SpellerSSL: Self-Supervised Learning with P300 Aggregation for Speller BCIs

Jiazhen Hong*
Emotiv Research
Melbourne, Australia
jiazhen@emotiv.com

Geoffrey Mackellar
Emotiv Research
Sydney, Australia
geoff@emotiv.com

Soheila Ghane
Emotiv Research
Melbourne, Australia
soheila@emotiv.com

Abstract

Electroencephalogram (EEG)-based P300 speller brain–computer interfaces (BCIs) face three main challenges: low signal-to-noise ratio (SNR), poor generalization, and time-consuming calibration. We propose *SpellerSSL*, a framework that integrates self-supervised learning (SSL) with P300 aggregation to address these issues. First, we introduce an aggregation strategy to enhance SNR. Second, a customized 1D U-Net backbone is pretrained on cross- and in-domain EEG to improve generalization, and then fine-tuned with a lightweight *ERP-Head* classifier for subject-specific P300 detection. Our evaluations on calibration time demonstrate that combining the aggregation strategy with SSL significantly reduces the calibration burden per subject and improves robustness across subjects. Experimental results show that SSL learns effective EEG representations in both in-domain and cross-domain, with in-domain achieving a state-of-the-art character recognition rate of 94% with only 7 repetitions and the highest information transfer rate (ITR) of 21.86 bits/min on the public II-B dataset. Moreover, in-domain SSL with P300 aggregation reduces the required calibration size by 60% while maintaining a comparable character recognition rate. To our knowledge, this is the first study to apply SSL to P300 spellers, highlighting its potential to improve both efficiency and generalization in speller BCIs and paving the way toward an EEG foundation model. The code is available at: https://github.com/Emotiv/SpellerSSL-NeurIPS2025

1 Introduction

Electroencephalogram (EEG)-based Brain–Computer Interfaces (BCIs), due to their non-invasive nature, low cost, and ease of use, have been widely adopted as assistive technologies for individuals with motor disabilities [1]. Among various paradigms, the P300 is an event-related potential (ERP) component that typically occurs around 300 ms after a target stimulus in an oddball paradigm [2]. It has been widely applied in speller BCI, known as the P300 Speller (visualized in Figure 1(a)).

Traditional P300 speller BCIs face three main challenges: (1) **Low signal-to-noise ratio (SNR):** The low SNR of EEG signals makes reliable single-trial P300 detection difficult [3]. Current systems often require multiple repetitions of each stimulus to achieve an accurate character recognition rate (CRR), which slows down spelling speed and reduces the information transfer rate (ITR) [4]. (2) **Poor generalization:** Existing methods used in P300 spellers include conventional machine learning algorithms such as stepwise linear discriminant analysis (SWLDA) [5] and support vector machines (SVMs) [6], as well as deep learning architectures such as convolutional neural networks (CNNs) [7], CapsNet with attention [8, 4], stacked Transformers [3], and image-based representations such as TimeSformer [9]. However, these approaches are task- or subject-specific and fail to provide

*Work done during internship at Emotiv Research.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Data on the Brain & Mind.

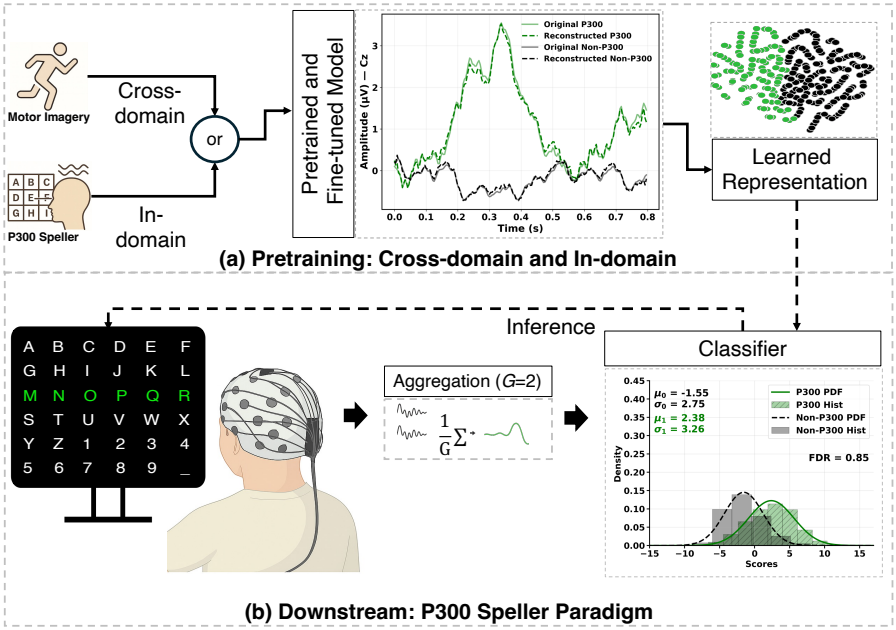


Figure 1: Overview of the proposed SpellerSSL.

generalizable representations of the P300 response. (3) **Time-consuming calibration:** Large inter-individual differences require a time-consuming and subject-specific calibration session before the speller can be used in real-time and online [10, 11].

In brief, the three main questions are: (1) how to obtain higher signal quality, (2) how to build more generalizable models, and (3) how to reduce calibration requirements.

Recent advances in self-supervised learning (SSL) and foundation models offer promising directions for addressing these challenges. In this work, we propose *SpellerSSL*, a P300 speller framework that integrates SSL with P300 aggregation strategies. Our contributions are summarized as follows:

- We introduce a P300 aggregation strategy that enhances signal quality in the training set, enabling the model to learn more reliable single-trial responses and achieve higher character recognition rates (CRRs) with fewer repetitions.
- We customize a 1D U-Net [12] as the backbone SSL model and show that an SSL model pretrained on both cross-domain and in-domain EEG data can enhance the generalization of learned representations. We also design a lightweight *ERP-Head* classifier for downstream detection of P300, which allows subject-specific adaptation.
- We evaluate calibration efficiency under reduced training set sizes and demonstrate that combining the aggregation strategy with a pretrained model enhances robustness, reducing calibration time without compromising accuracy in P300 speller BCIs.

To the best of our knowledge, this is the first work to explore SSL in P300 speller BCIs. We demonstrate that the proposed SpellerSSL improves both efficiency and generalization, paving the way for the future deployment of foundation models in practical P300 speller systems.

2 Method

Figure 1 provides an overview of SpellerSSL. In (a), self-supervised pretraining is performed with a full 1D U-Net backbone via a reconstruction task on either cross-domain or in-domain EEG (Section 2.1). In (b), downstream P300 detection is performed, where the learned representations at the U-Net bottleneck layer are fine-tuned with an ERP-Head classifier (Section 2.2). A P300 aggregation step, averaging G consecutive repetitions, is applied during fine-tuning to enhance the quality of single-trial signals (Section 2.3).

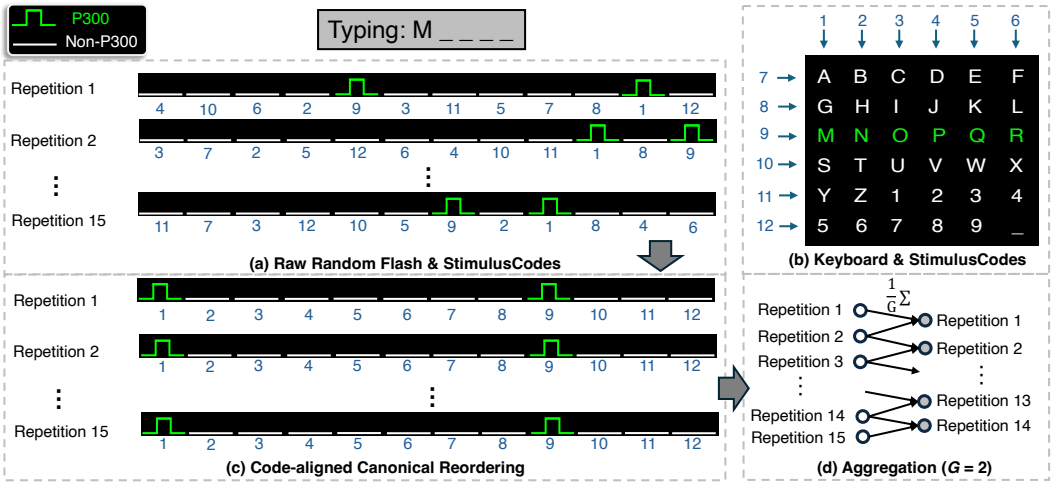


Figure 2: P300 aggregation for typing the character "M".

2.1 Pretraining

We customized 1D U-Net architecture (Appendix A.1) such that input sequences are zero-padded in time so that output length is a multiple of 16, matching the encoder's four successive downsamplings (each with stride 2). The model is trained for 200 epochs with a batch size of 64, using AdamW (initial learning rate 2.5×10^{-4}, weight decay 1×10^{-2}) and a OneCycle schedule [13] (maximum learning rate 5×10^{-4}, final learning rate 5×10^{-6}, 10% warmup, cosine decay). We optimize the model to reconstruct randomly masked EEG sequences (50% time masking, 0% channel masking).

Let the input EEG trial be $x \in \mathbb{R}^{C \times L}$. The masked input $\tilde{x}$ is passed through the model $f_\theta(\cdot)$ to obtain the reconstruction $\hat{x} = f_\theta(\tilde{x})$. We adopt a time–frequency consistency objective:

$$\mathcal{L}_{\text{SSL}} = \|x - \hat{x}\|_1 \; + \; \lambda \left\| \text{FFT}(x) - \text{FFT}(\hat{x}) \right\|_1, \tag{1}$$

where $\text{FFT}(\cdot)$ denotes the fast Fourier transform. The frequency-domain term encourages the encoder to capture spectral characteristics of EEG signals, complementing the time-domain reconstruction.

2.2 Downstream

After training, we extract the encoder's bottleneck feature map before upsampling, $b \in \mathbb{R}^{1024 \times \frac{L}{16}}$, and discard the decoder. The ERP-Head $\text{Head}_\phi(\cdot)$ operates directly on the sequence b and applies the following transformations:

$$\tilde{b} = \text{PointwiseConv}(b) \;\rightarrow\; \text{DepthwiseConv} \;\rightarrow\; \text{DilatedDepthwiseConv} \;\rightarrow\; \text{PointwiseConv}, \tag{2}$$

$$h = \text{GAP}(\tilde{b}) \in \mathbb{R}^D, \qquad z = Wh + c \in \mathbb{R}^2, \tag{3}$$

where PointwiseConv denotes a 1×1 projection and DepthwiseConv operates on temporal dimension.

We minimize the supervised cross-entropy loss

$$\mathcal{L}_{\text{sup}} = \text{CE}(z, y), \tag{4}$$

where $\mathcal{L}_{\text{sup}}$ denotes the supervised loss, and CE is the standard cross-entropy between predicted logits z and ground-truth labels y. A fixed positive vs. negative class weight ratio of $1:5$ is used to address class imbalance. When an SSL checkpoint is available, encoder-only weights up to the bottleneck are loaded, and the model is fine-tuned for 10 epochs on the training set. The detailed architecture of the ERP-Head is provided in Appendix A.2.

2.3 P300 Aggregation

Figure 2 summarizes the proposed aggregation pipeline for typing the character "M". Panel (a) shows the raw random flashes with their *StimulusCodes* [14] over R=15 repetitions; green segments

denote P300 (row/column hits), while white segments denote Non-P300 responses. Panel (b) illustrates the 6×6 speller with its code assignment; for "M", the target row/column are $(\text{row } 3,\ \text{col } 1)$, corresponding to codes $(9, 1)$. Panel (c) performs code-aligned canonical reordering within each repetition; the 12 trials are permuted to a fixed order k=1..12 by aligning to the 12 StimulusCodes. Panel (d) applies a sliding-window aggregation with group size G=2 (with overlap), where every two consecutive repetitions are averaged to form aggregated trials.

Notation. Each character selection is presented over R repetitions (here R=15), with 12 row/column flashes per repetition. Let $x_k^{(r)} \in \mathbb{R}^{C\times L}$ denote the EEG trial in repetition $r \in \{1, \ldots, R\}$ corresponding to the flash with canonical code index $k \in \{1, \ldots, 12\}$ (to be aligned in Step 1). Let $s_j^{(r)} \in \{1, \ldots, 12\}$ denote the StimulusCode of the j-th flash in temporal order within repetition r. For the typed character, define the P300 code set

$$\mathcal{K}_{\text{P300}} = \{k_{\text{row}},\, k_{\text{col}}\}, \quad k_{\text{row}} \in \{7, \ldots, 12\},\ k_{\text{col}} \in \{1, \ldots, 6\}. \tag{5}$$

which contains the row and column codes of that character (e.g., "M": $\mathcal{K}_{\text{P300}} = \{9, 1\}$). The per-code label is defined as

$$y[k] \;=\; \begin{cases} 1, & k \in \mathcal{K}_{\text{P300}} \quad \text{(P300)} \\ 0, & \text{otherwise} \quad \text{(Non-P300)} \end{cases} \qquad \text{for } k = 1, \ldots, 12, \tag{6}$$

so that each repetition contains exactly two P300 and ten Non-P300 flashes.

Step 1: Code-aligned canonical reordering. Within each repetition, we permute the 12 trials to a canonical order k=1..12 using the StimulusCode. Let π_r map the canonical code index k to its temporal position in repetition r, i.e.

$$s^{(r)}_{\pi_r(k)} \;=\; k', \qquad k' = 1, \ldots, 12. \tag{7}$$

We then form the ordered tensor

$$X^{(r)}_{\text{ord}}[k'] \;=\; x^{(r)}_{\pi_r(k)} \;\in\; \mathbb{R}^{C\times L}, \qquad X^{(r)}_{\text{ord}} \;\in\; \mathbb{R}^{12\times C\times L}, \tag{8}$$

where the subscript "ord" denotes canonical ordering by StimulusCodes.

Step 2: Sliding-window aggregation (codes retained). We aggregate G consecutive repetitions with a sliding window (overlap allowed), yielding windows indexed by $r = 1, \ldots, R - G + 1$. For each window and canonical code k, we average trials with the same code k across the G repetitions:

$$\bar{X}^{(r,G)}_{\text{ord}}[k'] \;=\; \frac{1}{G} \sum_{t=r}^{r+G-1} X^{(t)}_{\text{ord}}[k'], \qquad k' = 1, \ldots, 12, \tag{9}$$

so the aggregated window still contains 12 trials. The labels are preserved per code,

$$y^{(r,G)}[k'] \;\equiv\; y[k'] \;=\; \begin{cases} 1, & k' \in \mathcal{K}_{\text{P300}} \quad \text{(P300)} \\ 0, & \text{otherwise} \quad \text{(Non-P300)} \end{cases} \tag{10}$$

which preserves the original speller paradigm and maintains the two-P300 vs. ten-Non-P300 pattern in every window. Aggregation is applied only to calibration (training) data and is not performed during real-time or online operation.

3 Experiment

3.1 Dataset & Pre-processing

We use two publicly available datasets: PhysionetMI [15, 16] (motor imagery) and BCI Competition III-II [14] (P300 speller). PhysionetMI, accessed via the MOABB pipeline [17], serves as the cross-domain pretraining set. BCI Competition III-II is band-pass filtered at 0.1–60 Hz and consists of two subjects, generally split into two datasets, II-A and II-B [4]. We use II-A for in-domain pretraining and II-B for downstream evaluation.

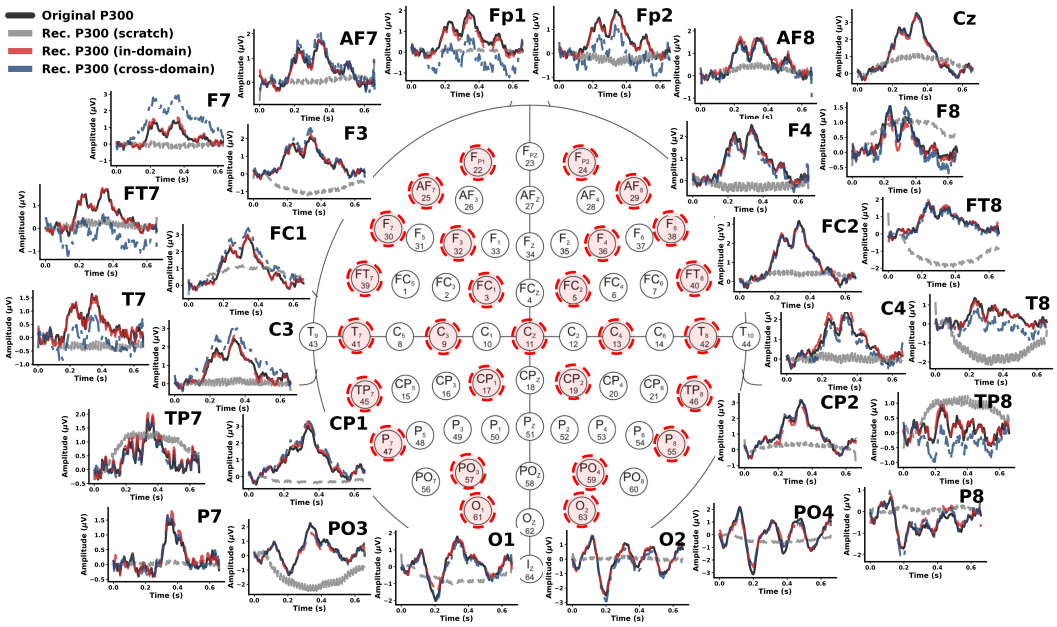


Figure 3: Reconstruction performance across 64 EEG channels. The montage layout is shown in the center, with representative channels displayed around it. Each plot compares the original P300 response (black) with reconstructions from models trained from scratch (gray), with in-domain pretraining (red), and with cross-domain pretraining (blue). More visualizations in Appendix C.

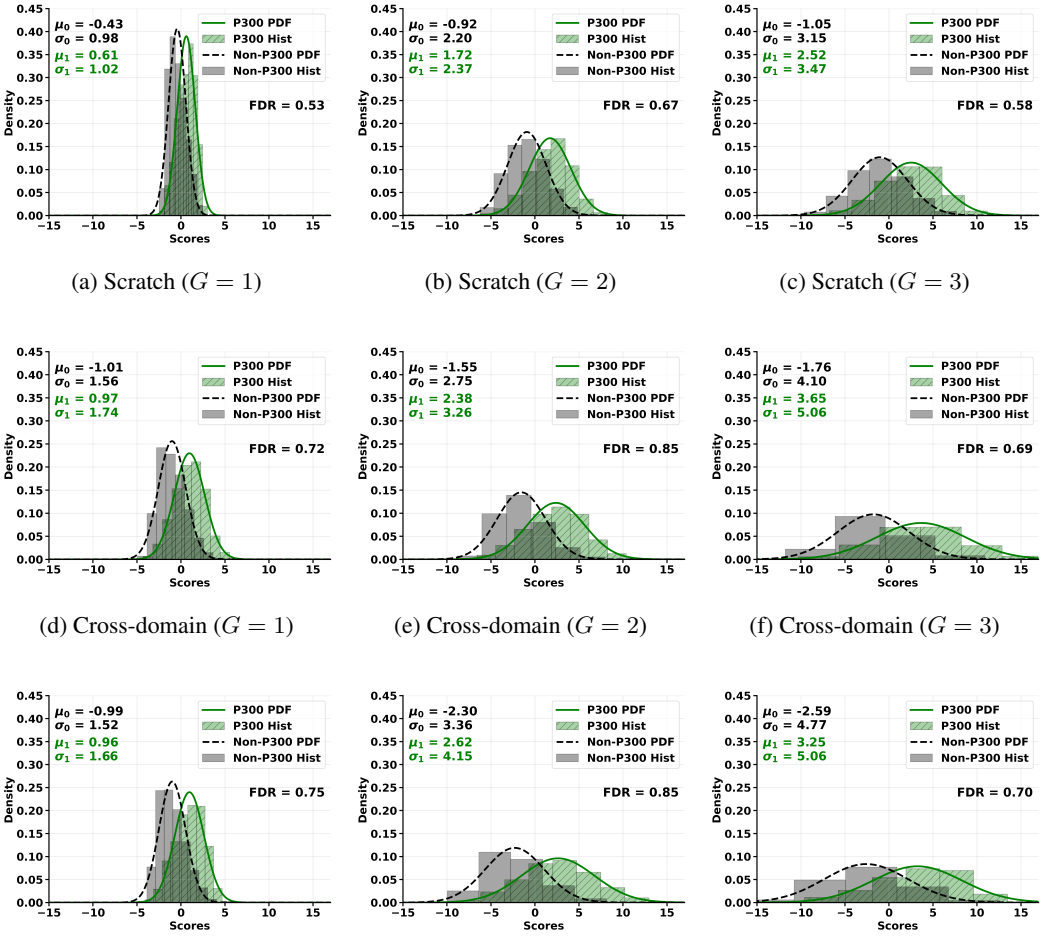
All datasets follow the 10–20 system with 64 channels in the same order. PhysionetMI provides 3-second trials at 160 Hz, while II-A/II-B provide 1-second trials at 240 Hz. To standardize, we resample all signals to 240 Hz and extract 667 ms post-stimulus segments (160 samples) per trial. PhysionetMI contains 69,760 trials across five classes, all used for cross-domain pretraining. II-A and II-B each contain 33,000 trials: in II-A, 15,300 (85 characters $\times$ 15 repetitions $\times$ 12 stimuli) form the calibration set and 18,000 (100 characters $\times$ 15 repetitions $\times$ 12 stimuli) form the testing set; all are used for in-domain pretraining. II-B is used for downstream tasks, with 85 calibration characters for fine-tuning and 100 testing characters for evaluation.

3.2 Experimental Setup

We evaluate three approaches: (1) *Scratch*: Random initialization and supervised training on the calibration set of single-subject P300 data (II-B), serving as the baseline. (2) *Cross-domain*: SSL pretraining on motor imagery EEG (PhysionetMI), followed by fine-tuning on the calibration set of single-subject P300 data (II-B). (3) *In-domain*: SSL pretraining on in-domain P300 EEG (II-A), followed by fine-tuning on the calibration set of a different subject's P300 data (II-B). P300 aggregation ($G \in \{1, 2, 3\}$) is applied only to calibration data (Sec. 2.3). Evaluation is on the independant II-B test set. The evaluation metrics include Character Recognition Rate (CRR), Accuracy, Binary F1-Score, Fisher's Discriminant Ratio (FDR), and Information Transfer Rate (ITR), detailed in Appendix B. All experiments are conducted using 32-bit mixed precision on a single NVIDIA RTX 6000 Ada GPU.

3.3 Reconstruction Results

Figure 3 shows reconstructed EEG signals under the three approaches. The ground-truth P300 responses (black) exhibit positive deflections at many frontal and central electrodes (e.g., Fp1, Fp2, Cz), while occipital sites such as O1 and O2 often show negative polarity, likely influenced by the reference montage. The model trained from scratch tends to yield noisy reconstructions, whereas in-domain pretraining yields signals closely aligned with the original waveforms. Interestingly, cross-domain pretraining on motor imagery data also recovers P300 morphology. For instance, at Cz and AF8 both in-domain (red) and cross-domain (blue) reconstructions match the ground truth, while at F7 and Fp1 the cross-domain model captures the trend but underestimates the amplitude.



(a) Scratch ($G = 1$) (b) Scratch ($G = 2$) (c) Scratch ($G = 3$)

(d) Cross-domain ($G = 1$) (e) Cross-domain ($G = 2$) (f) Cross-domain ($G = 3$)

(g) In-domain ($G = 1$) (h) In-domain ($G = 2$) (i) In-domain ($G = 3$)

Figure 4: Distributions of decision scores (log-odds) under three pretraining conditions: (a–c) training from scratch, (d–f) cross-domain pretraining, and (g–i) in-domain pretraining. For each condition, we evaluate three aggregation levels $G \in \{1, 2, 3\}$. Green curves and bars show P300 probability density function (PDF) and histogram (Hist), while black dashed curves and gray bars show Non-P300. The Fisher's Discriminant Ratio (FDR; higher is better) is reported.

To quantify these observations, we compute the mean squared error (MSE) averaged across all channels and trials. The scratch model yields an MSE of 161.38, compared to 3.26 for in-domain pretraining and 7.26 for cross-domain pretraining. These results demonstrate that self-supervised pretraining substantially improves EEG reconstruction quality, and that cross-domain pretraining can still learn transferable representations beneficial for downstream P300 tasks.

3.4 Downstream Results

Figure 4 shows decision score distributions for P300 vs. Non-P300 responses under different pretraining and aggregation settings. Without aggregation ($G = 1$), the two distributions are closer and less separable. Aggregation ($G = 2, 3$) generally increases the mean difference between target and non-target responses, thereby improving separability as reflected by higher FDR values. However, when $G = 3$, although the mean separation further increases, the variance of both classes also grows, leading to wider and noisier distributions and a drop in FDR compared to $G = 2$. For example, in scratch training, FDR improves from 0.53 ($G = 1$) to 0.67 ($G = 2$), but drops to 0.58 ($G = 3$). Overall, in-domain pretraining achieves the highest separability, while cross-domain remains competitive, with both reaching an FDR of 0.85 at $G = 2$.

Table 1: Downstream performance under different training and aggregation conditions.

pretraining	# G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Acc.	B-F1	FDR
Scratch	1	21	38	49	55	60	67	74	77	83	83	85	87	87	89	89	68.58	0.4312	0.53
Scratch	2	31	38	56	65	71	81	83	81	87	89	90	95	93	95	96	68.42	0.4481	0.67
Scratch	3	34	43	59	62	69	79	81	84	85	88	88	89	91	89	92	66.26	0.4310	0.58
Cross-domain	1	39	49	56	69	75	81	88	86	90	87	90	93	90	91	92	74.00	0.4751	0.72
Cross-domain	2	**44**	**65**	**74**	**75**	82	87	93	**92**	**95**	95	96	96	96	97	**97**	73.94	0.4937	**0.85**
Cross-domain	3	38	56	64	68	78	81	85	84	90	93	95	95	95	97	**97**	70.57	0.4657	0.69
In-domain	1	36	49	59	70	72	83	87	84	85	87	86	91	91	92	96	74.28	0.4779	0.75
In-domain	2	43	**65**	69	74	**84**	**89**	**94**	91	93	**96**	**98**	**99**	**98**	**99**	**97**	**76.18**	0.**5074**	**0.85**
In-domain	3	40	56	63	70	69	80	87	87	87	91	92	92	92	94	96	71.79	0.4689	0.70

Table 2: Downstream performance under reduced calibration.

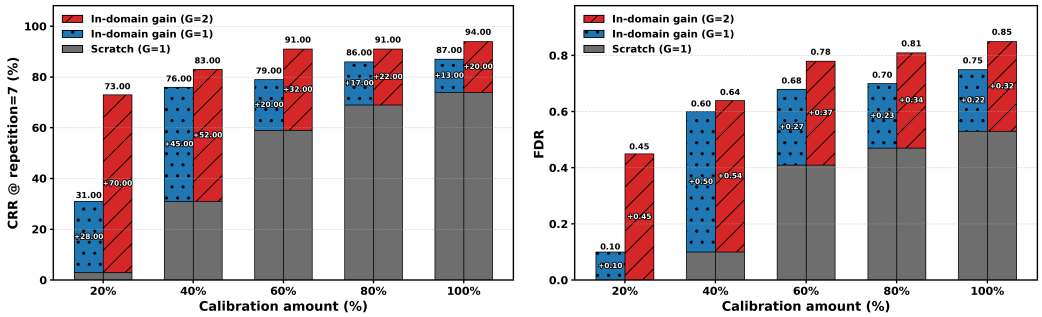
Calibration	# G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Acc.	B-F1	FDR
20% (Scratch)	1	3	2	0	4	1	2	3	3	3	4	6	4	4	5	5	59.68	0.2556	0
20%	1	10	6	17	23	28	28	31	36	38	42	38	36	44	43	46	67.24	0.3321	0.1
20%	2	29	39	45	50	61	68	73	69	70	77	75	82	80	81	81	68.11	0.4294	0.45
40% (Scratch)	1	10	6	17	23	28	28	31	36	38	42	38	36	44	43	46	67.24	0.3321	0.1
40%	1	30	41	49	64	65	74	76	77	82	81	81	89	85	85	86	72.50	0.4581	0.60
40%	2	34	43	51	64	74	76	83	83	86	88	87	88	87	90	92	65.37	0.4340	0.64
60% (Scratch)	1	27	36	50	53	57	64	59	63	66	73	72	77	75	74	77	71.22	0.4265	0.41
60%	1	35	43	50	59	66	73	79	79	83	87	88	91	89	90	91	72.34	0.4633	0.68
60%	2	37	59	72	77	82	87	91	92	93	96	96	97	96	97	98	74.82	0.4921	0.78
80% (Scratch)	1	23	36	45	56	62	64	69	67	69	76	79	84	81	83	85	68.41	0.4292	0.47
80%	1	37	51	56	64	73	81	86	84	88	91	91	91	92	90	92	73.91	0.4754	0.70
80%	2	43	60	76	78	88	87	91	92	91	96	98	96	94	96	97	71.42	0.4761	0.81
100% (Scratch)	1	21	38	49	55	60	67	74	77	83	83	85	87	87	89	89	68.58	0.4312	0.53
100%	1	36	49	59	70	72	83	87	84	85	87	86	91	91	92	96	74.28	0.4779	0.75
100%	2	43	65	69	74	84	89	94	91	93	96	98	99	98	99	97	76.18	0.5074	0.85

Table 1 compares character recognition rates (CRR), single-trial accuracy (Acc., %), binary F1, and FDR under three pretraining conditions across repetitions 1–15. CRRs above 90% are underlined, and the highest values are highlighted in bold. In-domain pretraining with $G = 2$ achieves the best overall performance, reaching 94% CRR at repetition 7 and the highest single-trial accuracy (76.18%) and B-F1 (0.5074). Cross-domain pretraining with $G = 2$ also performs competitively, reaching 93% CRR at repetition 7 and FDR = 0.85, demonstrating the transferability of SpellerSSL from motor imagery to P300 tasks and suggesting that pretrained or future foundation models can benefit P300 spellers. Scratch models improve with aggregation but remain consistently worse. Aggregation also accelerates recognition: with $G = 2$, both in-domain and cross-domain models surpass 90% CRR by repetition 7, whereas with $G = 1$ this threshold is only reached after repetition 11. Without aggregation and pretraining, the scratch model fails to exceed 90% CRR even at repetition 15. Excessive aggregation ($G = 3$) does not yield further gains, as the increased variance lowers FDR and slightly reduces accuracy. Overall, moderate aggregation ($G = 2$) offers the best trade-off, effectively denoising while retaining discriminative ERP features.

3.5 Calibration Reduction Analysis

Table 2 shows the effect of reduced calibration data on downstream performance, reporting results for aggregation $G = 1$ and $G = 2$, together with single-trial accuracy (Acc.), binary F1 (B-F1), and FDR. Scratch ($G = 1$) serves as baseline, and character recognition rates (CRR, %) above 90% are underlined. Without pretraining and aggregation (Scratch, $G = 1$), the model fails to capture discriminative features under limited calibration, yielding FDR ≈ 0 at 20%. In-domain pretraining ($G = 1$) raises FDR to 0.10, while combining pretraining with aggregation ($G = 2$) further boosts FDR to 0.45, enabling CRR to exceed 80% by repetition 12 even with only 20% calibration data. As calibration increases, both CRR and FDR improve across all settings, yet the gap between pretrained and scratch models remains substantial, highlighting the advantage of SpellerSSL in low-calibration conditions. At full calibration (100%), in-domain pretraining with $G = 2$ achieves best performance. More broadly, $G = 2$ consistently provides the best trade-off: for example, at 60% calibration, CRR reaches 91% at repetition 7 with FDR = 0.78, compared to only 0.41 for the scratch baseline.

Figure 5 visualizes the improvements of in-domain pretraining and P300 aggregation over the scratch baseline ($G = 1$ without aggregation). (a) shows CRR at repetition 7, where the earliest time CRR can exceed 90%. (b) presents the corresponding FDR values. Across all calibration levels, SpellerSSL with in-domain pretraining substantially improves early recognition and increases FDR, while combining pretraining with $G = 2$ aggregation further enhances performance. Overall,



(a) Character recognition rate (CRR) at repetition 7.

(b) Fisher's Discriminant Ratio (FDR).

Figure 5: Calibration reduction analysis using in-domain checkpoints. Baseline results from scratch ($G = 1$) are shown in gray, while gains from in-domain pretraining with $G = 1$ and $G = 2$ are shown in blue and red, respectively.

Table 3: Comparison of ITRs (bits/min) between SpellerSSL and state-of-the-art models.

Model	Ref.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ave.
ST-CapsNet	[4]	15.22	19.74	17.05	**18.06**	**17.5**	15.37	15.2	13.02	11.98	11.81	11.07	10.23	9.32	9.07	8.33	13.53
ERP-CapsNet	[8]	17.69	19.23	17.05	16.2	16.13	15.06	13.76	13.02	11.74	11.58	10.63	9.82	9.13	8.7	8.33	13.2
CNN-1	[7]	11.76	15.3	14.25	14.44	15.47	13.88	12.44	12.76	12.22	10.91	10.01	9.07	8.6	8.2	7.69	11.8
MCNN-1	[7]	14.04	16.74	15.43	13.1	14.83	13.32	13.49	13.55	11.98	11.13	10.84	10.02	9.32	8.53	8.0	12.29
S-LDA	[18, 8]	14.04	16.25	16.23	14.1	14.2	13.04	13.22	12.51	11.06	11.13	10.21	10.23	8.95	8.53	8.33	12.14
SWFP	[19, 4]	14.63	19.23	18.32	15.49	15.79	14.17	13.49	12.76	11.51	11.13	10.21	9.63	8.77	8.53	8.0	12.78
P3T	[3]	15.83	20.26	**20.52**	15.84	16.13	15.06	14.61	13.28	12.22	11.13	10.63	9.82	9.13	8.53	8.33	13.42
Cross-domain)		17.06	**21.86**	**20.52**	16.93	16.46	15.68	15.51	**13.55**	**12.97**	11.81	11.07	10.23	9.51	9.07	8.51	**14.05**
In-domain		16.44	**21.86**	18.32	16.57	17.15	**16.31**	**15.82**	13.28	12.46	12.06	**11.55**	**10.91**	**9.92**	**9.48**	8.51	14.04
In-domain (60%)		12.88	18.72	19.62	17.68	16.46	15.68	14.9	13.55	12.46	**12.06**	11.07	10.44	9.51	9.07	**8.69**	13.52

these results highlight that self-supervised pretraining reduces calibration requirements, and that moderate P300 aggregation ($G = 2$) provides additional gains, improving both recognition speed and discriminability in the P300 speller task.

3.6 Baseline Comparision

Table 3 compares SpellerSSL ($G = 2$) with state-of-the-art (SOTA) methods in terms of information transfer rate (ITR, bits/min). Details of ITR computation are provided in Appendix B.5. The best results are highlighted in bold, and average ITRs across repetitions are also reported to capture overall performance. The baselines include CNN-1 [7], MCNN-1 [7], ERP-CapsNet [8], ST-CapsNet [4], Spatially Weighted FLD-PCA (SWFP) [19, 4], Shrinkage LDA (S-LDA) [18, 8], and stacked Transformers (P3T) [3]. Overall, SpellerSSL achieves superior or comparable results in ITR. The peak ITR of 21.86 bits/min is obtained at repetition 2 by both cross-domain and in-domain SpellerSSL, surpassing all SOTA baselines. Cross-domain pretraining further achieves the highest average ITR (14.05 bits/min), slightly above in-domain (14.04 bits/min), highlighting the transferability of self-supervised pretraining even across paradigms. A complementary comparison of CRRs (%) across repetitions with SOTA models is provided in Appendix D.

4 Conclusion

We proposed SpellerSSL, a self-supervised framework for P300 speller BCIs that combines reconstruction-based pretraining with a lightweight ERP-Head classifier, together with a P300 aggregation strategy to improve ERP signal quality. Experiment results show that SpellerSSL outperforms or matches state-of-the-art methods in information transfer rate (ITR). In-domain SSL with moderate aggregation achieves the best overall results, reducing calibration requirements by up to 60% while maintaining the performance of prior models. Cross-domain SSL further demonstrates strong transferability, highlighting the potential of self-supervised EEG pretraining. These results establish SSL and aggregation as effective strategies for improving efficiency, robustness, and generalization in practical P300 speller BCIs, paving the way toward future EEG foundation models.

References

[1] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005, 2018.

[2] E Başar, C Başar-Eroglu, B Rosen, and A Schütt. A new approach to endogenous event-related potentials in man: relation between EEG and P300-wave. *International Journal of Neuroscience*, 24(1):1–21, 1984.

[3] Jiazhen Hong and Laleh Najafizadeh. P3T: A transformer model for enhancing character recognition rates in P300 speller systems. In *2024 58th Asilomar Conference on Signals, Systems, and Computers*, pages 514–518, 2024.

[4] Zehui Wang, Chuangquan Chen, Junhua Li, Feng Wan, Yu Sun, and Hongtao Wang. ST-CapsNet: Linking spatial and temporal attention with capsule network for P300 detection improvement. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:991–1000, 2023.

[5] William Speier, Corey Arnold, Nand Chandravadia, Dustin Roberts, Shrita Pendekanti, and Nader Pouratian. Improving P300 spelling rate using language models and predictive spelling. *Brain-Computer Interfaces*, 5(1):13–22, 2018.

[6] Faraz Akram, Mohamed K Metwally, Hee-Sok Han, Hyun-Jae Jeon, and Tae-Seong Kim. A novel P300-based BCI system for words typing. In *2013 International Winter Workshop on Brain-Computer Interface (BCI)*, pages 24–25. IEEE, 2013.

[7] Hubert Cecotti and Axel Graser. Convolutional neural networks for P300 detection with application to brain–computer interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):433–445, 2010.

[8] Ronghua Ma, Tianyou Yu, Xiaoli Zhong, Zhu Liang Yu, Yuanqing Li, and Zhenghui Gu. Capsule network for ERP detection in brain–computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:718–730, 2021.

[9] Jiazhen Hong and Laleh Najafizadeh. TopoEEG: A timesformer-based topographic image representation method for single-trial early detection of P300. In *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2025.

[10] Jiazhen Hong, Weinan Wang, and Laleh Najafizadeh. ChatBCI: A P300 speller BCI leveraging large language models for improved sentence composition in realistic scenarios. *preprint arXiv:2411.15395*, 2024.

[11] Nand Chandravadia, Shrita Pendekanti, Dustin Roberts, Robert Tran, Saarang Panchavati, Corey Arnold, Nader Pouratian, and William Speier. Comparing P300 flashing paradigms in online typing with language models. *PlOS ONE*, 20(2):e0303390, 2025.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[13] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[14] Dean Krusienski and Gerwin Schalk. BCI competition III challenge. Technical report, 2004. URL: https://www.bbci.de/competition/iii/desc_II.pdf.

[15] Ary L. Goldberger, Luis A. Nunes Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. EEG motor movement/imagery dataset. `https://physionet.org/content/eegmmidb/1.0.0/`, 2009. PhysioNet.

[16] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.

[17] Vinay Jayaram and Alexandre Barachant. MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011, 2018.

[18] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of ERP components—a tutorial. *NeuroImage*, 56(2):814–825, 2011.

[19] Adam D Gerson, Lucas C Parra, and Paul Sajda. Cortically coupled computer vision for rapid image search. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):174–179, 2006.

A Model Architectures

A.1 1D U-Net Backbone

Table 4 summarizes the customized 1D U-Net used for SSL pretraining. During downstream adaptation, we only use the *bottleneck* (encoder bottleneck, pre-decoder) feature map for classification, whereas the pretraining objective is applied over the entire reconstruction pipeline (encoder + bottleneck + decoder). Input sequences are zero-padded so that $L \bmod 16 = 0$.

Table 4: 1D U-Net backbone used for SSL pretraining. The feature map extracted for downstream is at the bottleneck (highlighted in bold).

Stage	**Channels (in → out)**	**Kernel**	**Stride**	**Output length**
Enc1 Conv×2	$C \to 64$	3	1	L
MaxPool	$64 \to 64$	2	2	$L/2$
Enc2 Conv×2	$64 \to 128$	3	1	$L/2$
MaxPool	$128 \to 128$	2	2	$L/4$
Enc3 Conv×2	$128 \to 256$	3	1	$L/4$
MaxPool	$256 \to 256$	2	2	$L/8$
Enc4 Conv×2	$256 \to 512$	3	1	$L/8$
MaxPool	$512 \to 512$	2	2	$L/16$
Bottleneck Conv×2	$512 \to 1024$	3	1	$L/16$
UpConv + Dec4 Conv×2	$1024 \to 512$	3	2	$L/8$
UpConv + Dec3 Conv×2	$512 \to 256$	3	2	$L/4$
UpConv + Dec2 Conv×2	$256 \to 128$	3	2	$L/2$
UpConv + Dec1 Conv×2	$128 \to 64$	3	2	L
Final Conv1 × 1	$64 \to C$	1	1	L

A.2 ERP-Head Classifier

The ERP-Head operates on the encoder bottleneck *sequence*

$$b \in \mathbb{R}^{1024 \times \frac{L}{16}},$$

discarding the decoder. It applies a pointwise projection, followed by a depthwise temporal convolution, a dilated depthwise convolution, and a final pointwise fusion. Global average pooling (GAP) over time yields a feature vector $h \in \mathbb{R}^D$, which is then passed to a fully connected layer for binary classification.

Table 5: ERP-Head (tconv) used for downstream P300 classification.

Layer	**Channels (in→out)**	**Kernel**	**Stride / Dilation**	**Output length**
Conv 1 × 1 (projection)	$1024 \to D$	1	1 / –	$L/16$
Depthwise Conv	$D \to D$	3	1 / 1	$L/16$
Dilated Depthwise Conv	$D \to D$	3	1 / 2	$L/16$
Conv 1 × 1 (fusion)	$D \to D$	1	1 / –	$L/16$
Global Avg. Pool (time)	$D \to D$	–	–	1
Fully Connected (classifier)	$D \to 2$	–	–	1

Notes. All convolutions use GELU activations; the first 1×1 projection is followed by BatchNorm1d. All convolutions use stride = 1 with padding chosen to preserve the temporal length (hence $L/16$ before GAP). "Output length" refers to the temporal dimension; the final fully connected layer outputs 2 logits (non-target vs. target) while the temporal length is 1 after GAP.

B Evaluation Metrics

Each character selection uses R=15 repetitions, with 12 flashes (6 rows and 6 columns) per repetition. Let $s_k^{(i)}$ be the classifier score for code k in repetition i. The cumulative score up to n repetitions is

$$S_k(n) = \sum_{i=1}^{n} s_k^{(i)}. \tag{11}$$

The predicted column c and row r are

$$c = \arg\max_{k \le 6} S_k(n), \qquad r = \arg\max_{k > 6} S_k(n), \tag{12}$$

and the recognized character is given by (r, c).

B.1 Character Recognition Rate (CRR)

Proportion of correctly recognized characters after n repetitions.

B.2 Single-trial Accuracy

Binary classification accuracy at the trial level, with labels

$$y[k] = \begin{cases} 1, & k \in \mathcal{K}_{\text{P300}} \text{ (P300)}, \\ 0, & \text{otherwise (Non-P300)}. \end{cases}$$

B.3 Binary F1-Score

For trial-level P300 vs. Non-P300 classification, the F1-score balances precision and recall:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{Precision} = \tfrac{TP}{TP+FP}, \ \text{Recall} = \tfrac{TP}{TP+FN}. \tag{13}$$

B.4 Fisher's Discriminant Ratio (FDR)

$$\text{FDR} = \frac{(\mu_{\text{P300}} - \mu_{\text{Non}})^2}{\sigma_{\text{P300}}^2 + \sigma_{\text{Non}}^2} \tag{14}$$

where μ and σ denote the mean and variance of scores for P300 vs. Non-P300 trials, respectively.

B.5 Information Transfer Rate (ITR)

Information transfer rate (ITR) quantifies communication efficiency in bits/min, balancing accuracy and speed [4]:

$$\text{ITR}(r) = \tfrac{60}{T(r)} \left[\log_2 N + A(r) \log_2 A(r) + (1 - A(r)) \log_2 \tfrac{1-A(r)}{N-1} \right], \tag{15}$$

where $A(r)$ is the CRR after r repetitions, N is the number of characters, and $T(r)$ is the average time per selection.

In this study, we follow the BCI Competition III-II dataset [14]. Each repetition includes 12 flashes, each lasting 100 ms with a 75 ms inter-stimulus interval, totaling 2.1 s. Together with a fixed 2.5 s pause between characters, the cumulative time after r repetitions is

$$T(r) = 2.5 + 2.1 \times r. \tag{16}$$

C Additional ERP Visualizations

To complement Figure 6 and 7, which provides an overview of reconstruction performance across all 64 EEG channels, we further present detailed visualizations of event-related potentials (ERPs) grouped by brain region. For each region (Frontal, Central, Parietal, Occipital, and Temporal), we

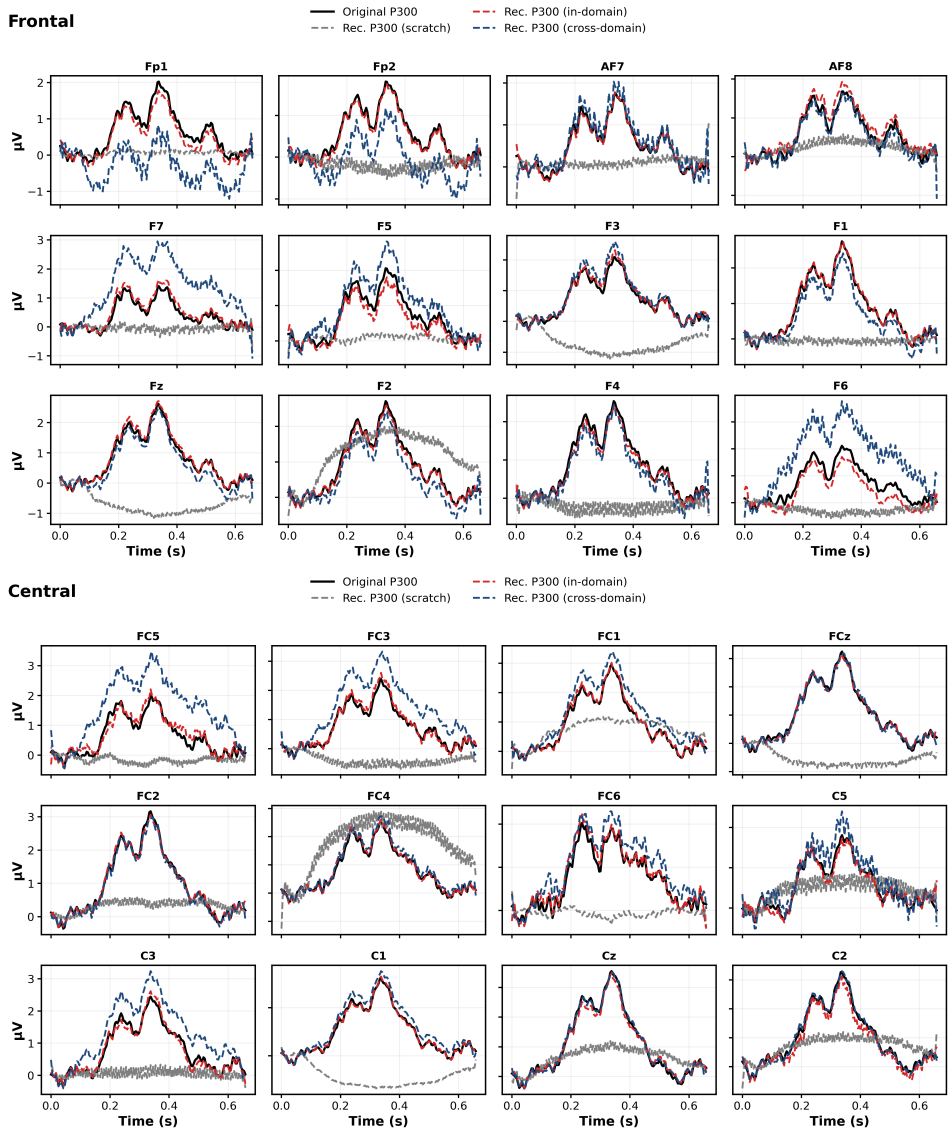


Figure 6: Region-wise reconstruction of P300 ERPs for (a) Frontal and (b) Central regions.

display up to 12 representative channels in a grid layout. Each subplot shows the original averaged P300 response (black) compared with reconstructions from three models: trained from scratch (gray), in-domain pretraining (red), and cross-domain pretraining (blue).

These region-wise plots highlight differences in reconstruction fidelity across scalp locations. In particular, in-domain pretraining achieves reconstructions closely aligned with ground truth across most regions, while cross-domain pretraining captures the overall waveform morphology but may underestimate amplitudes in some frontal and occipital sites. Scratch models generally fail to recover meaningful P300 waveforms. Together, these results demonstrate the robustness and transferability of SpellerSSL across spatially distributed brain regions.

In addition to the region-wise ERP comparisons, we also provide unified reconstructions across all channels to illustrate overall signal morphology. Figure 8 shows grand-average visualizations of P300 and Non-P300 responses under cross-domain and in-domain pretraining. Original signals are plotted alongside their reconstructions, with shaded regions denoting variability across channels. These plots complement the channel- and region-specific figures.

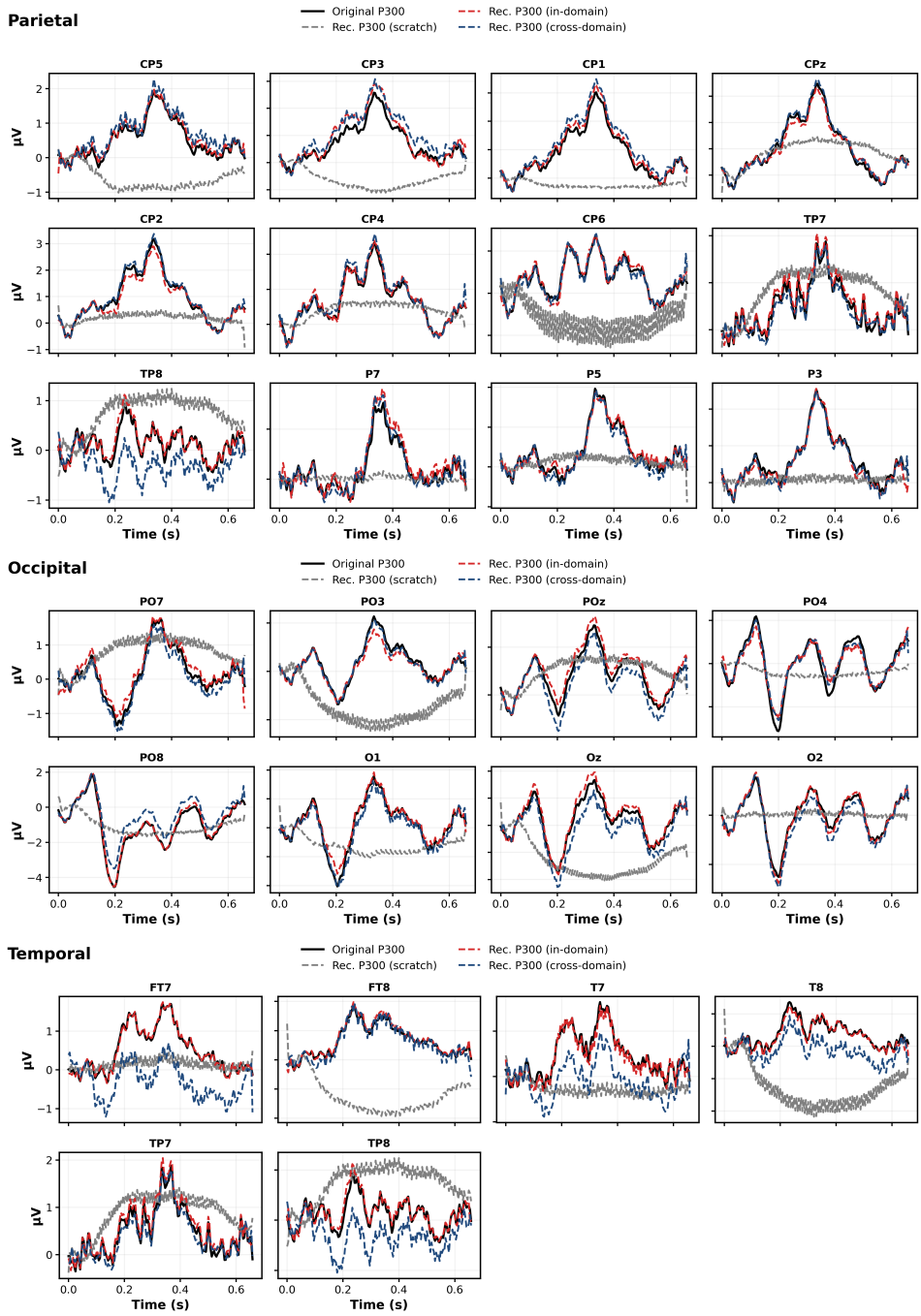
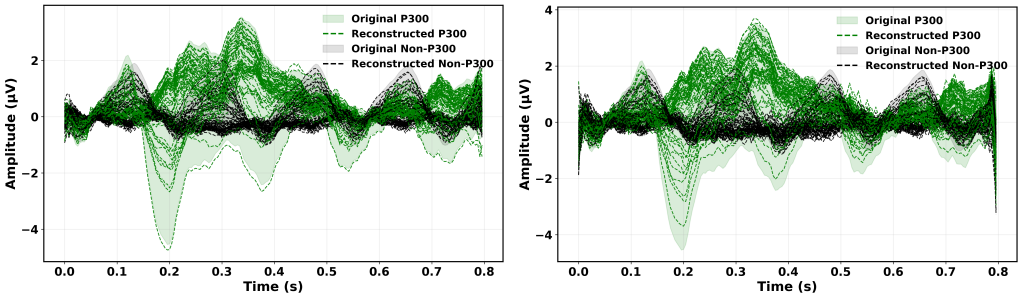


Figure 7: Region-wise reconstruction of P300 ERPs for (c) Parietal, (d) Occipital, and (e) Temporal regions.



(a) In-domain ERP across all channels.

(b) Cross-domain ERP across all channels.

Figure 8: Overall reconstruction visualizations of P300 and Non-P300 ERPs across all 64 channels under cross-domain and in-domain pretraining. Original responses (solid lines) are shown with reconstructions (dashed lines), with shaded regions indicating variability across channels.

Table 6: Comparison of CRRs (%) between SpellerSSL and state-of-the-art models.

Model	Ref.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ST-CapsNet	[4]	41	61	66	**78**	**85**	86	92	90	91	95	96	96	95	97	96
ERP-CapsNet	[8]	45	60	66	73	81	85	87	90	90	94	94	94	94	95	96
CNN	[7]	35	52	59	68	79	81	82	89	92	91	91	90	91	92	92
MCNN	[7]	39	55	62	64	77	79	86	92	91	92	95	95	95	94	94
S-LDA	[18, 8]	39	54	64	67	75	78	85	88	87	92	92	96	93	94	96
SWFP	[19, 4]	40	60	69	71	80	82	86	89	89	92	92	93	92	94	94
P3T	[3]	42	62	**74**	72	81	85	90	91	92	92	94	94	94	94	96
Cross-domain		**44**	**65**	**74**	75	82	87	93	**92**	**95**	95	96	96	96	97	**97**
In-domain		43	**65**	69	74	84	**89**	**94**	91	93	**96**	**98**	**99**	**98**	**99**	**97**
In-domain (60%)		37	59	72	77	82	87	91	92	93	**96**	96	97	96	97	**98**

D Additional Baseline Results

Table 6 reports a detailed comparison of character recognition rates (CRR, %) between SpellerSSL ($G = 2$) and representative state-of-the-art P300 speller models across repetitions 1–15. These results complement the ITR analysis in the main text (Table 3), providing an additional insight into recognition accuracy over increasing repetitions. CRRs above 90% are underlined, and the best values at each repetition are highlighted in bold.

Compared with classical approaches (e.g., S-LDA, SWFP) and recent deep models (e.g., ERP-CapsNet, ST-CapsNet, P3T), SpellerSSL achieves superior or comparable performance across most repetitions. Notably, early recognition (repetitions 1–3) is improved by SSL pretraining, with cross-domain and in-domain checkpoints showing faster accuracy gains than baselines. At later repetitions, in-domain SpellerSSL maintains the highest CRRs (up to 99%), confirming the benefit of self-supervised pretraining not only in terms of ITR efficiency but also in raw recognition accuracy.

These results further support the conclusion that SSL pretraining enhances both speed and accuracy, and that SpellerSSL can match or surpass existing state-of-the-art methods even under reduced calibration (e.g., in-domain 60%).