Appendix: Knowledge Aggregation Transformer Network for Multivariate Time Series Classification

Zhiwen Xiao, *Member, IEEE*, Huanlai Xing, *Member, IEEE*, Rong Qu, *Senior Member, IEEE*, Hui Li, Huagang Tong, Shouxi Luo, *Member, IEEE*, Jing Song, Li Feng, and Qian Wan

THEORETICAL ANALYSIS OF THE KNOWLEDGE AGGREGATION MECHANISM

In this section, we provide a detailed theoretical analysis of the knowledge aggregation mechanism within KATN. Unlike traditional fusion approaches, KATN employs an additive aggregation operation, integrating both local and global representations in a way that ensures feature-space alignment, gradient coherence, and enhanced model generalization. By examining task-oriented loss landscapes, gradient behavior, and the geometric properties of aggregated features, we demonstrate the superior advantages of KATN in comparison to existing fusion methods.

A. Feature Aggregation

Let $\mathbf{X}_i \in \mathbb{R}^{T\times D}$ represent the input multivariate time series for the i-th sample, where T is the number of time steps and D denotes the number of feature channels. At each transformer block, the outputs of the MResNet and multi-head attention network are denoted as:

$$\mathbf{O}_{i,MRN}^{j,k} \in \mathbb{R}^{T\times d_{i,MRN}^{j,k}}, \quad \mathbf{O}_{i,MHA}^{j,k} \in \mathbb{R}^{T\times d_{i,MHA}^{j,k}}, \tag{A1}$$

where, $\mathbf{O}_{i,MRN}^{j,k}$ and $\mathbf{O}_{i,MHA}^{j,k}$ are the outputs from the MResNet and multi-head attention network, respectively. To ensure consistency in dimensionality, we impose:

$$d_{i,MRN}^{j,k} = d_{i,MHA}^{j,k} = d. \tag{A2}$$

The aggregated output of the two networks is computed as:

$$\mathbf{O}_{i,AGG}^{j,k} = \mathrm{LN}\left(\mathbf{W}_{agg}^{j,k}\cdot\sigma\left(\mathbf{O}_{i,MRN}^{j,k}+\mathbf{O}_{i,MHA}^{j,k}\right)+\mathbf{b}_{agg}^{j,k}\right), \tag{A3}$$

where, $\sigma(\cdot)$ represents a non-linear activation function (e.g., GeLU), and $\mathbf{W}_{agg}^{j,k} \in \mathbb{R}^{d\times d}$ is a learnable affine projection matrix. This operation guarantees that the aggregated features remain aligned within a consistent semantic space.

B. Gradient Behavior and Optimization Dynamics

A critical aspect of the aggregation mechanism is the gradient flow, which influences the stability and efficiency of the optimization process. The gradient of the aggregation loss $\mathcal{L}$ with respect to the aggregated representation $\mathbf{O}_{i,AGG}^{j,k}$ is expressed as:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{O}_{i,AGG}^{j,k}} = \mathbf{G}^{j,k} \in \mathbb{R}^{T\times d}. \tag{A4}$$

The gradients with respect to the individual representations are:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{O}_{i,MRN}^{j,k}} = \mathbf{G}^{j,k}\cdot\mathbf{W}_{agg}^{j,k}\cdot\mathrm{diag}\left(\sigma'\left(\mathbf{O}_{i,MRN}^{j,k}+\mathbf{O}_{i,MHA}^{j,k}\right)\right), \tag{A5}$$

and similarly for $\mathbf{O}_{i,MHA}^{j,k}$:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{O}_{i,MHA}^{j,k}} = \mathbf{G}^{j,k}\cdot\mathbf{W}_{agg}^{j,k}\cdot\mathrm{diag}\left(\sigma'\left(\mathbf{O}_{i,MRN}^{j,k}+\mathbf{O}_{i,MHA}^{j,k}\right)\right). \tag{A6}$$

These gradient expressions reveal that the shared aggregation mechanism ensures a smooth and consistent gradient flow between the branches, mitigating the instability often observed in multi-branch architectures. In contrast, other fusion methods may encounter gradient inconsistencies, hindering optimization performance.

C. Geometric Regularization

The aggregated features $\mathbf{O}_{i,AGG}^{j,k}$ can be conceptualized as points on a low-dimensional manifold $\mathcal{M} \subseteq \mathbb{R}^{T\times d}$. To ensure smooth feature learning and avoid overfitting, we introduce a regularization term to penalize excessive curvature of the aggregation function:

$$\mathcal{R}_{curv}(\mathbf{O}_{i,AGG}^{j,k}) = \left\|\nabla_{\mathbf{X}_i}^2\mathbf{O}_{i,AGG}^{j,k}\right\|_F^2. \tag{A7}$$

The Jacobian of the aggregation operation is given by:

$$\mathbf{J}_{AGG}^{j,k} = \frac{\partial\mathbf{O}_{i,AGG}^{j,k}}{\partial\mathbf{X}_i} = \mathbf{W}_{agg}^{j,k}\cdot\left(\mathbf{J}_{MRN}^{j,k}+\mathbf{J}_{MHA}^{j,k}\right). \tag{A8}$$

By ensuring a well-conditioned Jacobian, KATN guarantees that the aggregated features remain on a smooth manifold, promoting stable learning dynamics and generalization.

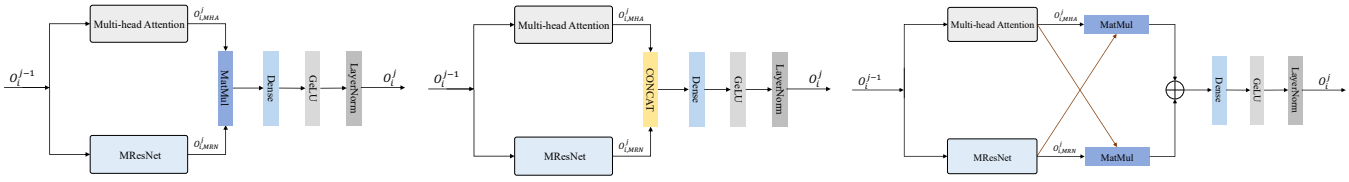
D. Convergence Analysis

We investigate the convergence behavior of the optimization process. The gradient of the total loss $\mathcal{L}$ with respect to any parameter $\theta \in \boldsymbol{\theta}$ is given by:

$$\frac{\partial\mathcal{L}}{\partial\theta} = \sum_{i=1}^{N}\frac{\partial\mathcal{L}}{\partial O_i}\cdot\frac{\partial O_i}{\partial\theta}, \tag{A9}$$

where, the partial derivative $\frac{\partial\mathcal{L}}{\partial O_i}$ is:

$$\frac{\partial\mathcal{L}}{\partial O_i} = -\frac{y_i}{O_i}+\frac{1}{1-O_i}. \tag{A10}$$



(a) KATN-MatMul (b) KATN-CONCAT (c) KATN-INTERN

Fig. A1. Differences between three feature aggregation methods.

TABLE A1
THEORETICAL COMPARISON OF FUSION METHODS

Fusion Method	Gradient Flow	Expressiveness	Generalization	Computational Cost
KATN	Smooth, stable	Moderate	High	Low
KATN-MatMul	Risk of vanishing gradients	High	Moderate	Low
KATN-CONCAT	Direct but expensive	Moderate	Moderate	High
KATN-INTERN	Complex but unstable	High	Moderate	Very High

Assuming that $\mathcal{L}(\boldsymbol{\theta})$ is smooth and its gradient is L-Lipschitz continuous:

$$\|\nabla\mathcal{L}(\boldsymbol{\theta}_1)-\nabla\mathcal{L}(\boldsymbol{\theta}_2)\| \leq L\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|, \quad \forall\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\in\mathbb{R}^d, \tag{A11}$$

the gradient descent algorithm converges at a rate of:

$$\mathcal{O}\left(\frac{1}{m}\right), \tag{A12}$$

where, m denotes the iteration index. This convergence is contingent on an appropriate learning rate and highlights the tractability of optimizing KATN.

E. Comparison with Existing Fusion Methods

To assess the efficacy of KATN's additive aggregation approach, we compare it with three other fusion variants: KATN-MatMul, KATN-CONCAT, and KATN-INTERN. These variants differ in how they combine the outputs of the MResNet and multi-head attention networks within each transformer block. The methods are as follows:

- **KATN**: The additive aggregation approach, where the outputs from the MResNet and multi-head attention networks are summed before being passed through a shared non-linearity and affine transformation. This ensures stable gradients, smooth feature alignment, and efficient optimization.
- **KATN-MatMul**: The multiplicative aggregation method, where the outputs from the MResNet and multi-head attention networks are element-wise multiplied before passing through the subsequent transformations, as illustrated in Fig. A1 (a).
- **KATN-CONCAT**: The concatenation-based fusion strategy, where the outputs from the two networks are concatenated along the feature dimension, as depicted in Fig. A1 (b).
- **KATN-INTERN**: The interactive aggregation method, where the two feature sets interact through a learned transformation, allowing for complex feature fusion, as shown in Fig. A1 (c).

Theoretical Analysis and Comparison:

- **Gradient Flow:**
  - In KATN-MatMul, the multiplicative interaction can lead to vanishing gradients, particularly when the input features are small.
  - KATN-CONCAT ensures direct gradient flow but increases computational cost due to the higher dimensionality of the concatenated features.
  - KATN-INTERN offers more flexibility in feature interaction, but its increased complexity can hinder efficient optimization.
  - KATN, with its additive aggregation, ensures smooth and stable gradient propagation across both branches, avoiding the pitfalls of vanishing gradients and computational inefficiencies.
- **Expressiveness:**
  - KATN-INTERN is the most expressive, as it allows for complex interactions between the features, but at the cost of higher computational overhead.
  - KATN-CONCAT provides moderate expressiveness but may face challenges in handling high-dimensional feature spaces.
  - KATN-MatMul, while expressive, suffers from the risks associated with multiplicative operations.
  - KATN, with its additive aggregation, strikes an optimal balance between expressiveness and efficiency, offering robust feature fusion without unnecessary complexity.
- **Generalization:**
  - KATN-INTERN has the potential for better generalization due to its more flexible feature fusion mechanism, though it may overfit in certain scenarios.
  - KATN-CONCAT may struggle with overfitting when the feature space is excessively high-dimensional.

- KATN-MatMul tends to offer moderate generalization but is highly dependent on the multiplicative interaction's stability.
- KATN, with its efficient aggregation strategy, generalizes effectively across various tasks, ensuring high performance even in heterogeneous setups.

The comparison between these variants clearly demonstrates that KATN offers the best balance of efficiency, gradient stability, and generalization across a range of tasks, as shown in Table A1.

In conclusion, KATN's additive aggregation approach outperforms other fusion methods in terms of gradient stability, computational efficiency, and generalization performance. This makes it the most suitable choice for solving various MTSC challenges.