**S.1 Model Architecture and Implementation of CVAE-GM**

**CVAE-GM Model**

The CVAE-GM model is designed to address the limitations of traditional Variational Autoencoders (VAE) in processing complex heterogeneous data by integrating a learnable Gaussian mixture prior and contrastive learning. This enhancement improves the model's generative capabilities and accuracy in disease classification tasks. By effectively capturing multimodal features and strengthening the correlation between features and labels, the model facilitates more precise subtype identification. This approach synergistically combines effective posterior distribution approximation via variational inference with the capacity of Gaussian mixture models to capture intricate data characteristics, ensuring that CVAE-GM can effectively discern and differentiate various features in highly heterogeneous medical datasets.

1. **Encoder**

To enhance the representational capacity of multi-band EEG functional connectivity, the CVAE-GM framework incorporates a data augmentation module (AugGen) within the encoder. AugGen generates structure-perturbed views for each frequency band, simulating natural variability in brain networks and improving encoding robustness. Inspired by graph contrastive learning methods (e.g., GraphCL1), it enables frequency-specific feature extraction via multi-view representations. The augmented graphs are encoded into latent variables for downstream representation learning and subtype modeling. The encoder’s core function is to map inputs to a latent probability distribution by learning its parameters (mean and variance), enabling effective and controllable modeling of complex brain data, as formulated below:

(1)

Where *x* represents the input EEG data, *z* denotes the latent variable distribution parameters, is the mean vector, and is the log variance vector corresponding to the latent variables derived from the input data *x*. The VAE samples the latent variable *z* from the distribution generated by the encoder for use in the decoder; however, this sampling process is non-differentiable and thus cannot be optimized directly. To address this limitation, the VAE employs the reparameterization trick, which transforms the sampling procedure into a differentiable operation, facilitating effective backpropagation. The sampling principle is as follows:

(2)

Where represents random noise sampled from a standard normal distribution, while and are the mean and standard deviation outputs from the encoder. The encoder generates the distribution parameters for the latent space and ensures that these parameters are as close as possible to the prior distribution. To address the issue of inadequate distribution matching in traditional methods, the VAE introduces variational inference, performing approximate inference by optimizing the Evidence Lower Bound (ELBO). The ELBO comprises a reconstruction loss that ensures the generated data is similar to the input data, along with the KL divergence, which measures the difference between the latent space distribution and the prior distribution. By combining these two components, the VAE effectively balances data reconstruction and distribution matching, enhancing the model's generative and representational capabilities. The principle of KL divergence is as follows:

(3)

(4)

Here, *d* represents the dimensionality of the latent variables. The role of the KL divergence is to ensure that the latent space distribution generated by the encoder is as close as possible to the prior distribution.

1. **Gaussian Mixture Latent Space**

The Gaussian mixture latent space captures multiple underlying patterns by representing latent variables as a linear combination of several Gaussian distributions. This architecture allows the model to express different distribution patterns within a unified latent space, effectively accommodating the complexities of diverse data characteristics. In multimodal data, each Gaussian component can represent distinct categories or features, enabling the model to generate a wide variety of data samples by learning the weights and parameters associated with each component. This results in enhanced flexibility and capability. The distribution of the latent variable *z* can be expressed as follows:

(5)

(6)

Here, *K* represents the number of mixture components, ​ denotes the weight of each Gaussian component, satisfying the condition . represents a Gaussian distribution with mean ​ and covariance ​, while *y* denotes the microstate label. The introduction of variational inference in the Gaussian mixture latent space aims to effectively approximate the posterior distribution by optimizing the evidence lower bound (ELBO), thereby enhancing the model's ability to capture multimodal characteristics of the data and improving its generative capability. The underlying principles are as follows:

(7)

Here, represents the inference network, which optimizes the variational lower bound by minimizing the KL divergence, as follows:

(8)

Here, represents the approximate true posterior distribution.

1. **Decoder**

The core task of the decoder is to transform the latent variable *z* and the conditional label *y* into generated data *x*, ensuring that the generated data is closely related to the conditional information for effective conditional generation. In CVAE-GM, the decoder is closely integrated with the Gaussian mixture model and significantly enhances the accuracy and diversity of generation through contrastive learning, helping the model better capture complex data distributions. The following presents the computational principles and key formulas of the decoder.

(9)

(10)

Where ​​ is the neural network for generating the mean, and ​​ is the neural network for generating the covariance. The reconstruction loss is used to maximize the similarity between the generated data and the real data, and its principle is as follows:

(11)

Combining with the KL divergence, the overall loss function is as follows:

(12)

1. **Contrastive Learning Module**

In CVAE-GM, the introduction of the contrastive learning module aims to enhance the correlation between features and labels, thereby improving the representation ability of multi-frequency data. Contrastive learning leverages the similarity between samples, bringing samples with the same labels closer together in the embedding space while pushing apart samples with different labels, making it particularly suitable for complex data such as EEG signals. Specifically, given the functional activity of a certain frequency band, the goal is to bring samples with similar energy fluctuations closer while distancing those with different fluctuations. This is achieved through noise contrastive estimation (NCE), facilitating the learning of specific brain functional network patterns. Embedding multi-frequency contrastive learning into the VAE framework enhances the expressiveness of self-supervised learning and effectively captures correlations between samples in a data-driven manner, overcoming the limitations of forced comparisons between labels in traditional VAE-GM. The decoder function decodes samples from the latent space into feature embeddings . We jointly learn the microstate label embeddings alongside . The contrastive loss function is as follows:

(13)

Here, *A* represents different frequency band comparisons, such as delta and theta, delta and alpha, as well as beta and gamma. The function measures the similarity between two embeddings, defined as , where is a parameter that controls the scaling of the dot product. and represent the feature embeddings and label embeddings, respectively. Our objective function also includes a supervised cross-entropy loss term, which is formulated as follows:

(14)

Here, is the activation function (sigmoid function). We use an objective function that encompasses all loss components to train the model. In summary, the ultimate objective to minimize is the total sum of the different losses, expressed as follows:

(15)

Here, *α* and *β* are weighting parameters. The model is trained using the Adam optimizer, optimizing *L*, and is evaluated across four different metrics: V-measure, normalized mutual information (NMI), adjusted Rand index (ARI), and silhouette coefficient (SC).

**Training Configuration of the CVAE-GM Model**

The CVAE-GM model was implemented in PyTorch and trained on an NVIDIA RTX 3090 GPU. Input features were absolute EEG power across six frequency bands (δ, slow θ, fast θ, α, β, γ), concatenated as multi-band vectors. Both the encoder and decoder consisted of three fully connected layers, with a latent space dimension of 64. A five-component Gaussian mixture prior was used, and a frequency-aware contrastive learning module was integrated with a two-layer MLP projection head (τ = 0.07).

The total loss combined reconstruction loss, KL divergence, contrastive loss, and cross-entropy loss, weighted by α = 0.5 and β = 1.0. The model was trained using the Adam optimizer (learning rate = 1e-3, batch size = 128, epochs = 200), with learning rate adjustment via Reduce the Learning Rate on a Plateau (patience = 10, factor = 0.5) and early stopping (patience = 20). Five-fold cross-validation was conducted on the internal dataset, and generalization was assessed on an independent external dataset. Evaluation metrics included ARI, NMI, V-measure, silhouette score, accuracy, sensitivity, and specificity.

**External Validation and Performance Evaluation**

To evaluate generalizability, the model was validated on an independent test set (n = 530) distinct from the training set (n = 1,419). Performance metrics including accuracy, sensitivity, and specificity were computed, and five-fold cross-validation was employed to enhance robustness. Comparative experiments with a baseline VAE model were conducted to assess improvements introduced by CVAE-GM.

**Clustering of latent functional network representations**

We used a conditional variational autoencoder (CVAE) model to extract individual-level latent representations of functional network features across the internal cohort (n = 1419). These representations were then subjected to K-means clustering to identify potential subtypes based on network distribution patterns. Clustering performance was evaluated using the Calinski–Harabasz (CH) index. The three-cluster solution (K=3) yielded the highest CH score (>250), suggesting optimal between-cluster separation and within-cluster cohesion (supplementary figure 1A).

To further explore the clustering dynamics, we visualised the distribution of network intensity within each cluster. The two-cluster solution resulted in a bimodal distribution, indicative of partial segregation between subgroups. In contrast, the three-cluster model revealed unimodal distributions within all clusters, supporting improved cluster stability and interpretability (supplementary figure 1B). These findings indicate that the CVAE-derived latent features captured biologically meaningful variability and allowed stable identification of participant subgroups based on functional network signatures.

[**Insert supplementary figure 1 near here**]

**S.2 Supplementary Result: Cortical network patterns of healthy controls and SSD subtypes**

Supplementary Figure 2 illustrates cortical network patterns in healthy controls (HCs) and the three subtypes of somatic symptom disorder (SSD) identified by the contrastive VAE clustering framework. **(A)** In HCs, cortical activation predominantly involved canonical regions of the *default mode network (DMN)*, including the medial prefrontal cortex (mPFC), posterior cingulate cortex (PCC), and precuneus, serving as a normative reference template for SSD subtypes. **(B)** In contrast, SSD patients exhibited three distinct subtype-specific cortical network alterations. Cluster 1 showed dominant disruptions within the *somatomotor network (SMN)*; Cluster 2 involved the *central executive network (CEN)*; and Cluster 3 revealed pronounced abnormalities within the *limbic network (LN)*. These subtype-specific deviation patterns highlight distinct neurobiological mechanisms underlying SSD heterogeneity relative to the healthy control network architecture.

[**Insert supplementary figure 2 near here**]

**Supplementary Reference**

1. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y. Graph contrastive learning with augmentations. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*). Curran Associates Inc. (2020).

图表

AI 生成的内容可能不正确。

**Supplementary Figure 1. Cluster number selection, subtype separation, and multi-frequency consistency evaluation.** (A) Model-based clustering with different numbers of clusters. The Calinski-Harabasz method suggests the optimal cluster number (K); (B) Clustering at K =2 and corresponding scalp network strength distributions. (C) Clustering at K=3, showing enhanced subtype separability with distribution histograms and overlap (OVL) indices across subtypes, indicating reduced distributional overlap. (D) Multi-frequency consistency analysis across delta, theta, alpha, beta, and gamma bands. Heatmaps report subtype stability metrics, including V-measure, ARI, NMI, and silhouette coefficient (SC), demonstrating consistent clustering solutions across frequency bands.

日历

AI 生成的内容可能不正确。

**Supplementary Figure 2. Cortical network patterns of healthy controls and SSD subtypes.** (A) Cortical activation maps of healthy controls, primarily reflecting the **default mode network (DMN)**, with representative regions such as the medial prefrontal cortex (mPFC) and posterior cingulate cortex (PCC), serve as a reference template for patient subtypes. (B) Cortical activation patterns of the three SSD subtypes identified by contrastive VAE clustering. **Cluster 1** shows dominant alterations in the somatomotor network (SMN), **Cluster 2** in the central executive network (CEN), and **Cluster 3** in the limbic network (LN). Subtype-specific deviations from the healthy control template highlight distinct neurobiological underpinnings of SSD heterogeneity.