MSAGAT-Net: Multi-Scale Adaptive Graph Attention Network for Spatiotemporal Forecasting

Architecture Diagrams

April 20, 2025

1 Network Architecture Overview

2 Key Architectural Components

2.1 Depthwise Separable Convolution for Temporal Feature Extraction

2.2 Efficient Adaptive Graph Attention Module

2.3 Dilated Multi-Scale Temporal Module

2.4 Progressive Prediction Module

3 Tensor Flow Throughout Network

4 Mathematical Formulation

The MSAGAT-Net model can be described mathematically as follows:

$$\boldsymbol{X} \in \mathbb{R}^{B \times T \times N} \quad \text{(Input time series)} \tag{1}$$

$$\boldsymbol{A} \in \mathbb{R}^{N \times N} \quad \text{(Adjacency matrix)} \tag{2}$$

$$\boldsymbol{F} = \text{TemporalFeatureExtraction}(\boldsymbol{X}) \quad \text{(Temporal features)} \tag{3}$$

$$\boldsymbol{H} = \text{FeatureProcessing}(\boldsymbol{F}) \quad \text{(Processed features)} \tag{4}$$

$$\boldsymbol{H'} = \text{EfficientGraphAttention}(\boldsymbol{H}, \boldsymbol{A}) \quad \text{(Spatial features)} \tag{5}$$

$$\boldsymbol{H''} = \text{MultiScaleTemporal}(\boldsymbol{H'}) \quad \text{(Multi-scale features)} \tag{6}$$

$$\hat{\boldsymbol{Y}} = \text{ProgressivePrediction}(\boldsymbol{H''}, \boldsymbol{X}_T) \quad \text{(Predictions)} \tag{7}$$

$$\mathcal{L} = \text{MSE}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) + \lambda \|\boldsymbol{\alpha}\|_1 \quad \text{(Loss function)} \tag{8}$$

where $\boldsymbol{X}_T$ represents the last observed values and $\lambda \|\boldsymbol{\alpha}\|_1$ is the attention regularization term that encourages sparse attention patterns for better interpretability.

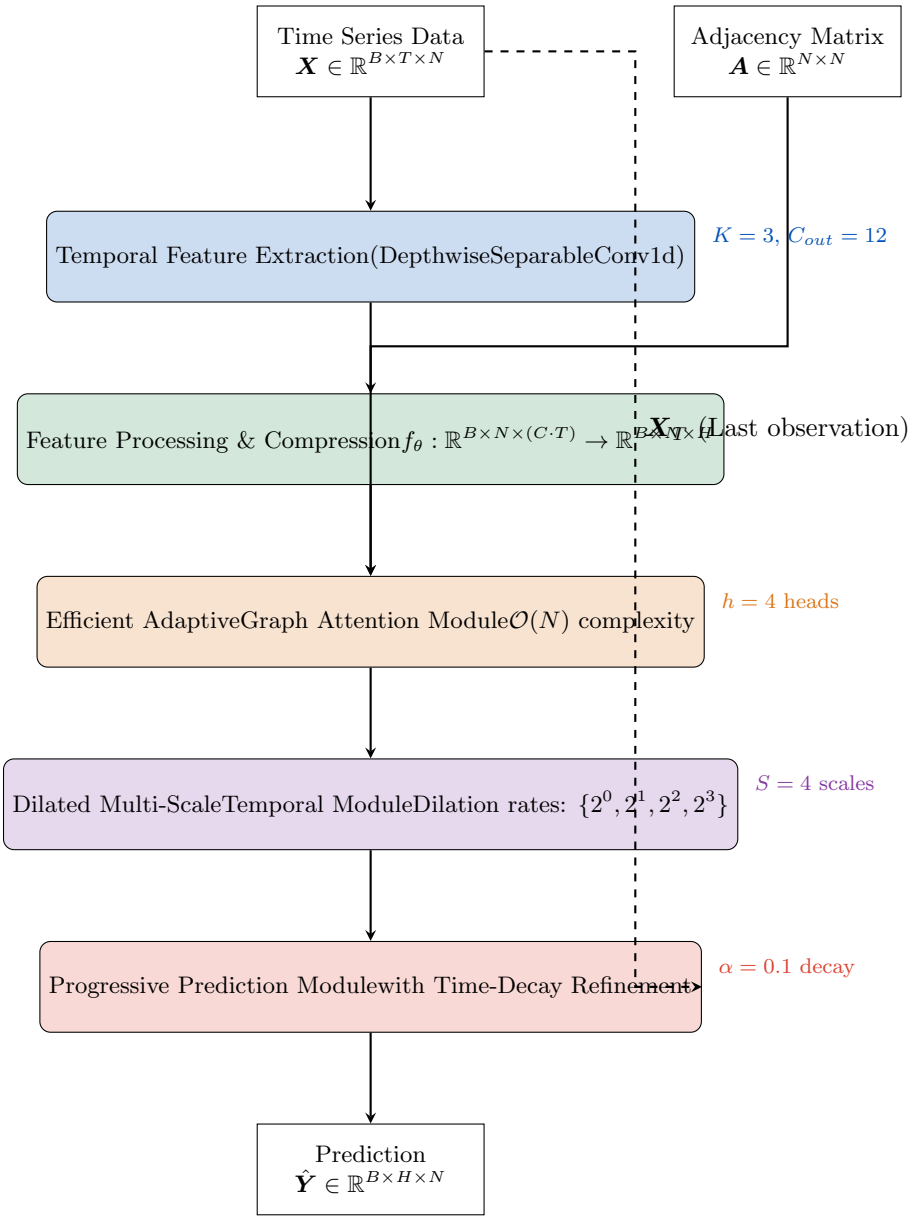

Figure 1: Overall architecture of MSAGAT-Net for spatiotemporal forecasting. The model processes time series data $\boldsymbol{X} \in \mathbb{R}^{B \times T \times N}$ and an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ through specialized modules for temporal and spatial feature extraction. B represents batch size, T is the input window size, N is the number of nodes, H is the hidden dimension, and the output $\hat{\boldsymbol{Y}}$ provides predictions for the next H time steps.

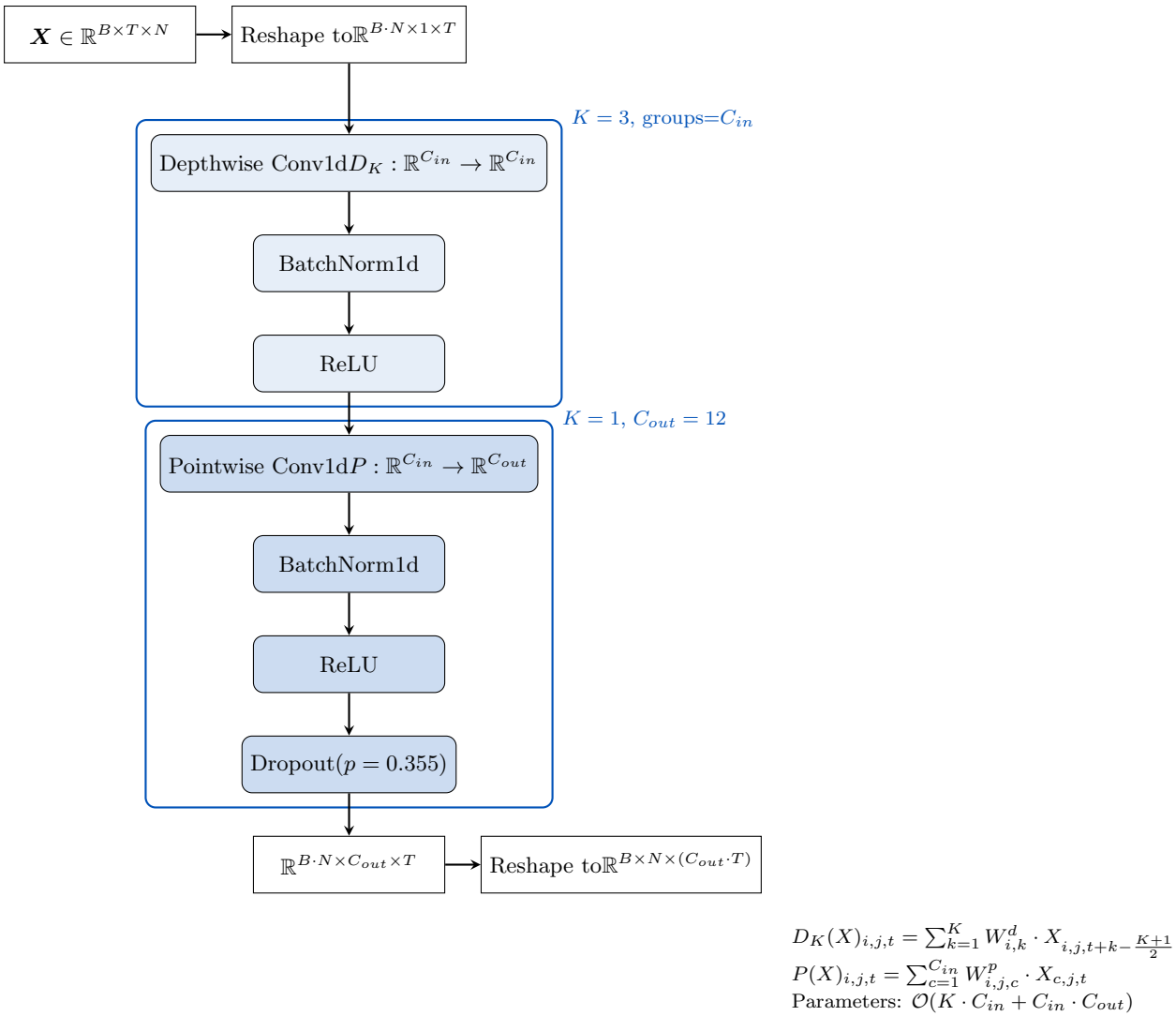

Figure 2: Depthwise Separable Convolution for temporal feature extraction. This factorized convolution operation first applies a depthwise convolution D_K with kernel size $K = 3$ that processes each channel independently, followed by a pointwise convolution P that mixes the channels. This factorization reduces the parameter count from $\mathcal{O}(K \cdot C_{in} \cdot C_{out})$ to $\mathcal{O}(K \cdot C_{in} + C_{in} \cdot C_{out})$ while maintaining strong representational capacity.

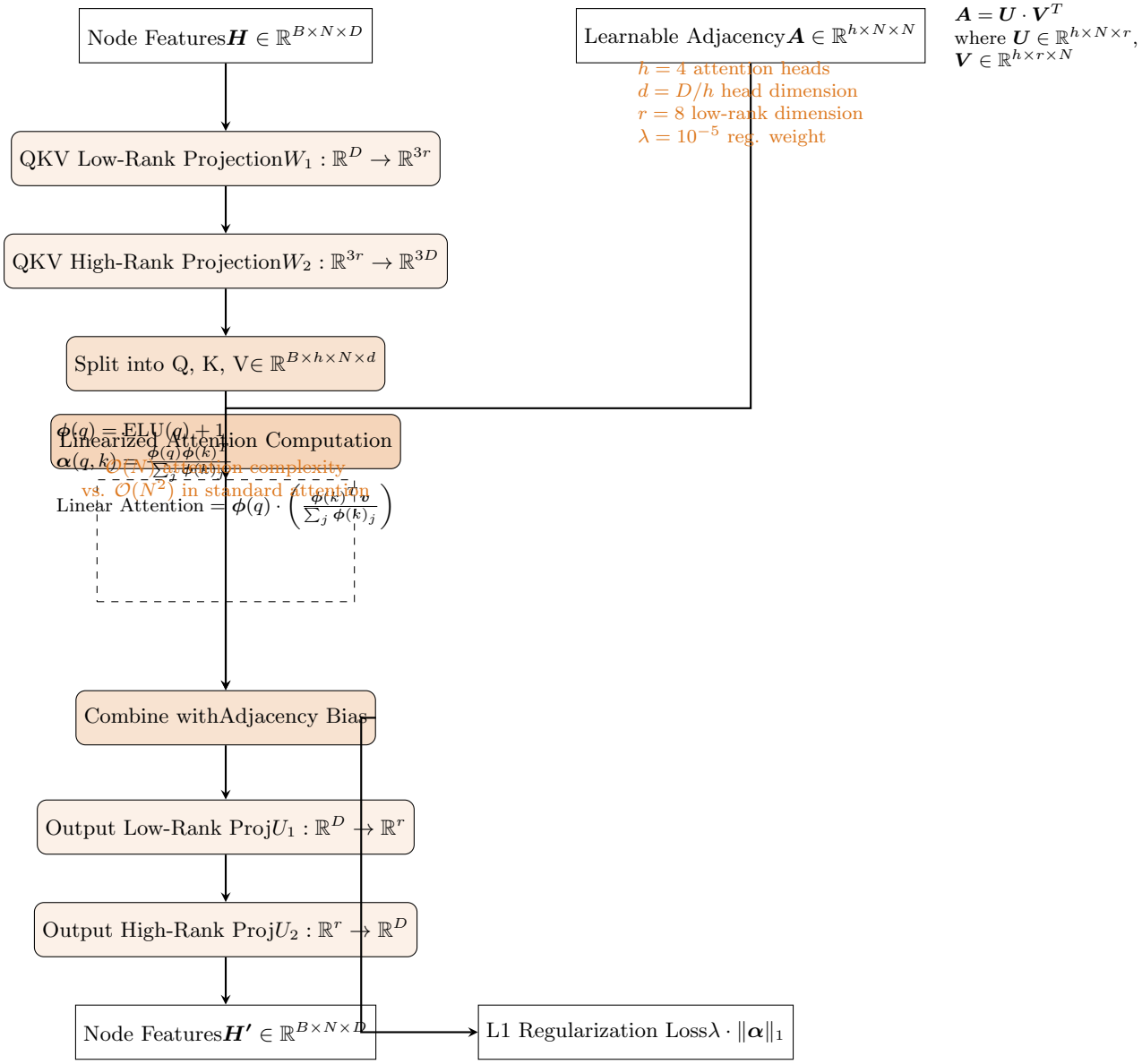

Figure 3: Efficient Adaptive Graph Attention Module with $\mathcal{O}(N)$ complexity and low-rank factorization. This module implements linear attention using the ELU+1 kernel trick, which transforms the quadratic complexity of standard attention to linear. The learnable adjacency matrix is factorized as $\boldsymbol{A} = \boldsymbol{U} \cdot \boldsymbol{V}^T$ to reduce memory requirements from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nr)$. All projection matrices are decomposed using low-rank factorization for parameter efficiency. The L1 regularization encourages sparse attention patterns for better interpretability.

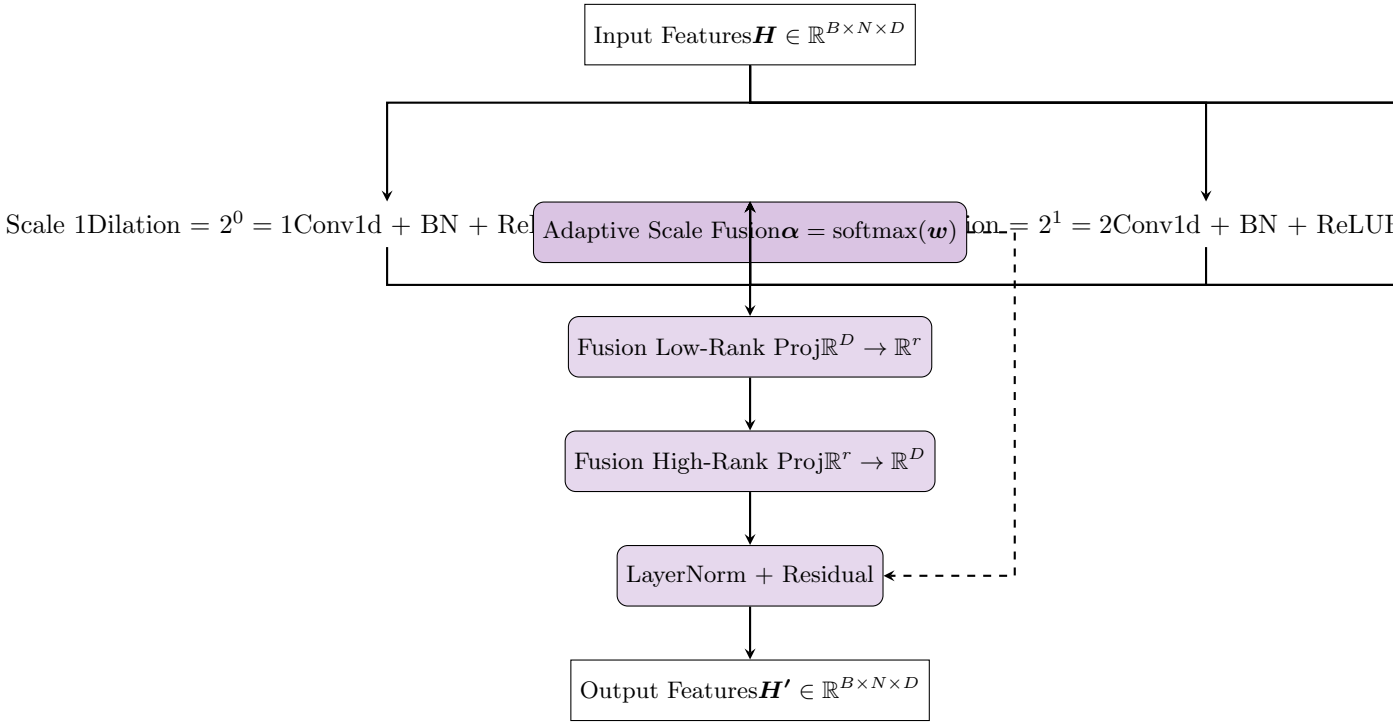

Figure 4: Dilated Multi-Scale Temporal Module with adaptive fusion of features from different time scales. This module captures temporal patterns at multiple resolutions by using parallel dilated convolutions with exponentially increasing dilation rates (1, 2, 4, 8). This creates an effective receptive field ranging from 3 to 17 time steps without requiring deep stacking of layers. The features from different scales are combined using learned weights that are dynamically adapted during training. A residual connection helps maintain gradient flow.

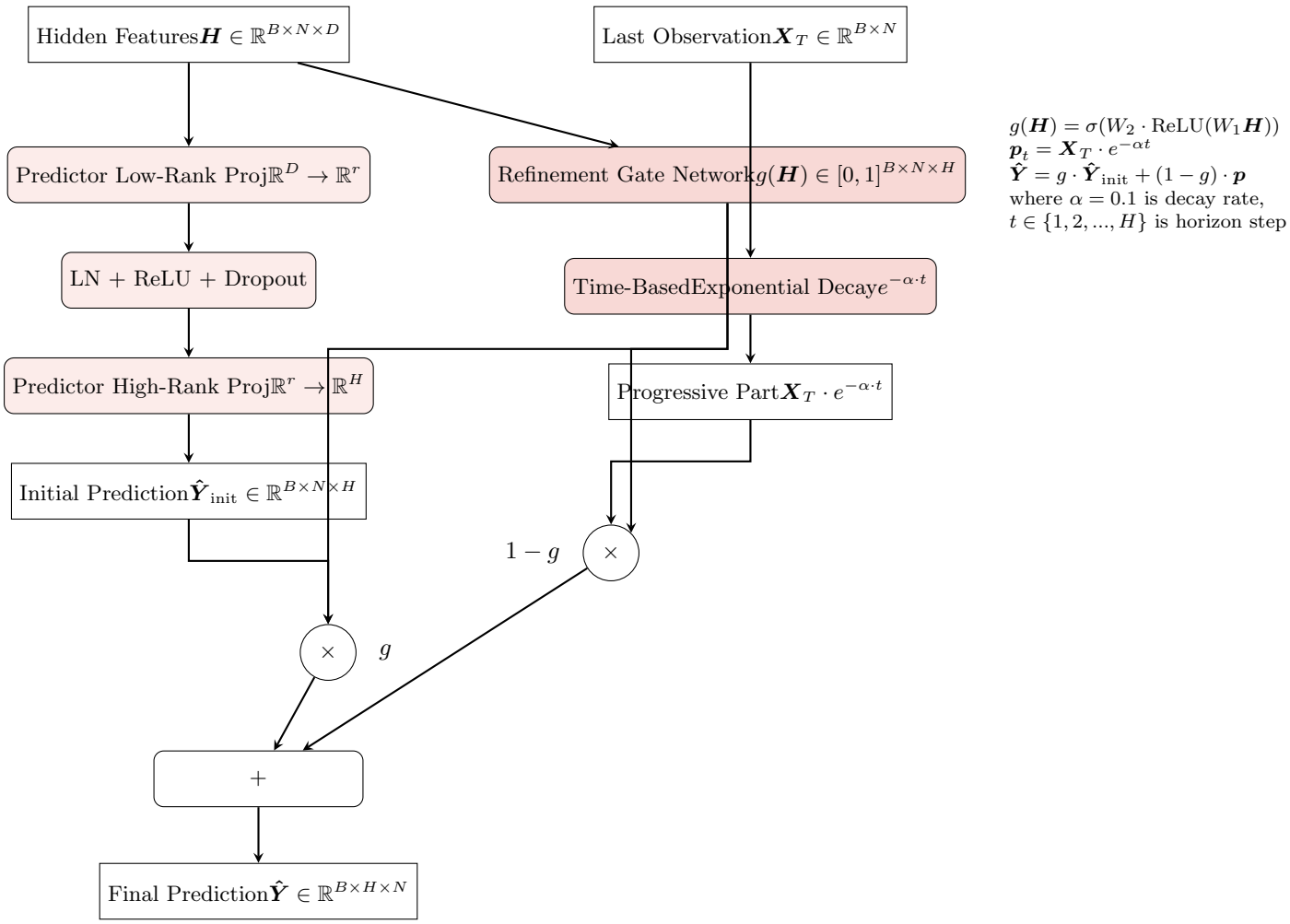

Figure 5: Progressive Prediction Module with adaptive gating and time-decay refinement. This module generates initial multi-step predictions that are then adaptively combined with a persistence forecast derived from the last observation $\boldsymbol{X}_T$. The gating network dynamically determines how much to rely on the model's prediction versus the persistence forecast for each node and time step. The persistence forecast is adjusted with an exponential decay factor $e^{-\alpha t}$ to reflect decreasing confidence in persistent patterns over longer horizons. This approach helps stabilize predictions, especially for shorter horizons.

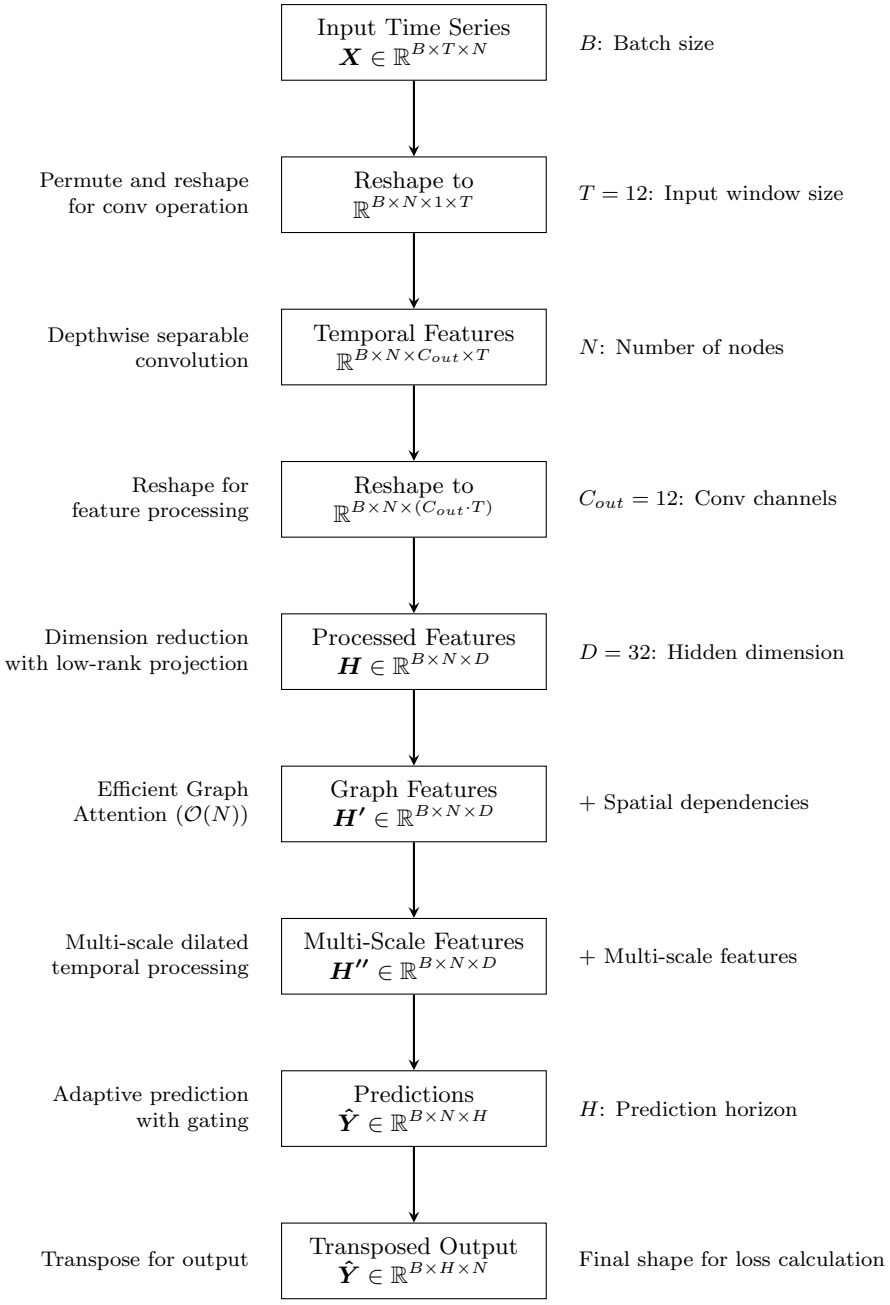

Figure 6: Detailed tensor flow through the MSAGAT-Net architecture showing dimensions and transformations at each stage. The network processes input time series data with shape $[B, T, N]$ through a sequence of specialized modules that extract and combine temporal and spatial features. Low-rank projections are used throughout the network to reduce parameters while maintaining model capacity. The final output tensor $[B, H, N]$ provides predictions for each node across the forecast horizon.