TVSRNet: A Unified Framework for Multivariate Time-Series Classification via Triple-View Sparsified Routing

Yunhe Guo[1], Zhengyang Zhou[1(✉)], Kedong Yan[2], and Yang Wang[1]

[1] University of Science and Technology of China, Hefei, China
guoyunhe@mail.ustc.edu.cn, {zzy0929,angyan}@ustc.edu.cn
[2] Suzhou Aerospace Information Research Institute, Suzhou, China
yankd@aircas.ac.cn

Abstract. Multivariate time-series classification exhibits diverse temporal mechanisms and strong cross-dataset heterogeneity, making it challenging to build a single model that generalizes across domains. Despite strong representational capacity, modern deep models often fail to consistently localize discriminative cues when datasets differ in variable relevance, spectral signatures, and interaction patterns across temporal stages. We present **TVSRNet** (**T**riple-**V**iew **S**parsified **R**outing **Net**work), a unified framework that adaptively selects discriminative information from three complementary views: channel, spectral, and interaction. TVSRNet injects a triple-view sparsifying residual branch into each backbone layer. The Channel view performs sample-wise budgeted channel allocation to emphasize class-relevant representations and suppress redundancy. The Spectral view leverages Fourier analysis to learn soft band masks with selective gains, enhancing informative frequencies while attenuating broadband noise. The Interaction view tokenizes sequences into patches and uses sparse Mixture-of-Experts routing to activate only a few experts, capturing stage-dependent heterogeneous dynamics. A bridge gate regulates residual injection for stable optimization and enables layer-wise, data-adaptive view balancing. Extensive experiments on **28** datasets show that TVSRNet achieves the best overall average rank of **2.12**, outperforming the strongest baseline by **1.72** rank points, and delivers the potential transfer capacity in cross-dataset few-shot transfer with an accuracy of **68.07**%, improving over the best baseline by **4.50**%. The learned channel selection, spectral emphasis, and interaction routing patterns are also highly interpretable.

Keywords: Multivariate time series · Sparse modeling · Mixture-of-experts

1 Introduction

Time-series classification aims to map an input sequence to a class label, and is a fundamental problem in many domains, including transportation, human

M. Ceci et al. (Eds.): ECML PKDD 2026, LNAI 16947, pp. 573–589, 2027.
https://doi.org/10.1007/978-3-032-37676-3_33

activity recognition and healthcare diagnosis [1,10,14]. Although these tasks share the same objective, existing methods are often developed in a domain-specific or task-specific manner, requiring repeated effort in architecture design, inductive-bias selection, and dataset-specific adaptation. This paradigm incurs considerable human and computational cost, and hinders the development of a general-purpose time-series classifier.

This raises an important question: *Can we build a unified time-series classification model that adapts to various domains with a common architecture?* Such a learner is attractive not only for efficiency, but also for transferability. If heterogeneous time-series tasks can be embedded into a shared learning path, common temporal regularities across domains may be absorbed into shared parameters, potentially improving robustness under limited supervision and benefiting low-resource generalization.

A key observation is that different datasets exhibit structural patterns through three common perspectives. Some rely more on informative channels, as in human activity recognition [1]. Some are better characterized by spectral patterns, as in ECG and speech-related tasks [7,14], and others are driven by time-varying inter-variable dependencies, as in traffic data [10]. Despite these differences, such cues can be consistently organized into a shared triple-view structure comprising channel, spectral, and interaction views. This triple-view structure provides a natural interface for unified modeling. However, the contribution of each view to the final classification varies across datasets, samples, and temporal stages, since different settings may rely on different views to different extents. As a result, directly combining all views may introduce redundancy and representation entanglement. This observation motivates a more concrete analysis of view-specific discriminative strength. We therefore quantify how strongly each view exposes class differences using a normalized separability score that compares between-class and within-class variability. View-wise scores are computed from simple statistics defined in the triple-view space, using channel energy/RMS for the Channel view, Welch band power for the Spectral view, and vectorized cross-channel Pearson correlations for the Interaction view. As shown in Fig. 1, datasets from different domains can all be described through the same triple-view structure, yet their dominant views differ substantially. These findings suggest that a unified classifier should follow a shared triple-view learning path while adaptively selecting the most informative evidence, rather than uniformly mixing all views.

This observation suggests that *sparsity is the key to unified modeling*. Rather than uniformly mixing all components, a unified model should selectively emphasize the most informative channels, frequency bands, or interaction pathways for each input, while retaining a shared triple-view learning framework.

Based on this insight, we propose **TVSRNet** (**T**riple-**V**iew **S**parsified **R**outing **Net**work), a unified framework for multivariate time-series classification. TVSRNet represents time series through three complementary views, namely **Channel**, **Spectral**, and **Interaction**. The Channel view performs sample-wise budgeted channel allocation to emphasize informative representa-

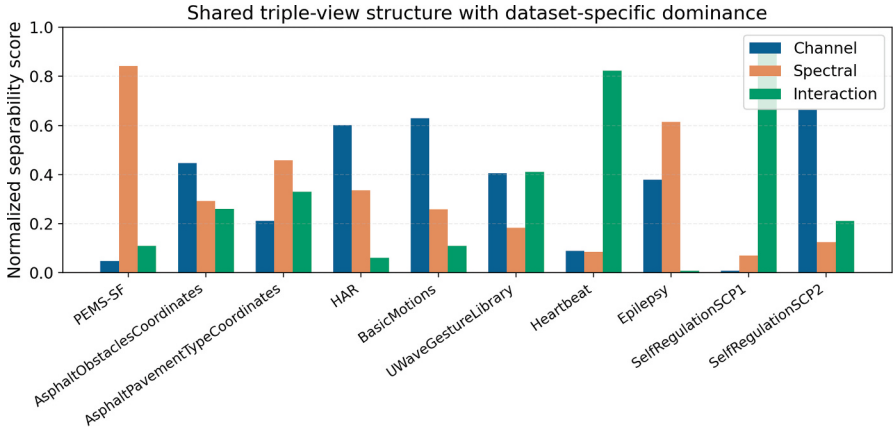


Fig. 1. Shared triple-view structure with dataset-specific dominance. We quantify view-wise class separability in the channel, spectral, and interaction views using channel energy/RMS, Welch band power, and cross-channel Pearson correlations. Normalized results show that all datasets share the same triple-view structure, but differ in their dominant view.

tions, the Spectral view learns soft band masks and gains to enhance discriminative frequency components, and the Interaction view adopts patch-level sparse Mixture-of-Experts routing to capture stage-dependent heterogeneous dynamics [6,9,16]. To reduce redundancy and promote efficient learning, these views are injected into each backbone layer through a sparsified residual branch. A bridging gate further controls the residual injection strength, enabling stable optimization and adaptive balancing across views.

Contributions. This paper makes the following contributions: (1) We study multivariate time-series classification from the perspective of unified learning, and show that patterns of datasets from various domains can be described through a shared triple-view structure: channel, spectral, and interaction. (2) We propose **TVSRNet**, a triple-view sparsified routing framework that unifies channel sparsification, spectral sparsification, and interaction sparsification within a residual-injection architecture. (3) Extensive experiments on **28** diverse datasets show that TVSRNet achieves the best overall average rank of **2.12**, surpassing the strongest baseline by **1.72** rank points, and delivers the potential transfer capacity in cross-dataset few-shot transfer with an accuracy of **68.07**%, improving over the best baseline by **4.50**%. TVSRNet also yields interpretable patterns in channel selection, spectral emphasis, and routing behavior.

2 Related Work

Multivariate Time Series Classification. Multivariate time-series classification (MTSC) remains challenging because discriminative information may arise from

channel saliency, temporal patterns, and cross-variable dependencies in different ways across datasets. The UEA archive has become a standard benchmark for MTSC evaluation [2], and large-scale comparisons have shown that no single method consistently dominates across datasets [15]. Representative approaches include deep models such as **InceptionTime**, which uses multi-scale temporal convolutions [5], and recent Transformer-style architectures such as **PatchTST** [13], **TimesNet** [21], and **Crossformer** [23], as well as strong traditional methods such as **ROCKET** [3] and **MiniROCKET** [4]. While effective, these methods typically emphasize one main inductive bias, such as multi-scale temporal modeling, patch tokenization, cross-dimension interaction, rather than jointly adapting channel relevance, spectral selectivity, and stage-wise interaction heterogeneity within a unified MTSC framework.

Sparse Deep Learning for Time Series. Sparsity is an important inductive bias for time-series learning because useful evidence is often concentrated in a subset of tokens or frequency components. Existing studies exploit this property from different perspectives, such as sparse attention in **Informer** [24] and frequency-domain modeling in **FEDformer** [25] and **FNet** [8]. While effective, these methods usually impose sparsity or compactness in only one representation space. In contrast, our method treats sparsity as a unified principle across channel, spectral, and interaction views.

Mixture-of-Experts in Time Series. Mixture-of-Experts (MoE) provides conditional computation by activating only a small subset of experts for each input. This idea is developed in representative models such as **Sparsely-Gated MoE** [16], **GShard** [9], and **Switch Transformer** [6], and has recently been introduced into time-series modeling by methods such as **MoLE** [12] and **Time-MoE** [17]. However, existing time-series MoE studies mainly focus on forecasting or model scaling. In contrast, our method uses sparse patch-level routing for MTSC and further integrates interaction routing with channel and spectral sparsification within a unified framework.

3 Method

3.1 Problem Formulation

We consider multivariate time-series classification on datasets from different domains. For a dataset $\mathcal{D} = \{(X^{(n)}, y^{(n)})\}_{n=1}^{|\mathcal{D}|}$, each sample $X^{(n)} \in \mathbb{R}^{T \times C}$ is a multivariate sequence with length T and channel dimensionality C, and $y^{(n)} \in \{1, \ldots, K\}$ is the corresponding class label. In practice, datasets from different domains may differ substantially in input dimensionality, temporal patterns, and variable semantics. Each input is therefore first projected into a latent space through an embedding layer, giving $H_0 = \mathrm{Embed}(X) \in \mathbb{R}^{T' \times d}$ with a shared hidden dimension d. Based on this representation, the task is to learn a mapping f_θ from the embedded sequence to a discriminative latent representation, on top of which a classifier predicts the dataset-specific class label. The

resulting representation is expected to capture transferable temporal structure while remaining sufficiently discriminative for classification under heterogeneous domains.

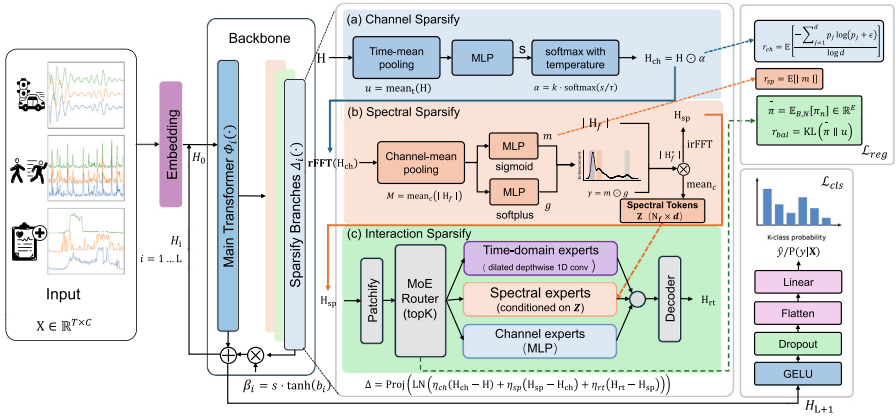


Fig. 2. Overall framework of TVSRNet. The model stacks a backbone encoder and, at each layer, injects a parallel sparse residual pathway through a bridging gate to progressively refine representations. The residual pathway consists of three complementary views: **(a)** *Channel sparsification*, which performs sample-adaptive budgeted channel reweighting to suppress redundancy; **(b)** *Spectral sparsification*, which applies differentiable band selection and gain shaping to denoise signals and enhance discriminative components, while producing spectral tokens as frequency context; **(c)** *Interaction sparsification*, which uses patch-wise top-k MoE routing over heterogeneous experts to capture stage-dependent interactions.

3.2 Overview

Multivariate time series present three recurring challenges: variable redundancy makes informative evidence unevenly distributed across channels, useful temporal structure is often concentrated in a few frequency bands, and different temporal stages may follow different interaction patterns. To address these issues within a shared architecture, we introduce a tri-view sparse residual branch that models *channel sparsity*, *spectral sparsity*, and *interaction sparsity*, and injects the resulting sparse increment into the backbone. As illustrated in Fig. 2, the model consists of a backbone transformation and a parallel sparse branch, where the latter provides a structured refinement signal through a bridging gate.

Specifically, the model stacks L hierarchical blocks. At layer i, given the hidden representation H_{i-1}, we compute a main transformation $\Phi_i(\cdot)$ together with a sparse residual branch $\Delta_i(\cdot)$, and fuse them as

$$H_i = \Phi_i(H_{i-1}) + \beta_i \cdot \Delta_i(H_{i-1}), \qquad i = 1, \ldots, L,$$

where the bridging gate

$$\beta_i = s \cdot \tanh(b_i)$$

controls the magnitude of residual injection, with b_i being a learnable scalar and s a fixed scaling constant. In this way, the sparse branch serves as a controllable refinement signal, improving optimization stability.

Concretely, given a layer input $H \in \mathbb{R}^{B\times T\times d}$, the sparse branch sequentially constructs channel-, spectral-, and interaction-aware representations, denoted by H_{ch}, H_{sp}, and H_{rt}. These view-specific refinements are then combined into a sparse increment

$$\Delta = \mathrm{Proj}\Big(\mathrm{LN}\big(\eta_{\mathrm{ch}}(H_{\mathrm{ch}} - H) + \eta_{\mathrm{sp}}(H_{\mathrm{sp}} - H_{\mathrm{ch}}) + \eta_{\mathrm{rt}}(H_{\mathrm{rt}} - H_{\mathrm{sp}})\big)\Big),$$

where η_{ch}, η_{sp}, and η_{rt} are learnable scalar gates that balance the contributions of the three views. After the final layer, the resulting representation is passed to a lightweight classification head to produce the class prediction. The detailed constructions of the three sparse components are introduced next.

Main Transformer Backbone. The *Main Transformer* in TVSRNet denotes the dense encoder path used as the backbone transformation $\Phi_i(\cdot)$. Given $H_{i-1} \in \mathbb{R}^{B\times T'\times d}$, each backbone layer follows a standard Transformer encoder block composed of temporal multi-head self-attention and a position-wise feed-forward network:

$$\bar{H}_i = \mathrm{LN}(H_{i-1} + \mathrm{MHSA}(H_{i-1})), \qquad \Phi_i(H_{i-1}) = \mathrm{LN}\big(\bar{H}_i + \mathrm{FFN}(\bar{H}_i)\big).$$

Here, $\mathrm{MHSA}(\cdot)$ models temporal dependencies among tokens, and $\mathrm{FFN}(\cdot)$ performs channel-wise nonlinear transformation.

3.3 Channel Sparsification via Sample-Wise Budget Allocation

In multivariate time series, discriminative evidence is often unevenly distributed across variables, and this heterogeneity naturally carries over to the latent representation. We therefore allocate a limited importance budget over latent channels so that the model emphasizes informative components while suppressing redundant ones.

Given a layer representation $H \in \mathbb{R}^{B\times T\times d}$, we first average it over time to obtain a channel summary:

$$u = \frac{1}{T}\sum_{t=1}^{T} H_{:,t,:} \in \mathbb{R}^{B\times d}.$$

A two-layer MLP then produces channel scores $s \in \mathbb{R}^{B\times d}$, and budget weights are obtained through a temperature-τ_{ch} softmax:

$$\alpha = k_{\mathrm{ch}} \cdot \mathrm{softmax}\left(\frac{s}{\tau_{\mathrm{ch}}}\right), \qquad \sum_{j=1}^{d} \alpha_j = k_{\mathrm{ch}},$$

where k_{ch} is the effective channel budget. The channel-refined representation is

$$H_{\mathrm{ch}} = H \odot \alpha.$$

This sample-wise budget allocation encourages the model to concentrate its capacity on a small subset of latent channels that are most useful for classification.

As a channel sparsity statistic, we compute the normalized entropy of the normalized weights $p = \alpha / \sum_j \alpha_j$:

$$r_{\mathrm{ch}} = \mathbb{E}\left[\frac{-\sum_{j=1}^{d} p_j \log(p_j + \epsilon)}{\log d}\right].$$

A smaller value of r_{ch} indicates a more concentrated channel allocation and therefore stronger channel sparsity.

3.4 Spectral Sparsification via Soft Band Masking and Gain Modulation

Discriminative temporal patterns are often concentrated in a limited set of frequency bands, while noise tends to be more broadband. Motivated by this observation, we perform adaptive band selection and modulation in the frequency domain, and then transform the refined representation back to the time domain.

We first apply a real FFT to the channel-refined representation:

$$H_f = \mathrm{rFFT}(H_{\mathrm{ch}}) \in \mathbb{C}^{B \times F \times d}, \qquad F = \left\lfloor \frac{T}{2} \right\rfloor + 1.$$

We then compute the mean spectral magnitude over channels as a compact frequency descriptor:

$$M = \mathrm{mean}_c(|H_f|) \in \mathbb{R}^{B \times F}.$$

Based on M, two lightweight networks produce a soft mask and a nonnegative gain:

$$m = \sigma(\mathrm{MLP}_{\mathrm{mask}}(M)) \in (0,1)^{B \times F}, \qquad v = \mathrm{softplus}(\mathrm{MLP}_{\mathrm{gain}}(M)) \in \mathbb{R}_{+}^{B \times F}.$$

The mask determines which frequency regions to emphasize, while the gain controls the amplification strength. We combine them as

$$\gamma = m \odot v,$$

share γ across channels, and modulate the spectrum:

$$H_f^* = H_f \odot \gamma[:, :, None], \qquad H_{\mathrm{sp}} = \mathrm{irFFT}(H_f^*) \in \mathbb{R}^{B \times T \times d}.$$

This factorization separates band selection from band enhancement, enabling selective spectral refinement with a lightweight parameterization.

The spectral sparsity statistic is

$$r_{\mathrm{sp}} = \mathbb{E}[|m|].$$

Minimizing this term encourages the mask to remain sparse so that the model focuses on a compact subset of informative frequency regions.

Spectral Tokens. The spectral branch also provides explicit frequency context for interaction routing. We compute the enhanced spectral magnitude

$$M^* = \mathrm{mean}_c(|H_f^*|) \in \mathbb{R}^{B\times F},$$

uniformly partition the frequency axis into N_f bands, average within each band, and project the result to the latent space:

$$Z \in \mathbb{R}^{B\times N_f\times d}.$$

These spectral tokens summarize coarse-grained frequency structure for subsequent routing.

3.5 Interaction Sparsity with Patch-Level MoE Routing

Different temporal stages may follow different mechanisms, so a single interaction pattern is often insufficient. We therefore partition the sequence into patches and activate only a small subset of experts for each patch, yielding sparse interaction structures and conditional computation.

Patch Tokenization. We split H_{sp} into non-overlapping patches of length P, with the number of patches

$$N = \left\lceil \frac{T}{P} \right\rceil.$$

Each patch is projected along the length dimension to obtain patch tokens

$$U \in \mathbb{R}^{B\times N\times d}.$$

Dynamic Mixture-of-Experts with Sparse Routing. Let the total number of experts be E, partitioned into temporal, frequency, and channel experts. Given the patch representation $U \in \mathbb{R}^{B\times N\times d}$, temporal experts use depthwise separable 1D convolutions over the patch dimension to model multi-scale temporal dynamics, frequency experts extract a global descriptor from the spectral tokens Z and broadcast it back to all patches, and channel experts apply position-wise two-layer MLPs to perform nonlinear channel mixing at each patch. Stacking all expert outputs yields an expert bank

$$\mathcal{E}(U, Z) \in \mathbb{R}^{B\times N\times E\times d}.$$

For each patch token u_n, the router computes

$$\ell_n = \frac{W_r\,\mathrm{LN}(u_n)}{\tau_r} \in \mathbb{R}^E, \qquad \pi_n = \mathrm{softmax}(\ell_n).$$

Only the top-k_r experts are retained and renormalized to obtain sparse routing weights $\widetilde{\pi}_n$, and the routed patch representation is

$$\widetilde{u}_n = \sum_{e=1}^{E} \widetilde{\pi}_{n,e}\,\mathcal{E}_{n,e} \in \mathbb{R}^d.$$

This produces $\widetilde{U} \in \mathbb{R}^{B\times N\times d}$, which is decoded back to the time domain as

$$H_{\mathrm{rt}} = \mathrm{Dec}(\widetilde{U}) \in \mathbb{R}^{B\times T\times d}.$$

Because routing is conditioned on patch representations, the model can adaptively select different interaction patterns across temporal stages.

Routing Balance Regularization. To avoid routing collapse to only a few experts, we compute the average routing distribution

$$\pi = \mathbb{E}_{B,N}[\pi_n] \in \mathbb{R}^{E},$$

and regularize it toward the uniform distribution:

$$r_{\mathrm{bal}} = \mathrm{KL}(\pi \| u).$$

This term encourages more balanced expert utilization and stabilizes sparse MoE training.

3.6 Training Objective: Classification Loss with Sparsity Regularizers

The training objective combines a classification term with the sparsity regularizers defined above. The classification loss is

$$\mathcal{L}_{\mathrm{cls}} = -\log \hat{p}(y \mid X).$$

Since the sparse branch influences the backbone through the layer-wise bridging gates, we weight each regularizer by the magnitude of the corresponding gate $|\beta_i|$ and average over layers:

$$\mathcal{L}_{\mathrm{reg}} = \lambda_{\mathrm{ch}}\mathbb{E}_i\left[|\beta_i|\, r_{\mathrm{ch}}^{(i)}\right] + \lambda_{\mathrm{sp}}\mathbb{E}_i\left[|\beta_i|\, r_{\mathrm{sp}}^{(i)}\right] + \lambda_{\mathrm{bal}}\mathbb{E}_i\left[|\beta_i|\, r_{\mathrm{bal}}^{(i)}\right].$$

The overall objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{reg}}.$$

4 Experiments

We evaluate **TVSRNet** on multivariate time-series classification benchmarks from multiple domains. We first report the main classification results, and then provide ablation and further analyses to better understand the proposed model. Code is available at: https://github.com/YunheGuo/TVSRNet/.

4.1 Experimental Setup

Datasets. Our benchmark contains **28 datasets**, including the **26 equal-length datasets** commonly used from the **UEA Multivariate Time Series Classification Archive** [2,15], together with two additional datasets, namely **PEMS-BAY** and **HAR** [1]. **PEMS-BAY** [10] is a widely used real-world traffic benchmark derived from the California Performance Measurement System (PeMS). We convert it into a multivariate time-series classification dataset in a PEMS-SF-style manner, where each day is treated as one sample, the multivariate traffic measurements over that day are used as the input sequence, and the day of the week is used as the class label. Since the original data are recorded at 5-minute intervals, each daily sample contains 288 time steps. For presentation, all datasets are grouped into four domains, namely **Traffic & Infrastructure** (2 datasets), **Human Activity & Motion** (10 datasets), **Physiological & Neuro Signals** (9 datasets), and **Audio, Speech & Other Sensing** (7 datasets). Detailed dataset names and results are reported in Table 1. We follow the official train/test split of each dataset.

Baselines and Evaluation Protocol. TVSRNet is compared with representative baselines from three categories, namely general-purpose sequence models, time-series-specific deep models, and non-deep-learning methods. The first category includes **Transformer** [19] and **iTransformer** [11]. The second category includes **Informer** [24], **Crossformer** [23], **TimesNet** [21], **LightTS** [22], and **PatchTST** [13]. The third category includes **Mini'ROCKET** [4], a strong non-deep-learning baseline based on random convolutional kernels. To ensure a fair comparison, the implementations of all deep learning baselines are adapted from the **Time-Series-Library** (**TSLib**) [18,20], an open-source benchmark covering multiple time-series analysis tasks, including classification. All methods are evaluated under the same train/test split, and **classification accuracy** (%) is used as the evaluation metric.

Implementation Details. All experiments are conducted on a single NVIDIA Tesla V100 GPU with 32 GB of memory. We use Adam for optimization, with the batch size set to 16 and the initial learning rate set to 0.001. The maximum number of training epochs is 60, and the patience for early stopping is set to 10. To accommodate heterogeneous datasets from different domains within a unified framework, each input is first projected into a shared latent space through an embedding layer. For TVSRNet, the patch length is set to $P = 4$, the channel budget is set to $k_{\text{ch}} = 32$, the number of spectral bands is set to $N_f = 16$, and the routing top-k is set to $k_r = 2$. The sparsity regularization weights are set to $\lambda_{\text{ch}} = 2 \times 10^{-5}$, $\lambda_{\text{sp}} = 2 \times 10^{-5}$, and $\lambda_{\text{bal}} = 5 \times 10^{-4}$. In the interaction branch, the numbers of temporal, frequency, and channel experts are set to 4, 2, and 2, respectively. All experiments are repeated three times, and the average results are reported.

4.2 Main Results

Table 1 reports the classification accuracy (%) on all benchmarks. Overall, TVSRNet achieves the best performance with an overall average rank of **2.12**, outperforming the strongest baseline **iTransformer** (**3.84**) by **1.72** rank points. TVSRNet also achieves the best domain-wise average rank in all four domains, namely **1.00** on Traffic & Infrastructure, **2.90** on Human Activity & Motion, **1.72** on Physiological & Neuro Signals, and **1.86** on Audio, Speech & Other Sensing.

The advantage of TVSRNet is particularly clear on datasets with strong multivariate redundancy or sparse discriminative structure. On *PEMS-BAY*, TVSRNet reaches **94.44**%, exceeding the second-best result of **91.66**% by **2.78** points. On *PEMS-SF*, it achieves **86.12**%, surpassing the second-best **85.54**% by **0.58** points. On *HandMovementDirection* it reaches **67.56**%, exceeding the second-best **63.84**% by **3.72** points. On speech-related datasets, TVSRNet achieves **98.11**% on *JapaneseVowels* and **98.54**% on *SpokenArabicDigits*, outperforming the second-best methods by **0.15** and **0.36** points, respectively.

The relative performance of the baselines is also in line with their architectural preferences, since different models are designed to emphasize different temporal patterns. Transformer and iTransformer provide generic sequence modeling, but do not explicitly capture sparse channel relevance, selective spectral evidence, and stage-dependent interaction heterogeneity within a unified framework. Time-series-specific models such as Informer, PatchTST, and TimesNet remain competitive on individual datasets, and MiniROCKET is especially strong on a few benchmarks such as *LSST* and *PhonemeSpectra*. Nevertheless, TVSRNet is more consistent overall, suggesting that triple-view sparsification provides a more suitable inductive bias for multivariate time-series classification across heterogeneous domains.

4.3 Ablation Study

We conduct ablation experiments on *PEMS-SF*, *JapaneseVowels*, and *HandMovementDirection* to evaluate the effects of tri-view modeling and sparsity design in TVSRNet. Specifically, we consider three single-view variants (*Only Channel*, *Only Spectral*, and *Only Interaction*) and three sparsity-removed variants (*w/o Channel sparsity*, *w/o Spectral sparsity*, and *w/o Interaction sparsity*).

As shown in Table 2, the full TVSRNet achieves the best results on all three datasets, with accuracies of **86.12%**, **98.11%**, and **67.56%**, respectively. In contrast, the single-view variants show clear performance drops. Their average accuracies decrease to **79.81%** (*Only Channel*), **79.59%** (*Only Spectral*), and **81.10%** (*Only Interaction*), indicating that no individual view is sufficient to replace the complete tri-view design. Among them, *Only Interaction* performs slightly better than the other two variants.

In addition, removing the sparsity mechanism leads to smaller but consistent degradations. The average accuracy drops from **83.93%** to **83.42%**, **83.07%**,

Table 1. Comparison with baseline methods on multivariate time-series classification datasets across four domains in terms of accuracy (%), with domain-wise and overall average ranks.

Domain	Dataset	TVSRNet	Transformer	iTransformer	Crossformer	Informer	TimesNet	LightTS	PatchTST	MiniROCKET
Traffic & Infrastructure	PEMS-SF	**86.12**	84.39	84.83	85.54	82.08	80.34	82.65	79.76	85.28
	PEMS-BAY	**94.44**	88.88	88.88	88.88	91.66	86.11	86.11	83.33	86.11
	Avg. Rank	1.00	4.50	4.00	3.00	4.50	7.50	6.50	9.00	5.00
Human Activity & Motion	HAR	**97.14**	92.40	92.86	91.72	92.90	93.07	94.41	86.46	**97.14**
	UWaveGestureLibrary	88.43	86.56	86.41	85.53	85.93	85.93	83.12	85.62	**94.04**
	BasicMotions	**100.00**	**100.00**	**100.00**	92.50	**100.00**	**100.00**	82.50	72.50	**100.00**
	RacketSports	**88.16**	87.50	87.24	76.50	86.84	86.84	77.63	76.97	85.26
	Cricket	**95.83**	94.44	94.62	93.06	93.05	**95.83**	88.88	**95.83**	94.61
	PenDigits	**99.28**	98.43	98.61	97.88	97.60	98.40	96.51	96.37	97.34
	Handwriting	34.00	34.82	35.07	23.88	32.00	32.59	21.29	28.82	**36.59**
	Libras	84.44	85.56	85.94	**87.22**	82.22	85.56	72.78	79.44	82.88
	NATOPS	91.11	**96.11**	95.84	88.33	93.33	92.22	91.66	78.89	91.67
	Epilepsy	98.20	89.13	89.46	84.06	89.86	93.47	86.23	94.55	**100.00**
	Avg. Rank	2.90	3.90	3.45	7.00	5.35	4.00	7.60	7.10	3.70
Physiological & Neuro Signals	HandMovementDirection	**67.56**	63.51	63.84	54.05	59.46	60.81	55.41	60.81	40.54
	SelfRegulationSCP1	89.76	89.41	89.68	90.10	90.44	89.41	**92.15**	85.32	91.47
	SelfRegulationSCP2	**55.56**	54.44	54.73	**55.56**	55.00	52.22	53.33	54.44	54.44
	MotorImagery	**65.50**	58.00	58.37	55.50	62.00	59.00	64.50	55.50	53.00
	FaceDetection	**69.54**	68.96	69.12	67.85	68.07	67.05	67.37	66.28	59.73
	FingerMovements	**64.00**	56.00	56.48	60.00	57.00	54.00	60.00	58.00	55.00
	Heartbeat	**79.02**	78.04	78.31	76.09	77.56	76.09	75.61	71.95	76.59
	AtrialFibrillation	46.67	33.33	33.74	**53.33**	40.00	33.33	46.67	33.33	13.33
	StandWalkJump	**66.67**	46.67	46.94	43.33	40.00	46.67	33.33	**66.67**	33.33
	Avg. Rank	1.72	5.22	3.89	4.67	4.33	6.56	5.06	6.28	7.28
Audio, Speech & Other Sensing	ArticularyWordRecognition	**99.00**	98.66	98.81	97.66	98.00	98.33	97.66	98.33	**99.00**
	JapaneseVowels	**98.11**	97.84	97.96	97.30	97.03	96.22	95.95	96.21	97.65
	SpokenArabicDigits	**98.54**	97.99	98.12	98.18	97.63	97.36	97.36	98.09	97.72
	PhonemeSpectra	14.08	12.94	13.07	13.63	12.35	13.84	13.90	14.11	**27.68**
	DuckDuckGeese	**64.00**	58.00	58.22	56.00	60.00	62.00	56.00	56.00	54.00
	EthanolConcentration	**41.55**	26.23	26.58	29.27	30.42	29.27	29.28	28.90	40.30
	LSST	41.04	41.04	41.28	34.75	34.55	40.02	37.43	51.54	**63.34**
	Avg. Rank	1.86	5.50	4.29	6.00	6.29	5.64	6.86	5.07	3.50
Overall Avg. Rank		2.12	4.77	3.84	5.71	5.20	5.48	6.52	6.46	4.89

and **83.55%** for *w/o Channel sparsity*, *w/o Spectral sparsity*, and *w/o Interaction sparsity*, respectively. These results show that the triple-view architecture provides the main performance gain, while sparsity further improves the representation quality.

4.4 Further Analysis

Cross-Dataset Few-Shot Transfer within the Same Domain. To evaluate whether TVSRNet can transfer useful inductive biases across related tasks, we conduct a cross-dataset few-shot transfer experiment within the same domain. Specifically, we first pretrain the model on the full training set of a source dataset, and then fine-tune it on only 20% of the training set of a target dataset from the same domain. The final evaluation is performed on the full test set of the target dataset. Since the source and target datasets correspond to different classification tasks, we transfer only the encoder parameters from the source model and randomly initialize the classification head for the target dataset during fine-tuning.

We conduct this experiment on three representative source-target pairs from different domains, namely *PEMS-BAY* → *PEMS-SF*, *SpokenArabicDigits* → *JapaneseVowels*, and *SelfRegulationSCP1* → *SelfRegulationSCP2*. As shown in

Table 2. Ablation study of TVSRNet from the perspectives of single-view modeling and sparsity removal.

Variant	PEMS-SF	JapaneseVowels	HandMovementDirection	Avg.
Full	**86.12**	**98.11**	**67.56**	**83.93**
Only Channel	82.34	95.62	61.48	79.81
Only Spectral	81.97	95.88	60.92	79.59
Only Interaction	83.45	96.71	63.14	81.10
w/o Channel sparsity	85.54	97.82	66.91	83.42
w/o Spectral sparsity	84.61	97.76	66.85	83.07
w/o Interaction sparsity	85.73	97.89	67.02	83.55

Table 3, TVSRNet achieves the best performance on all three transfer pairs, with an average accuracy of **68.07**%. This improves over **PatchTST** (**65.14**%) by **2.93** points, over **iTransformer** (**64.96**%) by **3.11** points, and over **Transformer** (**64.27**%) by **3.80** points. These results suggest that the proposed triple-view sparse design captures transferable structure beyond standard supervised fitting and is particularly beneficial in low-resource cross-dataset adaptation.

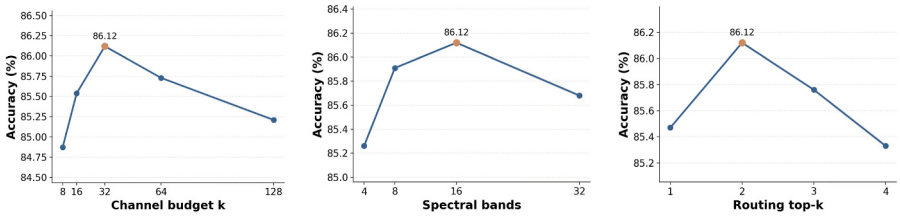
Table 3. Cross-dataset few-shot transfer accuracy (%) within the same domain.

Transfer Pair	Transformer	iTransformer	PatchTST	TVSRNet
PEMS-BAY → PEMS-SF	56.84	58.12	55.93	**60.47**
SpokenArabicDigits → JapaneseVowels	87.52	87.91	87.28	**90.41**
SelfRegulationSCP1 → SelfRegulationSCP2	48.44	48.86	52.22	**53.33**
Average	64.27	64.96	65.14	**68.07**

Sensitivity to Key Hyperparameters. We further analyze the sensitivity of TVSRNet to three key hyperparameters corresponding to its three sparse views, namely the channel budget k_{ch}, the number of spectral bands N_f, and the routing top-k k_r. These parameters control the sparsity strength of channel selection, the granularity of spectral decomposition, and the selectivity of expert routing. We conduct this analysis on *PEMS-SF* while keeping all other settings unchanged. The results are shown in Fig. 3.

As shown in Fig. 3(a), the performance on *PEMS-SF* first improves and then slightly declines as the channel budget k_{ch} increases, with the best behavior observed around $k_{\text{ch}} = 32$. When k_{ch} is too small, informative channels may be discarded together with redundant ones, whereas an overly large budget weakens the intended sparsification effect. A similar trend is observed in Fig. 3(b) for the number of spectral bands N_f, where the model performs best around $N_f = 16$. Too few spectral bands lead to coarse frequency summarization, while

too many bands reduce compactness and weaken spectral sparsity. Figure 3(c) further shows that the model performs best at a relatively small routing top-k k_r, with the most favorable result achieved at $k_r = 2$. Allowing too few experts limits its interaction modeling flexibility, whereas activating too many experts weakens sparse routing.

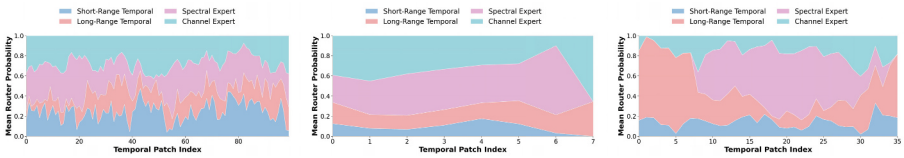


(a) k_{ch} (b) N_f (c) k_r

Fig. 3. Sensitivity analysis of three key hyperparameters on PEMS-SF.

Patch-Wise Expert Routing Patterns Across Datasets. To further understand interaction sparsity, we visualize the normalized mean router probability of different expert groups across temporal patches. As shown in Fig. 4, the routing patterns are clearly dataset-dependent rather than collapsing to a fixed expert preference. On **HandMovementDirection**, all expert groups remain active and fluctuate substantially across patches, indicating strong local heterogeneity and the need for adaptive switching among complementary interaction pathways. On **JapaneseVowels**, the routing pattern is smoother, with channel and spectral experts contributing more steadily, suggesting that the model relies more on relatively stable channel-frequency structure. On **PEMS-SF**, the routing pattern is also interpretable from the perspective of daily traffic routine: spectral experts become more important during periods with stronger regular periodicity, whereas temporal experts are more active when longer-range dynamics dominate. Overall, these results show that the proposed sparse router learns structured and interpretable expert allocation patterns that align well with the natural characteristics of different datasets.

Architectural and Computational Overhead. The proposed Triple-view design introduces an additional sparse residual branch on top of the backbone, consisting of sample-wise channel allocation, FFT-based spectral masking, and patch-level sparse MoE routing. This inevitably brings extra computation compared with very compact models such as LightTS, Transformer, Informer, and TimesNet. Nevertheless, the overhead is explicitly controlled by the sparsified formulation: the Channel view only applies a lightweight MLP to the temporal-pooled representation, the Spectral view uses shared soft masks over a limited number of frequency bands, and the Interaction view activates only the top-k_r experts



(a) HandMovementDirection (b) JapaneseVowels (c) PEMS-SF

Fig. 4. Patch-wise expert routing patterns across datasets.

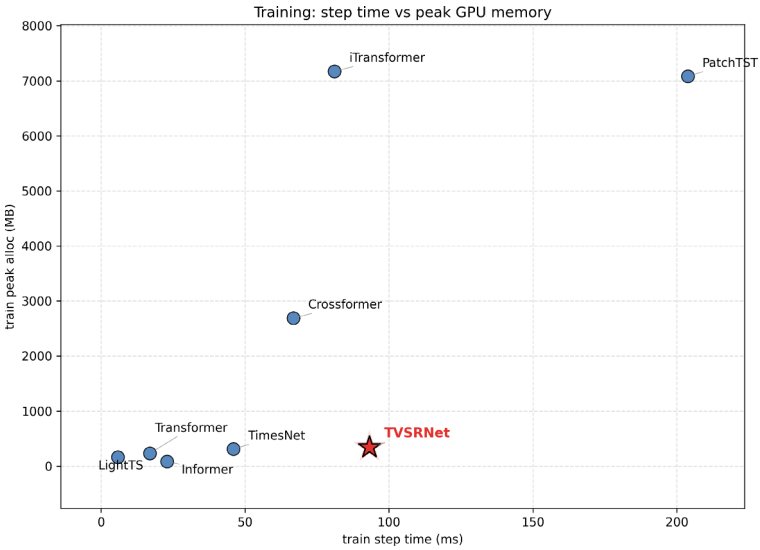


Fig. 5. Training efficiency comparison. We compare the average training step time and peak GPU memory allocation of different deep baselines. TVSRNet introduces a triple-view sparse residual branch, but keeps the memory footprint moderate compared with several Transformer-style baselines.

for each patch rather than evaluating all experts densely. In our implementation, we set $k_{\text{ch}} = 32$, $N_f = 16$, and $k_r = 2$, with 4 temporal experts, 2 frequency experts, and 2 channel experts. As shown in Fig. 5, TVSRNet requires about 93 ms per training step and around 350 MB peak GPU memory. Its step time is higher than lightweight baselines such as LightTS, Transformer, TimesNet, and Crossformer, but it remains far below PatchTST in runtime. More importantly, its memory usage is much lower than Crossformer, iTransformer, and PatchTST, which require approximately 2.7 GB, 7.2 GB, and 7.1 GB peak memory, respectively. Therefore, the Triple-view branch introduces a moderate computational overhead, while avoiding the severe memory cost of several Transformer-style baselines and providing substantial gains in accuracy, average rank, and few-shot transfer performance.

5 Conclusion

We presented TVSRNet, a unified framework for multivariate time-series classification based on triple-view sparsified routing over channel, spectral, and interaction representations. By combining sample-wise channel allocation, spectral band selection, and patch-level sparse expert routing within a residual-injection architecture, TVSRNet can adaptively emphasize the most informative evidence for different datasets and samples. Beyond classification, the framework can also be viewed as a knowledge discovery framework that jointly considers triple-view structure and sparsified representation, thereby supporting unified modeling and transfer across heterogeneous domains. Experiments on **28** datasets show that TVSRNet achieves the best overall average rank of **2.12**, outperforming the strongest baseline by **1.72** rank points, while cross-dataset few-shot transfer further demonstrates its advantage under limited supervision. Additional analyses also confirm the interpretability of the learned channel selection, spectral emphasis, and routing behavior, as well as the effectiveness of the proposed sparsity design. These results suggest that triple-view sparsification provides an effective inductive bias for unified multivariate time-series classification.

Acknowledgments. This paper is partially supported by the National Natural Science Foundation of China (No.62502488, No.12227901), Natural Science Foundation of Jiangsu Province (BK20240460), the grant from State Key Laboratory of Resources and Environmental Information System. The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

1. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: A public domain dataset for human activity recognition using smartphones. In: ESANN, vol. 3, pp. 3–4 (2013)
2. Bagnall, A., et al.: The UEA multivariate time series classification archive. arXiv preprint arXiv:1811.00075 (2018)
3. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. arXiv preprint arXiv:1910.13051 (2019)
4. Dempster, A., Schmidt, D.F., Webb, G.I.: MiniRocket: a very fast (almost) deterministic transform for time series classification. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD) (2021)
5. Fawaz, H.I., et al.: Inceptiontime: finding alexnet for time series classification. Data Min. Knowl. Disc. **34**(6), 1936–1962 (2020). https://doi.org/10.1007/s10618-020-00710-y
6. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. J. Mach. Learn. Res. **23**(120), 1–39 (2022)
7. Kudo, M., Toyama, J., Shimbo, M.: Multidimensional curve classification using passing-through regions. Pattern Recogn. Lett. **20**(11–13), 1103–1111 (1999)

8. Lee-Thorp, J., Ainslie, J., Eckstein, I., Ontañón, S.: Fnet: mixing tokens with fourier transforms. In: Proceedings of NAACL-HLT, pp. 4296–4313 (2022). https://doi.org/10.18653/v1/2022.naacl-main.319
9. Lepikhin, D., et al.: Gshard: scaling giant models with conditional computation and automatic sharding. In: International Conference on Learning Representations (2021)
10. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: International Conference on Learning Representations (ICLR 2018) (2018)
11. Liu, Y., et al.: iTransformer: inverted transformers are effective for time series forecasting. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=JePfAI8fah
12. Ni, R., Lin, Z., Wang, S., Fanti, G.: Mixture-of-linear-experts for long-term time series forecasting. In: Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, pp. 4672–4680 (2024)
13. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: long-term forecasting with transformers. In: International Conference on Learning Representations (2023)
14. Rajpurkar, P., Hannun, A.Y., Haghpanahi, M., Bourn, C., Ng, A.Y.: Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836 (2017)
15. Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **35**, 401–449 (2021). https://doi.org/10.1007/s10618-020-00727-3
16. Shazeer, N., et al.: Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
17. Shi, X., et al.: Time-MoE: billion-scale time series foundation models with mixture of experts. In: International Conference on Learning Representations (2025)
18. THUML: Time-series-library (2023). https://github.com/thuml/Time-Series-Library. Open-source benchmark codebase for general time series analysis
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
20. Wang, Y., Wu, H., Dong, J., Liu, Y., Long, M., Wang, J.: Deep time series models: a comprehensive survey and benchmark. arXiv preprint arXiv:2407.13278 (2024)
21. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: temporal 2D-variation modeling for general time series analysis. In: International Conference on Learning Representations (2023)
22. Zhang, T., et al.: Less is more: fast multivariate time series forecasting with light sampling-oriented MLP structures. arXiv preprint arXiv:2207.01186 (2022). https://doi.org/10.48550/arXiv.2207.01186
23. Zhang, Y., Yan, J.: Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: International Conference on Learning Representations (2023)
24. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
25. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In: Proceedings of the 39th International Conference on Machine Learning, pp. 27268–27286 (2022)