Knowledge and Information Systems (2026) 68:121
https://doi.org/10.1007/s10115-026-02744-y

RESEARCH

Semi-supervised time series classification via sequence neural process

Xin Song[1] · **Zhikui Chen**[1] · **Fangming Zhong**[1]

Received: 28 September 2025 / Revised: 28 January 2026 / Accepted: 19 March 2026

Abstract

Time series classification (TSC), essential for capturing temporal patterns in domains from healthcare to industrial monitoring, faces significant challenges under label scarcity. While deep semi-supervised methods mitigate annotation burdens, they remain vulnerable to epistemic uncertainty arising from insufficient evidence in low-label regimes. This data paucity exacerbates model overconfidence in spurious correlations, ultimately degrading generalization. To address this, we propose Sequence Neural Process with Multi-view Posterior Consistency (SNPMPC), a novel uncertainty-aware semi-supervised TSC framework based on Neural Processes (NPs). Motivated by the capacity of NPs for principled uncertainty quantification, SNPMPC fundamentally reformulates their optimization paradigm for sequence reasoning. It replaces the single global conditional distribution modeling with a sequence of conditional distributions, which dynamically captures evolving dependencies while inherently modeling epistemic uncertainty. Then, we introduce multi-view posterior alignment regularization to modify the distribution divergence regularization in the optimization paradigm of standard NPs that enforces distribution alignment between weak/strong augmented views of unlabeled data and the supervisory manifold of labeled instances, which injects richer signals to elevate latent variable quality. Extensive experiments demonstrate that the proposed approach can simultaneously quantify the epistemic uncertainty and significantly advance state-of-the-art classification accuracy.

Keywords Time series classification · Semi-supervised learning · Probabilistic model · Neural process

Xin Song, Zhikui Chen, and Fangming Zhong have contributed equally to this work.

✉ Zhikui Chen
zkchen@dlut.edu.cn

Xin Song
songxin72@mail.dlut.edu.cn

Fangming Zhong
fmzhong@dlut.edu.cn

[1] School of Software Technology, Dalian University of Technology, No. 2 Linggong Road, Dalian 116621, Liaoning, China

Published online: 11 April 2026

1 Introduction

Time series classification (TSC) represents a fundamental and widely studied problem in machine learning, focusing on assigning categorical labels to time-ordered sequences of real-valued observations. Due to its critical role in extracting meaningful patterns from temporal sequences, TSC has attracted substantial research interest in recent decades and has become indispensable in numerous real-world applications. Notably, in smart grid management, it facilitates electricity consumption forecasting and anomaly detection in power usage patterns. Furthermore, TSC plays a pivotal role in industrial safety and infrastructure protection, particularly in power system security, where it aids in fault diagnosis and predictive maintenance. The broad applicability and practical significance of TSC underscore its importance as a key research area in both academia and industry [1].

Significant research efforts have been devoted to developing TSC approaches, evolving from shapelet-based algorithms to deep learning algorithms. Contemporary state-of-the-art performance is predominantly achieved through deep learning algorithms, which demonstrate exceptional capability in extracting discriminative nonlinear features directly from raw temporal sequences [2–4]. Despite their advantages, these supervised deep learning algorithms have a fundamental limitation that they rely on large labeled datasets for effective training. This requirement poses substantial challenges in practical applications, as temporal sequences often exhibit intricate time-varying dynamic characteristics while demanding specialized domain knowledge for accurate annotation.

This challenge has propelled semi-supervised learning as a pivotal solution for TSC with limited labeled data [5]. Effective semi-supervised learning frameworks crucially leverage unlabeled instances through auxiliary learning tasks that capture supplementary temporal information. For instance, Eldele et al. [6] develop a contrastive learning architecture to preserve temporal dependencies between labeled and unlabeled instances with a temporal contrasting module and enhance the discriminability of feature representation with a context-aware feature distillation strategy. To develop a semi-supervised learning paradigm in TSC, TS-TFC treats the time domain and frequency domain of the same temporal sequences as two distinct views and employs pseudo-labels generated by one view to guide the training of the classifier of the other view [7]. Current semi-supervised time series classification (STSC) algorithms focus on learning temporal representations from a large amount of unlabeled sequences in the self-supervised learning framework. Nevertheless, prevailing STSC approaches exhibit critical limitations in handling model uncertainty, particularly epistemic uncertainty stemming from insufficient training evidence.

The intrinsic data scarcity in the semi-supervised learning configuration exacerbates model overfitting, where classifiers develop pathological confidence in spurious correlations. During inference, such models generate dangerously overconfident misclassifications due to the fragile learning decision boundaries. Similar overconfidence induced by the epistemic uncertainty has been widely reported in image-domain studies [8] and is expected to pose comparable risks in the STSC task. It underscores the imperative for uncertainty-aware learning frameworks that: (1) quantify epistemic uncertainty through probabilistic reasoning (e.g., Bayesian inference); (2) design entropy-based constraints to filter low-confidence unreliable pseudo-labels of unlabeled instances for classification. The entropy-based constraints embedded within our framework filter out low-confidence pseudo-labels and only retain those with high confidence, thus ensuring that pseudo-labels used in the learning process are both reliable and beneficial to model generalization.

To address this limitation, we explore a probabilistic uncertainty-aware learning framework for STSC with built-in uncertainty quantification. Neural Processes (NPs) emerge as a promising solution, which combines the flexibility of neural networks with the probabilistic rigor of Gaussian processes [9]. Specifically, NPs quantify epistemic uncertainty through variance or entropy derived from learned conditional Gaussian distributions. Sequential Neural Process (SNP) extends this capability by incorporating temporal state-transition models, enabling dynamic sequences of conditional distributions for time series analysis [10]. Nevertheless, SNP remains confined to the supervised regression prediction problem, failing to leverage uncertainty estimates for mitigating label scarcity. Moreover, SNP employs a step-by-step latent transition mechanism with a hidden unit to summarize sequential observations, which cannot capture long-range temporal dependencies, and uncertainty structures across multiple time steps. Recently, NPmatch demonstrates promising results in semi-supervised image domains through the single-pass uncertainty-aware pseudo-labeling strategy [11].

Despite their potential, directly applying NPs to STSC faces two main challenges. (1) The inherent time-varying nature of temporal concepts fundamentally violates the optimization paradigm imposed by standard NPs, which rely on a single global latent variable to model the conditional distribution of data. This single latent variable, sampled once one instance, is inherently inadequate for capturing temporal distribution shifts and evolving dependencies over time. (2) The scarcity of labeled data restricts the expressive power of learned latent variables, which leads to impoverished representations that fail to capture temporal dependencies hidden in entire input sequences. Consequently, this necessitates injecting richer, diverse learning signals to elevate latent variable quality beyond supervised constraints. Current approaches cannot simultaneously handle the above two challenges, which creates a critical gap in uncertainty-aware STSC methodologies.

In this article, we propose Sequence Neural Process with Multi-View Posterior Consistency (SNPMPC) as a foundational architecture for uncertainty-aware learning in dynamic systems, which open new avenues for probabilistic time series reasoning at the semi-supervised setting. First, SNPMPC reformulates the optimization paradigm of standard NPs at the STSC setting, which treats labeled input–output pairs as context points while dynamically expanding target points to include the whole dataset with a sequence of uncertainty-filtered pseudo-labels from unlabeled instances. Compared with conventional NPs that model the single conditional distribution $p(z)$, this novel formula learns a sequence of temporally conditional distributions $p(z_1)$, ..., $p(z_T)$. Thus, by explicitly learning the transition $p(z_t|z_{t-1})$ through latent variable sequences z_1, ..., z_T, the framework captures temporal evolution in dynamic systems. Simultaneously, considering that pseudo-labels are used in the optimization paradigm and low quality of pseudo-labels may affect the model performance, SNPMPC quantifies epistemic uncertainty through the entropy of predictions based on the sequence of latent variables and filters pseudo-labels with the quantified uncertainty.

To elevate latent variable quality based on unlabeled data, we develop a novel multi-view posterior alignment regularization. Specifically, our approach creates weak and strong augmented views of unlabeled data and aims to align the posterior distribution of weak/strong augmented views and the distribution manifold of labeled data, which enforces the latent variables to carry sufficient information to distinguish perturbations of unlabeled data. The proposed regularization modifies the optimization paradigm of standard NPs that simultaneously minimizes the conditional Gaussian distribution of labeled context points and that of weakly augmented unlabeled targets, and the corresponding divergence between labeled context points and strongly augmented unlabeled targets. Furthermore, we propose the implementation of SNPMPC and conduct experiments on five benchmark datasets, which outperform the state-of-the-art methods.

Overall, the main contributions of the paper are:

(1) We propose a novel semi-supervised TSC method based on NP, which is called SNPMPC. SNPMPC can estimate the uncertainty of the model prediction for TSC in a sequence stochastic process.
(2) To inject richer, diverse learning signals to elevate latent variable quality through the unsupervised constraint, we replace the distribution divergence term in the evidence low bound (ELBO) of NPs with the multi-view posterior alignment regularization, which enforces the posterior distributions of all unlabeled augmented views to simultaneously align with the distribution manifold of the labeled data.
(3) Extensive results on five benchmark datasets demonstrate that the proposed approach outperforms the state-of-the-art methods in terms of two standard evaluation metrics: Accuracy and MF1-Score.

2 Related works

This section organizes prior research through three lenses: time series classification, semi-supervised learning, and neural process architectures.

2.1 Time series classification

The widespread deployment of sensor networks has significantly increased the generation of time series data. This development has created a pressing need for effective time series classification (TSC) methods, leading researchers to focus on learning discriminative temporal feature representations [12–14].

Conventional feature-based approaches for TSC are primarily divided into two categories: shapelet-based and bag-of-patterns algorithms. The shapelet-based paradigm typically involves three key steps: (1) identifying discriminative subsequences (shapelets) through F-statistic optimization [15], (2) transforming raw sequences into feature vectors based on their minimum distances to selected shapelets [16], and (3) employing linear classifiers such as support vector machines (SVMs) for final prediction [17].

In contrast to shapelet-based algorithms, bag-of-words algorithms employ symbolic representation through three core processes: symbolic discretization, sliding window-based word extraction, and term frequency quantification. These methods build class-discriminative lexicons where characteristic term distributions encode temporal class signatures. The Symbolic Aggregate Approximation (SAX) [18] implements adaptive term frequency quantization strategies including normal distribution quantiles and time series-specific discretization. SAX-VSM [19] enhances the word extraction through TF-IDF weighted text representations derived from sliding window segments.

Although conventional TSC methods demonstrate strong performance, they require computationally intensive preprocessing steps for feature engineering and domain-specific expertise. To address these limitations, deep learning architectures have emerged as effective solutions that automatically learn discriminative patterns directly from raw temporal sequences, eliminating the need for manual feature construction or prior domain knowledge.

As fundamental deep learning algorithms, multilayer perceptrons (MLPs) remain widely adopted in TSC. Early work by Wang et al. demonstrated end-to-end MLP training for raw sequence analysis, establishing baseline performance metrics [3]. TCL [20] combines MLPs with nonlinear independent component analysis (ICA) for classification.

Besides MLP, researchers have also adapted the Convolutional Neural Network (CNN) architecture to capture the relationship between different variables within a single time step for TSC. This approach typically represents input sequences through channel-wise encoding analogous to image processing, where each timestep is treated as an independent channel. MCDCNN [21] introduces parallel convolutional pathways for multivariate inputs, employing dual convolutional blocks equipped with a ReLU activation layer and a pooling layer. MVCNN [22] employs multi-scale convolutional kernels to capture local temporal features. InceptionTime [23] combines five Inception modules to enhance model capacity. Nevertheless, MLP-based and CNN-based algorithms primarily focus on capturing local correlations in one time step, while neglecting global temporal ordering dependencies.

To model long-range dependencies, RNN-based and attention-based algorithms are proposed. Recurrent Neural Networks (RNNs) leverage the memory storage mechanism to learn global temporal information through the sequence-to-sequence architecture. S2SwA [24] employs bidirectional LSTM encoders to process variable-length inputs, coupled with decoders that generate fixed-length representations through unsupervised learning of sequence-independent patterns. Building on this foundation, SAE [25] implements an autoencoder framework using GRU units, enhanced by pretraining on large-scale unlabeled datasets to improve classification performance.

Compared with RNN-based algorithms, attention mechanisms capture long-range dependencies and extract rich contextual features through expansive receptive fields. MACNN [26] employs multi-scale convolutions to generate temporal feature maps, enabling to learn multi-scale temporal patterns along the time axis. The framework incorporates Squeeze-and-Excitation attention modules for feature recalibration, which dynamically suppresses useless information hidden in channels. FMLA [27] enhances local feature sensitivity of Transformer through deformable convolutional blocks combined with knowledge distillation techniques.

Despite demonstrating notable success in modeling long-range dependencies, RNN-based and attention-based algorithms confront the persistent challenge of requiring substantial annotated training data.

2.2 Semi-supervised learning

To solve this problem, researchers focus on semi-supervised time series classification (STSC) that learns supplementary information from unlabeled data.

Recent years have witnessed significant progress in semi-supervised learning algorithms, primarily focusing on pseudo-labeling and consistency regularization algorithms. Pseudo-labeling algorithms typically generate high-quality pseudo-labels for unlabeled data to augment training supervision. For instance, Lee et al. [28] propose assigning pseudo-labels to unlabeled instances through current model predictions to enhance training. Studies [29, 30] implemented label propagation by constructing nearest neighbor graphs incorporating both labeled and unlabeled data, based on the premise that adjacent instances share similar labels.

Consistency regularization algorithms integrate unsupervised regularization terms into the supervised loss function. The Π-Model [31] incorporates a consistency loss for unlabeled data, which ensures model predictions remain consistent across augmented variants. To address domain-specific augmentation requirements, Virtual Adversarial Training (VAT) [32] introduces adversarial perturbations to inputs that generate synthetic training targets. Building on these concepts, MixMatch [33] unifies multiple regularization approaches within a single framework, demonstrating state-of-the-art performance on classification benchmarks.

With the rapid development of semi-supervised learning algorithms, STSC has gained increasing attention. Jawed et al. [34] develop a joint training framework that combines supervised classification on labeled data with self-supervised forecasting across the entire dataset. Yue et al. [35] propose timestep-wise contrastive losses that simultaneously pull augmented temporal views closer while repelling dissimilar timesteps within sequences. To alleviate the model overfitting in low-data environments, Arunan et al. [36] propose to learn explanatory components of the time series (e.g., sensor factors, temporal factors) through the Intelligently Augmented Contrastive Tensor Factorization.

2.3 Neural processes

Neural Processes (NPs) constitute a family of probabilistic models that approximate stochastic processes by modeling conditional distributions through neural networks. Given input space $\mathcal{X}$ and output space $\mathcal{Y}$, a stochastic process is defined as a distribution over functions $f : \mathcal{X} \to \mathcal{Y}$. In particular, NP models a conditional prior $p(y_r|x_r) = p(y_r|f(x_r)) = p(y_r|z)p(z|g(x_c, z))$ for the target data (x_r, y_r), where latent variable z is parameterized by a neural network $g(x_c, z)$ and x_c represents a set of observed contexts. This framework models the conditional distribution as follows

$$p(y_r|x_c, y_c, x_r) = \int p(y_r|x_r, z)p(z|x_c, y_c)dz \tag{1}$$

NPs satisfy the Kolmogorov Extension Theorem (KET) [37] condition that includes two properties, exchangeability and consistency. Exchangeability: The conditional distribution between context data and target data is permutation invariant

$$p(y_r|x_c, y_c, x_r) = p(\pi(y_r)|\pi(x_c, y_c, x_r)) \tag{2}$$

where π is a permutation of data. Consistency: When a part of the input data is marginalized out, the resulting distribution is consistent

$$\begin{aligned} &p(y_r|x_c, y_c, x_r) = p(y_{rm}|x_{cm}, y_{cm}, x_{rm}) \\ &s.t. 1 \leq rm \leq r, 1 \leq cm \leq c \end{aligned} \tag{3}$$

The model parameters are learned by the following evidence low bound (ELBO) function

$$\begin{aligned} &\log p(y_r|x_c, y_c, x_r) \geq \mathbb{E}_{p(z|x_r)}[p(y_r|z, x_r) \\ &-KL(p(z|x_r)||p(z|x_r))] \end{aligned} \tag{4}$$

The principle underlying this function is to infer the target stochastic process from observed contexts. Due to the posterior's intractability, a variational approximation is employed, and a KL regularization term is incorporated into the loss objective to encourage the summary of observed contexts and target data to be close to each other.

NPs have demonstrated success across multiple domains, including regression, classification, and image completion. To our knowledge, this work presents the first application of NPs to tackle STSC challenges.

3 Methodology

In this section, we first provide a brief introduction to define the studied STSC problem and describe how to tackle it from the viewpoint of NP. Then, a detailed description of the proposed approach is presented.

3.1 Preliminaries

Let $\mathcal{T} = (X_{(1,1:T)}, Y_1), \ldots, (X_{(N,1:T)}, Y_N)$ denote a multivariate time series dataset, where $X_{(i,1:T)} = \{x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,T)}\}$ represents the input sequence of the i-th sample with corresponding label Y_i. Each observation $x_{(i,t)} \in \mathbb{R}^D$ at time step t contains D variables. The number of the whole dataset is N, and T is the number of time steps. In STSC, $\mathcal{T}$ is composed of a labeled subset $\mathcal{T}^L = (X_{(1,1:T)}, Y_1), \ldots, (X_{(M,1:T)}, Y_M)$ with M labeled samples ($M \ll N$) and an unlabeled subset $\mathcal{T}^U = X_{(M+1,1:T)}, \ldots, X_{(N,1:T)}$.

Existing methods typically learn a deterministic mapping function f for label prediction, which fails to account for model uncertainty. With fixed parameters, such functions exhibit limited flexibility when handling samples deviating from the training distribution. To overcome this limitation, we employ the NP framework to learn a conditional distribution $p(Y_{(1:N)}|X_{(1:N,1:T)}, z)$ parameterized by latent variable $z \sim \mathcal{N}(\mu, \Sigma)$, where μ denotes the mean vector and Σ the covariance vector. By treating labeled pairs $(X_{(1:M,1:T)}, Y_{1:M})$ as context points, the STSC task focuses on learning class-conditional distributions as follows

$$p(Y_{1:N}|X_{1:N,1:T}) = \int p(Y_{1:N}|X_{1:N,1:T}, z) \, p(z|(X_{1:M,1:T}, Y_{1:M}))dz \tag{5}$$

The obtained latent variable z models uncertainty through its Gaussian distribution. Unlike deterministic approaches that produce fixed outputs, the NP framework produces predictions via probabilistic sampling from a Gaussian distribution. Uncertainty quantification is subsequently achieved by computing the entropy of the resulting predictions.

Following the NP framework, labels of all training data are required for learning the conditional distribution $p(Y_{(1:N)}|X_{(1:N,1:T)}, z)$. Since the true labels $Y_{M+1:N}$ for unlabeled data are unavailable, we utilize pseudo-labels $\tilde{Y}_{M+1:N}$ generated by the pre-trained model. Model parameters are therefore optimized by maximizing the ELBO function over dataset $\mathcal{T}$

$$\begin{aligned} &\log p(Y_{1:N}|X_{1:N,1:T}) \geq \mathbb{E}_{q(z|\mathcal{T})}[\log(p(Y_{1:M}|z, X_{1:M,1:T})) \\ &+ \log(p(\tilde{Y}_{M+1:N}|z, X_{M+1:N,1:T})) \\ &- \log \tfrac{q(z|X_{M+1:N,1:T}, \tilde{Y}_{M+1:N})}{q(z|X_{1:M,1:T}, Y_{1:M})}] \end{aligned} \tag{6}$$

the total optimzie objective comprises three loss terms: (1) The first term corresponds to supervised learning using cross-entropy loss on labeled data. (2) The second term incorporates pseudo-label-based loss for unlabeled data. (3) The third term minimizes the Kullback–Leibler (KL) divergence between posterior distributions $q(z|X_{(1:M,1:T)}, Y_{1:M})$ and $q(z|X_{(M+1:N,1:T)}, \tilde{Y}_{M+1:N})$, which enforces consistency between labeled and unlabeled posterior distributions.

3.2 Sequence neural process with multi-view posterior consistency

3.2.1 Sequence neural process for multivariate time series

However, directly introducing the NP into the STSC faces two key limitations. First, standard NPs employ a single latent variable z across all data, which fails to capture long-range temporal dependencies among multiple time steps. They lack explicit modeling of temporal evolution in conditional distributions $p_{t-1} \rightarrow p_t$. To address this issue, a novel autoregressive

NP formula is designed to govern temporal dynamics

$$p(Y_{1:N}|X_{1:N,1:T}) = \prod_{t=1}^{T} \int p(Y_{1:N}|X_{(1:N,1:T)}, z_t) \\ p(z_t|z_{<t}, (X_{1:M,1:T}, Y_{1:M}))dz \tag{7}$$

where z_t represents the latent variable at the time step t, and $z_{<t}$ denotes historical latent variables. When $t = 1$, $p(z_t|z_{<t}, (X_{1:M,1:T}, Y_{1:M})) = 1$. This formula integrates the advantages of autoregressive modeling with the neural process framework, which aims to establish a sequence neural process that effectively captures temporal structures in class-conditional distributions.

Based on this, the posterior distribution over multivariate time series $q(z|X_{1:M,1:T}, Y_{1:M})$ and $q(z|X_{M+1:N,1:T}, \tilde{Y}_{M+1:N})$ can be approximated with the following temporal autoregressive factorization

$$q(z|X_{1:M,1:T}, Y_{1:M}) = \prod_{t=1}^{T} q(z_t|z_{<t})q(z_{<t}|X_{1:M,1:T}, Y_{1:M}) \tag{8}$$

$$q(z|X_{M+1:N,1:T}, \tilde{Y}_{M+1:N}) = \prod_{t=1}^{T} q(z_t|z_{<t})q(z_{<t}|X_{M+1:N,1:T}, \tilde{Y}_{M+1:N}) \tag{9}$$

With the above approximate posterior distributions, the ELBO function is reformulated as

$$\begin{aligned} &\log p(Y_{1:N}|X_{1:N,1:T}) \geq \frac{1}{T} \sum_{t=1}^{T} (\mathbb{E}_{p(z_t|\mathcal{T})}[\\ &\log(p(Y_{1:M}|z_t, X_{1:M,t})) + \log(p(\tilde{Y}_{M+1:N}|z_t, X_{M+1:N,t})) \\ &- \log \frac{q(z_{<t}|X_{M+1:N,t}, \tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t}, Y_{1:M})}]) \end{aligned} \tag{10}$$

when $t = 1$, $\frac{q(z_{<t}|X_{M+1:N,t}, \tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t}, Y_{1:M})} = 1$.

3.2.2 Multi-view posterior alignment regularization

Solely minimizing the KL divergence between $q(z_{<t}|X_{1:M,t}, Y_{1:M})$ and $q(z_{<t}|X_{M+1:N,t}, \tilde{Y}_{M+1:N})$ leads to suboptimal performance. This stems from insufficient learning of the distribution $q(z_{<t}|X_{1:M,t}, Y_{1:M})$ due to limited labeled data ($M \ll N$). We therefore propose multi-view posterior alignment regularization, which enforces the posterior distributions of all unlabeled augmented views to simultaneously align with the distribution manifold of the labeled data. Thus, our approach injects richer signals to elevate latent variable quality through unsupervised constraint.

Following the principle of consistency regularization [38], we employ weak and strong augmentations to process unlabeled data instances $X_{M+1:N,1:T}$, generating two distinct augmented views. Limited labels often lead to miscalibrated epistemic uncertainty in semi-supervised settings, allowing a model to maintain output consistency while internally inferring inconsistent latent temporal-invariant structure for the same prediction. Therefore, it is essential to design a novel posterior alignment regularization that reduces the inconsistency in feature distributions to pseudo-label noise. Specifically, $X^{U_w}_{M+1:N,1:T}$ and $X^{U_s}_{M+1:N,1:T}$

denote the weakly and strongly augmented views, respectively. The enhanced evidence lower bound (ELBO) incorporating multi-view posterior consistency can be expressed as

$$\begin{aligned}
&\log p(Y_{1:N}|X_{1:N,1:T}) \geq \tfrac{1}{T}\sum_{t=1}^{T}(\mathbb{E}_{q(z_t|\mathcal{T})}[\\
&\log(p(Y_{1:M}|z_t, X_{1:M,t})) + \log(p(\tilde{Y}_{M+1:N}|z_t, X^{U_w}_{M+1:N,t}))\\
&-\tfrac{1}{2}(\log\tfrac{q(z_{<t}|X^{U_w}_{M+1:N,t},\tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t},Y_{1:M})}\\
&+\log\tfrac{q(z_{<t}|X^{U_s}_{M+1:N,t},\tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t},Y_{1:M})})])
\end{aligned} \tag{11}$$

Furthermore, we propose to replace the conventional KL divergence term with geometric Jensen–Shannon (JS) divergence for computing distribution discrepancy loss. Unlike KL divergence, JS divergence satisfies the symmetry property and does not require absolute continuity. When both calculated distributions are Gaussian distributions, the sum of JS divergence can be calculated in closed form, which allows the optimization of the ELBO function in a computationally efficient way. For two distributions p and q, the geometric JS divergence objective is formulated as

$$\begin{aligned}
&D_{GJS}(p,q) =\\
&(1-\alpha)KL(p||G_\alpha(p,q)) + \alpha KL(q||G_\alpha(p,q))\\
&= (1-\alpha)\textstyle\int p(x)\log\frac{p(x)}{G_\alpha(p(x),q(x))}dx\\
&+\alpha\textstyle\int q(x)\log\frac{q(x)}{G_\alpha(p(x),q(x))}dx
\end{aligned} \tag{12}$$

where $\alpha \in [0,1]$ is the distribution weight of two distributions. $G_\alpha(p(x),q(x)) = p(x)^{1-\alpha}q(x)^\alpha$ is the weighted geometric mean. KL represents the Kullback–Leibler divergence function.

Under the NP theory, the inferred posterior distributions follow Gaussian forms: $q(z_{<t}|X_{1:M,t}, Y_{1:M}) \sim \mathcal{N}_{1,t}(\mu_{1,t}, \Sigma_{1,t})$, $q(z_{<t}|X^{U_w}_{M+1:N,t}, \tilde{Y}_{M+1:N}) \sim \mathcal{N}_{2,t}(\mu_{2,t}, \Sigma_{2,t})$, and $q(z_{<t}|X^{U_s}_{M+1:N,t}, \tilde{Y}_{M+1:N}) \sim \mathcal{N}_{3,t}(\mu_{3,t}, \Sigma_{3,t})$. Consequently, the distribution divergence term can be extended as

$$\begin{aligned}
&\log\tfrac{q(z_{<t}|X^{U_w}_{(M+1:N,t)},\tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t},Y_{1:M})} + \log\tfrac{q(z_{<t}|X^{U_s}_{M+1:N,t},\tilde{Y}_{M+1:N})}{q(z_{<t}|X_{1:M,t},Y_{1:M})}\\
&= D_{GLS}(\mathcal{N}_{2,t},\mathcal{N}_{1,t}) + D_{GLS}(\mathcal{N}_{3,t},\mathcal{N}_{1,t})\\
&= \tfrac{1}{2}(tr(\Sigma^{-1}_{\alpha_{u,t}}B^1_{2,t}) + tr(\Sigma^{-1}_{\alpha_{u,t}})B^1_{3,t})\\
&\quad +2(1-\alpha_{u,t})(A^T_{1,t}\Sigma^{-1}_{\alpha_{u,t}}A_{1,t} + \alpha_u(A^T_{2,t}\Sigma^{-1}_{\alpha_{u,t}}A_{2,t} + A^T_{3,t}\Sigma^{-1}_{\alpha_{u,t}}A_{3,t}))\\
&\quad +log[\tfrac{det[\Sigma_{\alpha_{u,t}}]}{det[\Sigma_{1,t}]^{1-\alpha_{u,t}}det[\Sigma_{2,t}]^{\alpha_{u,t}}}]\\
&\quad +log[\tfrac{det[\Sigma_{\alpha_{u,t}}]}{det[\Sigma_{1,t}]^{1-\alpha_u}det[\Sigma_{3,t}]^{\alpha_u}}]) - D^z
\end{aligned} \tag{13}$$

where D^z is the dimension of z, $det[]$ is the determinant, $\alpha_{u,t} = C_{avg1,t}/(C_{avg1,t} + C_{avg2,t} + C_{avg3,t})$. $C_{avg,t}$ represents the average predictions at current time step t based on different samples. $\Sigma_{\alpha_{u,t}} = ((1-\alpha_{u,t})\Sigma^{-1}_{1,t} + \frac{1}{2}\alpha_{u,t}(\Sigma^{-1}_{2,t} + \Sigma^{-1}_{3,t}))^{-1}$, and $\mu_{\alpha_{u,t}} = \Sigma_{\alpha_{u,t}}((1-\alpha_{u,t})\Sigma^{-1}_{1,t}\mu_{1,t} + \frac{1}{2}\alpha_{u,t}(\Sigma^{-1}_{2,t}\mu_{2,t} + \Sigma^{-1}_{3,t}\mu_{3,t}))$. $A_{1,t} = \mu_{\alpha_{u,t}} - \mu_{1,t}$. $A_{2,t}, A_{3,t}$ are defined analogously. $B^1_{2,t} = ((1-\alpha_{u,t})\Sigma_{1,t} + \alpha_{u,t}\Sigma_{2,t})$, $B^1_{3,t}$ is defined analogously.

3.3 Training and inference

This section demonstrates the workflow of the SNPMPC framework during the training and inference phases, which is demonstrated in Fig. 1.

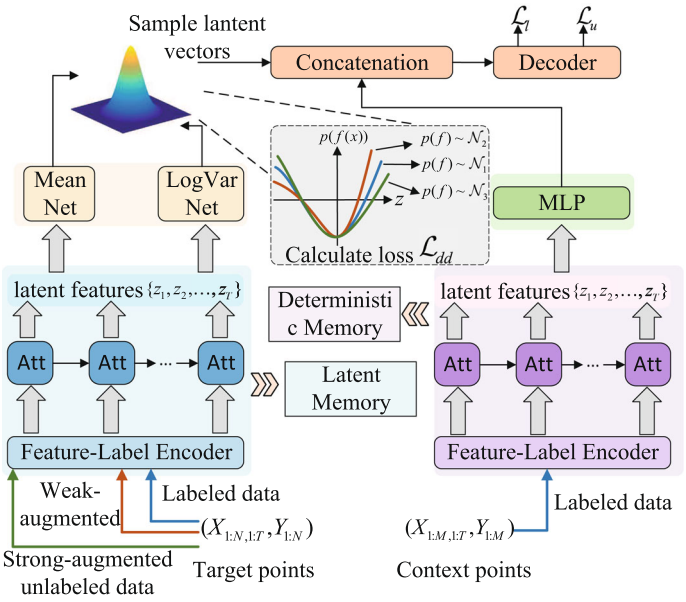


Fig. 1 The overall architecture of implementing SNPMPC

1)Training: Before training, unlabeled data requires an augmentation pre-processing. Strong augmentation employs splitting-and-shuffling operations to induce significant input perturbations, while weak augmentation applies scaling-and-shifting operations to preserve natural characteristics with subtle variations.

The SNPMPC framework consists of three components: the encoder, the NP module, and the decoder. At the beginning, the encoder extracts features from labeled data and both augmented views of unlabeled data. Features of labeled data $h_{(1:M,1:T)}$ combined with input labels $Y_{1:M}$ are used to establish context points. Target points comprise all available data $(h_{(1:O,1:T)}, \hat{Y}_{1:O})$. For unlabeled samples where $Y_{M+1:N}$ are unavailable, pseudo-labels $\tilde{Y}_{M+1:O}$ derived from high-confident weak-augmented predictions $\tilde{Y}_{M+1:O} = \{\tilde{y}_i | \tilde{y}_i = \sum_{j=1}^{K} C_i^{(U^w,j)} \log C_i^{(U^w,j)}, \tilde{y}_i > \omega\}$ are employed. $C_i^{U^w}$ denotes the prediction result of weak-augmented unlabeled samples based on the NP module and the decoder, which will be described in the following paragraph. K is the number of classes. Because pseudo-labels generated by a pre-trained model on scarce data could potentially be subject to overfitting and overconfidence, our approach introduces a selection mechanism to quantify and mitigate uncertainty. We assume that the number of selected pseudo-labels is O and O changes during the training process in accordance with the variations of the selecting operation. The confidence coefficient ω is set to 0.4 in the experiment. Subsequently, the NP module processes these context and target points separately.

The NP module comprises two pathways: a latent path and a deterministic path. The latent path processes target points $(h_{(1:M,1:T)}, Y_{1:M})$, $(h^{U_w}_{(M+1:M+O,1:T)}, \tilde{Y}_{M+1:M+O})$, $(h^{U_s}_{M+1:M+O,1:T}, \tilde{Y}_{M+1:M+O})$ through two MLP layers to produce latent vectors $s_{1:M,1:T}$, $s^{U_w}_{M+1:M+O,1:T}$, $s^{U_s}_{M+1:M+O,1:T}$. Then, we employ a mean aggregator to average these representations that enforces exchangeability and consistency constraints. Concurrently, a latent memory bank $\mathcal{M}_{lat}$ stores these vectors through a first-in-first-out update strategy,

$s_{(1:O,1:T)} \rightarrow \mathcal{M}_{lat}$. Based on the obtained latent vectors, we build an autoregressive model f_{ag} (implemented via a transformer architecture) to learn a sequence of mean and variance vectors across multiple time steps,

$$\mu_{1,1:T} = f_{ag}(s_{1:N,1:T}), \quad \Sigma_{1,1:T} = f_{ag}(s_{1:N,1:T}) \tag{14}$$

$$\mu_{2,1:T} = f_{ag}(s^{U_w}_{M+1:M+O,1:T}), \quad \Sigma_{2,1:T} = f_{ag}(s^{U_w}_{M+1:M+O,1:T} \tag{15}$$

$$\mu_{3,1:T} = f_{ag}(s^{U_s}_{M+1:M+O,1:T}), \quad \Sigma_{3,1:T} = f_{ag}(s^{U_s}_{M+1:M+O,1:T}) \tag{16}$$

After that, the mean and variance vectors are used for sampling latent variables $z_{1:M,1:T} \sim \mathcal{N}_{1,T}(\mu_{1,1:T}, \Sigma_{1,1:T})$, $z^{U_w}_{M+1:M+O,1:T} \sim \mathcal{N}_{2,T}(\mu_{2,1:T}, \Sigma_{2,1:T})$, $z^{U_s}_{M+1:M+O,1:T} \sim \mathcal{N}_{3,T}(\mu_{3,1:T}, \Sigma_{3,1:T})$ via the reparameterization trick.

The deterministic path employs analogous processing to the latent path, which utilizes a mean aggregator and an autoregressive model to produce deterministic vectors $r_{1:M,1:T}$ of context points. These vectors are stored in a deterministic memory bank $\mathcal{M}_{\text{deter}}, r_{(1:M,1:T)} \rightarrow \mathcal{M}_{\text{deter}}$. Unlike the latent path, this path directly outputs average deterministic vectors without requiring reparameterization

$$r_{\text{avg},1:T} = \frac{1}{M}\sum_{i=1}^{M}(r_{(i,1:T)}) \tag{17}$$

Finally, the decoder concatenates deterministic vectors with sampled latent variables and then makes a sequence of predictions through an MLP for labeled data and unlabeled data as follows,

$$C_{1:M} = f_{de}([h_{1:M,1:T}, z_{1:M,1:T}, r_{avg,1:T}]) \tag{18}$$

$$C^{U_w}_{M+1:M+O} = f_{de}([h^{U_w}_{(M+1:M+O,1:T)}, z^{U_w}_{M+1:M+O,1:T}, r_{avg,1:T}]) \tag{19}$$

$$C^{U_s}_{M+1:M+O} = f_{de}([h^{U_s}_{(M+1:M+O,1:T)}, z^{U_s}_{M+1:M+O,1:T}, r_{avg,1:T}]) \tag{20}$$

$[\cdot]$ represents a feature concatenation operation. Prediction uncertainty is quantified via the entropy of outputs $\sum_{k=1}^{K} C^{(U^w,k)} \log C^{(U^w,k)}$. The training objective combines three loss functions: two cross-entropy loss terms and a distribution divergence loss term. The first loss term, the cross-entropy loss of labeled data, is as follows

$$\mathcal{L}_l = \frac{1}{M}\sum_{i=1}^{M} cross\_entropy(C_i, Y_i) \tag{21}$$

where C_i is the prediction of the i-th sample. The second loss is a cross-entropy loss between strong predictions of augmented unlabeled view and selected pseudo-labels based on weak-augmented unlabeled view

$$\mathcal{L}_u = \frac{1}{O}\sum_{i=M+1}^{O} cross\_entropy(C^{U_s}_i, \tilde{Y}_i) \tag{22}$$

where $\tilde{Y}_i$ represents the pseudo-label of i-th sample. To minimize the divergence among posterior distributions, the distribution divergence loss is designed

$$\mathcal{L}_{dd} = \frac{1}{T}\sum_{t} D_{GLS}(\mathcal{N}_{2,t}, \mathcal{N}_{1,t}) + D_{GLS}(\mathcal{N}_{3,t}, \mathcal{N}_{1,t}) \tag{23}$$

Overall, the goal of our approach is to minimize the following loss function

$$\mathcal{L} = \lambda_1(\mathcal{L}_l + \mathcal{L}_u) + \lambda_2 \mathcal{L}_{dd} \tag{24}$$

where λ_1 and λ_2 are the hyperparameters of cross-entropy loss and distribution divergence loss, respectively.

Algorithm 1 The workflow of SNPMPC framework

Require: Labeled dataset $\mathcal{T}^L = \{X_{(1:M,1:T)}, Y_{1:M}\}$, Unlabeled dataset $\mathcal{T}^U = X^{U_w}_{(M+1:N,1:T)}, X^{U_s}_{(M+1:N,1:T)}$, Test sample $x_{*,1:T}$, Training epochs Epo
Ensure: Test prediction results C_*
while $e < Epo$ **do**
  Extract features by the encoder $h_{(1:M,1:T)}, h^{U_w}_{(M+1:N,1:T)}, h^{U_s}_{(M+1:N,1:T)}$
  Use the model output $C^{U_w}_{(M+1:N,1:T)}$ to generate pseudo-labels $\tilde{Y}_{M+1:N}$ and select O samples with low uncertainty
  Calculate deterministic vectors $r_{(1:M,1:T)}$, and Store them into the deterministic memory bank $r_{(1:M,1:T)} \rightarrow \mathcal{M}_{\text{deter}}$
  while $t < T$ **do**
    Take $\{h_{(1:M,t)}, Y_{1:M}\}$ as inputs to calculated $s_{(1:M,t)}$
    Take $\{h^{U_w}_{(M+1:O,t)}, \tilde{Y}_{M+1:O}\}$ and $\{h^{U_s}_{(M+1:O,t)}, \tilde{Y}_{M+1:O}\}$ as inputs to calculate $s^{U_w}_{(M+1:O,t)}$ and $s^{U_s}_{(M+1:O,t)}$
    Calculate $\mu_{1,t}, \Sigma_{1,t}, \mu_{2,t}, \Sigma_{2,t}, \mu_{3,t}, \Sigma_{3,t}$
    Generate $z_{1,t}, z_{2,t}, z_{3,t}$ by the reparameterization trick and Store $s_{(1:O,t)}$ into the latent memory bank
  end while
  Calculate the prediction $C_{1:O}$
  Take Eq.(24) as the total loss function to train the model
end while
extract feature for test samples h_*
obtain deterministic vector $\bar{r}_{1:T}$
while $t < T$ **do**
  obtain latent vector s_t from memory bank
  Calculate $\bar{\mu}_t$ and $\bar{\Sigma}_t$
  generate $\bar{z}_t$ by the reparameterization trick
end while
concatenate feature vectors h_*, $\bar{r}_{1:T}$, $\bar{z}_{1:T}$
Output the prediction C_*

2)Inference: The inference phase of the SNPMPC framework involves processing a given sequence and generating classification predictions while maintaining an uncertainty-aware mechanism. This section describes the step-by-step approach for inference, highlighting how the model extracts features, makes predictions, and estimates uncertainty using the posterior distributions.

First, the encoder processes input test samples and produces temporal features $h_* = f_{en}(x_{(*,1:T)})$. Unlike the training stage, there are no additional data augmentation operations before encoding.

Next, the NP module is applied to sequentially predict the posterior distribution of latent variables. Instead of directly extracting latent variables and deterministic feature vectors from inputs, the deterministic features and latent vectors are obtained by a memory mechanism during inference. Given the latent memory bank $\mathcal{M}_{lat}$, latent vectors for test samples are

defined as

$$s_{(1:E,1:T)} \leftarrow \mathcal{M}_{lat} \tag{25}$$

$$\bar{s}_{1:T} = MeanAgg(s_{(1:E,1:T)}) \tag{26}$$

where $MeanAgg$ denotes the summation and averaging operation. E is the memory size. Similarly, the deterministic vectors also are obtained from the deterministic memory bank

$$r_{(1:E,1:T)} \leftarrow \mathcal{M}_{deter} \tag{27}$$

$$\bar{r}_{1:T} = MeanAgg(r_{(1:E,1:T)}) \tag{28}$$

These memory banks can be viewed as a set of context representations rather than a sequence, which are not queried sequentially but set-wise. This aligns with the core NP assumption that feature vectors are order-invariant. In other words, latent vectors and deterministic vectors contain the global information of samples, rather than the specific information of a single sample.

Then, latent vectors are then sampled from the Gaussian distribution parameterized by the predicted mean and variance. This sampling is performed using the reparameterization trick to ensure differentiability

$$\bar{\mu}_{1:T} = f_{ag}(\bar{s}_{1:T}), \quad \bar{\Sigma}_t = f_{ag}(\bar{s}_{1:T}) \tag{29}$$

$$\bar{z}_{1:T} = \bar{\mu}_{1:T} + \bar{\Sigma}_{1:T} \odot \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{30}$$

where ϵ is a noise term sampled from a standard normal distribution.

After that, these vectors are then concatenated to produce a class probability vector

$$C_* = f_{de}([h_*;\ \bar{r}_{1:T};\ \bar{z}_{1:T}]) \tag{31}$$

This fusion of deterministic and stochastic components allows the model to leverage both explicit features and latent dependencies.

4 Experiments

We evaluate the proposed approach on multiple time series datasets, presenting comprehensive experimental settings, comparative methods, and results analysis.

4.1 Datasets

Our experiments are conducted on three publicly available real-world datasets, including HAR, Epilepsy, and UCR datasets, which cover different time series applications.

1) Human Activity Recognition: HAR [39] comprises smartphone accelerometer and gyroscope readings, which include 6 activities.
2) Epilepsy Seizure Prediction: Epilepsy [40] contains 23.6-second EEG recordings from 500 subjects. We treat four classes that do not include epileptic seizures as a single class and establish a binary classification task.
3) UCR Repository Datasets: 3 UCR datasets are selected for evaluation. The Wafer dataset contains semiconductor fabrication process measurements with two classes. Ford comprises 9367 automotive sensor recordings (4921 normal/4446 fault samples). StarLightCurves is a dataset with 1024-length phase-aligned light curves from 1000 stars.

4.2 Evaluation metrics

Two evaluation metrics (accuracy and macro-averaged F1-score (MF1)) are employed to verify model performance on the three datasets. Accuracy quantifies the proportion of correct predictions, providing a straightforward measure of overall performance. The MF1 metric calculates the harmonic mean of precision and recall across all classes, which is defined as follows

$$\text{MF1 - score} = \frac{1}{\text{CA}} \sum_{i=1}^{\text{CA}} \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i} \tag{32}$$

where CA is the number of classes, $precision_i$ and $recall_i$ represent the precision and recall of i-th class.

4.3 Implement details

The proposed approach is an end-to-end classification model, which contains three parts: the encoder, the NP module, and the decoder. The encoder is constituted by three convolutional layers. The kernel size of each convolutional layer is set to 8. The NP module consists of three parts: the auto-regressive model, the latent path, and the deterministic path. The auto-regressive model extracts temporal features for the following distribution learning, which is built by a transformer. The transformer has 4 heads and 4 residual modules. The latent path and the deterministic path have 2 layers, and each layer is an MLP layer. For the latent path, we use two MLP layers as the mean and logvar net, respectively, to learn the conditional Gaussian distribution over latent variables. The decoder is composed of an MLP layer and a linear layer. The SNPMPC framework operates efficiently on GPUs with 3.55 Mflops. The model size consists of 0.69 million parameters.

The ratio of training, validation, and testing samples is 60%, 20%, and 20%. The batch size of labeled samples is set to 8, and the batch size of unlabeled samples is set to 56. The proposed approach is trained for 100 epochs by the SGD optimizer with a learning rate of 3e-2, weight decay of 3e-4, and momentum of 0.9. For the unlabeled sample augmentation, we adopt permutation as the strong augmentation and scaling as the weak augmentation. The strong augmentation sets the number of segments to 12 for the Epilepsy dataset, to 20 for the sleep-EDF dataset, and to 10 for all other datasets. The weak augmentation sets the scaling ratio to 2. The default setting of loss hyperparameters of SNPMPC is $\lambda_1 = 1$, $\lambda_2 = 0.01$. The confidence coefficient ω is set to 0.4. On the HAR dataset, the average time per epoch is 0.24 min, with a total training time of 25 min for convergence. Inference time for a single sample is 1.9 ms, demonstrating the model's efficiency in both training and deployment.

4.4 Experimental results

To evaluate the performance of SNPMPC, we compare it with five self-supervised methods (Random Initialization, Supervised, SSL-ECG [41], CPC [42], SimCLR [5]) and six semi-supervised methods (Mean-Teacher [43], DivideMix [44], SemiTime [45], FixMatch [46], TS-TCC [6], CA-TCC [6]). It is worth noting that SSL-ECG, CPC, SimCLR, and TS-TCC all have the pre-training process over fully unlabeled data.

Experimental results presented in Tables 1 and 2 demonstrate the superior performance of SNPMPC at 1% and 5% labeled data settings. With only 1% labeled data, SNPMPC achieves

Table 1 Performance comparison for semi-supervised time series classification on HAR and Epilepsy datasets

Datasets	HAR		Epilepsy	
Methods	Accuracy	MF1-score	Accuracy	MF1-score
1% of labeled data				
Random Init	39.8±3.8	34.7±5.0	70.3±2.1	66.2±2.6
Supervised	44.9±6.7	41.0±6.7	76.1±0.7	74.8±0.4
SSL-ECG	60.0±4.0	54.0±6.0	89.3±0.4	86.0±0.3
CPC	65.4±1.4	63.8±1.7	88.9±1.1	85.8±0.3
SimCLR	65.8±0.7	64.3±0.9	88.3±1.5	84.0±1.0
TS-TCC	70.5±0.3	69.5±0.5	91.2±0.5	89.2±0.2
Mean-Teacher	75.9±1.9	74.0±2.8	91.5±0.3	90.6±0.6
DivideMix	76.5±0.7	75.4±1.0	90.9±0.7	89.4±1.4
SemiTime	77.6±1.1	76.3±0.9	91.6±0.3	90.8±0.6
FixMatch	76.4±2.5	75.6±2.8	93.2±0.2	92.2±0.5
CA-TCC	77.3±0.6	76.2±0.1	92.0±0.1	91.9±0.1
SNPMPC	**85.4±1.1**	**85.1±1.2**	**95.4±0.3**	**92.5±0.5**
5% of labeled data				
Random Init	49.6±2.5	45.8±2.0	75.5±3.6	70.5±3.3
Supervised	52.8±1.5	50.9±0.2	83.4±0.7	80.4±0.7
SSL-ECG	63.7±5.3	58.6±7.4	92.8±0.2	89.0±0.3
CPC	75.4±2.1	74.7±2.5	92.8±0.3	90.2±0.5
SimCLR	75.8±1.4	74.9±1.5	74.9±1.5	89.2±1.0
TS-TCC	77.6±1.8	76.7±1.7	93.1±0.3	93.7±0.6
Mean-Teacher	88.2±1.2	88.1±1.2	94.0±0.4	93.6±0.7
DivideMix	89.1±2.0	89.1±1.3	93.9±0.6	93.4±1.1
SemiTime	87.6±1.3	87.1±0.8	94.0±0.5	93.0±0.9
FixMatch	87.6±0.3	87.3±0.4	93.7±1.4	92.4±0.3
CA-TCC	88.3±0.4	88.3±0.3	94.5±0.1	**94.0±0.1**
SNPMPC	**88.4±0.5**	**88.5±0.5**	**95.9±0.2**	93.5±0.4

The bold values represent the highest / optimal values

statistically significant improvements across all 5 benchmark datasets. The largest performance gaps occur in the HAR dataset (8.1% accuracy and 8.9% MF1-score improvement over the second-best method) and StarLightCurves dataset (4.5% accuracy and 9.7% MF1-score gains). Moderate yet consistent improvements are observed across other datasets, including 5.8% accuracy and 6.4% MF1-score gains on Ford, 3.4% accuracy improvement on Epilepsy, and 1.6% MF1-score enhancement on Wafer. When the labeled data proportion increases to 5%, SNPMPC maintains performance leadership on 5 datasets while showing negligible performance gaps (exhibiting a marginal 0.5% MF1-score deficit on the Epilepsy dataset, 0.9% Accuracy, and 0.9% MF1-score deficit on the Epilepsy dataset). Notably, SNPMPC verifies the effectiveness for temporal dependency modeling, while CA-TCC achieves comparable performance on the Ford and Epilepsy datasets.

Additionally, comparative analysis reveals two critical patterns. First, contrastive methods (e.g., FixMatch, CA-TCC) systematically outperform pretext-based approaches (e.g., SSL-ECG), confirming the importance of data augmentation for unlabeled data consistency.

Table 2 Performance comparison for semi-supervised time series classification on the UCR dataset

Datasets	Wafer		Ford		StarLightCurves	
Methods	Accuracy	MF1-score	Accuracy	MF1-score	Accuracy	MF1-score
1% of labeled data						
Random Init	90.6±1.6	58.1±2.1	52.5±1.8	47.5±6.4	83.9±1.6	65.4±3.6
Supervised	91.9±1.3	67.6±9.2	51.9±2.6	48.0±3.6	78.8±0.9	71.4±0.1
SSL-ECG	93.4±0.5	76.1±2.4	64.4±6.2	60.5±7.9	78.3±0.9	72.0±0.8
CPC	93.5±0.4	78.4±1.5	66.8±3.1	65.0±3.9	80.8±1.4	74.4±0.6
SimCLR	93.8±0.2	78.5±1.1	50.9±1.3	49.8±2.2	80.6±0.6	71.6±0.2
TS-TCC	93.2±0.8	76.7±4.6	72.7±0.9	71.9±1.0	86.0±0.4	79.2±0.7
Mean-Teacher	94.7±0.2	84.7±0.3	65.9±2.8	65.8±2.8	79.4±0.5	77.7±0.6
DivideMix	93.2±0.5	82.0±0.8	54.5±2.8	54.1±3.2	79.0±0.5	77.2±0.4
SemiTime	94.4±0.6	84.4±1.2	67.6±2.2	67.5±2.3	79.5±0.5	77.8±0.6
FixMatch	95.0±0.4	84.8±1.2	56.7±5.9	55.4±6.9	77.2±0.3	71.6±0.1
CA-TCC	95.1±0.3	85.1±0.6	73.8±1.5	73.0±1.8	85.8±0.7	77.8±0.5
SNPMPC	**95.6±0.4**	**86.7±0.7**	**79.6±0.1**	**79.4±0.0**	**90.3±0.3**	**87.5±0.8**
5% of labeled data						
Random Init	91.2±1.2	65.5±8.2	51.3±3.2	48.2±5.3	74.2±1.4	69.8±4.1
Supervised	94.6±0.3	83.9±0.6	60.5±2.8	58.8±3.7	81.8±0.8	71.4±4.1
SSL-ECG	94.9±0.3	84.5±0.7	71.7±3.1	69.8±3.8	82.6±1.3	74.5±1.3
CPC	92.5±0.4	79.4±0.8	86.3±0.8	86.2±0.8	89.1±1.0	84.5±0.8
SimCLR	94.8±0.2	83.3±0.6	63.0±3.0	60.7±4.2	84.2±1.3	74.0±2.3
TS-TCC	93.2±0.4	81.2±0.7	86.1±1.5	85.9±1.6	89.6±0.2	82.7±0.9
Mean-Teacher	94.4±0.7	83.8±1.4	64.6±3.8	62.7±5.5	84.9±2.0	83.9±1.4
DivideMix	94.7±0.6	84.6±1.5	60.2±5.6	57.9±7.1	85.6±2.8	84.1±2.1
SemiTime	95.0±0.4	84.7±1.0	65.0±4.9	62.6±7.1	84.6±4.8	83.8±3.7
FixMatch	94.9±0.6	84.4±1.2	62.7±5.8	60.7±7.5	84.1±2.0	77.5±3.0
CA-TCC	95.8±0.2	85.2±0.6	**88.2±0.4**	**88.2±0.4**	88.8±0.7	81.1±2.0
SNPMPC	**97.5±0.1**	**93.0±0.0**	87.3±0.2	87.3±0.3	**92.8±0.1**	**90.4±0.2**

Second, models that aim to capture global temporal dependencies (CA-TCC, SNPMPC) achieve improvements over other baselines.

To analyze the relationship between predictive uncertainty and classification accuracy, we present a quantitative analysis in Fig. 2. This observation aligns with the theoretical framework of NPs, wherein enhanced label availability strengthens the model posterior estimation and reduces the model uncertainty. Samples with high uncertainty range exhibit lower accuracy compared to those with low uncertainty. It verifies that uncertainty quantification can be treated as a selection criterion for choosing pseudo-labels of unlabeled samples.

Compared with CA-TCC, the number of overconfidence incorrect classifications made by our framework explicitly is much lower, which is shown in Fig. 5. These results empirically validate our claim in the introduction that existing semi-supervised TSC models tend to produce highly overconfident predictions under the label scarcity setting, often exceeding 0.85 confidence even when incorrect.

As shown in Fig. 3, the learned latent variables are visualized through t-SNE dimensionality reduction on the HAR and Epilepsy datasets to evaluate the classification performance

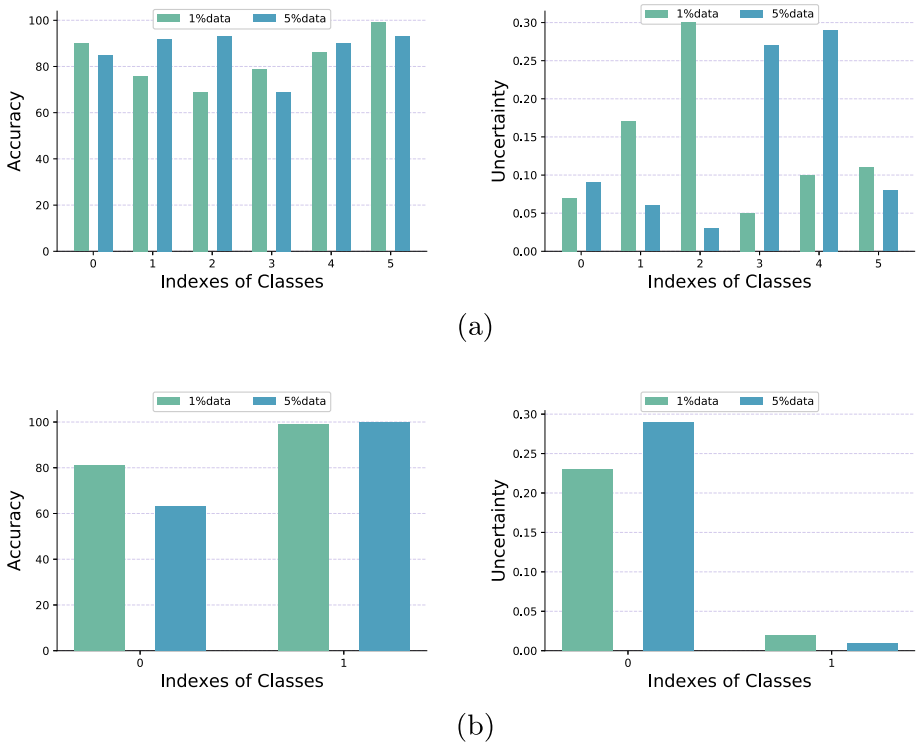


Fig. 2 Analysis of average class-wise uncertainty and accuracy

of SNPMPC. The latent representations produced by standard NP collapse into a narrow region with substantial overlap between classes, indicating impoverished representations under label-scarce conditions. In contrast, SNPMPC yields well-separated clusters, demonstrating richer and more informative latent structures. At 1% labeled data setting, the t-SNE projections reveal discernible separation between classes, with distinct cluster formation patterns emerging across the classification task.

For verifying the classification generalization of the proposed approach, time series classification experiments on three large-scale datasets (UCIHAR [47], MotionSense (MS) [48], WISDOM [49]) are conduct. Nine related works including FreRA [50], InfoMin [51], InfoTS [52], AutoTCL [53], TS2Vec [35], TNC [54], TS-TCC [6], TF-C [55], and SoftCLT [56], are selected as compared baselines. As shown in Table 3, SNPMPC consistently outperforms recent TSC baselines and obtains 0.976, 0.984, and 0.981 Accuracy on three datasets, respectively, demonstrating that our uncertainty-aware posterior modeling offers complementary benefits beyond masked modeling and frequency-based augmentation.

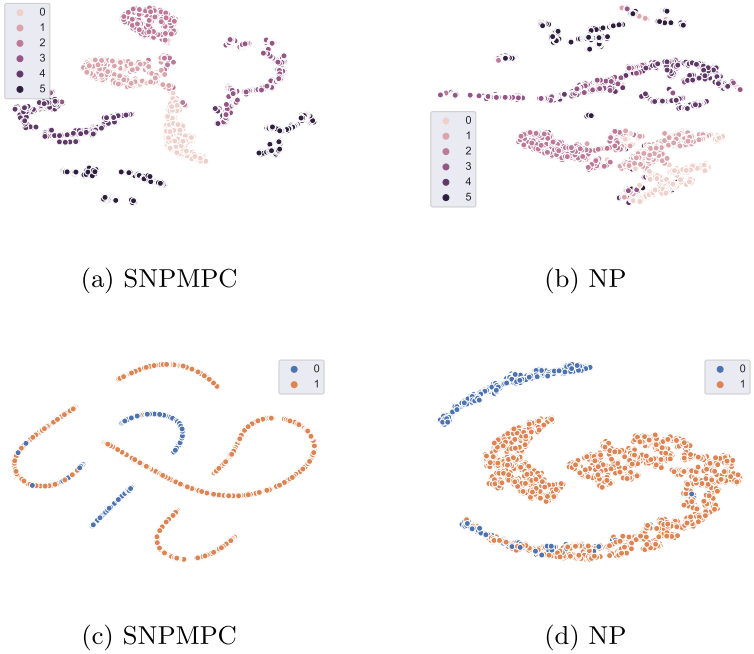
4.5 Ablation study

We now report our ablation studies on the HAR and Epilepsy datasets.

To assess the methodological necessity of the multi-view posterior consistency (MPC) component in SNPMPC, we conduct an ablation study comparing its performance against the conventional KL divergence loss prevalent in NPs and KL-based posterior consistency

Table 3 Performance comparison for time series classification on three large-scale datasets by accuracy

Datasets	SNPMPC	FreRA	InfoMin	InfoTS	AutoTCL	TS2Vec	TNC	TS-TCC	TF-C	SoftCLT
UCIHAR	0.976	0.975	0.967	0.967	0.697	0.959	0.568	0.924	0.875	0.961
MS	0.984	0.982	0.71	0.967	0.691	0.945	0.526	0.915	0.811	0.962
WISDOM	0.981	0.972	0.959	0.915	0.760	0.939	0.543	0.889	0.839	0.952



(a) SNPMPC (b) NP

(c) SNPMPC (d) NP

Fig. 3 Visualization of the learned latent variables through t-SNE. The subfigures **a** and **b** are the visualization on the HAR dataset. The subfigures **c** and **d** are the visualization on the Epilepsy dataset

loss (KPC). Quantitative evaluation on the HAR and Epilepsy datasets reveals statistically significant improvements when employing MPC versus KL divergence. On HAR, MPC achieves 86.20% accuracy and 86.38% MF1-score at 1% labeled data setting, while Epilepsy results show 95.78% accuracy and 93.10% MF1-score, as detailed in Fig. 4. The performance gap demonstrates that the multi-view posterior consistency loss enhances the classification performance through effective model prediction distribution learning.

The hyperparameter sensitivity analysis on the Human Activity Recognition (HAR) dataset (Fig. 6) systematically evaluates the impact of regularization coefficients λ_1 (cross-entropy loss weight) and λ_2 (multi-view posterior consistency loss weight) at 1% and 5% labeled data setting, where the loss weight varying from 0.001 to 100. As shown in Fig. 5, when fixing $\lambda_2 = 1$, λ_1=0.01 yields peak accuracy and MF1 scores, with performance decaying exponentially beyond $\lambda_1 > 10$. Conversely, when fixing $\lambda_2 = 1$, $\lambda_2 = 0.01$ achieves optimal results. This differential sensitivity between λ_1 and λ_2 suggests the consistency loss exhibits broader operational tolerance (Fig. 6).

5 Conclusion

STSC remains fundamentally challenged by epistemic uncertainty under extreme label scarcity, where conventional deterministic models often exhibit pathological confidence in spurious temporal correlations. This work addresses the critical need for probabilistic frameworks capable of quantifying uncertainty while leveraging unlabeled data in the STSC task. We propose SNPMPC, which consists of two key innovations. First, SNPMPC replaces

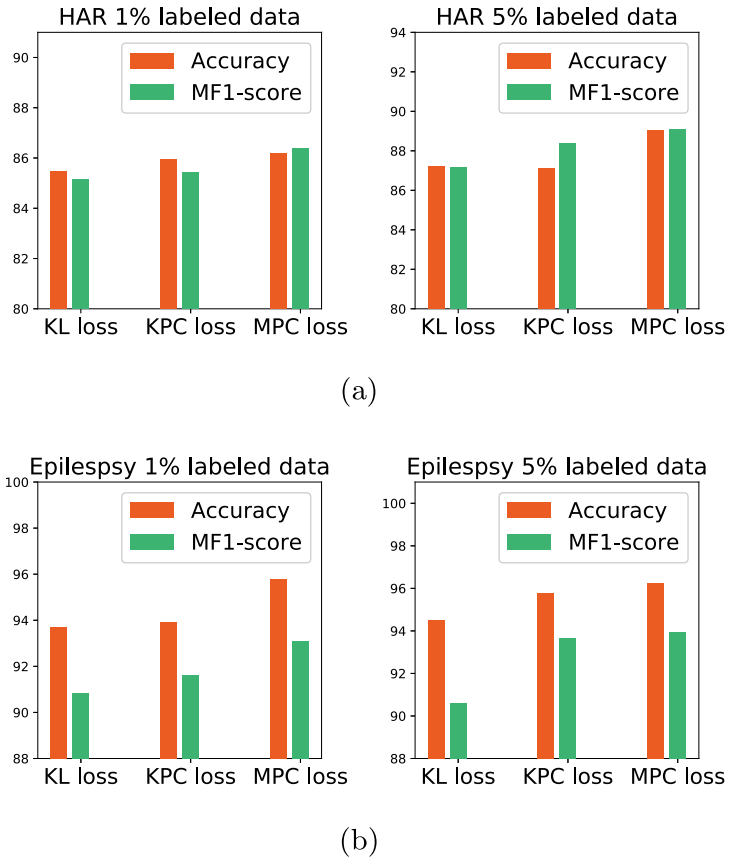


Fig. 4 Performance comparison with KL loss and cross-view posterior consistency loss on HAR and Epilepsy datasets at 1% and 5% labeled data setting

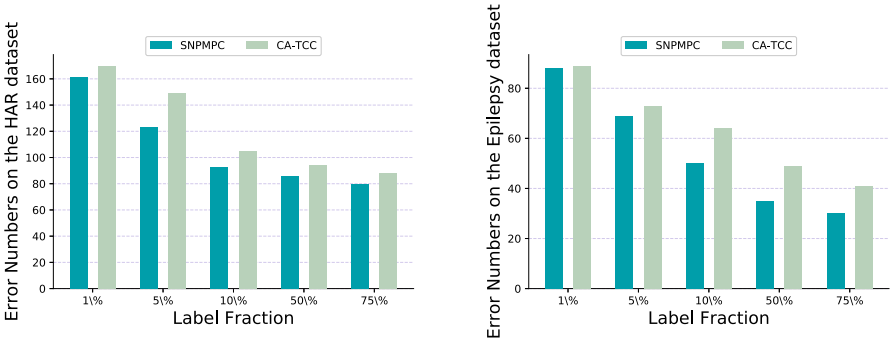
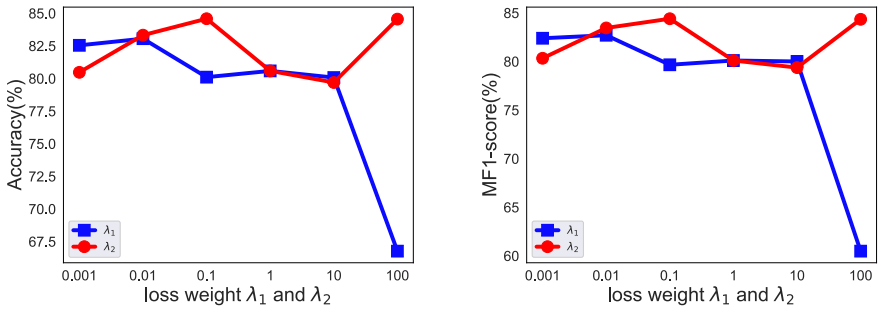
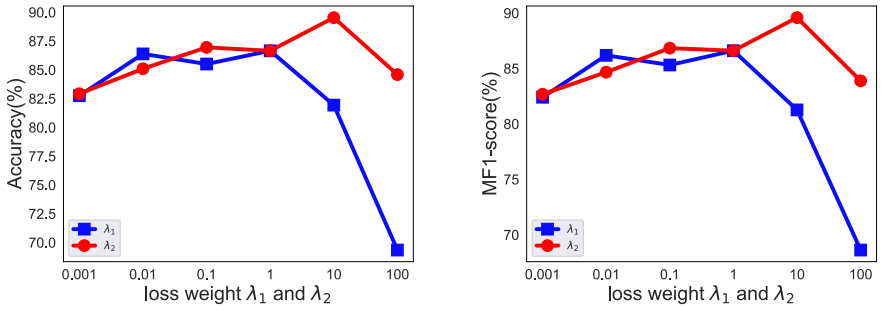


Fig. 5 Analysis of overconfidence incorrect classifications (the probability of overconfidence predictions is set to be greater than 0.85)



(a) 1% labeled data



(b) 5% labeled data

Fig. 6 Sensitivity analysis on HAR dataset at 1% and 5% labeled data setting

deterministic mappings with temporally conditioned Neural Processes that model latent variables through autoregressive formulations to capture evolving distributional shifts. This proposed neural process introduces uncertainty-aware learning by using entropy-based constraints, which ensures the pseudo-labels used during training do not introduce noise or bias. Then, a novel KL-based alignment is designed to enforce distributional invariance between weak/strong augmented views and the supervisory manifold, injecting richer signals to elevate latent variable quality. Empirical evaluations demonstrate that SNPMPC achieves consistent accuracy improvements and provides calibrated uncertainty estimates strongly correlated with prediction errors.

Author Contributions Xin Song and Zhikui Chen conceived this study, designed the computational algorithms, wrote the program code, and wrote the manuscript. Fangming Zhong proposed some valuable suggestions and guided the experiments.

Funding This work is supported in part by Science and Technology Planning Project of Liaoning Province (2023JH26/10100008), in part by the Fundamental Research Funds for the Central Universities (DUT22RC(3)011) and in part by the Applied Basic Research Programs of Liaoning Province 2023JH2/101300092.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Ma F, Wang C, Zeng Z (2023) SVM-based subspace optimization domain transfer method for unsupervised cross-domain time series classification. Knowl Inf Syst 65(2):869–897
2. Foumani NM, Tan CW, Webb GI, Salehi M (2024) Improving position encoding of transformers for multivariate time series classification. Data Min Knowl Disc 38(1):22–48
3. Wang Z, Yan W, Oates T (2017) Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 international joint conference on neural networks (IJCNN), pp 1578–1585
4. Jhin SY, Shin H, Kim S, Hong S, Jo M, Park S, Park N, Lee S, Maeng H, Jeon S (2024) Attentive neural controlled differential equations for time-series classification and forecasting. Knowl Inf Syst 66(3):1885–1915
5. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol 119, pp 1597–1607
6. Eldele E, Ragab M, Chen Z, Wu M, Kwoh C-K, Li X, Guan C (2023) Self-supervised contrastive representation learning for semi-supervised time-series classification. IEEE Trans Pattern Anal Mach Intell 45(12):15604–15618
7. Liu Z, Ma Q, Ma P, Wang L (2023) Temporal-frequency co-training for time series semi-supervised learning. Proc AAAI Conf Artif Intell 37:8923–8931
8. Liu P, Liu J (2025) When confidence fails: revisiting pseudo-label selection in semi-supervised semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 21874–21884
9. Garnelo M, Schwarz J, Rosenbaum D, Viola F, Rezende DJ, Eslami SMA, Teh YW (2018) Neural Processes. arXiv:1807.01622
10. Singh G, Yoon J, Son Y, Ahn S (2019) Sequential neural processes. In: Advances in neural information processing systems, vol 32
11. Wang J, Lukasiewicz T, Massiceti D, Hu X, Pavlovic V, Neophytou A (2022) NP-match: when neural processes meet semi-supervised learning. In: Proceedings of the 39th international conference on machine learning. Proceedings of machine learning research, vol 162, pp 22919–22934
12. Zheng Y, Si Y-W, Wong R (2021) Feature extraction for chart pattern classification in financial time series. Knowl Inf Syst 63(7):1807–1848
13. Shu W, Yao Y, Lyu S, Li J, Chen H (2021) Short isometric shapelet transform for binary time series classification. Knowl Inf Syst 63(8):2023–2051
14. Raza A, Kramer S (2020) Accelerating pattern-based time series classification: a linear time and space string mining approach. Knowl Inf Syst 62(3):1113–1141
15. Bostrom A, Bagnall A (2015) Binary shapelet transform for multiclass time series classification. In: Big data analytics and knowledge discovery, Cham, pp 257–269
16. Cheng Z, Yang Y, Jiang S, Hu W, Ying Z, Chai Z, Wang C (2023) Time2Graph+: bridging time series and graph representation learning via multiple attentions. IEEE Trans Knowl Data Eng 35(2):2078–2090
17. Liu Z, Pei W, Lan D, Ma Q (2024) Diffusion language-shapelets for semi-supervised time-series classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 38, pp 14079–14087
18. Lin J, Keogh E, Wei L, Lonardi S (2007) Experiencing SAX: a novel symbolic representation of time series. Data Min Knowl Disc 15(2):107–144
19. Senin P, Malinchik S (2013) SAX-VSM: interpretable time series classification using SAX and vector space model. In: 2013 IEEE 13th international conference on data mining, pp 1175–1180
20. Hyvarinen A, Morioka H (2016) Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In: Advances in neural information processing systems, vol 29
21. Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL (2014) Time series classification using multi-channels deep convolutional neural networks. In: Web-age information management, Cham, pp 298–310
22. Liu C-L, Hsaio W-H, Tu Y-C (2019) Time series classification with multivariate convolutional neural network. IEEE Trans Industr Electron 66(6):4788–4797

23. Ismail Fawaz H, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller P-A, Petitjean F (2020) InceptionTime: finding AlexNet for time series classification. Data Min Knowl Disc 34(6):1936–1962
24. Tang Y, Xu J, Matsumoto K, Ono C (2016) Sequence-to-sequence model with attention for time series classification. In: 2016 IEEE 16th international conference on data mining workshops (ICDMW), pp 503–510
25. Malhotra P, TV V, Vig L, Agarwal P, Shroff G (2017) TimeNet: pre-trained deep recurrent neural network for time series classification. arXiv preprint arXiv:1706.08838
26. Chen W, Shi K (2021) Multi-scale attention convolutional neural network for time series classification. Neural Netw 136:126–140
27. Zhao B, Xing H, Wang X, Song F, Xiao Z (2023) Rethinking attention mechanism in time series classification. Inf Sci 627:97–114
28. Lee DH, *et al.* (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML, vol 3, pp 896
29. Kamnitsas K, Castro D, Folgoc LL, Walker I, Tanno R, Rueckert D, Glocker B, Criminisi A, Nori A (2018) Semi-supervised learning via compact latent space clustering. In: Proceedings of the 35th international conference on machine learning. Proceedings of machine learning research, vol 80, pp 2459–2468
30. Iscen A, Tolias G, Avrithis Y, Chum O (2019) Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
31. Laine S, Aila T (2016) Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242
32. Miyato T, Maeda S-I, Koyama M, Ishii S (2019) Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans Pattern Anal Mach Intell 41(8):1979–1993
33. Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel CA (2019) MixMatch: a holistic approach to semi-supervised learning. In: Advances in neural information processing systems, vol 32
34. Jawed S, Grabocka J, Schmidt-Thieme L (2020) Self-supervised learning for semi-supervised time series classification. In: Advances in knowledge discovery and data mining, Cham, pp 499–511
35. Yue Z, Wang Y, Duan J, Yang T, Huang C, Tong Y, Xu B (2022) Ts2vec: towards universal representation of time series. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, pp 8980–8987
36. Arunan A, Qin Y, Li X, Yuen C (2025) Intelligently augmented contrastive tensor factorization empowering multi-dimensional time series classification in low-data environments. Exp Syst Appl 287:127889
37. Øksendal B (2003) Stochastic differential equations. In: Stochastic differential equations: an introduction with applications, Berlin, Heidelberg, pp 65–84
38. Zhang X, Tan Z, Lu F, Yan R, Liu J (2024) Adaptive semi-supervised learning from stronger augmentation transformations of discrete text information. Knowl Inf Syst 66(8):4609–4629
39. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: The European symposium on artificial neural networks
40. Andrzejak R, Lehnertz K, Mormann F, Rieke C, David P, Elger C (2002) Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. Phys Rev E: Stat, Nonlin, Soft Matter Phys 64:061907
41. Sarkar P, Etemad A (2022) Self-supervised ECG representation learning for emotion recognition. IEEE Trans Affect Comput 13(3):1541–1554
42. Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. arXiv: 1807.03748
43. Tarvainen A, Valpola H (2017) Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in neural information processing systems, vol 30
44. : Li J, Socher R, Hoi SCH (2020) DivideMix: learning with noisy labels as semi-supervised learning. ArXiv : abs/2002.07394
45. Fan H, Zhang F, Wang R, Huang X, Li Z (2021) Semi-supervised time series classification by temporal relation prediction. In: ICASSP 2021 - 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 3545–3549
46. Sohn K, Berthelot D, Carlini N, Zhang Z, Zhang H, Raffel CA, Cubuk ED, Kurakin A, Li CL (2020) FixMatch: simplifying semi-supervised learning with consistency and confidence. In: Advances in neural information processing systems, vol 33, pp 596–608
47. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: Ambient assisted living and home care, Berlin, Heidelberg, pp 216–223

48. Malekzadeh M, Clegg RG, Cavallaro A, Haddadi H (2019) Mobile sensor data anonymization. In: Proceedings of the international conference on internet of things design and implementation, New York, USA, pp 49–58
49. Kwapisz J, Weiss G, Moore S (2010) Activity recognition using cell phone accelerometers. SIGKDD Explor 12:74–82
50. Tian T, Miao C, Qian H (2025) Frera: a frequency-refined augmentation for contrastive learning on time series classification. In: Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining V.2, New York, USA, pp 2835–2846
51. Tian Y, Sun C, Poole B, Krishnan D, Schmid C, Isola P (2020) What makes for good views for contrastive learning. Adv Neural Inf Process Syst 33:6827–6839
52. Luo D, Cheng W, Wang Y, Xu D, Ni J, Yu W, Zhang X, Liu Y, Chen Y, Chen H, Zhang X (2023) Time series contrastive learning with information-aware augmentations. Proc AAAI Conf Artif Intell 37:4534–4542
53. Zheng X, Wang T, Cheng W, Ma A, Chen H, Sha M, Luo D (2024) Parametric augmentation for time series contrastive learning. In: The twelfth international conference on learning representation, Vienna, Austria
54. Tonekaboni S, Eytan D, Goldenberg A (2021) Unsupervised representation learning for time series with temporal neighborhood coding. In: 9th international conference on learning representations
55. Zhang X, Zhao Z, Tsiligkaridis T, Zitnik M (2022) Self-supervised contrastive pre-training for time series via time-frequency consistency. In: Advances in neural information processing systems
56. Lee S, Park T, Lee K (2024) Soft contrastive learning for time series. In: The Twelfth international conference on learning representations, Vienna, Austria

Xin Song received the B.S. degree in Software Engineering from Dalian University of Technology, China, in 2020. He is a Ph.D. candidate in the School of Software Technology, Dalian University of Technology. Her research interests include computer vision and machine learning.

Zhikui Chen received his Ph.D. degree in Digital Signal Processing and M.S. degree in Mechanics from Chongqing University, China, in 1998 and 1993, respectively. He obtained his B.S. degree in the Department of Mathematics and Computer Science from Chongqing Normal University, China. He is working as a full professor at Dalian University of Technology, China. His research interests include big data processing, mobile cloud computing, ubiquitous network and its computing.

Fangming Zhong received the B.S. degree, the M.S. degree and the PhD degree in software engineering from Dalian University of Technology, in 2012, 2014 and 2018, respectively. He is an associate professor in the School of Software Engineering at Dalian University of Technology, China. His research interests include multimodal learning, cross-modal retrieval, cross-modal hashing and zero-shot learning.