. Hutton. Cognitive control of saccadic eye movements. *Brain and Cognition*, 68(3):327–340, 2008. doi: 10.1016/j.bandc.2008.08.021.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020. doi: 10.1007/s10618-020-00710-y.

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*. OpenReview.net, 2024.

Weibang Jiang, Yansen Wang, Bao-Liang Lu, and Dongsheng Li. NeuroLM: A universal multitask foundation model for bridging the gap between language and EEG signals. In *International Conference on Learning Representations*. OpenReview.net, 2025.

Christoph Klein, Reinhold Rauh, and Monica Biscaldi. Cognitive correlates of anti-saccade task performance. *Experimental Brain Research*, 203(4):759–764, 2010. doi: 10.1007/s00221-010-2276-5.

Dmitry Lagun, Cecelia Manzanares, Stuart M. Zola, Elizabeth A. Buffalo, and Eugene Agichtein. Detecting cognitive impairment by eye movement analysis using automatic classification algorithms. *Journal of Neuroscience Methods*, 201(1):196–203, 2011. doi: 10.1016/j.jneumeth.2011.06.027.

Hua Li, Anjiao Peng, Wanlin Lai, Junru Wu, Shuming Ji, Dan Hu, Shujuan Chen, Chenxing Zhu, Qiulei Hong, Mingsha Zhang, and Lei Chen. Impacts of education level on Montreal Cognitive Assessment and saccades in community residents from Western China. *Clinical Neurophysiology*, 161:27–39, 2024. doi: 10.1016/j.clinph.2024.02.017.

Bin Lin, Zongjian Li, Yuwei Niu, Kaixiong Gong, Yunyang Ge, Yunlong Lin, Mingzhe Zheng, JianWei Zhang, Miles Yang, Zhao Zhong, Liefeng Bo, and Li Yuan. iFSQ: Improving FSQ for image generation with 1 line of code. *CoRR*, abs/2601.17124, 2026. doi: 10.48550/arXiv.2601.17124. arXiv preprint.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893. Curran Associates, Inc., 2022. doi: 10.52202/068431-0718.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*. OpenReview.net, 2019.

Wenbo Ma and Mingsha Zhang. Multiple step saccades are generated by internal real-time saccadic error correction. *Frontiers in Neuroscience*, 17:1112655, 2023. doi: 10.3389/fnins.2023.1112655.

Wenbo Ma, Min Li, Junru Wu, Zhihao Zhang, Fangfang Jia, Mingsha Zhang, Hagai Bergman, Xuemei Li, Zhipei Ling, and Xin Xu. Multiple step saccades in simply reactive saccades could serve as a complementary biomarker for the early diagnosis of Parkinson's disease. *Frontiers in Aging Neuroscience*, 14:912967, 2022. doi: 10.3389/fnagi.2022.912967.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*. OpenReview.net, 2024.

Akane Oyama, Shuko Takeda, Yuki Ito, Tsuneo Nakajima, Yoichi Takami, Yasushi Takeya, Koichi Yamamoto, Ken Sugimoto, Hideo Shimizu, Munehisa Shimamura, Taiichi Katayama, Hiromi Rakugi, and Ryuichi Morishita. Novel method for rapid assessment of cognitive impairment using high-performance eye-tracking technology. *Scientific Reports*, 9(1):12932, 2019. doi: 10.1038/s41598-019-49275-x.

Leihao Sha, Hua Li, Anjiao Peng, Huajun Yang, Xin Liu, Hongjian Zhao, Wenbo Ma, Qiulei Hong, Yusha Tang, Mingsha Zhang, and Lei Chen. Diagnostic value of saccades in mild cognitive impairment: a community-based study. *The Journals of Gerontology, Series A: Biological Sciences and Medical Sciences*, 81(2):glaf264, 2026. doi: 10.1093/gerona/glaf264.

Harshinee Sriram, Cristina Conati, and Thalia Field. Classification of Alzheimer's disease with deep learning on eye-tracking data. In *Proceedings of the 25th International Conference on Multimodal Interaction*, pp. 104–113. ACM, 2023. doi: 10.1145/3577190.3614149.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10078–10093. Curran Associates, Inc., 2022. doi: 10.52202/068431-0732.

Gonzalo Uribarri, Simon Ekman von Huth, Josefine Waldthaler, Per Svenningsson, and Erik Fransén. Deep learning models accurately classify Parkinson's disease from eye-tracking fixation data. *International Journal of Medical Informatics*, 220:106626, 2026. doi: 10.1016/j.ijmedinf.2026.106626.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Josefine Waldthaler, Mikkel C. Vinding, Allison Eriksson, Per Svenningsson, and Daniel Lundqvist. Neural correlates of impaired response inhibition in the antisaccade task in Parkinson's disease. *Behavioural Brain Research*, 422:113763, 2022. doi: 10.1016/j.bbr.2022.113763.

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*. OpenReview.net, 2025.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*. OpenReview.net, 2023.

Fangyu Zuo, Peiguang Jing, Jinglin Sun, Jizhong Duan, Yong Ji, and Yu Liu. Deep learning-based eye-tracking analysis for diagnosis of Alzheimer's disease using 3D comprehensive visual stimuli. *IEEE Journal of Biomedical and Health Informatics*, 28(5):2781–2793, 2024. doi: 10.1109/jbhi.2024.3365172.

A Dataset settings

A.1 Dataset composition and source collections

A.1.1 Data acquisition

The acquisition protocol described by Sha et al. (2026) used a 1,000 Hz infrared video eye tracker (EM-2000 series, Chengdu Jasmines BioTech, China) and a 27-inch LCD (1,920×1,080 pixels; 100 Hz). Participants used a chinrest at a viewing distance of 57 cm, with their eyes aligned to the screen's horizontal midline. Calibration and validation accompanied four tasks: PS (pro-saccade), AS (anti-saccade), MGS (memory-guided saccade), and DS (double-step saccade). These task families are also described by Li et al. (2024). Figure A1 summarizes the four paradigms; task instructions and indexed sequences are detailed below.

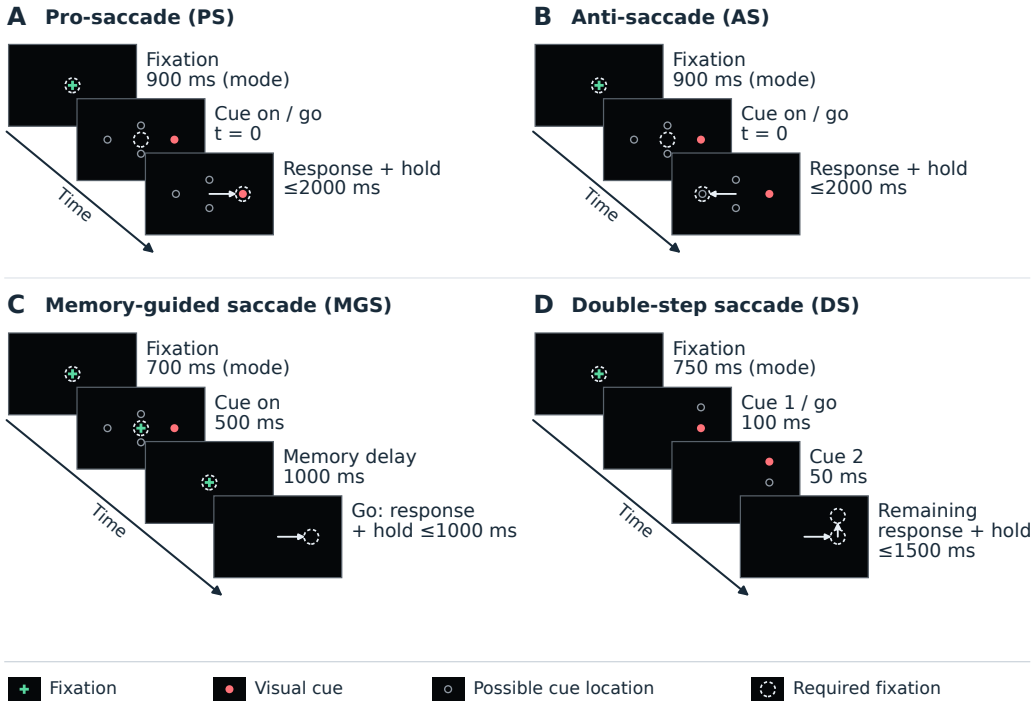


Figure A1: The four acquisition paradigms and their stimulus-response sequences. Crosses mark fixation and filled circles mark visual cues; arrows indicate instructed gaze shifts. Timing labels summarize the indexed event sequences, and spatial positions are schematic. Detailed timing conventions appear in Appendix A.4.

PS (Pro-saccade). A trial begins with central fixation. A peripheral target then appears as the fixation cue disappears, and the participant shifts gaze toward the visible target and holds fixation there. This visually guided response provides a baseline for the other paradigms. Response latency, saccade amplitude, and landing error characterize how rapidly and accurately the eye reaches the target.

AS (Anti-saccade). The stimulus sequence resembles PS, but the participant must suppress the reflexive response toward the visible cue and instead look to its mirror location across fixation. The goal location is not marked by a visible target. The recorded response can include a correct anti-saccade, a reflexive direction error, and a subsequent correction, supporting measures of inhibitory control and spatial accuracy.

MGS (Memory-guided saccade). A peripheral cue appears briefly while central fixation is maintained. The cue then disappears, followed by a memory delay during which the participant must

continue fixating centrally. Removal of the fixation cue signals a saccade to the remembered location, without a visible target. Complete successful MGS events in our source audit had a 500 ms cue and a 1,000 ms memory delay. The response captures spatial memory, response occurrence and velocity, premature responses, and landing accuracy.

DS (Double-step saccade). Two peripheral locations are cued in sequence, and the participant makes two ordered gaze shifts toward those locations. The first movement changes the starting position for the second, coupling sequential execution with spatial updating. In our audited DS records, fixation offset coincides with the first cue onset, so a response need not wait until the second cue disappears. Source events confirm a combined 150 ms cue interval; the prepared stimulus stream encodes 100/50 ms, but the source event table does not record the intermediate switch. First-step latency and amplitude, step precision, and final landing error characterize performance.

A.1.2 PRETRAINING DATA

Pretraining draws on 4,255 subjects (counted by dataset identifiers) and approximately 500 h of four-task source recordings in the common 1,000 Hz representation. Shared binocular samples are counted once and duplicate file copies excluded.

Participants range in age from 18 to 93 years and include 2,677 female and 1,578 male subjects, spanning adolescence, adulthood, and older age. The source metadata also capture varied educational backgrounds and marital statuses.

The model uses three synchronized four-channel streams: stimulus (fixation-cue presence, stimulus presence, horizontal location, and vertical location), and left- and right-eye streams (horizontal gaze, vertical gaze, pupil area, and a binary blink indicator). Gaze and pupil area are normalized; missing observations and padding have separate validity masks.

A.1.3 FINE-TUNING DATA

Fine-tuning uses clinical cohorts and control recordings collected under the four acquisition paradigms described above. The source collections differ in clinical composition and recording coverage. Subject counts use dataset identifiers. Clinical label definitions and evaluation splits are detailed in Appendices A.2 and A.3.

AD (Alzheimer's disease). The AD collection includes participants with Alzheimer's disease and a control group, supporting binary classification of the AD cohort versus controls. Together, these groups comprise 252 subjects and 37,808 trials, with a total recording duration of 22.63 h.

MCI (mild cognitive impairment). The MCI collection comprises participants labeled with mild cognitive impairment and a control group. Their recordings support the downstream task of distinguishing MCI from controls. The collection contains 383 subjects, 58,279 trials, and 36.85 h of recordings.

PD (Parkinson's disease and related conditions). The PD dataset supports multiclass classification among controls, parkinson-spectrum participants, and tremor-spectrum participants. It comprises 417 subjects and 55,385 trials, with a total recording duration of 35.43 h.

MIG (migraine). The migraine collection includes a migraine-labeled clinical group and its control group, used for binary migraine-versus-control classification. It contains 152 subjects and 23,327 trials, totaling 13.23 h of recordings.

DETOX (detoxification-center cohort). This collection includes participants from a detoxification-center cohort and a control group. The available source labels identify cohort membership. It contains 91 subjects and 13,704 trials, with a total duration of 8.11 h, and supports binary cohort-versus-control classification.

EPI (epilepsy). The epilepsy collection includes an epilepsy-labeled clinical group and a control group. These recordings support binary epilepsy-versus-control classification and comprise 572 subjects, 89,969 trials, and 49.56 h of recordings.

All durations are rounded independently. Source control labels are retained; no additional health-status criteria are inferred from cohort membership.

Disease diagnoses were established by clinicians at the contributing hospitals using clinical evaluations and relevant assessments, including standardized rating scales where applicable. Fine-tuning labels follow these source-provided diagnoses and cohort assignments.

A.2 Clinical labels

Clinical labels identify a subject's cohort, whereas acquisition-task labels specify the instruction followed during a recording. MCI, AD, Detox, epilepsy and migraine are binary classification tasks: each separates the corresponding source-labeled clinical cohort from controls. MCI denotes mild cognitive impairment; AD denotes Alzheimer's disease. Detox denotes the detoxification cohort. Epilepsy and migraine retain their respective source cohort labels. These labels supervise downstream classification; they are not used to generate tokenizer or masked-pretraining targets.

PD denotes a multiclass task with three classes: control, parkinson-spectrum and tremor-spectrum labels. The parkinson-spectrum class combines the source PD and movement-disorder groups. The tremor-spectrum class combines tremor and essential tremor. Thus, PD evaluates both control-versus-clinical separation and discrimination between the two clinical groupings. All cohorts use the same acquisition paradigms. The encoder learns task-conditioned responses during pretraining, and subject-level fine-tuning associates those responses with clinical labels.

A.3 Recording and evaluation organization

Source collections and their composition are described in Appendix A.1. Recordings with at least one usable eye are eligible for pretraining.

We divide the pretraining corpus by subject into nominal 90:5:5 training, validation and reserved test partitions. No subject occurs in more than one of these pretraining partitions. The reserved pretraining test partition is not used for parameter updates.

Within each downstream cohort, training, validation and test subjects are disjoint, with a nominal 64:16:20 allocation. Downstream evaluation requires usable recordings from all four acquisition tasks: at least 16 trials per task for training and four per task for validation or testing. Subject-level pooling combines these complementary responses into one clinical prediction. Only downstream training subjects supply labels for supervised fine-tuning; validation labels select checkpoints, and test labels are used only for final scoring. This is a transductive classification protocol without test-label leakage.

A.4 Event timing and recording quality

Figure A1 summarizes timings derived from the recorded events. The initial fixation durations show the observed modes, including acquisition; some recordings use a shorter fixation period. We show response and hold as a single window because the event records do not distinguish their boundary. Each upper bound marks the observed end of the normal task sequence, including trials with unsuccessful responses. These windows are measured from cue onset for prosaccade and antisaccade, fixation offset for memory-guided saccade, and second-cue offset for double saccade. The double-saccade cue durations follow the source encoding, with their combined duration also confirmed by source events.

Each eye has Valid, Blink and Missing quality labels. Pupil-area reductions and closing or reopening transitions identify blink intervals. Tracking loss, frozen raw channels and transient artifacts identify missing intervals. The dataset used here includes guarded gaze filtering inherited from signal preprocessing, with no additional filtering during data repacking. A fourth-order 75-Hz Butterworth low-pass filter is applied forward and backward within stable valid segments of at least 40 ms. Filtering excludes 25-ms neighborhoods around detected saccades and 30-ms neighborhoods around blink or missing samples. A 10-ms raised-cosine blend limits edge transitions. If filtering introduces or removes a detected saccade, affected samples are restored; unresolved event changes trigger a return to the unfiltered trial. Quality labels, pupil channels and stimulus channels remain unchanged. This procedure targets tracking noise while preserving detected rapid movements and quality boundaries. Normalization and patch-level validity rules are specified in Appendix B.3.

B MODEL SETTINGS

B.1 MODEL CONFIGURATIONS

Table 2 lists the encoder depth, embedding dimension, feed-forward width, attention heads and parameter count for each StimEM configuration. Parameter counts include the input embeddings, encoder and four masked-code prediction heads; they exclude the separately trained tokenizer and the downstream classifier. All configurations use 40-sample patches at 1,000 Hz, corresponding to 40 ms and a maximum input duration of 5.12 s per trial.

Table 2: StimEM configurations. FFN denotes the intermediate feed-forward width; hidden size is the token embedding dimension. Parameters are in millions, including masked-prediction heads and excluding the tokenizer.

Model	Layers	FFN	Hidden size	Heads	Params (M)
StimEM-S	8	756	256	8	7.7
StimEM-M	10	960	320	8	14.4
StimEM-L	12	1152	384	8	24.3
StimEM-XL	16	1536	512	8	56.2

B.2 METHOD HYPERPARAMETERS

We use $C_S = C_E = 4$, $P = 40$, and $N_{\max} = 128$. Encoder depths, widths, and head counts are listed in Table 2; the tokenizer decoder has $L_D = 3$ layers. FSQ uses $D_q = 4$ and levels $(\ell_1, \ldots, \ell_{D_q}) = (9, 7, 5, 5)$, giving $K = 1{,}575$ codes. The feature head has $F = 38$ outputs, with two count features excluded from the loss. Waveform coefficients are $\lambda_x = \lambda_y = 1$ and $\lambda_{\text{area}} = \lambda_{\text{blink}} = 0.1$, with the gaze term averaged jointly over valid horizontal and vertical samples. We set $\lambda_{\text{feat}} = 0.0015$ and $\alpha_j = 1.5$ for continuous features or 0.25 for binary features. Short masking uses $B_s = 15$ and $[m_s^-, m_s^+] = [1, 3]$ patches; long masking uses $B_l = 5$ and $[m_l^-, m_l^+] = [4, 6]$. We set $g = 1$, $\rho = 0.85$, and $r_m = 0.5$. For the reference dual-scale setup, each masked-pretraining batch contains 64 trials per GPU, with two masked views per trial, giving 128 trial views. Batch construction for the masking comparison is detailed in Appendix C.4. Batch sizes and training schedules are listed in Table 3; subject-level trial sampling is described in Appendix B.8.

B.3 PREPROCESSING AND OPTIMIZATION

We normalize gaze coordinates using fixed horizontal and vertical reference ranges. Pupil area uses robust normalization per subject and eye; when observations are sparse, pooled or global statistics are substituted instead, and extreme values are clipped. The scope of the area statistics follows Appendix A.3. Tokenizer eye tokens require at least 50% non-missing samples, and paired-eye masking requires both eyes to meet the stricter 85% threshold.

The reference tokenizer uses a separately trained stimulus-conditioned encoder with the same architecture and dimensions as StimEM-L. Its decoder shares the encoder's hidden size, FFN width and number of attention heads, with depth specified in Appendix B.2. We use AdamW (Loshchilov & Hutter, 2019) throughout training, with the schedules in Table 3. The encoder and classifier use the same fine-tuning learning rate.

B.4 ARCHITECTURE AND TOKENIZER IMPLEMENTATION

Quality embeddings. During masked pretraining, the quality term in Equation 1 is $\eta_i^s = b_{\kappa(\nu_i^s)}$ for eye tokens, where ν_i^s is the non-missing sample fraction and κ bins this fraction. It is zero for stimulus tokens and outside masked pretraining.

Standard block operations. In Equation 2, the feed-forward network is

$$\text{FFN}(Z) = [(ZW_a) \odot \text{SiLU}(ZW_g)]W_o. \tag{12}$$

Table 3: Optimization settings. Batch sizes are per GPU; the masked-pretraining batch size counts masked trial views.

Setting	Tokenizer	Masked pretraining	Fine-tuning
Optimizer	AdamW	AdamW	AdamW
Peak learning rate	2×10^{-4}	3×10^{-4}	5×10^{-5}
Minimum learning rate	2×10^{-5}	5×10^{-5}	5×10^{-6}
Schedule	Cosine	Cosine	Cosine
Warm-up	2,000 steps	1,000 steps	1 epoch
Training duration	40,000 steps	20,000 steps	15 epochs
Adam β_1, β_2	0.9, 0.95	0.9, 0.95	0.9, 0.95
Gradient clipping	1.0	1.0	1.0
Batch size	128 trials	128 trial-views	4 subjects

The matrices W_a, W_g project to the feed-forward width, W_o projects back to d, and $\odot$ denotes element-wise multiplication. Each head projects normalized states to $Q_h = ZW_h^Q$, $K_h = ZW_h^K$, and $V_h = ZW_h^V$. With H_a heads of dimension $d_h = d/H_a$,

$$\mathrm{SCA}(Z; A) = \mathrm{Concat}(O_1, \ldots, O_{H_a})W^O, \tag{13}$$

where O_h is the output of Equation 3 for head h. Token-wise normalization and feed-forward operations preserve the separation between stimulus states and ocular information.

FSQ and supervision details. The implicit code space contains $K = \prod_{k=1}^{D_q} \ell_k$ combinations. We use $q_{\mathrm{ST}} = u + \mathrm{sg}(q - u)$, where sg stops gradients (Bengio et al., 2013). FSQ uses fixed scalar grids and requires neither a learned-codebook loss nor a commitment loss. The stimulus is neither discretized nor reconstructed. Missing and padded samples are excluded from waveform losses; gaze and pupil-area losses additionally exclude blink frames. Behavioral supervision uses the available continuous and binary features, excluding count features. Channel and feature-type weights are given in Appendix B.2.

B.5 Trial-level behavioral features

The tokenizer's feature head has 38 outputs, listed in Table 4. Of these, 21 are binary indicators, 15 are continuous measurements and two are count features. The two count outputs are retained for interface compatibility but excluded from the loss; therefore, at most 36 feature dimensions contribute to supervision. Only task-applicable, available targets contribute for each trial, so a trial is not supervised on every output.

Continuous targets use training-set median and median absolute deviation (MAD) statistics: when the MAD is positive, values are centered, divided by $1.4826\,\mathrm{MAD} + 10^{-8}$ and clipped to $[-5, 5]$. Binary targets remain 0/1 indicators. Continuous and binary targets use smooth-L1 and binary cross-entropy losses, respectively; their weights are specified in Appendix B.2. The units in the table describe targets before normalization.

B.6 Paired dual-scale mask sampling

For each trial, the short-span view uses B_s blocks with lengths in $[m_s^-, m_s^+]$ and the long-span view uses B_l blocks with lengths in $[m_l^-, m_l^+]$, where $m_s^+ < m_l^-$. The masks are independently sampled and are not complements. Within a view, blocks do not overlap and are separated by at least g unmasked patches where feasible. Both eyes must meet a non-missing threshold ρ at selected positions. The nominal masking ratio is r_m; realized coverage depends on trial length and the available paired-eye positions. Short- and long-span views are stacked along the batch axis. Reference values appear in Appendix B.2, and ablation settings in Appendix C.7.

The coordinate probabilities in Equation 9 define $p_\theta(z_i^e \mid \widetilde{X}_v, S) = \prod_{k=1}^{D_q} p_{\theta,k}(z_{i,k}^e \mid \widetilde{X}_v, S)$, so summing coordinate cross-entropies yields the code's negative log-likelihood in Equation 10.

Table 4: Task-specific behavioral outputs of the tokenizer. Binary entries are flags. †Count outputs are excluded from the training loss. Only available targets for the corresponding task are supervised.

Task	Feature	Type	Original unit
Prosaccade	Nonresponse	Binary	0/1
	Response latency	Continuous	ms
	First-step amplitude	Continuous	deg
	Saccade count	Count†	count
	Multistep response	Binary	0/1
	Blink	Binary	0/1
	Fixation break	Binary	0/1
Antisaccade	Accuracy	Binary	0/1
	First spatial error	Continuous	deg
	Reflexive error	Binary	0/1
	Correction after reflexive error	Binary	0/1
	Blink	Binary	0/1
	Fixation break	Binary	0/1
	Final landing error	Continuous	deg
	Reflexive saccade count	Count†	count
	Corrective latency	Continuous	ms
	Corrective landing error	Continuous	deg
	Error	Binary	0/1
Memory-guided	Accuracy	Binary	0/1
	Nonresponse	Binary	0/1
	Mean velocity	Continuous	deg/s
	Peak velocity	Continuous	deg/s
	First-step amplitude	Continuous	deg
	First spatial error	Continuous	deg
	Correction after premature error	Binary	0/1
	Blink	Binary	0/1
	Fixation break	Binary	0/1
	Final landing error	Continuous	deg
	Error	Binary	0/1
Double-step	Accuracy	Binary	0/1
	Nonresponse	Binary	0/1
	First-step latency	Continuous	ms
	First-step amplitude	Continuous	deg
	Mean step precision	Continuous	deg
	Blink	Binary	0/1
	Fixation break	Binary	0/1
	Final landing error	Continuous	deg
	Error	Binary	0/1

B.7 WITHOUT-TOKENIZER PRETRAINING

The without-tokenizer variant applies the tokenizer's supervision objectives directly during masked pretraining. Encoder, masking and optimization settings follow the reference masked-pretraining configuration. The decoder and its waveform and behavioral-feature prediction heads use the same architecture as the tokenizer. The encoder, decoder and heads are trained together during pretraining, without a separately trained tokenizer or discrete-code targets.

Waveform reconstruction is computed only at masked eye-patch positions, with the channel-specific validity exclusions in Appendix B.4. The decoder's classification state predicts the task-applicable, available trial-level features described in Appendix B.5. Reconstruction and feature losses use the tokenizer objective in Equation 7, with the same pointwise losses and weights in Appendix B.2.

B.8 SUBJECT-LEVEL FINE-TUNING

Initialization and trial sampling. We initialize the encoder and patch embeddings from the masked-pretraining checkpoint and replace the code-prediction heads with a clinical classifier. The tokenizer is not used during fine-tuning or inference. At each epoch, we sample $K_{\mathrm{trial}} = 16$ trials without replacement from each of $J = 4$ acquisition tasks for each training subject, giving a bag of 64 trials. Bags are resampled each epoch. The reference configuration updates all encoder layers, patch embeddings and classifier parameters according to Table 3.

Subject prediction. For subject u, a shared classifier with a 128-dimensional hidden layer, GELU and dropout 0.3 produces trial-level logits $o_{u,t,r}$ for trial r from acquisition task t. We aggregate these logits hierarchically:

$$o_u = \frac{1}{J} \sum_{t=1}^{J} \frac{1}{|\mathcal{B}_{u,t}|} \sum_{r \in \mathcal{B}_{u,t}} o_{u,t,r}, \tag{14}$$

where J is the number of acquisition tasks and $\mathcal{B}_{u,t}$ contains trials from task t. Averaging logits within each task and then across tasks prevents differences in trial count from dominating the subject prediction.

Classification losses. Binary tasks use subject-level binary cross-entropy, with the positive-class weight determined by the training-subject class counts. PD uses two sigmoid heads: clinical versus control, and tremor-spectrum versus parkinson-spectrum conditional on a clinical label. The first head is trained on all training subjects; the second is trained only on the two clinical groups. The two losses receive equal weight, with class-frequency weighting computed separately within each head. If p_d is the clinical probability and p_t is the conditional tremor-spectrum probability, the three class probabilities are $1 - p_d$, $p_d(1 - p_t)$ and $p_d p_t$ for control, parkinson-spectrum and tremor-spectrum, respectively.

Checkpoint selection and evaluation. Validation uses all eligible trials for each subject with the same pooling rule. We select the checkpoint with the highest validation AUROC for binary tasks or macro one-versus-rest AUROC for PD, then evaluate it on the designated test subjects using all their eligible trials. Test labels do not enter optimization or checkpoint selection. Learning rates, batch size and the optimization schedule are given in Table 3; eligibility rules appear in Appendix A.3.

B.9 Baseline comparison settings

We compare against six supervised time-series models: TCN, TimesNet, Non-stationary Transformers (NST), CNN-Transformer, InceptionTime, and ConvTran. These baselines include convolutional, Transformer-based, and hybrid architectures. Each model uses 30 training epochs, an effective batch of four subjects and dropout 0.3. Checkpoints are selected by validation AUROC for binary tasks and macro one-versus-rest AUROC for PD. Reported values summarize the individual runs with seeds 42, 43 and 44.

The input and screening settings match StimEM: normalized signals at 1,000 Hz, 40-sample temporal patches, a 5,120-sample limit, explicit padding masks and the four-task eligibility rules above. Training resamples 16 trials per task for each subject, while evaluation uses all eligible trials. Baselines average trial features within each task and then average the four task features before classification.

C ABLATION SETTINGS

The following subsections describe configurations for the main-text studies, followed by supplementary ablations in the order of their results in Appendix D. The without-tokenizer implementation is specified in Appendix B.7.

C.1 PRETRAINING DATA AND MODEL SIZE

We evaluate the four configurations in Table 2 using 12.5%, 25%, 50% and 100% of the pretraining data. The data subsets use seed 42, and the pretraining validation split remains fixed. All four model sizes use targets from the same frozen 12-layer FSQ tokenizer, so changing encoder capacity does not require training a new tokenizer. The 0% condition skips masked pretraining and trains the downstream model from random initialization.

All nonzero data fractions are pretrained for the same effective number of epochs. We scale the number of training updates with the size of each subset and adjust the warm-up schedule accordingly. The downstream tasks, metrics and three-seed evaluation follow the main-text protocol.

C.2 TOKENIZER LOSS SETTINGS

We train three tokenizers using the reference FSQ levels, encoder and decoder dimensions. Joint supervision uses the objective and weights in Equation 7 and Appendix B.2. Waveform only sets the feature-loss weight to zero; Features only sets the waveform-loss weight to zero. All other loss weights remain unchanged.

Each tokenizer is trained and then frozen to produce targets for a StimEM-L encoder with paired dual-scale masking and factorized cross-entropy. Both stages follow the training schedules in Table 3. We then fine-tune and evaluate each configuration on MCI and PD using seeds 42, 43 and 44. Settings for the without tokenizer comparison are given in Appendix B.7, with results in Appendix D.2.

C.3 ATTENTION MECHANISM SETTINGS

Stim-conditioned attention restricts stimulus queries to stimulus keys while allowing ocular and CLS queries to access all valid tokens. Full-attention removes this restriction while retaining padding masks. Cross-attention uses aligned stimulus/left-eye/right-eye mixing followed by separate temporal attention for stimulus tokens and for the CLS/eye sequence in each block. It uses two attention modules per block rather than the single module in the joint-attention configurations.

All three masked encoders use 12 layers, hidden width 384, eight heads and FFN width 1,152, with paired dual-scale masking and 20,000 pretraining steps. For each attention variant, we use the corresponding Transformer form in both the tokenizer and the masked-pretraining encoder. We first train its FSQ tokenizer and then freeze it to provide targets for the matching encoder. This keeps stimulus-response processing consistent across the two stages. The comparison therefore evaluates the complete two-stage attention configuration. The cross-attention tokenizer uses a 5,000-step warm-up, compared with 2,000 steps for the reference tokenizer. Downstream evaluation uses the same MCI and PD tasks and three-seed protocol.

C.4 MASKING STRATEGY

For the temporal masking comparison in the main text, Dual uses 15 short blocks of 1 to 3 patches and five long blocks of 4 to 6 patches in separate views. Single draws spans from 1 to 6 patches. Short and Long restrict spans to 1 to 3 and 4 to 6 patches, respectively. Single uses the symmetric span-length sampling scheme, whereas Short and Long sample uniformly within their respective ranges. Random masks individual patches without requiring contiguous spans. Single, Short, Long and Random use a nominal masking ratio of 0.5 and one masked view per trial; Dual constructs separate short- and long-span views. All variants mask aligned positions in both eyes and leave stimulus tokens visible. They use the reference frozen FSQ tokenizer, the StimEM-L encoder, factorized cross-entropy and 20,000 masked-pretraining steps.

Dual samples 64 trials per GPU at each training step. Each trial is used twice, once with a short-span mask and once with a long-span mask, yielding $64 \times 2 = 128$ masked trial views. Single, Short, Long and Random each use 128 trials per GPU with one masked view per trial. These five settings therefore use the same effective batch size of 128 masked trial views per GPU, consistent with Table 3. The comparison evaluates the combined training design.

C.5 Masked prediction objective

Factorized CE sums four cross-entropies over coordinate alphabets of sizes 9, 7, 5 and 5. Joint CE replaces these heads with a single 1,575-class head over complete codes. Both objectives use the same frozen FSQ targets.

C.6 Tokenizer target comparison

We compare scalar, continuous and vector-quantized targets to test how target construction affects transfer (Figure A3). FSQ (Mentzer et al., 2024) and iFSQ (Lin et al., 2026) use four scalar grids with levels $[9, 7, 5, 5]$ and 1,575 possible codes. FSQ bounds coordinates with $\tanh(x)$; iFSQ replaces this map with $2\sigma(1.6x) - 1$, where σ is the logistic sigmoid, before rounding on the same grids. Both use coordinate-wise categorical prediction.

Our continuous autoencoder (AE) follows the reconstruction-based bottleneck principle of Hinton & Salakhutdinov (2006). It produces an unquantized four-dimensional latent, which the masked encoder predicts with MSE after LayerNorm (Ba et al., 2016). Our VQ-VAE variant follows the learned vector-quantization formulation of van den Oord et al. (2017), using a dictionary of 1,575 vectors of dimension 64 and discrete code prediction. The latent dimensions and prediction objectives are specific to these experiments. The comparison concerns the complete target-and-prediction configurations.

C.7 Masking intensity

We vary masking intensity by changing the number of blocks. Nominal ratios of 0.4, 0.5 and 0.6 use 10 short plus 4 long, 15 short plus 5 long, and 18 short plus 6 long blocks, respectively, with span ranges of 1 to 3 and 4 to 6 patches. Where feasible, at least one unmasked patch separates adjacent blocks.

C.8 Code-space size

The level sets $[9, 7, 5]$, $[9, 7, 5, 5]$ and $[9, 7, 7, 5, 5]$ define 315, 1,575 and 11,025 codes, respectively. These configurations use three, four and five coordinates. All variants use scalar quantization and coordinate-wise prediction.

C.9 Extent of downstream fine-tuning

All variants train the classifier. Top 4 updates layers 9 to 12, Top 8 updates layers 5 to 12, and Top 12 updates all 12 encoder layers; each keeps the remaining layers and patch embeddings frozen. All + embedding updates the full encoder and the patch embeddings.

D SUPPLEMENTARY EXPERIMENTAL RESULTS

D.1 BASELINE COMPARISONS ON ADDITIONAL COHORTS

We extend the baseline comparison to AD, Detox, epilepsy and migraine in Tables 5 and 6. The subject-level evaluation protocol is the same as for MCI and PD. StimEM-L outperforms every non-StimEM baseline across the reported metrics on these cohorts. The advantage is therefore not confined to the tasks presented in the main text and supports the use of the pretrained representation across the evaluated clinical cohorts.

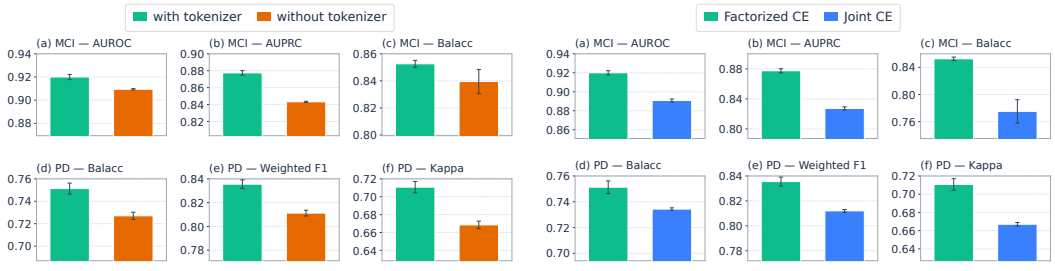
Table 5: Baseline comparison on AD and Detox. Values are means with variability terms defined in the evaluation protocol. **Bold** marks the highest mean in each metric column across all listed models; ties are all bold.

	AD			Detox		
Model	AUROC	AUPRC	BalAcc	AUROC	AUPRC	BalAcc
TCN	0.9346 ± 0.00286	0.8882 ± 0.00438	0.7374 ± 0.00760	0.8256 ± 0.02354	0.7647 ± 0.02072	0.7593 ± 0.02426
TimesNet	0.9432 ± 0.00204	0.9104 ± 0.00352	0.7593 ± 0.01244	0.8704 ± 0.00404	0.8536 ± 0.00346	0.7222 ± 0.02500
NST	0.9310 ± 0.00310	0.8977 ± 0.00358	0.7374 ± 0.00648	0.9444 ± 0.00092	0.8754 ± 0.00460	0.8704 ± 0.00578
CNNTransformer	0.9159 ± 0.00466	0.8747 ± 0.00626	0.7188 ± 0.01082	0.8951 ± 0.00326	0.8606 ± 0.00858	0.8194 ± 0.00734
InceptionTime	0.9425 ± 0.00050	0.9109 ± 0.00114	0.7698 ± 0.00656	0.7068 ± 0.00914	0.6640 ± 0.01934	0.5046 ± 0.00160
ConvTran	0.9109 ± 0.00238	0.8664 ± 0.00506	0.7500 ± 0.00000	0.9012 ± 0.00510	0.8564 ± 0.00194	0.7222 ± 0.03094
StimEM-S	0.9490 ± 0.00296	0.9367 ± 0.00372	0.8567 ± 0.01128	0.9367 ± 0.00174	0.8887 ± 0.00314	0.8796 ± 0.00346
StimEM-M	0.9770 ± 0.00234	0.9660 ± 0.00348	0.8779 ± 0.00270	0.9522 ± 0.00244	0.9298 ± 0.00486	0.8981 ± 0.00692
StimEM-L	0.9770 ± 0.00212	0.9651 ± 0.00244	0.8987 ± 0.01268	**0.9676** ± 0.00130	**0.9539** ± 0.00246	0.8935 ± 0.00130
StimEM-XL	**0.9885** ± 0.00108	**0.9799** ± 0.00180	**0.9217** ± 0.00180	0.9583 ± 0.00076	0.9368 ± 0.00116	**0.9259** ± 0.00524

Table 6: Baseline comparison on Epilepsy and Migraine.

	Epilepsy			Migraine		
Model	AUROC	AUPRC	BalAcc	AUROC	AUPRC	BalAcc
TCN	0.8547 ± 0.00134	0.8427 ± 0.00212	0.7687 ± 0.00236	0.6096 ± 0.01150	0.5209 ± 0.01116	0.5583 ± 0.01156
TimesNet	0.8167 ± 0.00140	0.7798 ± 0.00164	0.7499 ± 0.00376	0.6678 ± 0.02472	0.6147 ± 0.02068	0.5828 ± 0.00172
NST	0.8209 ± 0.00070	0.7721 ± 0.00124	0.7536 ± 0.00232	0.8706 ± 0.00806	0.8548 ± 0.00762	0.7477 ± 0.01456
CNNTransformer	0.8216 ± 0.00064	0.7953 ± 0.00250	0.7578 ± 0.00194	0.7564 ± 0.00398	0.7514 ± 0.00478	0.6958 ± 0.00842
InceptionTime	0.8424 ± 0.00308	0.8055 ± 0.00260	0.7594 ± 0.00336	0.5618 ± 0.01140	0.5306 ± 0.00024	0.5105 ± 0.01410
ConvTran	0.8153 ± 0.00178	0.7839 ± 0.00278	0.7478 ± 0.00510	0.5594 ± 0.02668	0.5355 ± 0.01364	0.5519 ± 0.01796
StimEM-S	0.8539 ± 0.00210	0.8193 ± 0.00100	0.7822 ± 0.00204	0.8683 ± 0.00476	0.8001 ± 0.00448	0.7267 ± 0.01024
StimEM-M	0.8854 ± 0.00118	0.8507 ± 0.00092	0.8017 ± 0.00302	0.9079 ± 0.00348	0.8776 ± 0.00028	0.8007 ± 0.01088
StimEM-L	0.8931 ± 0.00062	0.8562 ± 0.00256	0.8191 ± 0.00474	**0.9394** ± 0.00270	0.9020 ± 0.00448	0.8339 ± 0.01130
StimEM-XL	**0.8941** ± 0.00052	**0.8596** ± 0.00056	**0.8386** ± 0.00164	0.9301 ± 0.00262	**0.9045** ± 0.00290	**0.8543** ± 0.00456

D.2 TOKENIZER-BASED TARGETS VERSUS DIRECT PREDICTION



(a) With and without tokenizer **(b) Prediction objectives**

Figure A2: Supplementary pretraining comparisons on MCI and PD. **(a)** With tokenizer versus without tokenizer. **(b)** Factorized coordinate prediction versus joint-code prediction.

The tokenizer-based variant achieves higher mean performance on every reported metric (Figure A2(a)). Both variants use waveform and behavioral-feature supervision: one uses it to train a separate tokenizer, and the other applies it directly during masked pretraining. The results favor the tokenizer-based formulation as a whole; both the prediction targets and the pretraining loss differ. Experimental settings are detailed in Appendix B.7.

D.3 Masked prediction objective

Factorized prediction outperforms joint-code prediction across the reported metrics (Figure A2(b)). With factorized CE, codes that share a coordinate value contribute to learning the same category. Joint CE instead treats each complete combination as a separate class. Sharing supervision across coordinate values may make better use of the structure imposed by FSQ. The encoder can still combine stimulus and ocular information before the prediction heads, while each head solves a smaller categorical problem. These results favor a prediction objective that follows the tokenizer's coordinate-wise construction. Experimental settings are detailed in Appendix C.5.

D.4 Tokenizer target comparison

FSQ gives the best results across the reported metrics. The other methods show a task-dependent ordering: continuous AE exceeds VQ-VAE on MCI, whereas VQ-VAE performs better on PD. Thus, replacing continuous targets with discrete ones is not sufficient on its own to improve transfer.

Because latent dimensions and prediction losses also differ, these results compare the complete target-and-prediction formulations. The comparison does not isolate quantization alone.

FSQ also outperforms iFSQ across the reported metrics. iFSQ was proposed to improve quantization-bin utilization through a distribution-matching bounding function in image generation (Lin et al., 2026). Here, the downstream results favor the original FSQ mapping for ocular targets. Establishing whether this difference arises from code usage or response separability would require a separate analysis. Experimental settings are detailed in Appendix C.6.

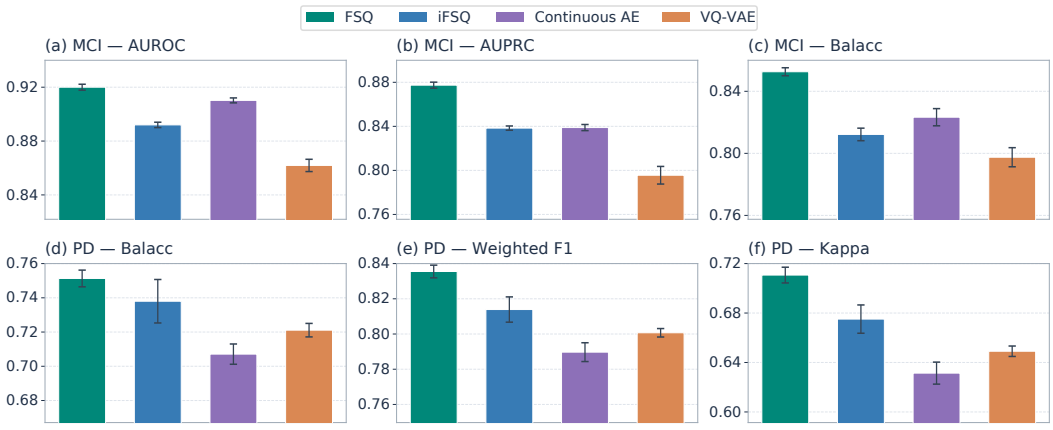


Figure A3: Comparison of FSQ, iFSQ, continuous AE and VQ-VAE on MCI and PD.

D.5 Masking intensity

The intermediate masking ratio gives the best results on both tasks (Table 7). With lighter masking, nearby observations may make local interpolation sufficient for many predictions. Heavier masking discourages this shortcut but can remove the context needed to recover the hidden response. The intermediate setting appears to make prediction demanding without depriving the encoder of useful evidence. We use it as the reference, although realized coverage varies with the valid eye positions in each trial. Experimental settings are detailed in Appendix C.7.

Table 7: Masking-intensity ablation. Variability terms follow the evaluation protocol; bold marks the highest mean in each metric.

	MCI			PD		
Configuration	AUROC	AUPRC	BA	BA	Weighted F1	Kappa
0.4	0.9016 ± 0.00320	0.8527 ± 0.00396	0.8492 ± 0.00510	0.7216 ± 0.00664	0.8046 ± 0.00384	0.6585 ± 0.00564
0.5	**0.9200** ± 0.00220	**0.8774** ± 0.00272	**0.8526** ± 0.00258	**0.7513** ± 0.00488	**0.8355** ± 0.00360	**0.7107** ± 0.00636
0.6	0.9090 ± 0.00102	0.8629 ± 0.00194	0.8394 ± 0.00508	0.7090 ± 0.00992	0.7962 ± 0.00550	0.6462 ± 0.00800

D.6 Code-space size

The intermediate configuration outperforms both the smaller and larger code spaces (Table 8). A small code space may merge responses that carry different behavioral information. Enlarging it allows finer distinctions, but may also preserve variation that is difficult to predict or unhelpful for classification. This pattern suggests that finer quantization does not necessarily produce better pretraining targets. We retain the four-coordinate configuration, which gives the strongest results on both cohorts. The comparison changes coordinate dimension as well as code count, so it evaluates their combined effect. Experimental settings are detailed in Appendix C.8.

Table 8: FSQ level configurations and code-space sizes. Variability terms follow the evaluation protocol; bold marks the highest mean in each metric.

	MCI			PD		
Configuration	AUROC	AUPRC	BA	BA	Weighted F1	Kappa
$[9, 7, 5]$ (315)	0.8987 ± 0.00140	0.8451 ± 0.00250	0.8307 ± 0.00128	0.7455 ± 0.00266	0.8226 ± 0.00210	0.6873 ± 0.00398
$[9, 7, 5, 5]$ (1,575)	**0.9200** ± 0.00220	**0.8774** ± 0.00272	**0.8526** ± 0.00258	**0.7513** ± 0.00488	**0.8355** ± 0.00360	**0.7107** ± 0.00636
$[9, 7, 7, 5, 5]$ (11,025)	0.8937 ± 0.00446	0.8314 ± 0.00764	0.8162 ± 0.00410	0.7471 ± 0.00118	0.8166 ± 0.00098	0.6749 ± 0.00174

D.7 Extent of downstream fine-tuning

Table 9: Extent of encoder and patch-embedding fine-tuning for StimEM-L. Variability terms follow the evaluation protocol; bold marks the highest mean in each metric.

	MCI			PD		
Configuration	AUROC	AUPRC	BA	BA	Weighted F1	Kappa
Top 4	0.8827 ± 0.00124	0.8314 ± 0.00246	0.7716 ± 0.00262	0.6981 ± 0.00442	0.7979 ± 0.00268	0.6481 ± 0.00408
Top 8	0.9020 ± 0.00128	0.8618 ± 0.00232	0.8146 ± 0.00708	0.7190 ± 0.00868	0.8129 ± 0.00540	0.6751 ± 0.00850
Top 12	0.9106 ± 0.00112	0.8646 ± 0.00116	0.8442 ± 0.00104	0.7325 ± 0.00688	0.8184 ± 0.00384	0.6854 ± 0.00526
All + embedding	**0.9200** ± 0.00220	**0.8774** ± 0.00272	**0.8526** ± 0.00258	**0.7513** ± 0.00488	**0.8355** ± 0.00360	**0.7107** ± 0.00636

Performance improves as more of the pretrained encoder is fine-tuned (Table 9). Adapting the full encoder and patch embeddings gives the strongest results. The pretrained representation is therefore most effective when it can be adjusted to the downstream task.

This difference is consistent with the change in learning objective. Pretraining asks the model to infer missing response structure, whereas clinical supervision asks it to emphasize information that separates subjects. Updating only the upper layers limits how much the earlier signal features can adapt to this new objective. Full fine-tuning allows both those features and their temporal integration to change, and updating the patch embeddings extends adaptation to the input representation. Together with Figure 3, the results support using pretraining for initialization followed by end-to-end fine-tuning. These results motivate further exploration of parameter-efficient fine-tuning to achieve comparable performance with fewer trainable parameters. Experimental settings are detailed in Appendix C.9.

D.8 PRETRAINING LEARNING CURVES

Figure A4 compares learning curves for the different encoder sizes. All models use the full pretraining dataset, the same frozen tokenizer, paired dual-scale masking and factorized cross-entropy. We report prediction loss, masked-code accuracy and factorized joint perplexity (PPL), allowing the models to be compared on a shared target space.

Larger encoders attain lower loss and PPL and higher masked-code accuracy. The curves improve rapidly at first and more gradually later in training. Because the targets are fixed, the differences reflect how well each encoder infers missing responses from the available context. This pattern suggests that greater capacity helps model the dependencies needed for masked prediction. PPL expresses the loss on an exponential scale, while accuracy tracks whether the most likely code matches the target. The downstream experiments in Section 4.4 assess how these pretraining gains translate into clinical classification performance.

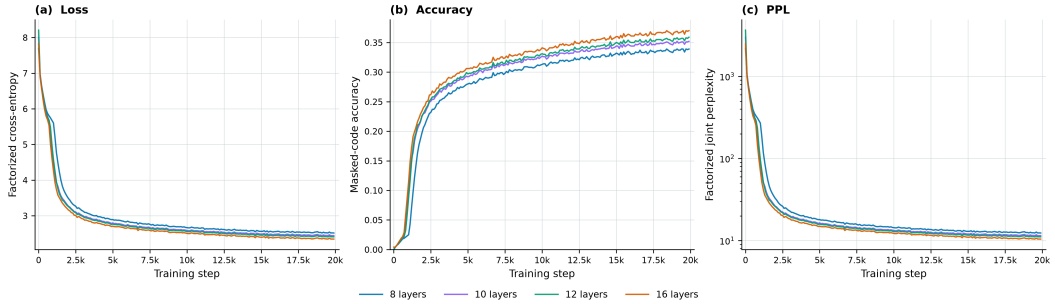


Figure A4: Training dynamics for the four StimEM configurations using the same frozen tokenizer targets: (a) factorized cross-entropy, (b) masked-code accuracy, and (c) factorized joint perplexity, computed as the exponential of the loss and displayed on a logarithmic axis. The legend gives encoder depth: 8, 10, 12 and 16 layers correspond to StimEM-S, StimEM-M, StimEM-L and StimEM-XL, respectively.

D.9 QUALITATIVE TOKENIZER RECONSTRUCTION

Figure A5 shows waveform and behavioral-feature reconstruction for a healthy subject performing an antisaccade trial. Columns show the two eyes, and rows show horizontal gaze, vertical gaze and pupil area on a shared frame axis. Gaze is measured in degrees; pupil area and continuous task features use robust normalization. The lower panels separate these features from binary labels and predicted probabilities. Waveform traces omit missing and blink frames, and feature panels include the available non-count targets.

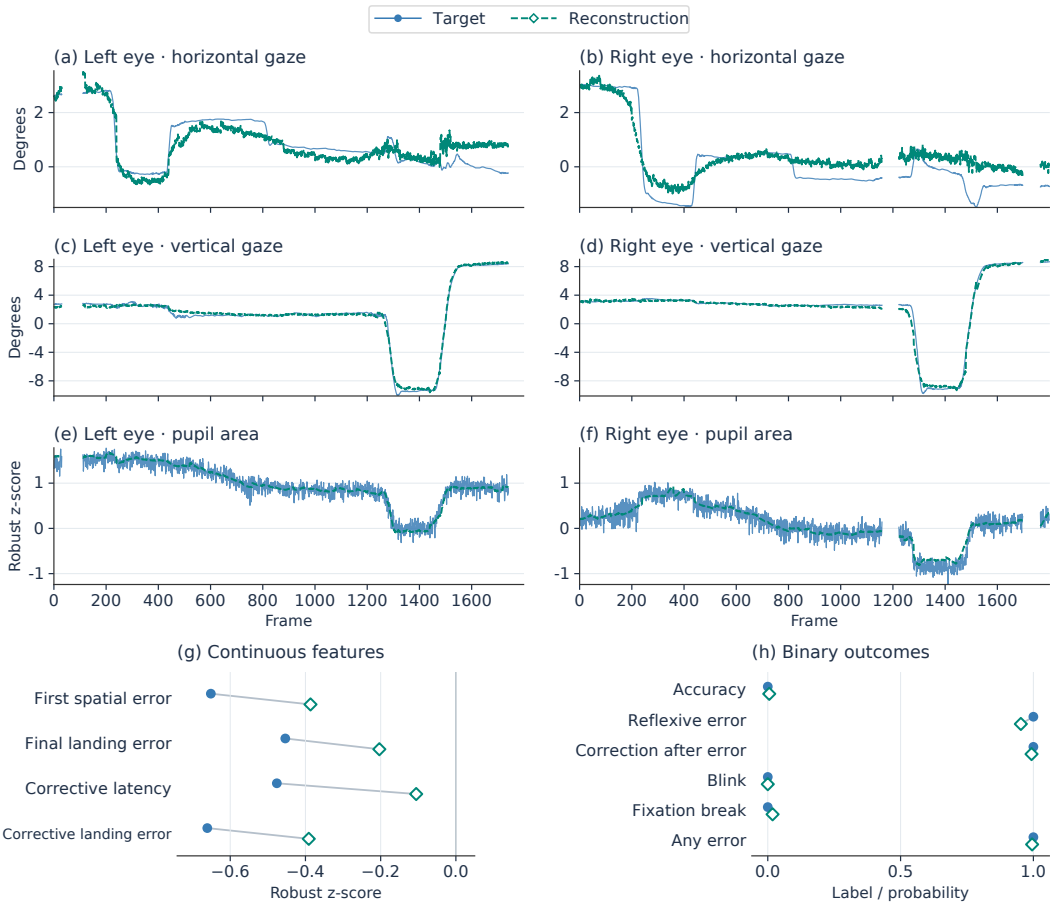


Figure A5: Waveform and task-feature reconstruction for a healthy subject performing an antisaccade trial. Solid blue traces and filled circles denote targets; dashed teal traces and open diamonds denote reconstructions. The two eye columns share frame and row-specific value axes. Continuous features and binary outcomes use separate scales.

The reconstructed waveforms follow the main gaze transitions in both eyes and the slower changes in pupil area. The vertical traces also reproduce the large late excursion and subsequent upward movement. Pupil traces are smoother than the targets but retain their broad temporal pattern. This example suggests that the compressed codes preserve both rapid gaze changes and more sustained ocular dynamics. Supervising the channels together retains these different aspects of the same response.

The behavioral predictions provide a complementary view of the trial. The subject first makes a reflexive error and then corrects it, and the predicted probabilities agree with those labels. The continuous predictions also preserve the direction of the normalized spatial-error and corrective-latency targets. A healthy control can make an error on an individual trial, so clinical status and task execution should not be conflated. Here, joint supervision connects the reconstructed trajectory with the behavioral sequence it represents.