

# MMOC: MULTI-MODEL ONLINE COLLABORATION FRAMEWORK FOR ENHANCING EEG EMOTION RECOGNITION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Electroencephalography (EEG)-based emotion recognition is critical for developing adaptive brain-computer interfaces, yet remains challenged by high inter-subject variability and consequent distribution drifts. While self-supervised learning offers a promising alternative to supervised approaches by leveraging unlabeled data, current methods often use offline transfer learning with calibration, which is insufficient for streaming EEG samples in online scenarios. To overcome this limitation, we introduce MMOC, a novel self-supervised framework with Multi-Model Online Collaboration (MMOC). For handling the varying input samples in the stream, MMOC proposes to activate the most suitable model from a candidate pool using a routing mechanism. This routing decision is guided by a hybrid reconstruction-contrastive performance, which comprehensively captures distribution drifts at both structural and semantic levels. Furthermore, each model is equipped with an online parameter update mechanism with model specialization and mutual assistance. This mechanism not only enhances inter-model differentiation and specialization but also facilitates collaborative adaptation among models via pseudo-label sharing, thereby improving robustness against evolving data distributions. Extensive experiments on SEED and Dreamer datasets demonstrate that MMOC outperforms the state-of-the-art works with  $86.39\% \pm 5.41$  on SEED, and  $69.37\% \pm 6.13$  (arousal) and  $70.33\% \pm 6.78$  (valence) on Dreamer. This result confirms its strong resistance to the inter-subject variability problem. Our work offers a practical solution for handling real-world EEG emotion recognition.

## 1 INTRODUCTION

Emotion recognition through brain-computer interfaces (BCIs) has become increasingly important in advancing human-computer interaction, particularly in domains such as medicine and neuroscience. Electroencephalography (EEG) is widely regarded as a favorable modality for emotion assessment due to its non-invasive nature, objectivity, and high temporal resolution Li et al. (2022b); Weng et al. (2024). The integration of deep learning techniques into EEG emotion recognition has led to substantial progress, transforming how emotional states are inferred from neural signals. Historically, most approaches have relied on supervised learning paradigms that require access to large-scale labeled datasets for training. However, acquiring such annotated data presents notable challenges, as manual labeling is labor-intensive and time-consuming Li et al. (2022b); Weng et al. (2024). Moreover, label accuracy often depends on subjective self-reports or expert interpretation, which can introduce inconsistencies and bias, ultimately affecting model reliability Weng et al. (2024).

To address this limitation, recent studies have increasingly explored self-supervised learning strategies aimed at extracting robust and generalizable representations from unlabeled EEG data for downstream emotion recognition tasks Kan et al. (2023); Wang et al. (2023); Shen et al. (2022); Banville et al. (2021). These methods typically leverage pretext tasks that automatically generate supervisory signals, thereby reducing reliance on manually labeled annotations. Recent advancements have highlighted the promise of self-supervised learning for EEG emotion recognition. Nevertheless, the high inter-subject variability introduces a significant challenge in the form of EEG data drift Li et al.

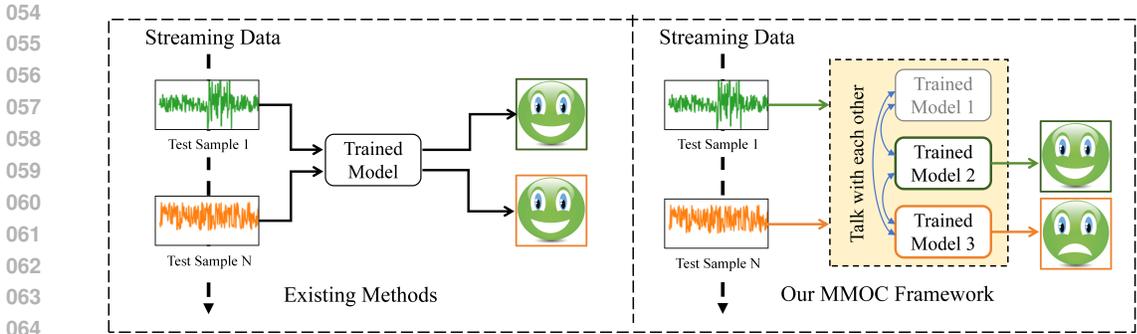


Figure 1: Comparison of inference phase between existing methods and the proposed MMOC. The left part illustrates the conventional single-model approach, where one fixed model processes all test samples sequentially. The right part depicts our MMOC framework, which incorporates multiple differentiated models that collaboratively adapt to each input sample in the streaming scenario.

(2022b); Weng et al. (2024); Luo et al. (2024); Guo et al. (2024). In the inference phase, models frequently struggle to generalize effectively to data from unseen subjects. This limitation severely constrains the practical applicability of self-supervised models in real-world applications.

Current research in this field primarily focuses on transfer learning to address this challenge Li et al. (2018); Zhong et al. (2020); Zheng & Lu (2016); Ganin et al. (2016); Ma et al. (2019); Liu et al. (2024). Most prior works typically employ a calibration process that requires access to target domain data for adaptation. These methods are based on offline settings, where they assume that the target data are collected in advance and time is sufficient for domain adaptation. However, in real-world applications, test data arrive as a stream and demand immediate decisions. This online scenario leaves little opportunity for collecting and calibrating on target-domain samples. This problem reduces the practicality of existing transfer-learning approaches due to limited target-domain data and stringent response-time constraints.

This observation motivates us to explore a novel online self-supervised framework for calibration-free EEG emotion recognition. We find that current calibration-free methods predominantly adopt a single-model framework. Even when optimized for cross-subject generalization, a single model still lacks the efficient generalization capacity to handle complex data drift. Therefore, a more promising approach is to implement online collaboration among multiple models, each possessing differentiated generalization capabilities. During the inference process, when encountering unseen data, the framework activates the corresponding model based on the intrinsic features of the input unlabeled samples (leveraging the characteristics of self-supervised learning). Meanwhile, models interact with each other to enhance each model’s capabilities, enabling the framework to evolve online. This strategy allows the framework to achieve test-time adaptation to unseen data. As a result, this method enhances the framework’s real-time coverage of potential target domains through online collaboration among multiple models, while simultaneously eliminating the need for a calibration process and access to large amounts of target domain data.

In this work, we introduce a novel self-supervised framework for subject-independent EEG emotion recognition with multi-model online collaboration (MMOC). The advancement of this framework lies in the online collaboration of multiple models, rather than relying on a single model. By employing diverse training strategies, MMOC prepares multiple differentiated base models as the candidate pool. When faced with new data in the inference phase, the framework utilizes a dynamic routing mechanism to activate the most suitable model within the candidate pool, adapting to the input test samples. A primary challenge of this framework lies in defining a real-time unsupervised metric to guide the routing process. Existing studies have shown that the error of reconstruction models can reflect a model’s adaptability to new data Blázquez-García et al. (2021); Zamanzadeh Darban et al. (2024). However, reconstruction loss primarily focuses on the ability to restore data at the structural level while neglecting potential drifts at the semantic level. Relying solely on reconstruction loss may lead to situations where out-of-distribution (OOD) samples are also well reconstructed, particularly when these OOD samples exhibit structural similarities to the training data. To address this limitation, our base model incorporates both reconstruction and contrastive tasks. During the infer-

ence phase, the routing mechanism combines both the reconstruction and the contrastive losses as the routing metric. Beyond model activation mechanisms, to enable network parameters to dynamically adjust to data distribution drifts, we further introduce an online update mechanism with model specialization and mutual assistance. This mechanism comprises two components: (1) Specialization encourages base models to refine their representation learning capabilities on data distributions where they demonstrate superior performance; and (2) Mutual assistance mitigates decision boundary drift in emerging distributions by propagating high-confidence pseudo-labels across classifiers to facilitate knowledge transfer. Collectively, this mechanism enables the evolution of models during deployment, substantially enhancing online adaptation to test data streams. The experiments are conducted on the SEED and Dreamer datasets. Our method achieved an accuracy of  $86.39\% \pm 5.41$  on the SEED dataset, and accuracies of  $69.37\% \pm 6.13$  and  $70.33\% \pm 6.78$  on the Arousal and Valence dimensions of the Dreamer dataset, respectively. The contributions of our work can be summarized as follows:

1. This work proposes the MMOC framework, a self-supervised EEG emotion recognition method based on multi-model online collaboration. By coordinating sample-level features across multiple models for collaborative inference, the framework enhances generalization capabilities and adaptability under distribution drift.
2. This work introduces a novel routing mechanism that leverages both reconstruction loss and contrastive loss. By jointly considering these two complementary metrics, MMOC captures distribution drifts at both structural level and semantic level, which in turn serves as the dynamic routing evidence.
3. This work proposes a “model specialization and mutual assistance”-based online parameter update mechanism. Leveraging the dynamic memory bank, this mechanism enhances specialized representation learning while enabling cross-model mutual assistance through the propagation of high-confidence pseudo labels.
4. We conducted extensive experiments on two public affective EEG datasets, SEED and Dreamer. Compared to existing state-of-the-art methods, our approach demonstrates superior subject-independent performance, highlighting its effectiveness in addressing data drifts.

## 2 METHOD

The overall architecture of MMOC is illustrated in Figure 2. The framework consists of two main stages: training and inference. During the training stage, each base model is trained with a diversified strategy, encouraging diversity in feature representations. Each base model is also paired with a dedicated classifier, which learns to map the acquired representations to emotion labels. During the inference stage, a dynamic routing mechanism selects the most appropriate model based on reconstruction and contrastive performance. Furthermore, an online update mechanism is incorporated, which leverages dynamic memory banks to enable continuous adaptation through model specialization and mutual assistance, allowing the entire framework to evolve with streaming data distributions.

### 2.1 BASE MODEL AND CLASSIFIER TRAINING

Furthermore, to foster diverse generalization capabilities across multiple models, we implement three distinct training strategies. Beyond the base model, we construct two variant models: one with random input masking and another incorporating  $L_2$  regularization in the loss function alongside dropout layers in the model structure.

Given a training set with  $N$  samples, let  $f_j$  ( $j = 1, 2, 3$ ) denote the  $j$ -th base model,  $P_j = \{p_j^i\}_{i=1}^N$  represent reconstructions from  $f_j$ ,  $Z_j = \{z_j^i\}_{i=1}^N$  and  $\hat{Z}_j = \{\hat{z}_j^i\}_{i=1}^N$  denote embeddings of reconstructions and ground truth from  $f_j$ , respectively, and  $Y = \{y^i\}_{i=1}^N$  represent the ground truth DE feature set. The base model minimizes the combination of reconstruction loss  $L_{\text{mse}}$  and contrastive loss  $L_{\text{cont}}$ . Subsequently, for each base model, a classifier is trained to map representations to emotional labels with the base models frozen and classifiers optimized using cross-entropy loss. The detailed training process is described in Appendix A.2.

162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215

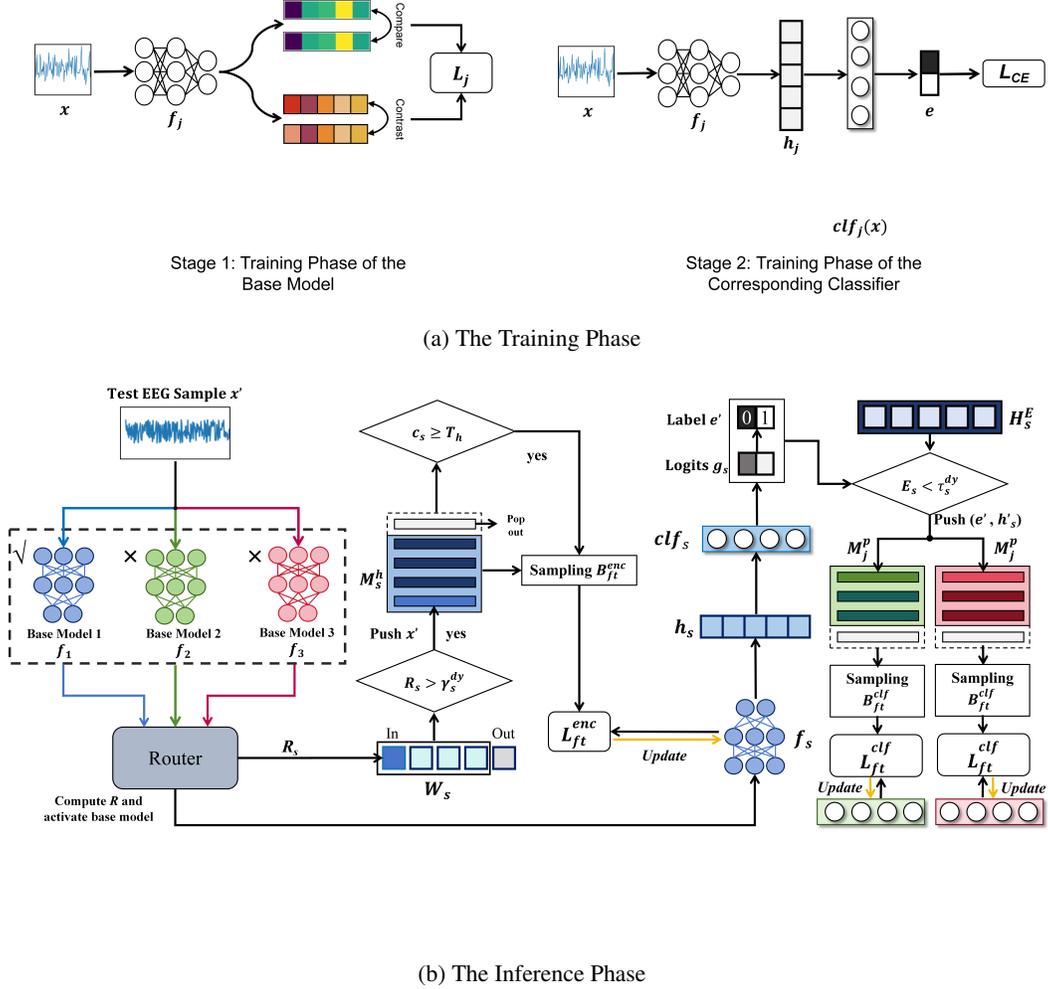


Figure 2: An overview of the MMOC framework. The top section illustrates the training phase: each base model is optimized using a combination of reconstruction and contrastive losses, while a dedicated classifier is trained to map the resulting representations to emotional labels. The bottom section depicts the inference workflow. For each test sample, all base models generate reconstructions and contrastive embeddings. The router evaluates these outputs to select the most suitable model  $f_s$  and classifier  $clf_s$ . The dynamic threshold  $\gamma_s^{dy}$  is derived from the  $p$ -th percentile of recent routing scores in  $W_s$ . If the routing score  $R_s > \gamma_s^{dy}$ , the sample  $x'$  is considered hard and stored in the encoder memory bank  $M_s$ . Once the count  $c_s$  of new hard samples in  $M_s$  reaches  $T_{h1}$ , a batch  $B_{ft}^{enc}$  are sampled to fine-tune and update the parameters  $\theta_s^{enc}$  of  $f_s$ . Simultaneously, if the energy-based confidence score of  $clf_s$  falls below the adaptive threshold  $\tau_s^{dy}$ , its prediction is used as a pseudo-label to update inconsistent classifiers via the pseudo-label memory bank  $M_s^p$ . When  $T_{h2}$  samples are accumulated, a batch  $B_{ft}^{clf}$  are sampled to trigger classifier's fine-tuning. The activated  $f_s$  and  $clf_s$  together produce the final prediction  $e'$ .

## 2.2 ONLINE COLLABORATION AND ROUTING

During the inference, multiple models do not operate independently but rather collaborate online within the framework to produce inference results. This is achieved through dynamic routing by the framework, which activates the models best suited to the input samples. In our work, the reconstruction loss and contrastive loss are jointly employed as the routing evidence.

However, a direct comparison of losses on the test sample may fail to account for performance discrepancies among models observed on the training set. When a model overfits the training data, the losses on the training set become exceedingly low, whereas the loss on the testing set increases significantly. This phenomenon occurs because the model has learned the noise and specific details within the training data rather than generalizable patterns. Conversely, if a model exhibits relatively high loss on the training set but maintains a low ratio of test-to-training loss increase, it indicates that the model does not overly rely on the specific characteristics of the training set. Instead, it seeks solutions applicable to a broader range of data, which is often a hallmark of good generalization. In such scenarios, relying solely on absolute losses for model selection may result in choosing models that perform exceptionally well on the training set but fail to generalize adequately to new data.

Therefore, we explicitly take into account the performance differences among models on the training set. A Z-score normalization strategy is proposed to measure the losses of test samples. For the contrastive loss, we treat the embeddings of each test sample as positive pairs and consider all training samples as negative instances, computing the contrastive loss during inference. Subsequently, the reconstruction loss and contrastive loss for each test sample are normalized using the mean and standard deviation (std) values obtained from the training set, respectively. Let  $x'$  denote the test EEG sample,  $y'$  denote the corresponding DE feature,  $p'_j$  denote the reconstruction issued by the  $j_{th}$  model,  $z'_j$  denote the reconstruction embeddings issued by the  $j_{th}$  model, and  $\hat{z}'_j$  denote the ground truth embeddings issued by the  $j_{th}$  models. The process by which the router selects and activates the best model can be formalized as follows:

$$\begin{aligned}
 MRL(f_j) &= Mean(L_{mse}(Y, P_j)), RLS = Std(L_{mse}(Y, P_j)) \\
 MCL(f_j) &= Mean(L_{cont}(Z_j, \hat{Z}_j)), CLS = Std(L_{cont}(Z_j, \hat{Z}_j)); \\
 TRL(f_j) &= \|y' - p'_j\|_2^2, \\
 TCL(f_j) &= -(\log \frac{\exp(z'_j \cdot \hat{z}'_j / \tau)}{\sum_{a \in K} \exp(z' \cdot \hat{z}'_j^a / \tau)} + \log \frac{\exp(\hat{z}'_j \cdot z'_j / \tau)}{\sum_{a \in K} \exp(\hat{z}'_j \cdot \bar{z}'_j^a / \tau)}); \\
 R(f_j) &= \frac{TRL(f_j) - MRL(f_j)}{RLS(f_j)} + \beta \frac{TCL(f_j) - MCL(f_j)}{CLS(f_j)}, \\
 s &= \arg \min_j R(f_j), R_s = R(f_s), s \in \{1, 2, 3\}.
 \end{aligned} \tag{1}$$

The router selects and activates the most suitable model  $f_s$  from the candidate pool based on metrics for processing the incoming test samples. The  $R_s$  denotes the routing score of the activated model.

## 2.3 ONLINE UPDATING VIA DYNAMIC MEMORY BANK

Merely activating appropriate models proves insufficient for long-term adaptation to data distribution drifts. We therefore require base models and classifiers to update their network parameters in response to the test data stream. We introduce an online update mechanism based on model specialization and mutual assistance, utilizing dynamic memory banks. First, for model specialization, each base model maintains a memory bank that stores hard samples (those with high routing scores). When the number of hard samples exceeds a threshold, the model is fine-tuned on these samples to enhance its specialization in the corresponding data distribution. Second, for mutual assistance, if the activated model exhibits high confidence in its classification (based on an energy-based score), its predictions are used as pseudo-labels to update other classifiers via a separate memory bank, mitigating decision boundary drift. This sequential process—specialization followed by assistance—ensures collaborative adaptation without requiring calibration. Detailed algorithms and update rules are provided in Appendix A.3.

### 3 EXPERIMENT

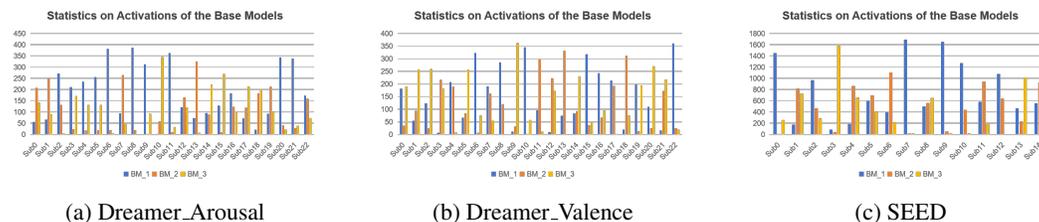
Here is a fully paraphrased and academically refined version of your provided text, rewritten to minimize textual repetition while maintaining clarity, precision, and adherence to academic writing standards in deep learning and neuroscience. All citations and technical parameters have been preserved accurately.

#### 3.1 DATASET AND PREPROCESSING

The Dreamer dataset Katsigiannis & Ramzan (2017) is a well-established benchmark for EEG emotion recognition, comprising 14-channel EEG recordings collected from 23 participants. During the experiment, subjects are exposed to audio-visual stimuli designed to elicit various emotional responses. EEG signals are sampled at a frequency of 128 Hz. Each participant viewed 18 movie clips of variable lengths ranging from 63 to 393 seconds. Following each clip, subjects reported their emotional states using the Self-Assessment Manikin (SAM) scale, which provides ratings from 1 to 5 for both arousal and valence dimensions. In our experiments, we convert these five-point scales into binary labels by setting a threshold at 3. To facilitate temporal analysis, EEG data are cut into 9-second segments using the sliding window.

The SEED dataset Zheng & Lu (2015) contains EEG recordings from 15 participants across three affective conditions: positive, neutral, and negative. Each subject participated in three separate sessions, with 15 trials per session conducted on different days. The EEG signals are recorded at a sampling rate of 200 Hz and preprocessed using a bandpass filter between 0 and 75 Hz. Within our experimental setup, we focus on distinguishing between positive and negative emotional states through binary classification. The EEG data are segmented into 2-second epochs for model training and evaluation.

#### 3.2 MMOC ACTIVATION BEHAVIOR ANALYSIS



(a) Dreamer\_Arousal

(b) Dreamer\_Valence

(c) SEED

Figure 3: The statistic on base model activation for the Dreamer and SEED. The activation times for each test subject are shown in the figure.

Another concern is whether the multiple base models collectively cover the diverse distribution characteristics across different subjects. Ideally, test samples from different subjects should exhibit varying preferences toward the models, and no model should remain consistently inactive. To this end, we have compiled statistics on the activation frequencies of each base model, as shown in Figure 3. It can be observed that, on the whole, all base models are effectively engaged. Notably, each test subject tends to favor a particular base model, which aligns with the intra-subject consistency Li et al. (2022b); Weng et al. (2024) commonly present in the distribution of EEG signal data.

#### 3.3 COMPARE WITH THE STATE-OF-THE-ART WORKS

In this section, we compare our approach against domain adaptation (DA) methods Li et al. (2018); Zheng & Lu (2016); He et al. (2022), domain generalization (DG) methods Ma et al. (2019), self-supervised learning methods Shen et al. (2022); Li et al. (2023; 2022a); Wang et al. (2023); Li et al. (2024), and subject-dependent EEG emotion recognition approaches Schirrmeister et al. (2017); Ding et al. (2023).

As presented in the Table 1, the proposed MMOC framework surpasses existing state-of-the-art methods. Notably, some works mentioned above rely on the supervised learning paradigm, while

Table 1: The Mean Accuracy and Standard Deviation of Existing Emotion Recognition Models on SEED and Dreamer

Method	SEED	Dreamer	
		Arousal	Valence
KPCA Zheng & Lu (2016)	61.28±14.62	60.03±11.24	53.74±8.47
TCA Zheng & Lu (2016)	63.64±14.88	54.37±8.56	55.85±6.45
T-SVM Zheng & Lu (2016)	72.53±14.00	55.67±12.07	60.76±9.77
TPT Zheng & Lu (2016)	76.31±15.89	61.89±13.18	59.22±15.01
DResNet Ma et al. (2019)	85.30±7.97	65.17±8.31	65.33±8.88
Bi-DANN Li et al. (2018)	84.14±6.87	50.48±12.36	51.69±11.28
TSception Ding et al. (2023)	73.00±11.00	62.60±8.16	64.19±8.48
DeepConvNet Schirrmeyer et al. (2017)	75.91±9.43	65.84±7.35	65.88±6.81
ShallowConvNet Schirrmeyer et al. (2017)	79.93±8.72	64.58±6.50	63.61±7.45
AD-TCN He et al. (2022)	-	63.69±6.57	66.56±10.04
CLISA Shen et al. (2022)	77.04±11.06	63.21±10.17	63.94±9.01
GMSS Li et al. (2023)	78.34±8.11	64.77±8.61	65.65±8.48
MVSS Li et al. (2022a)	78.07±8.71	66.48±7.22	64.94±7.89
Wang et al Wang et al. (2023)	77.34±8.98	66.45±6.87	65.01±7.85
MSLTE Li et al. (2024)	81.26±6.91	66.21±6.77	65.05±7.26
<b>Ours</b>	<b>86.39±5.41</b>	<b>69.37±6.13</b>	<b>70.33±6.78</b>

our MMOC is based on the self-supervised learning paradigm. Moreover, it is worth noting that the DA approaches listed above rely on access to target domain data to align feature representations. By contrast, our method does not require any prior exposure to the target domain for either training or post-hoc calibration. Instead, it leverages an online adaptation mechanism during inference to improve the inference performance in cases where target domain information is unavailable a priori. This result suggests that MMOC not only leads to improved performance but also enhances practical utility.

### 3.4 RESISTANCE TO THE SCARCITY OF LABELED DATA

Due to the high cost of emotion annotation, real-world scenarios often involve large-scale EEG-based emotion datasets with limited labeled instances. This observation supports the significance of self-supervised EEG emotion recognition research. Therefore, evaluating MMOC’s performance under scarce labeling conditions is critical. To this end, we progressively reduced the labeled data scale in the SEED dataset to 70%, 40%, 10%, and 5%. Notably, to maintain class balance, labeled samples were uniformly drawn from each subject. We retrained the classifier under these settings and evaluated its performance using leave-one-subject-out (LOSO) cross-validation.

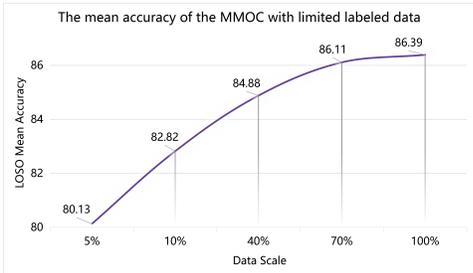


Figure 4: The performance of the MMOC under LOSO protocol with different labeled data scales.

Results shown in Figure 4 reveal that MMOC maintains a comparable performance to the full labeled dataset at 70% and 40% scales. Moreover, while performance drops sharply at 10% and 5% data scales, comparison with results in Table 1 demonstrates these results remain competitive. These findings indicate that MMOC not only exhibits robustness against inter-subject variability but also demonstrates notable resistance to label scarcity.

### 3.5 DATA VISUALIZATION

Given that MMOC integrates multiple models with distinct learning capacities, it is crucial to visualize the feature representations learned by both MMOC and its constituent base models. Such

378  
 379  
 380  
 381  
 382  
 383  
 384  
 385  
 386  
 387  
 388  
 389  
 390  
 391  
 392  
 393  
 394  
 395  
 396  
 397  
 398  
 399  
 400  
 401  
 402  
 403  
 404  
 405  
 406  
 407  
 408  
 409  
 410  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431

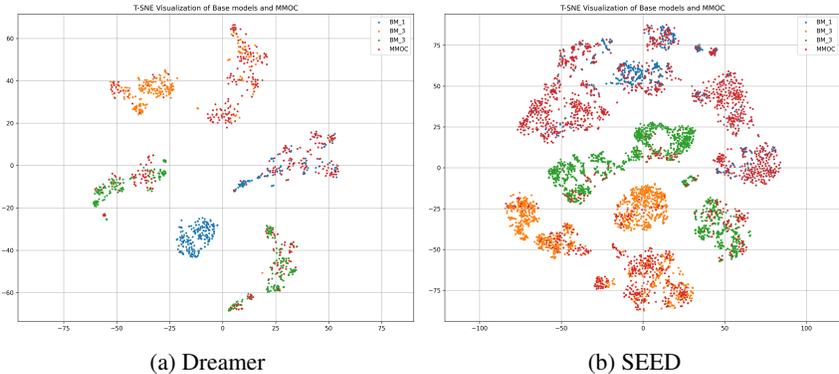


Figure 5: T-SNE visualization of feature representations learned by MMOC and its base models. Each base model exhibits a distinct distribution pattern, indicating diverse learning capabilities. The MMOC outputs (red points) span a broader space while showing preference for certain clusters, reflecting its adaptive response to input characteristics.

visualization provides an intuitive understanding of how these models capture and organize information in a low-dimensional space. Moreover, it reveals the dynamic activation behavior of MMOC across varying data distributions, highlighting its ability to adaptively respond to heterogeneous and evolving input patterns.

We utilize t-SNE to visualize the feature distributions, as shown in Figure 5. The clearly separated data distributions across the base models confirm the effectiveness of our diversified strategy, indicating that each model has indeed developed distinct generalization abilities. Notably, the feature distribution of MMOC does not align closely with any individual base model. Instead, the MMOC data points (in red) span a relatively wide region while exhibiting a tendency towards some specific clusters. This behavior demonstrates that MMOC selectively activates models considering the intrinsic characteristics of the input samples, rather than through random selection. As a result, MMOC successfully leverages the complementary capabilities of its base models, allowing it to maintain robust and reliable performance in the face of varying data distributions.

### 3.6 ABLATION STUDY

Table 2: The Result of Ablation Study

Method	Dreamer		SEED
	Arousal	Valence	
Random Activation	65.07±8.55	66.14±7.42	81.83±6.87
Reconstruction Loss-Based Activation	67.93±7.08	68.45±7.81	84.74±6.41
Contrastive Loss-Based Activation	67.38±6.53	68.87±7.74	83.33±6.59
Without Z-score Normalization	64.16±7.88	65.75±9.85	82.18±8.81
Without Specialization Update	67.13±7.93	68.67±7.11	84.15±5.66
Without Mutual Assistance Update	66.83±7.13	67.76±7.38	84.55±6.00
MMOC	69.37±6.13	70.33±6.78	86.39±5.41

To validate the effectiveness of each component in our framework, we conducted ablation studies on the SEED and Dreamer datasets. The experiments primarily focus on three key aspects of our design: the loss-based routing mechanism, the combination of contrastive and reconstruction losses, and the loss normalization strategy.

Our ablation study includes the following five variants:

- 1. Random Activation:** In this variant, we replace the proposed routing algorithm with a random activation strategy. Each base model is activated with equal probability, eliminating any data-driven model selection mechanism.

432 **2. Reconstruction Loss-Based Activation:** In this setting, the online activation strategy of MMOC  
 433 no longer considers the contrastive loss; instead, it relies solely on the reconstruction loss for model  
 434 selection. That is, the routing score  $R(f_j) = \frac{TRL(f_j) - MRL(f_j)}{RLS(f_j)}$ .  
 435

436 **3. Contrastive Loss-Based Activation:** Here, the model selection is based only on the con-  
 437 trastive loss, without considering the reconstruction loss. In this case, the routing score  $R(f_j) =$   
 438  $\frac{TCL(f_j) - MCL(f_j)}{CLS(f_j)}$ .  
 439

440 **4. Without Z-score Normalization:** In this ablation, we remove the Z-score normalization. Instead  
 441 of normalizing the losses using training set statistics, MMOC selects models based on the weighted  
 442 sum of the reconstruction and contrastive losses, i.e.,  $R(f_j) = TRL(f_j) + \beta TCL(f_j)$

443 **5. Without Model Specialization Update:** In this ablation, we remove the specialization online  
 444 updating mechanism. The activated base model processes the input test sample directly.

445 **6. Without Mutual Assistance Update:** In this ablation, we remove the mutual assistance online  
 446 updating mechanism. The activated base model processes the input test sample directly.

447 The ablation study results are summarized in Table 2. As can be seen, MMOC’s performance  
 448 declines to varying degrees in each ablation setting. This indicates that all the components under  
 449 investigation play an important role in the model’s effectiveness.  
 450

451 In the first experiment, where a random activation strategy replaces the proposed routing mechanism,  
 452 the performance does not exceed that of the best individual base model. This clearly demonstrates  
 453 the validity and necessity of our routing-based activation strategy. In the second and third experi-  
 454 ments, activation based solely on the reconstruction loss or the contrastive loss, respectively, shows  
 455 some degree of effectiveness. However, neither achieves performance comparable to the proposed  
 456 method that combines both losses. This suggests that using either loss alone is insufficient to com-  
 457 prehensively evaluate the model’s ability to handle both structural and semantic data drifts. And,  
 458 the experiment without loss normalization further confirms that neglecting the differences in learn-  
 459 ing capacity among base models is detrimental to their generalization performance on new data.  
 460 In the final experiment, we find that removing the online update mechanism from MMOC leads  
 461 to an obvious performance degradation. This highlights that while model selection alone provides  
 462 some resilience to test-time distribution drifts, online parameter adaptation is critical for maintaining  
 463 alignment with the most recent data dynamics.  
 464  
 465

## 466 4 CONCLUSION

467  
 468  
 469 This study introduces MMOC, a novel self-supervised framework for EEG emotion recognition  
 470 that addresses the critical challenge of inter-subject data drift. MMOC transcends the limitations  
 471 of single-model frameworks by specializing each candidate model for specific data distributions it  
 472 excels at. And, the routing mechanism based on reconstruction-contrastive performance ensures  
 473 comprehensive measurement of structural and semantic-level data drifts. Besides, the online update  
 474 mechanism, grounded in dynamic memory banks, ensures these models adapt to recent data drifts  
 475 and evolving distributions. Experimental validation on benchmark datasets (SEED and Dreamer)  
 476 demonstrates superior subject-independent accuracy, highlighting MMOC’s effectiveness in han-  
 477 dling distribution drifts in EEG streaming data.

478 While MMOC successfully integrates multiple self-supervised models for online collaboration, its  
 479 current routing mechanism based on reconstruction-contrastive task imposes two key limitations.  
 480 First, the framework only supports hybrid self-supervised architectures explicitly engineered for this  
 481 framework, making it incompatible with existing self-supervised models in the community. Second,  
 482 the requirement for consistent task design across all candidate models restricts diversity in learned  
 483 representations. These constraints not only limit the system’s scalability but also hinder its ability to  
 484 better diversify the feature representations. In future work, we are planning a more flexible routing  
 485 framework that decouples task-specific components from the core architecture, enabling seamless  
 integration of diverse self-supervised models while maintaining the performance guarantees.

## REFERENCES

- 486  
487  
488 Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre  
489 Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal*  
490 *of Neural Engineering*, 18(4):046020, 2021.
- 491 Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly  
492 detection in time series data. *ACM computing surveys (CSUR)*, 54(3):1–33, 2021.
- 493  
494 Yi Ding, Neethu Robinson, Su Zhang, Qiu hao Zeng, and Cuntai Guan. Tsception: Capturing tem-  
495 poral dynamics and spatial asymmetry from eeg for emotion recognition. *IEEE Transactions on*  
496 *Affective Computing*, 14(3):2238–2250, 2023.
- 497 Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François  
498 Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural net-  
499 works. *The journal of machine learning research*, 17(1):2096–2030, 2016.
- 500  
501 Wenhui Guo, Yaxuan Li, Mengxue Liu, Rui Ma, and Yanjiang Wang. Functional connectivity-  
502 enhanced feature-grouped attention network for cross-subject eeg emotion recognition.  
503 *Knowledge-Based Systems*, 283:111199, 2024.
- 504  
505 Zhipeng He, Yongshi Zhong, and Jiahui Pan. An adversarial discriminative temporal convolutional  
506 network for eeg-based cross-domain emotion recognition. *Computers in biology and medicine*,  
507 141:105048, 2022.
- 508 Haoning Kan, Jiale Yu, Jiabin Huang, Zihe Liu, Heqian Wang, and Haiyan Zhou. Self-supervised  
509 group meiosis contrastive learning for eeg-based emotion recognition. *Applied Intelligence*, pp.  
510 1–19, 2023.
- 511  
512 Stamos Katsigiannis and Naeem Ramzan. Dreamer: A database for emotion recognition through  
513 eeg and ecg signals from wireless low-cost off-the-shelf devices. *IEEE journal of biomedical and*  
514 *health informatics*, 22(1):98–107, 2017.
- 515  
516 Guangqiang Li, Ning Chen, Yixiang Niu, Zhangyong Xu, Yuxuan Dong, Jing Jin, and Hongqin Zhu.  
517 Mslte: multiple self-supervised learning tasks for enhancing eeg emotion recognition. *Journal of*  
518 *Neural Engineering*, 21(2):024003, 2024.
- 519  
520 Rui Li, Yiting Wang, Wei-Long Zheng, and Bao-Liang Lu. A multi-view spectral-spatial-temporal  
521 masked autoencoder for decoding emotions with self-supervised learning. In *Proceedings of the*  
522 *30th ACM International Conference on Multimedia*, pp. 6–14, 2022a.
- 523  
524 Xiang Li, Yazhou Zhang, Prayag Tiwari, Dawei Song, Bin Hu, Meihong Yang, Zhigang Zhao,  
525 Neeraj Kumar, and Pekka Marttinen. Eeg based emotion recognition: A tutorial and review. *ACM*  
*Computing Surveys*, 55(4):1–57, 2022b.
- 526  
527 Yang Li, Wenming Zheng, Yuan Zong, Zhen Cui, Tong Zhang, and Xiaoyan Zhou. A bi-hemisphere  
528 domain adversarial neural network model for eeg emotion recognition. *IEEE Transactions on*  
529 *Affective Computing*, 12(2):494–504, 2018.
- 530  
531 Yang Li, Ji Chen, Fu Li, Boxun Fu, Hao Wu, Youshuo Ji, Yijin Zhou, Yi Niu, Guangming Shi,  
532 and Wenming Zheng. Gmss: Graph-based multi-task self-supervised learning for eeg emotion  
533 recognition. *IEEE Transactions on Affective Computing*, 14(3):2512–2525, 2023. doi: 10.1109/  
TAFFC.2022.3170428.
- 534  
535 Shuaiqi Liu, Zeyao Wang, Yanling An, Bing Li, Xinrui Wang, and Yudong Zhang. Da-capsnet: A  
536 multi-branch capsule network based on adversarial domain adaption for cross-subject eeg emotion  
537 recognition. *Knowledge-Based Systems*, 283:111137, 2024.
- 538  
539 Gang Luo, Yutong Han, Weichu Xie, Fuze Tian, Lixian Zhu, Kun Qian, Xiaowei Li, Shuting Sun,  
and Bin Hu. Gcd-jfse: Graph-based class-domain knowledge joint feature selection and ensemble  
learning for eeg-based emotion recognition. *Knowledge-Based Systems*, pp. 112770, 2024.

- 540 Bo-Qun Ma, He Li, Wei-Long Zheng, and Bao-Liang Lu. Reducing the subject variability of eeg  
541 signals with adversarial domain generalization. In *Neural Information Processing: 26th Interna-*  
542 *tional Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings,*  
543 *Part I 26*, pp. 30–42. Springer, 2019.
- 544 Robin Tibor Schirrmester, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin  
545 Glasstetter, Katharina Eggenberger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and  
546 Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization.  
547 *Human brain mapping*, 38(11):5391–5420, 2017.
- 548 Xinke Shen, Xianggen Liu, Xin Hu, Dan Zhang, and Sen Song. Contrastive learning of subject-  
549 invariant eeg representations for cross-subject emotion recognition. *IEEE Transactions on Affec-*  
550 *tive Computing*, 2022.
- 551 Xingyi Wang, Yuliang Ma, Jared Cammon, Feng Fang, Yunyuan Gao, and Yingchun Zhang. Self-  
552 supervised eeg emotion recognition models based on cnn. *IEEE Transactions on Neural Systems*  
553 *and Rehabilitation Engineering*, 31:1952–1962, 2023. doi: 10.1109/TNSRE.2023.3263570.
- 554 Weining Weng, Yang Gu, Shuai Guo, Yuan Ma, Zhaohua Yang, Yuchen Liu, and Yiqiang  
555 Chen. Self-supervised learning for electroencephalogram: A systematic survey. *arXiv preprint*  
556 *arXiv:2401.05446*, 2024.
- 557 Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep  
558 learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):1–42, 2024.
- 559 Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for eeg-  
560 based emotion recognition with deep neural networks. *IEEE Transactions on autonomous mental*  
561 *development*, 7(3):162–175, 2015.
- 562 Wei-Long Zheng and Bao-Liang Lu. Personalizing eeg-based affective models with transfer learn-  
563 ing. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pp.  
564 2732–2738, 2016.
- 565 Peixiang Zhong, Di Wang, and Chunyan Miao. Eeg-based emotion recognition using regularized  
566 graph neural networks. *IEEE Transactions on Affective Computing*, 13(3):1290–1301, 2020.

## 571 A APPENDIX

### 572 A.1 LLM USAGE

573 Large language models were used in this work to polish the writing.

### 574 A.2 DETAILED BASE MODEL AND CLASSIFIER TRAINING

575 As we mentioned above, a variety of training strategies are implemented to encourage the develop-  
576 ment of diverse generalization abilities among the multiple base models. In addition to the original  
577 base model, we construct two other variant models. In the first variant, we employed random mask-  
578 ing of the input. We generate a mask  $m \in [0, 1]^{C \times T}$  for each input  $x \in R^{C \times T}$ . And then, we  
579 obtain the mask sample  $x_{mask} = x \odot m$ , where  $\odot$  represents element-wise multiplication. In the  
580 second variant, we added  $L2$  regularization to the loss and dropout layer to the model structure.  
581 Each base model outputs three components: the latent representation, the reconstructed output, and  
582 the high-dimensional embedding, as follows:

$$\begin{aligned}
 583 \quad h_1, p_1, z_1, \hat{z}_1 &= f_1(x), \\
 584 \quad h_2, p_2, z_2, \hat{z}_2 &= f_2(x_{mask}), \\
 585 \quad h_3, p_3, z_3, \hat{z}_3 &= f_3(x).
 \end{aligned} \tag{2}$$

586 The  $f_1$ ,  $f_2$  and  $f_3$  denote the three base models,  $h_1$ ,  $h_2$  and  $h_3$  denote the learned representations,  
587  $p_1$ ,  $p_2$  and  $p_3$  denote the reconstructed DE features,  $z_1$ ,  $z_2$  and  $z_3$  denote the embeddings of recon-  
588 structions in the high-dimensional space, and  $\hat{z}_1$ ,  $\hat{z}_2$  and  $\hat{z}_3$  denote the embeddings of ground truth  
589 in the high-dimensional space.

And then, the training objective of the base model is to minimize the combination of reconstruction and contrastive loss. Assume the training set has  $N$  samples. Let  $P_j = \{p_j^i | i = 1, 2, \dots, N\}$ ,  $j \in \{1, 2, 3\}$  denote the reconstructions from the  $j$ th model,  $Z_j = \{z_j^i | i = 1, 2, \dots, N\}$ ,  $j \in \{1, 2, 3\}$  denote the embeddings of reconstructions from the  $j$ th model,  $\hat{Z}_j = \{\hat{z}_j^i | i = 1, 2, \dots, N\}$ ,  $j \in \{1, 2, 3\}$  denote the embeddings of ground truth from the  $j$ th model, and  $Y = \{y^i | i = 1, 2, \dots, N\}$  denote the ground truth DE feature set.

$$\begin{aligned} L_1 &= \min_{\theta_1} L_{mse}(P_1, Y) + \beta L_{cont}(Z_1, \hat{Z}_1), \\ L_2 &= \min_{\theta_2} L_{mse}(P_2, Y) + \beta L_{cont}(Z_2, \hat{Z}_2), \\ L_3 &= \min_{\theta_3} L_{mse}(P_3, Y) + \beta L_{cont}(Z_3, \hat{Z}_3) + \lambda \|\theta_3\|_2^2. \end{aligned} \quad (3)$$

The  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  denote the parameters of the  $f_1$ ,  $f_2$ , and  $f_3$ .  $L_{mse}$  denotes the mean squared error (MSE),  $L_{cont}$  denotes the contrastive loss,  $\beta$  is a hyperparameter that controls the balance of reconstruction loss and contrastive loss, and  $y$  denotes the ground truth DE feature of  $x$ .

As for the contrastive loss, for each model, we merge the two sets  $Z_j$  and  $\hat{Z}_j$  into a new set  $\bar{Z}_j$  with a size of  $2N$ . In  $\bar{Z}_j$ , we combine reconstruction embedding  $\bar{z}_j^{2k-1}$  and ground truth embedding  $\bar{z}_j^{2k}$ ,  $k \in \{1, \dots, N\}$ , as positive pairs. We use  $K = \{1, \dots, 2N\}$  to denote the index set. The  $p_j(k)$  denotes the index of the positive sample of  $\bar{z}_j^k$ . Subsequently, the contrastive loss  $L_{cont}$  is defined as shown in the following.

$$L_{cont}(Z_j, \hat{Z}_j) = - \sum_{k \in K} \log \frac{\exp(\bar{z}_j^k \cdot \bar{z}_j^{p_j(k)} / \tau)}{\sum_{a \in A(k)} \exp(\bar{z}_j^k \cdot \bar{z}_j^a / \tau)}, \quad (4)$$

where  $A_j(k) = \{a | a \in K, \bar{z}_j^a \neq \bar{z}_j^k\}$ , and  $\tau$  is a scalar temperature parameter.

In stage 2, we set up a classifier for each base model to establish a mapping from the representations to the emotional labels.

$$e_j = \text{softmax}(clf_j(h_j)). \quad (5)$$

In this process, the base models are frozen, and the classifiers are trained to minimize the classification loss. Here, we adopt the cross-entropy loss to optimize the classifiers.

### A.3 DETAILED ONLINE UPDATING

We implement a 'model specialization and mutual assistance' style online parameter update mechanism. This mechanism can be regarded as a deepening of the multi-model online collaboration conception: (1) 'Specialization' identifies and promotes the most competent base models to refine their representational learning capabilities through test data streams; and (2) 'Mutual assistance' assists weaker classifiers in adapting to decision boundary drift by leveraging high-confidence predictors to propagate knowledge across the model ensemble. Collectively, this mechanism ensures synchronized evolution during deployment, substantially enhancing robustness to continuous distribution drifts.

We first introduce the base model parameter update strategy, i.e., model specialization. This approach encourages base models to further adapt to their respective specialized data distributions through online updates, thereby reinforcing the functional differentiation among them. For this reason, we adopt a multi-memory bank strategy that preserves distinct representational characteristics across models. Each base model  $f_j$  maintains an independent memory bank  $M_j^h$  implemented as a fixed-size FIFO queue with capacity  $C_M$ . This design ensures constant memory usage while recording the hard and most recent samples. When the  $f_j$  is activated, the memory bank stores raw test EEG input  $x'$  that represents hard samples into the  $M_j^h$  during inference. We use the  $M_s$  to denote the memory bank of the selected model. The memory update policy follows a selective strategy where only samples exceeding a dynamic threshold are included:

$$M_s^h \leftarrow \text{Push}(M_s^h, x') \quad \text{if} \quad R_s > \gamma_s^{\text{dy}} \quad (6)$$

where  $\gamma_s^{\text{dy}}$  represents the adaptive threshold for model  $f_s$ .

The threshold  $\gamma_s^{\text{dynamic}}$  rules the selection of hard samples. It is computed dynamically based on each model’s recent performance history as the activated base model. We maintain a sliding window  $W_j$  that records the most recent  $K$  routing scores for each model when it is activated. The adaptive threshold for the activated base model  $f_s$  is computed as:

$$\gamma_s^{\text{dy}} = Q_{p_1}(W_s), \quad (7)$$

where  $Q_{p_1}$  denotes the  $p_1$ -th percentile of the scores in the window. This adaptive threshold ensures that the system maintains sensitivity to distribution drifts while remaining robust to occasional outliers.

The fine-tuning process is initiated through an evidence-based triggering mechanism. Let  $T_h$  denote the threshold number of new hard samples required to trigger fine-tuning. For each model  $f_j$ , we maintain a counter  $c_j$  that increments when a new hard sample is added to  $M_j^h$ . Fine-tuning is triggered when sufficient evidence of distribution drift has been accumulated:

$$c_j \leftarrow c_j + 1, \quad \text{TriggerFineTune()} \quad \text{iff} \quad c_s \geq T_{h1}. \quad (8)$$

Upon triggering,  $c_j$  is reset to zero, and the fine-tuning process randomly samples from  $M_s^h$  to form the fine-tuning batch  $B_{ft}^{\text{enc}}$ . The contrastive loss is computed using the stored samples, same as Eq. equation 4:

$$\begin{aligned} Z', \hat{Z}' &\leftarrow f_s(B_{ft}^{\text{enc}}), \\ L_{ft}^{\text{enc}} &= L_{\text{cont}}(Z', \hat{Z}'). \end{aligned} \quad (9)$$

Our parameter unfreezing strategy adheres to the following principles. Since lower layers capture generic features, we primarily fine-tune higher-level components. To maintain computational efficiency and mitigate catastrophic forgetting, only a minimal set of parameters is updated. Furthermore, we selectively unfreeze the components most sensitive to current distributional drifts. Consequently, during fine-tuning, we unfreeze and update the following parameters:

$$\theta_s^{\text{enc}} = \{\theta_s^{\text{projector}}, \theta_s^{\text{norm}}, W_s^{\text{out}}, b_s^{\text{out}}\} \subset \theta_s. \quad (10)$$

These parameters are optimized using the Adam optimizer with a reduced learning rate  $\alpha_{enc}^{\text{ft}}$ :

$$\theta_s^{\text{enc}} \leftarrow \theta_s^{\text{enc}} - \alpha_{enc}^{\text{ft}} \cdot \nabla_{\theta_s^{\text{enc}}} L_{ft}^{\text{enc}} \quad (11)$$

Besides, routing scores are derived from the self-supervised task (reconstruction-contrastive) and are inherently tied to proxy tasks. They fail to directly reflect decision confidence in the downstream emotion classification task. Therefore, we also need to optimize classifiers online to address the issue of decision boundary adaptation. We introduce a mutual assistance mechanism that utilizes high-confidence predictions from strong models as pseudo labels to optimize other classifiers exhibiting prediction inconsistencies.

We retain both the logits and predicted labels from the selected classifier  $clf_s$ :

$$\begin{aligned} g_s &= clf_s(h_s), \\ e'_s &= \text{softmax}(g_s). \end{aligned} \quad (12)$$

where  $h_s$  is the representation from the  $f_s$ ,  $g_s$  is the logit vector, and  $e'_s$  is the issued prediction. Following [NeurIPS20], we adopt an energy-based function to measure the confidence score of the selected classifier.

$$E_s = -T \cdot \log \sum_{c=1}^C \exp\left(\frac{g_s[c]}{T}\right), \quad (13)$$

where  $C$  is the number of classes,  $g_s[c]$  are the logits for class  $c$  from the classifier  $clf_s$ , and  $T$  is a temperature scaling factor. During training, we compute and store all the energy scores for three classifiers,  $H_j^E$ , in the training dataset. In the testing phase, the  $W_s^E$  helps us identify the dynamic threshold for high-confidence prediction.

If the energy score  $E_s$  falls below the  $p_2$ -th percentile of  $H_s^E$ , we consider the classifier’s prediction to be highly confident.

$$E_s < \tau_s^{\text{dy}}, \tau_s^{\text{dy}} = Q_{p_2}(H_s^E). \quad (14)$$

Then the predictions are used as pseudo-labels to supervise the online update of other classifiers. Analogous to the earlier design, we maintain a dedicated pseudo-label dynamic memory  $M_j^p$  bank for each classifier. A consistency check is performed to identify the inconsistent classifiers. When a pseudo-label disagrees with a classifier’s own prediction, the classifier is regarded as the inconsistent classifier. And the pseudo-label is pushed into its pseudo-label memory bank.

$$M_j^p \leftarrow \text{Push}(M_j^p, (e'_s, h'_j)) \quad \text{if } e'_j \neq e'_s, \quad (15)$$

where  $e'_j$  are the predictions issued by the other  $clf_j$ , and  $h'_j$  are the representations issued by the  $f_j$ .

Once a memory bank accumulates  $T_{h2}$  new samples, we sample a batch  $B_{ft}^{clf}$  from it and optimize the corresponding classifier  $clf_j$  using the cross-entropy loss.

$$\begin{aligned} L_{clf}^{ft} &= \text{Crossentropy}(B_{ft}^{clf}(e'_s), clf_j(B_{ft}^{clf}(h'_j))), \\ \theta_j^{clf} &\leftarrow \theta_j^{clf} - \alpha_{clf}^{ft} \cdot \nabla_{\theta_j^{clf}} L_{ft}^{clf}, \end{aligned} \quad (16)$$

where  $L_{ft}^{clf}$  is the loss for fine-tuning, and  $\theta_j^{clf}$  are the parameters of the classifier  $clf_j$ . Merely activating appropriate models proves insufficient for long-term adaptation to data distribution drifts. We therefore require base models and classifiers to update their network parameters in response to the test data stream. To achieve this, we implement a ‘model specialization and mutual assistance’ style online parameter update mechanism. This mechanism can be regarded as a deepening of the multi-model online collaboration conception: (1) ‘Specialization’ identifies and promotes the most competent base models to refine their representational learning capabilities through test data streams; and (2) ‘Mutual assistance’ assists weaker classifiers in adapting to decision boundary drift by leveraging high-confidence predictors to propagate knowledge across the model ensemble. Collectively, this mechanism ensures synchronized evolution during deployment, substantially enhancing robustness to continuous distribution drifts.

We first introduce the base model parameter update strategy, i.e., model specialization. This approach encourages base models to further adapt to their respective specialized data distributions through online updates, thereby reinforcing the functional differentiation among them. For this reason, we adopt a multi-memory bank strategy that preserves distinct representational characteristics across models. Each base model  $f_j$  maintains an independent memory bank  $M_j^h$  implemented as a fixed-size FIFO queue with capacity  $C_M$ . This design ensures constant memory usage while recording the hard and most recent samples. When the  $f_j$  is activated, the memory bank stores raw test EEG input  $x'$  that represents hard samples into the  $M_j^h$  during inference. We use the  $M_s$  to denote the memory bank of the selected model. The memory update policy follows a selective strategy where only samples exceeding a dynamic threshold are included:

$$M_s^h \leftarrow \text{Push}(M_s^h, x') \quad \text{if } R_s > \gamma_s^{\text{dy}} \quad (17)$$

where  $\gamma_s^{\text{dy}}$  represents the adaptive threshold for model  $f_s$ .

The threshold  $\gamma_s^{\text{dynamic}}$  rules the selection of hard samples. It is computed dynamically based on each model’s recent performance history as the activated base model. We maintain a sliding window  $W_j$  that records the most recent  $K$  routing scores for each model when it is activated. The adaptive threshold for the activated base model  $f_s$  is computed as:

$$\gamma_s^{\text{dy}} = Q_{p1}(W_s), \quad (18)$$

where  $Q_{p1}$  denotes the  $p_1$ -th percentile of the scores in the window. This adaptive threshold ensures that the system maintains sensitivity to distribution drifts while remaining robust to occasional outliers.

The fine-tuning process is initiated through an evidence-based triggering mechanism. Let  $T_h$  denote the threshold number of new hard samples required to trigger fine-tuning. For each model  $f_j$ , we maintain a counter  $c_j$  that increments when a new hard sample is added to  $M_j^h$ . Fine-tuning is triggered when sufficient evidence of distribution drift has been accumulated:

$$c_j \leftarrow c_j + 1, \quad \text{TriggerFineTune}() \quad \text{iff } c_s \geq T_{h1}. \quad (19)$$

756 Upon triggering,  $c_j$  is reset to zero, and the fine-tuning process randomly samples from  $M_s^h$  to form  
 757 the fine-tuning batch  $B_{ft}^{enc}$ . The contrastive loss is computed using the stored samples, same as  
 758 Eq. equation 4:

$$\begin{aligned} 759 Z', \hat{Z}' &\leftarrow f_s(B_{ft}^{enc}), \\ 760 L_{ft}^{enc} &= L_{cont}(Z', \hat{Z}'). \end{aligned} \quad (20)$$

761 Our parameter unfreezing strategy adheres to the following principles. Since lower layers cap-  
 762 ture generic features, we primarily fine-tune higher-level components. To maintain computational  
 763 efficiency and mitigate catastrophic forgetting, only a minimal set of parameters is updated. Fur-  
 764 thermore, we selectively unfreeze the components most sensitive to current distributional drifts.  
 765 Consequently, during fine-tuning, we unfreeze and update the following parameters:

$$766 \theta_s^{enc} = \{\theta_s^{projector}, \theta_s^{norm}, W_s^{out}, b_s^{out}\} \subset \theta_s. \quad (21)$$

767 These parameters are optimized using the Adam optimizer with a reduced learning rate  $\alpha_{enc}^{ft}$ :

$$768 \theta_s^{enc} \leftarrow \theta_s^{enc} - \alpha_{enc}^{ft} \cdot \nabla_{\theta_s^{enc}} L_{ft}^{enc} \quad (22)$$

769 Besides, routing scores are derived from the self-supervised task (reconstruction-contrastive) and  
 770 are inherently tied to proxy tasks. They fail to directly reflect decision confidence in the downstream  
 771 emotion classification task. Therefore, we also need to optimize classifiers online to address the issue  
 772 of decision boundary adaptation. We introduce a mutual assistance mechanism that utilizes high-  
 773 confidence predictions from strong models as pseudo labels to optimize other classifiers exhibiting  
 774 prediction inconsistencies.

775 We retain both the logits and predicted labels from the selected classifier  $clf_s$ :

$$\begin{aligned} 776 g_s &= clf_s(h_s), \\ 777 e'_s &= \text{softmax}(g_s). \end{aligned} \quad (23)$$

778 where  $h_s$  is the representation from the  $f_s$ ,  $g_s$  is the logit vector, and  $e'_s$  is the issued prediction.  
 779 Following [NeurIPS20], we adopt an energy-based function to measure the confidence score of the  
 780 selected classifier.

$$781 E_s = -T \cdot \log \sum_{c=1}^C \exp\left(\frac{g_s[c]}{T}\right), \quad (24)$$

782 where  $C$  is the number of classes,  $g_s[c]$  are the logits for class  $c$  from the classifier  $clf_s$ , and  $T$  is  
 783 a temperature scaling factor. During training, we compute and store all the energy scores for three  
 784 classifiers,  $H_j^E$ , in the training dataset. In the testing phase, the  $W_s^E$  helps us identify the dynamic  
 785 threshold for high-confidence prediction.

786 If the energy score  $E_s$  falls below the  $p_2$ -th percentile of  $H_s^E$ , we consider the classifier’s prediction  
 787 to be highly confident.

$$788 E_s < \tau_s^{dy}, \tau_s^{dy} = Q_{p_2}(H_s^E). \quad (25)$$

789 Then the predictions are used as pseudo-labels to supervise the online update of other classifiers.  
 790 Analogous to the earlier design, we maintain a dedicated pseudo-label dynamic memory  $M_j^p$  bank  
 791 for each classifier. A consistency check is performed to identify the inconsistent classifiers. When a  
 792 pseudo-label disagrees with a classifier’s own prediction, the classifier is regarded as the inconsistent  
 793 classifier. And the pseudo-label is pushed into its pseudo-label memory bank.

$$800 M_j^p \leftarrow \text{Push}(M_j^p, (e'_s, h'_j)) \quad \text{if } e'_j \neq e'_s, \quad (26)$$

801 where  $e'_j$  are the predictions issued by the other  $clf_j$ , and  $h'_j$  are the representations issued by the  
 802  $f_j$ .

803 Once a memory bank accumulates  $T_{h_2}$  new samples, we sample a batch  $B_{ft}^{clf}$  from it and optimize  
 804 the corresponding classifier  $clf_j$  using the cross-entropy loss.

$$\begin{aligned} 805 L_{clf}^{ft} &= \text{Crossentropy}(B_{ft}^{clf}(e'_s), clf_j(B_{ft}^{clf}(h'_j))), \\ 806 \theta_j^{clf} &\leftarrow \theta_j^{clf} - \alpha_{clf}^{ft} \cdot \nabla_{\theta_j^{clf}} L_{ft}^{clf}, \end{aligned} \quad (27)$$

807 where  $L_{ft}^{clf}$  is the loss for fine-tuning, and  $\theta_j^{clf}$  are the parameters of the classifier  $clf_j$ .

#### 810 A.4 DETAILED BASE MODEL AND CLASSIFIER STRUCTURE

811  
812 Our model consists of two main components: a Differential Entropy (DE) reconstruction module  
813 and a contrastive learning module. The DE reconstruction module aims to learn multi-band DE  
814 representations from raw EEG signals across multiple frequency bands. The contrastive learning  
815 module, on the other hand, is designed to align the learned representations with their corresponding  
816 ground truth DE values through an encoder-projector architecture based on self-attention.

817 The DE reconstruction module employs a multi-branch depth-wise 1-D convolutional structure with  
818 different kernel sizes to capture multi-scale features from the input raw EEG data  $x$ . Each branch  
819 processes the input using a depth-wise convolution.

$$820 g_b = \text{Conv1D}_{\text{depth-wise}}(x; k_b), \quad (28)$$

821 where  $k_b$  denotes the kernel size of the  $b$ -th branch. After feature extraction, each branch is passed  
822 through a Multi-Layer Perceptron (MLP) to map the extracted features into DE values for that spe-  
823 cific frequency band:

$$824 p_b = \text{MLP}(g_b), \quad (29)$$

825 where  $p_b$  represents the reconstructed DE values for the  $i$ -th frequency band. Finally, the DE values  
826 from all branches are concatenated to form the final multi-band DE reconstruction:

$$827 p = \text{concat}(\{p_b\}), p \in \mathbb{R}^{C \times d_b}, \quad (30)$$

828 where  $d_b$  denotes the number of frequency bands.

829 The contrastive learning module takes both the reconstructed DE values  $p$  and the ground truth DE  
830 values  $y$  as inputs. It consists of an encoder and a projector. The encoder is built upon a self-attention  
831 mechanism. First, we perform the learnable position coding and linear layer mapping on the inputs:

$$832 E = p_{in} + P, \quad (31)$$

833 where  $p_{in}$  is either reconstruction  $p$  or ground truth  $y$ , and  $P$  denotes the positional encoding ma-  
834 trix. Then, a self-attention mechanism is applied along the channel dimension to capture spatial  
835 dependencies:

$$836 Q = EW_Q^\top, \quad K = EW_K^\top, \quad V = EW_V^\top;$$

$$837 A = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V;$$

$$838 H_1 = \text{LayerNorm}(A + E); \quad (32)$$

$$839 H_2 = \text{ReLU}(H_1W_{\text{ff1}} + b_{\text{ff1}})W_{\text{ff2}} + b_{\text{ff2}};$$

$$840 H_3 = \text{LayerNorm}(H_2 + H_1);$$

$$841 H = H_3W_{\text{out}} + b_{\text{out}}, W_{\text{out}} \in \mathbb{R}^{d_k \times d_b}.$$

842  $W$  with various subscripts are learnable parameters, and  $b$  with various subscripts are biases.  $d_k$  is  
843 the hidden dimension.

844 Since the contrastive module takes both the reconstructed output and the ground truth as inputs,  
845 it generates two corresponding representations,  $H$  and  $\hat{H}$ , respectively. An MLP architecture is  
846 utilized to project the learned representations  $H$  and  $\hat{H}$  into a high-dimensional space, resulting in  
847 embeddings  $z$  and  $\hat{z}$ .

$$848 z_1 = \text{ReLU}(H_{\text{eith}}W_1 + b_1), W_1 \in \mathbb{R}^{(C \times d_b) \times d_{em1}};$$

$$849 z_{\text{eith}} = z_1W_2 + b_2, W_2 \in \mathbb{R}^{d_{em1} \times d_{em2}}, \quad (33)$$

850 where  $H_{\text{eith}}$  is either  $H$  or  $\hat{H}$ , and  $z_{\text{eith}}$  is either  $z$  or  $\hat{z}$ .

851 These two representations are then flattened and concatenated to form the final representation  $h$ .

852 We adopt a three-layer MLP as the classifier. The rectified linear unit (ReLU) activation functions  
853 are applied between each pair of consecutive layers. The numbers of hidden units in the two hidden  
854 layers are denoted as  $d_{h1}$  and  $d_{h2}$ , respectively.

## A.5 IMPLEMENTATION DETAILS

To assess the inter-subject generalization performance of our method, we employ a leave-one-subject-out cross-validation (LOSOCV) evaluation protocol. In this protocol, data from one subject are used as the test set while the remaining data serve as the training set. This process is repeated iteratively so that each subject acts as the test set once. Final results are reported as the average accuracy and standard deviation across all folds, providing a robust estimate of subject-independent performance.

Table 3: The Performance of Each Base Model

Method	Dreamer		SEED
	Arousal	Valence	
Base model 1	66.10±6.35	67.01±7.39	82.18±6.97
Base model 2	65.85±8.08	66.89±8.81	81.93±7.33
Base model 3	66.38±6.98	67.79±7.77	83.39±6.87
MMOC	69.37±6.13	70.33±6.78	86.39±5.41

All experiments are conducted on an NVIDIA RTX 2080Ti GPU. During training, the temperature parameter  $\tau$  is set to 0.07, L2 penalty  $\lambda = 10^{-5}$ , and the balancing factor  $\beta = 0.9$ . The DE features are calculated on the  $\delta$ ,  $\theta$ ,  $\alpha$ ,  $\beta$ , and  $\gamma$  frequency bands for SEED, and  $\theta$ ,  $\alpha$ , and  $\beta$  for Dreamer. Thus, the  $d_b = 5$  for SEED, and the  $d_b = 3$  for Dreamer. The kernel sizes of  $k_b$  are  $\in \{401, 51, 25, 15, 7\}$ , corresponding to the five bands respectively. The  $d_k = 64$ ,  $d_{em1} = 256$ ,  $d_{em2} = 128$ ,  $d_{h1} = 62$ , and  $d_{h2} = 30$ . The batch size is set to 128, and the base models are trained for 200 epochs with early stop. The base models are optimized using the Adam optimizer with a learning rate of 0.001. For the classification training, the Adam optimizer minimizes the cross-entropy loss with a learning rate of 0.0001 over 50 training epochs.

Additionally, we detail the hyperparameters governing the online inference and adaptation process. For the routing mechanism, the sliding window size for tracking recent scores  $K$  is set to 32, and the percentile  $p_1$  for calculating the dynamic threshold  $\gamma_s^{dy}$  is 85. Each model’s memory bank  $M_s$  has a capacity  $C_M$  of 16 for Dreamer and 32 for SEED. The fine-tuning process is triggered when the counter of new hard samples  $c_s$  reaches the threshold  $T_{h1} = 4$  for Dreamer and 8 for SEED, at which point a batch of  $B_{ft} = 16$  for Dreamer and 32 for SEED samples is randomly drawn for encoder updating. The fine-tuning learning rate  $\alpha_{enc}^{ft}$  is set to  $5 \times 10^{-5}$ . For classifier mutual assistance, the energy score percentile  $p_2$  is 15, and the update threshold  $T_{h2}$  is also 4 for Dreamer and 8 for SEED. The fine-tuning learning rate  $\alpha_{clf}^{ft}$  is set to  $1 \times 10^{-5}$ .

## A.6 BASE MODEL PERFORMANCE EVALUATION

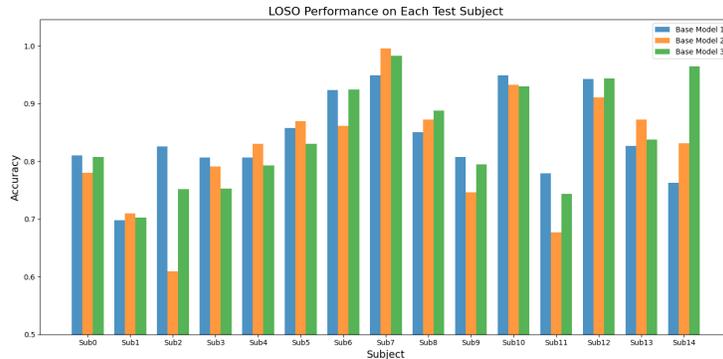


Figure 6: The LOSOCV performance of the base models on each test subjects.

In this work, we aim to propose a multi-model online collaboration framework for self-supervised learning along with its corresponding routing mechanism. To achieve this, we integrate multiple

918 base models into our framework. In this case, it is necessary to demonstrate that the performance  
919 improvements observed in MMOC stem from the online collaboration of multiple models rather  
920 than merely the robustness of individual base models. Consequently, we compare the performance  
921 of MMOC against each individual base model, as illustrated in Table 3.

922 As can be seen, the MMOC framework based on the three base models achieves better cross-subject  
923 performance than any individual model alone. This demonstrates that MMOC effectively integrates  
924 diverse base models and leverages its routing strategy to enhance performance on test samples.  
925

926 Moreover, we present the accuracy of each base model across different test subjects on the SEED  
927 dataset in Figure 6. It can be observed that the base models exhibit obvious performance differences  
928 on some test subjects. For example, on Sub 2, Base Model 1 achieves the best performance, while  
929 Base Model 2 performs significantly worse than the others. On Sub 14, Base Model 3 yields the  
930 highest accuracy, Base Model 2 achieves the second-best result, and Base Model 1 performs the  
931 worst. This observation demonstrates that our diverse training strategy effectively differentiates the  
932 generalization capabilities of the base models. The observed variation in generalization performance  
933 across base models under different data distributions supports the rationale for model collaboration.  
934 By combining multiple models with complementary strengths, the framework achieves broader cov-  
935 erage of unknown distributions.  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971