

Debiased Contrastive Learning with
multi-resolution Kolmogorov-Arnold Network for
Gravitational Wave Glitch Detection **Appendix**

1 Wasserstein distance and Wasserstein loss L_{wass} computation

The direct computation of the Wasserstein distance can be challenging due to its optimization over the space of joint distributions. We implement the Wasserstein distance more practically by using Sinkhorn divergence [4]. Sinkhorn divergence introduces entropy regularization to the optimal transport problem, making it efficient to compute while retaining sensitivity to distributional characteristics.

Given two distributions μ and ν , with corresponding samples $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$, the Sinkhorn divergence is defined as:

$$W_\epsilon(\mu, \nu) = \min_{\gamma \in \Gamma(\mu, \nu)} \sum_{i,j} \gamma_{ij} c(x_i, y_j) + \epsilon \cdot \text{KL}(\gamma \| \mu \otimes \nu), \quad (1)$$

where:

- $\Gamma(\mu, \nu)$: Set of all joint distributions with marginals μ and ν .
- γ_{ij} : Transport plan between x_i and y_j .
- $c(x_i, y_j)$: Cost function, often the squared Euclidean distance $\|x_i - y_j\|^2$.
- ϵ : Regularization parameter for entropy smoothing.
- $\text{KL}(\gamma \| \mu \otimes \nu)$: Kullback-Leibler divergence regularizing the transport plan.
- $\mu \otimes \nu$: Independent product of the distributions μ and ν .

1.1 Wasserstein loss $L_{\text{wass}}(f(x^+), f(x^-)) = \text{Sinkhorn-divergence}(f(x^+), f(x^-))$

Our Wasserstein loss L_{wass} is defined as

$$L_{\text{wass}}(f(x^+), f(x^-)) = \min_{\gamma \in \Gamma(f(x^+), f(x^-))} \sum_{i,j} \gamma_{ij} c(f(x_i^+), f(x_j^-)) + \epsilon \cdot \text{KL}(\gamma \| f(x^+) \otimes f(x^-)), \quad (2)$$

where:

- $\Gamma(f(x^+), f(x^-))$: Set of all joint distributions between the embeddings of positive samples $f(x^+)$ and negative samples $f(x^-)$.
- γ_{ij} : Transport plan between $f(x_i^+)$ and $f(x_j^-)$.
- Cost function $c(f(x_i^+), f(x_j^-))$: $\frac{1}{2} \|f(x_i^+) - f(x_j^-)\|^2$.
- ϵ (0.01): Regularization parameter for entropy smoothing .
- $\text{KL}(\gamma \| f(x^+) \otimes f(x^-))$: Kullback-Leibler divergence regularizing the transport plan.
- $f(x^+) \otimes f(x^-)$: Independent product of the embeddings' distributions.

2 Proof of Theorem 1.

Theorem 1 (Robustness of wDCL to Data Imbalance):

$\mathcal{L}_{\text{wdcl}}$ is robust to data imbalance than $\mathcal{L}_{\text{debiased}}$.

Proof.

Let $\mathcal{L}_{\text{wdcl}}$ represent the Wasserstein-based Debiased Contrastive Loss, and $\mathcal{L}_{\text{debiased}}$ represent the standard Debiased Contrastive Loss, we will demonstrate that the Wasserstein distance term in $\mathcal{L}_{\text{wdcl}}$ provides a more stable and representative measure of dissimilarity between distributions, especially under data imbalance.

1. Sensitivity of $\mathcal{L}_{\text{debiased}}$ to Data Imbalance:

The Debiased Contrastive Loss $\mathcal{L}_{\text{debiased}}$ is defined as:

$$\mathcal{L}_{\text{debiased}} = \mathbb{E}_{(x, x^+) \sim p_{\text{pos}}} \left[-\log \frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau} + \sum_{i=1}^M \left(e^{f(x)^\top f(x_i^-)/\tau} - \gamma e^{2f(x)^\top f(x_i^-)/\tau} \right)} \right] \quad (3)$$

This loss function relies on pointwise similarities between embeddings $f(x_i^-)$ and $f(x)$. Under data imbalance, where negative samples x_i^- dominate, the pointwise similarities become biased, resulting in gradient updates that do not reflect the true data structure!

It means that the variance of the gradient updates under data imbalance becomes higher: $\text{Var}(\nabla_f \mathcal{L}_{\text{debiased}})$ is large due to this overrepresentation.

2. Robustness of the Wasserstein Distance:

The Wasserstein distance $W(\mu, \nu)$ between two probability distributions μ and ν over a metric space \mathcal{X} is defined as:

$$W(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y) d\gamma(x, y), \quad (4)$$

where $\Gamma(\mu, \nu)$ is the set of all couplings (joint distributions) with marginals μ and ν , and $d(x, y)$ is a distance metric.

a. Sensitivity to Distribution Geometry: The Wasserstein distance captures global structural differences by considering optimal mass transport between distributions, not pointwise similarities.

b. Robustness to Data Imbalance: The Wasserstein distance evaluates the entire distribution's transport plan, making it less influenced by sample imbalance and mitigating negative sample overrepresentation.

3. Wasserstein-based Debiased Contrastive Loss $\mathcal{L}_{\text{wdcl}}$:

The Wasserstein-based Debiased Contrastive Loss is defined as:

$$\mathcal{L}_{\text{wdcl}}(f, x, \alpha, \beta) = \lambda \mathcal{L}_{\text{wass}}(f(x^+), f(x(\alpha)^-)) - \beta \mathcal{L}_{\text{N-pair}}(f(x), f(x^+), f(x(\alpha)^-)), \quad (5)$$

where $\mathcal{L}_{\text{wass}}$ represents the Wasserstein distance between positive and negative samples.

4. Gradient Stability and Generalization Comparison:

Gradient Stability: The gradient for $\mathcal{L}_{\text{wdcl}}$ with respect to the model parameters f is given by:

$$\nabla_f \mathcal{L}_{\text{wdcl}} = \lambda \nabla_f \mathcal{L}_{\text{wass}} - \beta \nabla_f \mathcal{L}_{\text{N-pair}}. \quad (6)$$

The Wasserstein term involves integration over the distributions, leading to smoother gradients:

$$\text{Var}(\nabla_f \mathcal{L}_{\text{wdcl}}) < \text{Var}(\nabla_f \mathcal{L}_{\text{debiased}}). \quad (7)$$

Generalization: Generalization error is given by the expected difference between the true data distribution P_{data} and the model's learned distribution Q_{model} :

$$\mathbb{E}_{x \sim P_{\text{data}}}[\mathcal{L}(f(x))] - \mathbb{E}_{x \sim Q_{\text{model}}}[\mathcal{L}(f(x))]. \quad (8)$$

For $\mathcal{L}_{\text{wdcl}}$, this difference is minimized, as it reflects the global structure of the data.

5. Mathematical Justification:

For $\mathcal{L}_{\text{debiased}}$: The gradient with respect to $f(x)$ is influenced by individual negative samples $f(x_i^-)$. Overrepresentation of negative samples leads to biased gradient updates.

For $\mathcal{L}_{\text{wdcl}}$: The Wasserstein term involves integration over distributions:

$$\nabla_f \mathcal{L}_{\text{wass}} \propto \int_{\mathcal{X}} (\nabla_f f(x^+) - \nabla_f f(x_i^-)) d\gamma(x^+, x_i^-), \quad (9)$$

leading to smoother gradients and less sensitivity to imbalance.

By incorporating the Wasserstein distance, $\mathcal{L}_{\text{wdcl}}$ smooths the effect of imbalanced samples and better captures the global structure of the data, resulting in:

- **More Stable Optimization:** Gradients are less volatile:

$$\text{Var}(\nabla_f \mathcal{L}_{\text{wdcl}}) < \text{Var}(\nabla_f \mathcal{L}_{\text{debiased}}). \quad (10)$$

- **Better Generalization:** The model learns embeddings that reflect the true data distribution:

$$\mathbb{E}_{x \sim P_{\text{data}}} [\mathcal{L}(f(x))] - \mathbb{E}_{x \sim Q_{\text{model}}} [\mathcal{L}(f(x))] \text{ is minimized for } \mathcal{L}_{\text{wdcl}}. \quad (11)$$

Therefore, mathematically, $\mathcal{L}_{\text{wdcl}}$ is more robust to data imbalance than $\mathcal{L}_{\text{debiased}}$.

3 Proof of Theorem 2

Theorem 2 Let $\mathcal{F}_{\text{KAN-W}}$, $\mathcal{F}_{\text{KAN-S}}$, and \mathcal{F}_{MLP} be the hypothesis classes of KAN with wavelet basis functions, B-spline basis functions, and MLP (Multilayer Perceptron), respectively. The norm-based Rademacher complexity of these function classes satisfies the following inequality:

$$\mathcal{R}_n(\mathcal{F}_{\text{KAN-W}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{KAN-S}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{MLP}}), \quad (12)$$

, where \prec denotes a strict inequality.

Proof.

We begin with the empirical Rademacher complexity for a function class \mathcal{F} over a sample $S = \{x_1, \dots, x_n\}$:

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(x_i) \right], \quad (13)$$

where σ_i are Rademacher random variables taking values in $\{-1, 1\}$ with equal probability, and $f(x_i) \in \mathcal{F}$ represents the function applied to sample x_i .

The Rademacher complexity can be bounded based on the norm of the hypothesis class, using the inequality:

$$\mathcal{R}_n(\mathcal{F}_\psi) \leq \frac{\lambda}{\sqrt{n}} \cdot \mathbb{E} [\|\psi\|_{\mathcal{H}_\psi}], \quad (14)$$

where $\|\psi\|_{\mathcal{H}_\psi}$ is the norm of the basis function ψ in the appropriate Hilbert space \mathcal{H}_ψ , and λ is a constant.

We compare the norms of the basis functions:

1. **Wavelet Basis ψ_w :** Wavelet functions have compact support and exhibit localization in both time and frequency. The H^1 norm of a wavelet basis function is given by:

$$\|\psi_w\|_{H^1} = \int_{-\infty}^{\infty} (|\psi_w(x)|^2 + |\nabla \psi_w(x)|^2) dx.$$

Since wavelets are localized, this norm is relatively small, leading to a lower Rademacher complexity.

2. **B-Spline Basis ψ_s :** Spline basis functions are smoother but more global than wavelets. Their H^1 norm is given by:

$$\|\psi_s\|_{H^1} = \int_0^1 (|\psi_s(x)|^2 + |\nabla \psi_s(x)|^2) dx. \quad (15)$$

B-Splines typically have larger norms because they spread over larger intervals and require more parameters, leading to a higher complexity compared to wavelets.

3. **MLP Functions:** MLPs, with many parameters, exhibit high expressivity but also have very large norms due to the number of layers and parameters. Therefore, the Rademacher complexity of MLPs grows significantly faster than that of wavelet and spline functions.

To rigorously quantify these differences, we apply Dudley's entropy integral:

$$\mathcal{R}_n(\mathcal{F}) \leq \frac{12}{\sqrt{n}} \int_0^\infty \sqrt{\log N(\epsilon, \mathcal{F}, \|\cdot\|)} d\epsilon, \quad (16)$$

where $N(\epsilon, \mathcal{F}, \|\cdot\|)$ is the covering number of \mathcal{F} with ϵ -balls under the norm $\|\cdot\|$. Since wavelets require fewer terms to represent functions, the covering number is smaller for $\mathcal{F}_{\text{KAN-W}}$, followed by $\mathcal{F}_{\text{KAN-S}}$, and then \mathcal{F}_{MLP} .

Thus, integrating the bounds gives:

$$\mathcal{R}_n(\mathcal{F}_{\text{KAN-W}}) \leq \frac{12}{\sqrt{n}} \int_0^\infty \sqrt{\log N(\epsilon, \mathcal{F}_{\text{KAN-W}}, \|\cdot\|)} d\epsilon, \quad (17)$$

with the same inequality holding for $\mathcal{F}_{\text{KAN-S}}$ and \mathcal{F}_{MLP} .

Therefore, by combining norm-based bounds and entropy integrals, we conclude:

$$\mathcal{R}_n(\mathcal{F}_{\text{KAN-W}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{KAN-S}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{MLP}}) \quad (18)$$

4 Proof of Theorem 3

Theorem 3: Let $\mathcal{F}_{\text{dcMltR-KAN-W}}$, $\mathcal{F}_{\text{dcMltR-KAN-S}}$, and $\mathcal{F}_{\text{dc-MLP}}$ represent the hypothesis classes of dcMltR-KAN with wavelet basis, B-spline basis, and dc-MLP model, respectively. The upper-bound on the generalization error for these models satisfies:

$$\mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-W}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-S}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dc-MLP}}), \quad (19)$$

where $\mathcal{E}_{\text{gen}}(\cdot)$ denotes the generalization error, and \prec signifies strict inequality.

Proof.

1. Preliminaries

We are to prove that the upper bound on the generalization error for the models satisfies: $\mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-W}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-S}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dc-MLP}})$, where:

- $\mathcal{F}_{\text{dcMltR-KAN-W}}$ is the hypothesis class of the dcMltR-KAN model with wavelet basis functions.
- $\mathcal{F}_{\text{dcMltR-KAN-S}}$ is the hypothesis class of the dcMltR-KAN model with B-spline basis functions.
- $\mathcal{F}_{\text{dc-MLP}}$ is the hypothesis class where MltR-KAN is replaced by an MLP.

The generalization error $\mathcal{E}_{\text{gen}}(\mathcal{F})$ measures the difference between the expected loss and the empirical loss for a hypothesis class \mathcal{F} :

$$\mathcal{E}_{\text{gen}}(\mathcal{F}) = \mathbb{E}_{f \sim \mathcal{F}}[L_{\text{expected}}(f) - L_{\text{empirical}}(f)]. \quad (20)$$

The Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ of a hypothesis class \mathcal{F} with sample size n is a measure of its capacity, reflecting how well the class can fit random noise:

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\sigma, X} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(x_i) \right], \quad (21)$$

where σ_i are independent Rademacher variables taking values ± 1 with equal probability, and $X = \{x_1, \dots, x_n\}$ is the sample.

2. Relate generalization error to Rademacher complexity

We can have the relationships between the generalization error and the Rademacher complexity:

$$\mathcal{E}_{\text{gen}}(\mathcal{F}) \leq 2\mathcal{R}_n(\mathcal{F}) + \epsilon(n, \delta), \quad (22)$$

where $\epsilon(n, \delta)$ is a term that diminishes as the sample size n increases and confidence level δ is considered. Since $\epsilon(n, \delta)$ is common for all models (assuming the same n and δ), the primary factor influencing the generalization error is the Rademacher complexity $\mathcal{R}_n(\mathcal{F})$.

3. Apply Theorem 2

From Theorem 2, we have the ordering of Rademacher complexities:

$$\mathcal{R}_n(\mathcal{F}_{\text{dcMltR-KAN-W}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{dcMltR-KAN-S}}) \prec \mathcal{R}_n(\mathcal{F}_{\text{dc-MLP}}). \quad (23)$$

Since the generalization error is directly proportional to the Rademacher complexity, we have the ordering of generalization errors follows the same strict inequalities:

$$\mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-W}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dcMltR-KAN-S}}) \prec \mathcal{E}_{\text{gen}}(\mathcal{F}_{\text{dc-MLP}}). \quad (24)$$

This result indicates that the dcMltR-KAN model with wavelet basis functions has a strictly lower upper bound on the generalization error compared to the versions with B-spline basis. This is because Wavelet Basis Functions offer a sparse representation and capture localized features effectively, leading to a more constrained hypothesis class with lower complexity.

5 Ablation studies of dcMltR-KAN with four wavelets

Table 1: Ablation Study: Top-1 Accuracy and D-Index for Different Methods (Datasets: O1, O2, and O3)

| Dataset | Method | Top-1 Accuracy (mean \pm std) | D-Index (mean \pm std) |
|---------|-----------------------------|---------------------------------|--------------------------|
| O1 | Baseline - CNN | 0.9288 | 1.9027 |
| | <i>Ablation Components:</i> | | |
| | <i>w/ wDCL</i> | 0.9219 ± 0.0014 | 1.9187 ± 0.0015 |
| | <i>w/ MltR-KAN</i> | 0.9254 ± 0.0069 | 1.9174 ± 0.0025 |
| | dcMltR-KAN | | |
| | <i>Haar</i> | 0.9817 ± 0.0017 | 1.9936 ± 0.0009 |
| | <i>Mexican Hat</i> | 0.9804 ± 0.0016 | 1.9929 ± 0.0009 |
| | <i>Db₄</i> | 0.9772 ± 0.0014 | 1.9916 ± 0.0005 |
| O2 | Baseline - CNN | 0.9155 | 1.8576 |
| | <i>Ablation Components:</i> | | |
| | <i>w/ wDCL</i> | 0.8887 ± 0.0015 | 1.8154 ± 0.0014 |
| | <i>w/ MltR-KAN</i> | 0.8850 ± 0.0059 | 1.9272 ± 0.0035 |
| | dcMltR-KAN | | |
| | <i>Haar</i> | 0.9799 ± 0.0072 | 1.9832 ± 0.0059 |
| | <i>Mexican Hat</i> | 0.9731 ± 0.0028 | 1.9776 ± 0.0023 |
| | <i>Db₄</i> | 0.9744 ± 0.0076 | 1.9811 ± 0.0056 |
| O3 | Baseline - CNN | 0.8363 | 1.8175 |
| | <i>Ablation Components:</i> | | |
| | <i>w/ wDCL</i> | 0.8888 ± 0.0008 | 1.9293 ± 0.0004 |
| | <i>w/ MltR-KAN</i> | 0.8639 ± 0.0018 | 1.8830 ± 0.0012 |
| | dcMltR-KAN | | |
| | <i>Haar</i> | 0.9009 ± 0.0019 | 1.9377 ± 0.0010 |
| | <i>Mexican Hat</i> | 0.9005 ± 0.0007 | 1.9126 ± 0.0007 |
| | <i>Db₄</i> | 0.9045 ± 0.0005 | 1.9399 ± 0.0003 |
| | <i>Sym₄</i> | 0.9010 ± 0.0019 | 1.9378 ± 0.0012 |

6 Proof of Proposition 1

Proposition 1: $\mathcal{L}_{\text{wdcl}}$ with FNE is lower than the loss without FNE: $\mathbb{E}_{(x, x(\alpha)^-)} [\mathcal{L}_{\text{wdcl}}^{\text{FNE}}] < \mathbb{E}_{(x, x^-)} [\mathcal{L}_{\text{wdcl}}^{\text{no FNE}}]$, where α is the elimination ratio.

Statement: Let $\mathcal{L}_{\text{wdcl}}^{\text{FNE}}(x)$ denote the Wasserstein Debiased Contrastive Loss (wDCL) with False Negatives Elimination (FNE), and $\mathcal{L}_{\text{wdcl}}^{\text{no FNE}}(x)$ denote the wDCL without FNE. Then, under the assumption that the set of negative samples after FNE is a proper subset of the original negative samples, and that the removed negatives are those with the highest similarity to the anchor sample x , we have:

$$\mathbb{E}_{(x, x^+, x(\alpha)^-)} [\mathcal{L}_{\text{wdcl}}^{\text{FNE}}(x)] < \mathbb{E}_{(x, x^+, x^-)} [\mathcal{L}_{\text{wdcl}}^{\text{no FNE}}(x)], \quad (25)$$

where α is the elimination ratio, x^- are negative samples, and $x(\alpha)^-$ are the negative samples after applying FNE.

Proof.

Let x be an anchor sample, x^+ its positive counterpart, and \mathcal{N} the set of all negative samples.

Define $\mathcal{N}(\alpha) \subset \mathcal{N}$ as the set after FNE, where the top α fraction of negatives most similar to x are removed.

Let f be the encoder mapping samples to normalized embeddings $\mathbf{h} = f(x)$, $\mathbf{h}^+ = f(x^+)$, and $\mathbf{h}^- = f(x^-)$.

The N-pair contrastive loss without FNE is:

$$\mathcal{L}_{\text{N-pair}}^{\text{no FNE}} = -\log \left(\frac{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau}}{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau} + \sum_{x^- \in \mathcal{N}} e^{\mathbf{h}^\top \mathbf{h}^- / \tau}} \right). \quad (26)$$

With FNE, it becomes:

$$\mathcal{L}_{\text{N-pair}}^{\text{FNE}} = -\log \left(\frac{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau}}{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau} + \sum_{x^- \in \mathcal{N}(\alpha)} e^{\mathbf{h}^\top \mathbf{h}^- / \tau}} \right). \quad (27)$$

Since $\mathcal{N}(\alpha) \subset \mathcal{N}$ and the most similar negatives are removed, we have:

$$\sum_{x^- \in \mathcal{N}(\alpha)} e^{\mathbf{h}^\top \mathbf{h}^- / \tau} < \sum_{x^- \in \mathcal{N}} e^{\mathbf{h}^\top \mathbf{h}^- / \tau}. \quad (28)$$

This implies:

$$\frac{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau}}{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau} + \sum_{x^- \in \mathcal{N}(\alpha)} e^{\mathbf{h}^\top \mathbf{h}^- / \tau}} > \frac{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau}}{e^{\mathbf{h}^\top \mathbf{h}^+ / \tau} + \sum_{x^- \in \mathcal{N}} e^{\mathbf{h}^\top \mathbf{h}^- / \tau}}. \quad (29)$$

Since $-\log(x)$ is a decreasing function, it follows that:

$$\mathcal{L}_{\text{N-pair}}^{\text{FNE}} < \mathcal{L}_{\text{N-pair}}^{\text{no FNE}}. \quad (30)$$

For the Wasserstein loss $\mathcal{L}_{\text{wass}}$, removing negatives closest to x may increase the distance:

$$\mathcal{L}_{\text{wass}}^{\text{FNE}} \geq \mathcal{L}_{\text{wass}}^{\text{no FNE}}. \quad (31)$$

The total loss difference is:

$$\Delta\mathcal{L} = \mathcal{L}_{\text{wdcl}}^{\text{FNE}} - \mathcal{L}_{\text{wdcl}}^{\text{no FNE}} = \lambda (\mathcal{L}_{\text{wass}}^{\text{FNE}} - \mathcal{L}_{\text{wass}}^{\text{no FNE}}) - \beta (\mathcal{L}_{\text{N-pair}}^{\text{FNE}} - \mathcal{L}_{\text{N-pair}}^{\text{no FNE}}). \quad (32)$$

Since $\mathcal{L}_{\text{N-pair}}^{\text{FNE}} < \mathcal{L}_{\text{N-pair}}^{\text{no FNE}}$ (from Equation 30), the second term in Equation 32 is negative. The first term is non-negative due to Equation 31.

By choosing β sufficiently large relative to λ , the decrease in N-pair loss outweighs any increase in Wasserstein loss, ensuring $\Delta\mathcal{L} < 0$.

Taking expectations over the data distribution:

$$\mathbb{E} [\mathcal{L}_{\text{wdcl}}^{\text{FNE}}] = \mathbb{E} [\mathcal{L}_{\text{wdcl}}^{\text{no FNE}} + \Delta\mathcal{L}] < \mathbb{E} [\mathcal{L}_{\text{wdcl}}^{\text{no FNE}}], \quad (33)$$

since $\Delta\mathcal{L} < 0$.

Therefore, we have

$$\mathbb{E}_{(x, x^+, x(\alpha)^-)} [\mathcal{L}_{\text{wdcl}}^{\text{FNE}}(x)] < \mathbb{E}_{(x, x^+, x^-)} [\mathcal{L}_{\text{wdcl}}^{\text{no FNE}}(x)].$$

7 Proof of Proposition 2

The expected WDCL loss with Similarity-Based Weighting (SBW) is lower than without SBW:

$$\mathbb{E}_{(x, \mathbf{v}_i)} [\mathcal{L}_{\text{wdcl}}^{\text{SBW}}] < \mathbb{E}_{(x, x^-)} [\mathcal{L}_{\text{wdcl}}^{\text{no SBW}}], \quad (34)$$

where \mathbf{v}_i is the aggregated feature vector from the top k most similar samples via SBW.

Proof

1. WDCL Loss Function:

The Weighted Decoupled Contrastive Loss (WDCL) for a sample x is defined as:

$$\mathcal{L}_{\text{wdcl}} = -\log \left(\frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau} + \sum_{x^-} w(x, x^-) e^{f(x)^\top f(x^-)/\tau}} \right), \quad (35)$$

where:

- $f(x)$ is the feature representation of sample x .
- x^+ is a positive sample associated with x .
- x^- are negative samples.
- $w(x, x^-)$ is the weight assigned to each negative sample.
- τ is a temperature parameter.

2. Effect of SBW:

- **With SBW:** Focuses on the top k most similar negatives, aggregating them into \mathbf{v}_i and assigning appropriate weights.
- **Without SBW:** Considers a larger set of negatives, often with equal weighting.

3. Comparison of Denominators:

- **With SBW:**

$$D_{\text{SBW}} = e^{f(x)^\top f(x^+)/\tau} + w_{\text{SBW}} \cdot e^{f(x)^\top \mathbf{v}_i/\tau}, \quad (36)$$

where w_{SBW} is the aggregated weight for the negative \mathbf{v}_i .

- **Without SBW:**

$$D_{\text{no SBW}} = e^{f(x)^\top f(x^+)/\tau} + \sum_{x^-} e^{f(x)^\top f(x^-)/\tau}. \quad (37)$$

4. Key Observation:

- **Aggregated Negatives:** SBW’s aggregation leads to a more informative negative \mathbf{v}_i , but the overall denominator D_{SBW} grows less than $D_{\text{no SBW}}$.
 - **Denominator Size:** A smaller denominator in SBW means the fraction inside the logarithm is larger.
5. **Implication on Loss:** Since the negative logarithm function is decreasing, a larger fraction results in a lower loss:

$$\mathcal{L}_{\text{wdcl}}^{\text{SBW}} < \mathcal{L}_{\text{wdcl}}^{\text{no SBW}}. \quad (38)$$

6. **Expectation over Data:**

Taking expectations over the data distribution confirms the inequality:

$$\mathbb{E}_{(x, \mathbf{v}_i)} [\mathcal{L}_{\text{wdcl}}^{\text{SBW}}] < \mathbb{E}_{(x, x^-)} [\mathcal{L}_{\text{wdcl}}^{\text{no SBW}}]. \quad (39)$$

Thus, by focusing on the most informative negatives and weighting them appropriately, SBW reduces the expected WDCL loss compared to not using SBW.

8 Proof of Proposition 3

The total loss in the Wasserstein Debiased Contrastive Learning (wDCL) framework is a sum over all resolution levels:

$$\mathcal{L}_{\text{wdcl}} = \sum_{l=1}^L \omega^{(l)} \mathcal{L}^{(l)}, \quad (40)$$

where:

- $\mathcal{L}^{(l)}$ is the loss at resolution level l .
- $\omega^{(l)} \geq 0$ are learned weights adjusting the contribution of each level.

1. Applying SBW at Each Level Reduces Loss:

From Proposition 2, we know that applying SBW to the feature representations reduces the expected loss at a single resolution level:

$$\mathbb{E}_{(x, \mathbf{v}_i^{(l)})} [\mathcal{L}^{(l), \text{SBW}}] < \mathbb{E}_{(x, x^-)} [\mathcal{L}^{(l), \text{no SBW}}], \quad (41)$$

where:

- $\mathbf{v}_i^{(l)}$ is the SBW-refined feature vector at level l .
- $\mathcal{L}^{(l), \text{SBW}}$ is the loss at level l with SBW.
- $\mathcal{L}^{(l), \text{no SBW}}$ is the loss at level l without SBW.

2. Summing Over All Levels:

Since the inequality holds at each level l , we can multiply both sides by the non-negative weights $\omega^{(l)}$ and sum over all levels:

$$\sum_{l=1}^L \omega^{(l)} \mathbb{E}_{(x, \mathbf{v}_i^{(l)})} [\mathcal{L}^{(l), \text{SBW}}] < \sum_{l=1}^L \omega^{(l)} \mathbb{E}_{(x, x^-)} [\mathcal{L}^{(l), \text{no SBW}}]. \quad (42)$$

3. Expressing the Overall Expected Loss:

The left side represents the overall expected loss with SBW applied:

$$\mathbb{E}_{x_i \sim p_{\text{data}}} [\mathcal{L}_{\text{wdcl}}^{\text{SBW}}] = \sum_{l=1}^L \omega^{(l)} \mathbb{E}_{(x, \mathbf{v}_i^{(l)})} [\mathcal{L}^{(l), \text{SBW}}]. \quad (43)$$

Similarly, the right side is the overall expected loss without SBW:

$$\mathbb{E}_{x_i \sim p_{\text{data}}} [\mathcal{L}_{\text{wdcl}}^{\text{no SBW}}] = \sum_{l=1}^L \omega^{(l)} \mathbb{E}_{(x, x^-)} [\mathcal{L}^{(l), \text{no SBW}}]. \quad (44)$$

As such, combining the above, we have:

$$\mathbb{E}_{x_i \sim p_{\text{data}}} [\mathcal{L}_{\text{wdcl}}^{\text{SBW}}] < \mathbb{E}_{x_i \sim p_{\text{data}}} [\mathcal{L}_{\text{wdcl}}^{\text{no SBW}}]. \quad (45)$$

This inequality demonstrates that applying SBW before MItR-KAN across all resolution levels reduces the overall expected wDCL loss compared to not applying SBW.

9 Visualization of the explainability enhancement process in MltR-KAN

The CNN encoder initially extracts high-level features from the normalized SNR data, which are then decomposed by the Haar wavelet into approximation (cA) and detail coefficients (cD1, cD2), which capture the global and local data behaviors of the SNR feature after CNN. This provides a multi-resolution view of the learned representation, enhancing the transparency and interpretability of the feature extraction process.

The combined use of a CNN encoder followed by Haar wavelet transformation helps us clearly see what features are being learned from the SNR data. The CNN extracts high-level features, while the Haar wavelet further breaks down these features into explainable components, covering both broad trends and finer details. This multi-stage process makes the learned representation more transparent and easier to understand, enhancing explainability.

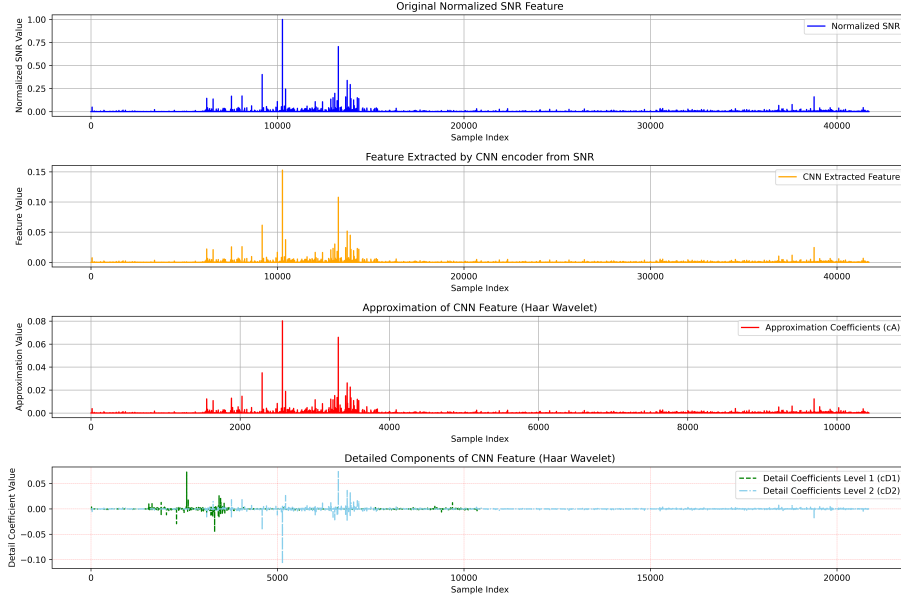


Figure S1: Visualization of the explainability enhancement process in MltR-KAN for the SNR feature from gravitational wave O1 data. The original normalized SNR data is processed by a CNN encoder to extract high-level features. The learned CNN feature is subsequently decomposed using Haar wavelet transformation, resulting in both approximation (cA) and detail coefficients (cD1, cD2), which provide a multi-resolution view of the learned representation, enhancing transparency and interpretability of the feature extraction process

10 Impact of False Negative Elimination (FNE) on Hierarchical Loss During Training under MltR-KAN

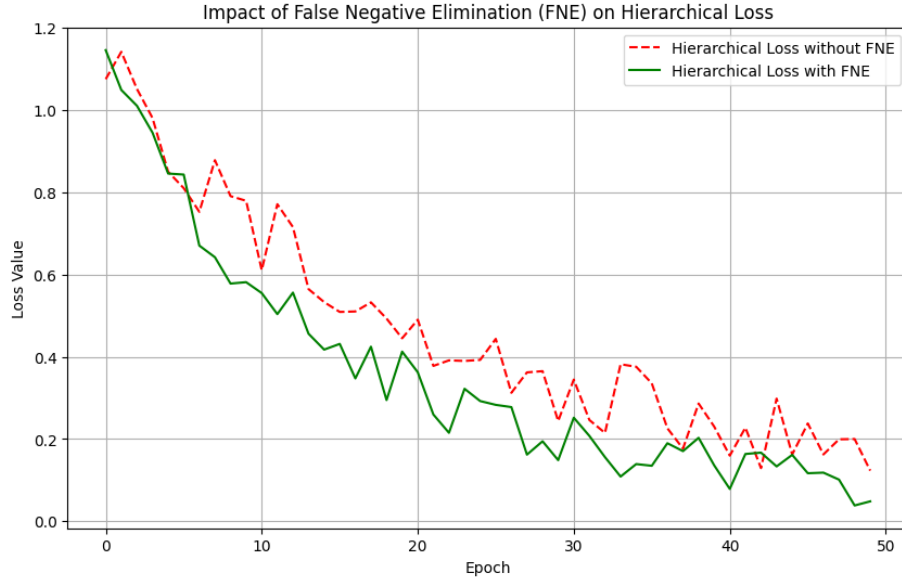


Figure S2: Simulated impact of False Negative Elimination (FNE) on Hierarchical Loss During Training. This figure compares the hierarchical loss values for models trained with and without the False Negative Elimination (FNE) process over 50 epochs. The green line represents the model incorporating FNE, while the red dashed line shows the model without FNE. The model with FNE exhibits a consistently lower loss, indicating that FNE helps to effectively minimize false negatives, leading to enhanced learning and improved convergence during training

11 Impact of Similarity-Based Weighting (SBW) on Hierarchical Loss During Training under MltR-KAN

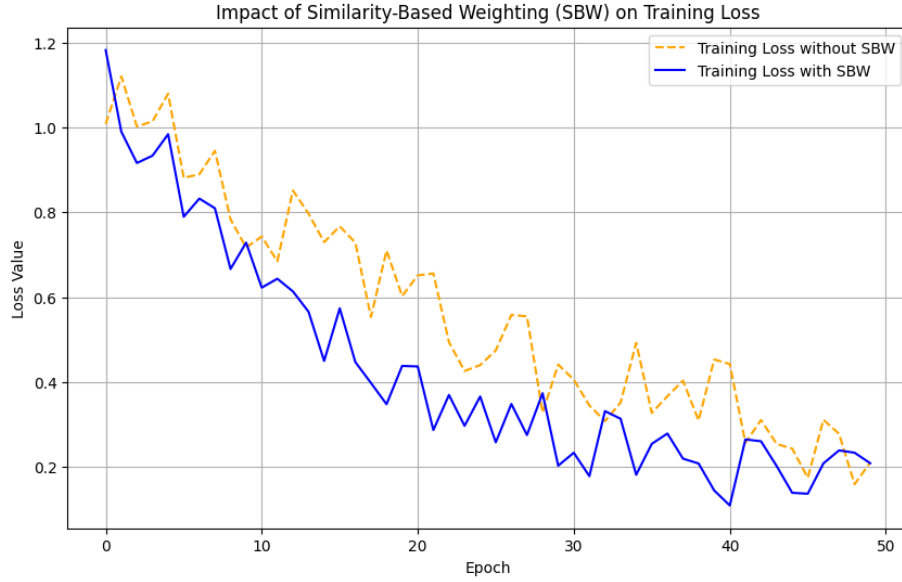


Figure S3: Simulated impact of Similarity-Based Weighting (SBW) on Training Loss. This figure illustrates the effect of incorporating SBW into a simulated training process. The blue line represents the training loss with SBW, while the orange dashed line shows the training loss without SBW. It is evident that using SBW results in a more rapid decline in training loss, indicating enhanced model convergence and efficiency. The reduced overall loss with SBW suggests better feature representation, ultimately contributing to improved model performance.

12 Baseline comparisons

12.1 CPC(Contrastive Predictive Coding) Result

Contrastive Predictive Coding (CPC) [7] is an unsupervised learning method to extract robust high-level representations from sequential data such as audio, images, text, and reinforcement learning trajectories. The CPC architecture combines an encoder and an autoregressive model to capture temporal or spatial dependencies, encoding input data into a compact latent space that emphasizes essential features while filtering noise. An autoregressive model then processes these encoded representations to create a context vector, preserving temporal relationships and summarizing the information necessary for future predictions. Using a contrastive loss function, specifically InfoNCE, CPC maximizes the mutual information between the context vector and subsequent data, refining its ability to predict future sequence elements. However, CPC has limitations: it is best suited to sequential data, relying on temporal or spatial coherence; it is sensitive to the quality of negative samples, which are essential for effective contrastive learning; and, while it captures broad contextual information, it may overlook finer details.

Table 2: Performance Metrics for Dataset O1, Dataset O2, and Dataset O3

| Metric | Dataset O1 | Dataset O2 | Dataset O3 |
|-----------|------------|------------|------------|
| Accuracy | 0.775202 | 0.525758 | 0.510133 |
| Precision | 0.773208 | 0.499888 | 0.455736 |
| Recall | 0.775202 | 0.525758 | 0.510133 |
| F1 Score | 0.771829 | 0.475751 | 0.453775 |
| D-index | 1.877014 | 1.696813 | 1.638236 |

12.2 TS-TCC(Time-Series Representation Learning via Temporal and Contextual Contrasting) Result

Time-Series Representation Learning via Temporal and Contextual Contrasting (TS-TCC) [3] is an unsupervised framework designed to extract powerful representations from time-series data, which makes it especially effective in scenarios with limited labeled data. By generating two augmented views of the input, one with weaker augmentations and the other with stronger augmentations, TSC learns temporal dependencies by predicting future segments of one view using the context of the other. This cross-view prediction strengthens the model’s ability to handle variations from augmentation and capture essential patterns. The contextual contrasting module of TS-TCC further enhances learning by maximizing similarity between contexts of the same sample and minimizing similarity with others, promoting discriminative and generalizable representations. However, TS-TCC demands high computational power due to its use of augmented views and an autoregressive model, and it can be sensitive

to hyperparameters. Additionally, while capturing general temporal patterns effectively, TS-TCC may underperform on tasks that require very fine-grained or specialized features.

Table 3: TS-TCC: Performance Metrics for Dataset O1, Dataset O2, and Dataset O3

| Metric | Dataset O1 | Dataset O2 | Dataset O3 |
|---------------|-------------------|-------------------|-------------------|
| Accuracy | 0.980205 | 0.976989 | 0.843326 |
| Precision | 0.980719 | 0.977677 | 0.840732 |
| Recall | 0.980205 | 0.976989 | 0.843326 |
| F1 Score | 0.980179 | 0.977055 | 0.839187 |
| D-index | 1.967096 | 1.984084 | 1.819134 |

12.3 SimCLR (Simple Contrastive Learning of Representations) Result

SimCLR [1] is a self-supervised framework for learning visual representations, reducing contrastive learning by removing complex architectures and memory banks in favor of large batch sizes and strong enhancements. Train by maximizing agreement between two augmented views of the same image, generated through a data augmentation module that applies transformations such as cropping and color distortion. These views, forming a positive pair, pass through an encoder and projection head to a latent space where contrastive loss aligns similar images. This approach allows SimCLR to achieve performance close to fully supervised models on datasets such as ImageNet. However, SimCLR requires large batch sizes, making it computationally demanding, and its performance heavily depends on carefully chosen augmentations. While strong at capturing general visual features, SimCLR may miss fine details that other, more task-specific methods can capture.

Table 4: SimCLR: Performance Metrics for Dataset O1, Dataset O2, and Dataset O3

| Metric | Dataset O1 | Dataset O2 | Dataset O3 |
|---------------|-------------------|-------------------|-------------------|
| Accuracy | 0.969796 | 0.966020 | 0.828186 |
| Precision | 0.971033 | 0.968858 | 0.826369 |
| Recall | 0.969796 | 0.966020 | 0.828186 |
| F1 Score | 0.969475 | 0.966591 | 0.822599 |
| D-index | 1.971161 | 1.948001 | 1.826083 |

12.4 Fully-supervised deep learning Models

To leverage the time-series structure of the data from all three observing runs (O1, O2, O3), we begin by sorting the dataset chronologically, using earlier data points to train the models and later points to test. Since our dataset is heavily imbalanced, we ensure that both the training and testing sets reflect the same label distribution to maintain a fair performance evaluation across all deep learning models.

For model testing, we split the data, dedicating 80% to training and the remaining 20% to testing. The following machine-learning models were implemented:

- **GAN-DNN Classifier [5]:** This model employs a Generative Adversarial Network (GAN) consisting of a generator and a discriminator to augment the dataset with synthetic samples. The generator network takes random noise as input and produces synthetic data samples, utilizing two dense layers with LeakyReLU activation and batch normalization to stabilize training. The discriminator, structured to classify both real and synthetic samples, has two dense layers with LeakyReLU activation followed by a final dense layer with softmax activation to output labels. The GAN generates 20,000 synthetic samples with three additional labels to balance the original dataset. The final labeled dataset, combining real and synthetic samples, is used for classification training with categorical cross-entropy as the loss function.
- **CNN:** This Convolutional Neural Network (CNN) is designed for sequential data classification. It begins with an input layer that preserves the original shape of the sequence. Two 1D convolutional layers with 64 filters and a kernel size of 3 apply ReLU activation while maintaining the sequence length. The output is flattened and then passed through two dense layers with 64 neurons and ReLU activations, which identify complex patterns. Finally, a softmax output layer, with neurons equal to the target classes, provides class probabilities for classification.
- **Gated Recurrent Unit (GRU) [2]:** This GRU model consists of three layers with 128, 256, and 128 neurons, respectively. Each GRU layer is followed by a dropout layer with rates of 0.1, 0.2, and 0.3. The GRU cells include an update gate and a reset gate, both with sigmoid activation. The update gate controls the balance between the previous hidden state and the current node's hidden state, while the reset gate controls the degree of forgetting of the previous hidden state in calculating the new candidate state. The model ends with a dense output layer that uses softmax activation for class probability output, optimized with categorical cross-entropy.
- **Residual Networks (ResNet) [6]:** A ResNet-50 model is implemented, starting with an initial convolutional layer (64 filters, stride of 2) followed

by batch normalization, ReLU activation, and a max-pooling layer (pool size of 3, stride of 2). The main architecture includes four stages of bottleneck blocks with configurations [3, 4, 6, 3]. Each bottleneck block reduces dimensions, applies a convolution, and then restores dimensions with shortcut connections between the input and output of each block. Batch normalization and ReLU activation are applied throughout. Down-sampling occurs at the start of each new stage by adjusting the stride. The model concludes with global average pooling and a dense output layer with softmax activation to produce class probabilities. Categorical cross-entropy is used as the loss function for multiclass classification.

- **Transformer [8]:** This model utilizes a Transformer architecture with a multi-head attention mechanism, configured with 32 heads alongside feed-forward layers. Each Transformer block includes a multi-head attention layer and a feed-forward neural network consisting of dense layers with ReLU activation. Layer normalization is applied both before and after the feed-forward network, while dropout layers are included after the attention and feed-forward layers for regularization. After attention and feed-forward processing, the output is flattened and passed through dense layers for final classification.

Each model was trained for 100 epochs, experimenting with different learning rates (1e-3, 1e-4, 1e-5) and batch sizes (64, 128, 256, 512). The optimal model configuration was selected based on the highest accuracy and D-index, ensuring it did not overfit the training data.

13 Silhouette analysis of O1, O2, and O3 data before and after dcMltR-KAN

Table 5: Silhouette analysis under UMAP

| Data | n_neighbors (UMAP) | Silhouette Score (K-Mean clustering) |
|--------------------------|--------------------|--------------------------------------|
| Original O1 data | 5 | 0.1847 |
| | 10 | 0.2490 |
| | 15 | 0.1991 |
| | 20 | 0.3094 |
| | 30 | 0.2457 |
| | 50 | 0.2787 |
| O1 data after dcMltR-KAN | 5 | 0.3958 |
| | 10 | 0.5028 |
| | 15 | 0.5307 |
| | 20 | 0.5319 |
| | 30 | 0.5219 |
| | 50 | 0.5401 |
| Original O2 data | 50 | 0.2293 |
| | 60 | 0.2323 |
| | 70 | 0.2139 |
| | 80 | 0.2088 |
| | 90 | 0.1966 |
| | 100 | 0.2328 |
| O2 data after dcMltR-KAN | 50 | 0.4754 |
| | 60 | 0.4963 |
| | 70 | 0.5130 |
| | 80 | 0.4748 |
| | 90 | 0.5125 |
| | 100 | 0.5041 |
| Original O3 data | 50 | -0.0807 |
| | 60 | -0.0181 |
| | 70 | -0.0733 |
| | 80 | -0.0583 |
| | 90 | 0.1428 |
| | 100 | 0.1213 |
| O3 data after dcMltR-KAN | 50 | 0.4317 |
| | 60 | 0.4450 |
| | 70 | 0.4291 |
| | 80 | 0.4237 |
| | 90 | 0.4393 |
| | 100 | 0.4391 |

Note: UMAP is applied to original O1/O2/O3 and their corresponding data after dcMItR-KAN before Kmeans

14 dcMltR-KAN results on EMOB and ablation study

Table S5: dcMltR-KAN results on EMOB and ablation study

| Method | Top1 Accuracy (mean \pm std) | D-Index (mean \pm std) |
|-----------------------------|--------------------------------|--------------------------|
| <i>Ablation components:</i> | | |
| <i>w/o wDCL</i> | 0.8503 ± 0.0277 | 1.9042 ± 0.0181 |
| <i>w/o MltR-KAN</i> | 0.8379 ± 0.0103 | 1.8955 ± 0.0067 |
| dcMltR-KAN | | |
| <i>Mexican-hat</i> | 0.9326 ± 0.0035 | 1.9573 ± 0.0022 |
| <i>Sym4</i> | 0.9186 ± 0.0055 | 1.9483 ± 0.0035 |
| <i>Db4</i> | 0.9180 ± 0.0036 | 1.9478 ± 0.0023 |
| <i>Haar</i> | 0.8866 ± 0.0061 | 1.9278 ± 0.0040 |

15 Preprocessing and Feature Extraction for EMODB data

The EMODB dataset consists of raw mono audio files, each sampled at 16,000 Hz and approximately two seconds in duration. The audio files were first blocked into small chunks of audio signals, i.e., windowing, where each window has a length of 1024 samples (block size) and is spaced by hop of 512 samples (hop size). For each windowed segment, we extracted features such as Mel Frequency Cepstral Coefficients (MFCC) (first 14 coefficients), spectral centroid, spectral bandwidth, spectral contrast, spectral rolloff, Zero-Crossing Rate (ZCR), Root Mean Square Energy (RMS), and fundamental frequency (F0). Table 1 lists the dimensions of each feature. After feature extraction for each window, we computed two statistics, mean and standard deviation, to represent the overall characteristics of the audio file by aggregating all the instantaneous features. Figure 1 illustrates the preprocessing and feature extraction process.

Table 1. Audio Dataset Features

| Features | Feature Dim. for Each Windowed Segment | Aggregated Feature Dim. for Each File |
|--------------------|--|---------------------------------------|
| MFCC | 14 | 28 |
| Spectral Centroid | 1 | 2 |
| Spectral Bandwidth | 1 | 2 |
| Spectral Contrast | 7 | 14 |
| Spectral Rolloff | 1 | 2 |
| Zero-Crossing Rate | 1 | 2 |
| RMS Energy | 1 | 2 |
| F ₀ | 1 | 2 |

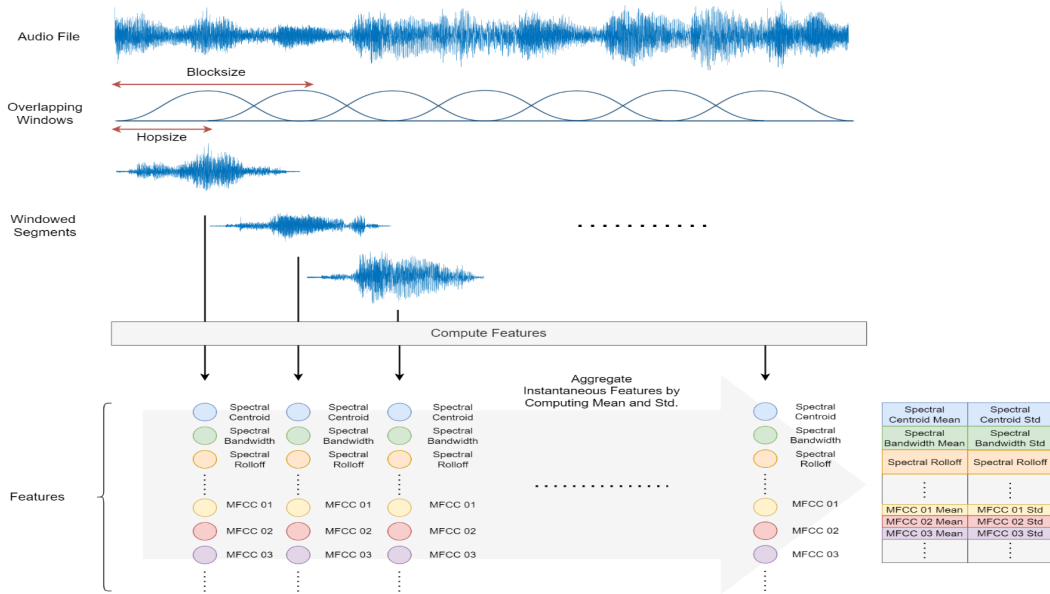


Figure S1. Feature Extraction of EMODB data. Each audio file was divided into smaller segments. We then computed the features for each segment as detailed in Table 1. After all features are extracted for each window, we aggregated all these instantaneous features by computing mean and standard deviation to represent the audio file.

References

- [1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.
- [2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.
- [3] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting, 2021.
- [4] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.
- [5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [7] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.
- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.