MSTGAT-Net: A Multi-Scale Temporal Graph Attention Network for Spatiotemporal Forecasting

Author Names Institution
Email: author@institution.edu

***Abstract*—**

***Index Terms*—Deep Learning, Graph Neural Networks, Multi-head Attention, Time Series Analysis, Epidemic Forecasting, Adaptive Graph Learning, Multi-scale Feature Fusion, Spatiotemporal Prediction**

I. Introduction

II. Literature Review

III. Methodology

A. Problem Formulation

Let us consider N geographical regions (e.g., cities, counties, states, or NHS regions in England) as nodes in a graph. Historical epidemic data are represented as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$, where $\mathbf{x}_t \in \mathbb{R}^N$ denotes the observed values in all N regions at time step t. Each individual element $x_{i,t}$ represents the epidemic measure (e.g. ventilator bed occupancy or positive case count) for the region i at time t.

For each specific region i, its temporal sequence is represented as $\mathbf{x}^i = [x_{i,1}, x_{i,2}, \ldots, x_{i,T}] \in \mathbb{R}^T$. This dual representation allows us to analyse both the spatial patterns (across regions at a specific time) and temporal patterns (within a region across time).

Our primary objective is to predict future epidemic values for all regions on a specific time horizon h steps ahead. Formally, given the historical data up to time t, we want to predict:

$$\mathbf{x}_{t+h} = [x_{1,t+h}, x_{2,t+h}, \ldots, x_{N,t+h}]^T \tag{1}$$

For practical forecasting, we employ a sliding window approach with a fixed-length lookback period w. At any current time step t, we use the most recent observations w $[\mathbf{x}_{t-w+1}, \mathbf{x}_{t-w+2}, \ldots, \mathbf{x}_t]$ to predict $\mathbf{x}_{t+h}$.

The spatial relationships between regions are encoded in a graph structure $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathbf{A})$, where $\mathscr{V} = \{v_1, v_2, \ldots, v_N\}$ represents the set of regions, $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ denotes the connections between regions and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix. Each element a_{ij} of $\mathbf{A}$ quantifies the strength of the relationship between regions v_i and v_j. The adjacency matrix can be constructed based on various criteria, such as geographical proximity, transportation networks, or healthcare referral patterns. For example, in the context of epidemic forecasting, the adjacency matrix can reflect the mobility patterns of individuals between regions or the referral pathways of patients between healthcare facilities.

The forecasting task can be formalised as learning a function f that maps recent historical data and graph structure to future predictions.

$$\mathbf{x}_{t+h} = f([\mathbf{x}_{t-w+1}, \mathbf{x}_{t-w+2}, \ldots, \mathbf{x}_t], \mathscr{G}; \Theta) \tag{2}$$

where Θ represents the learnable parameters of our forecasting model.

The challenge lies in designing this function f to effectively capture both spatial dependencies between regions and temporal patterns within regions, whilst remaining computationally tractable and robust to the noisy and incomplete nature of epidemic data. Our approach, detailed in the following sections, addresses this challenge through a novel neural network architecture that combines graph attention mechanisms with multi-scale processing.

B. Feature Extraction

The first component of the MSAGAT-Net architecture is the feature extraction module, which transforms the raw time-series data into meaningful feature representations whilst maintaining computational efficiency. Given the input time-series data $\mathbf{X} = [\mathbf{x}_{t-w+1}, \mathbf{x}_{t-w+2}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{N \times w}$ for the N regions over a lookback window of w time steps, we need to extract features that capture relevant temporal patterns for each region. This is achieved through a combination of depthwise separable convolutions and low-rank projections, which allow for efficient feature extraction while reducing the risk of overfitting. The idea of using depthwise separable convolutions is inspired by the success of this approach in computer vision tasks, where it has been shown to significantly reduce the number of parameters and computational complexity while maintaining high performance chollet2017xception. Our adaptation of this technique draws from the work of li2023multi and yu2022traffic, who have successfully applied depthwise separable convolutions to feature extraction rather than the standard convolutional approach. This allows us to efficiently capture the temporal dynamics of epidemic data without incurring the high computational costs associated with traditional convolutional architectures.

1) Depthwise Separable Convolutions: We employ the depthwise separable convolutions to extract temporal features

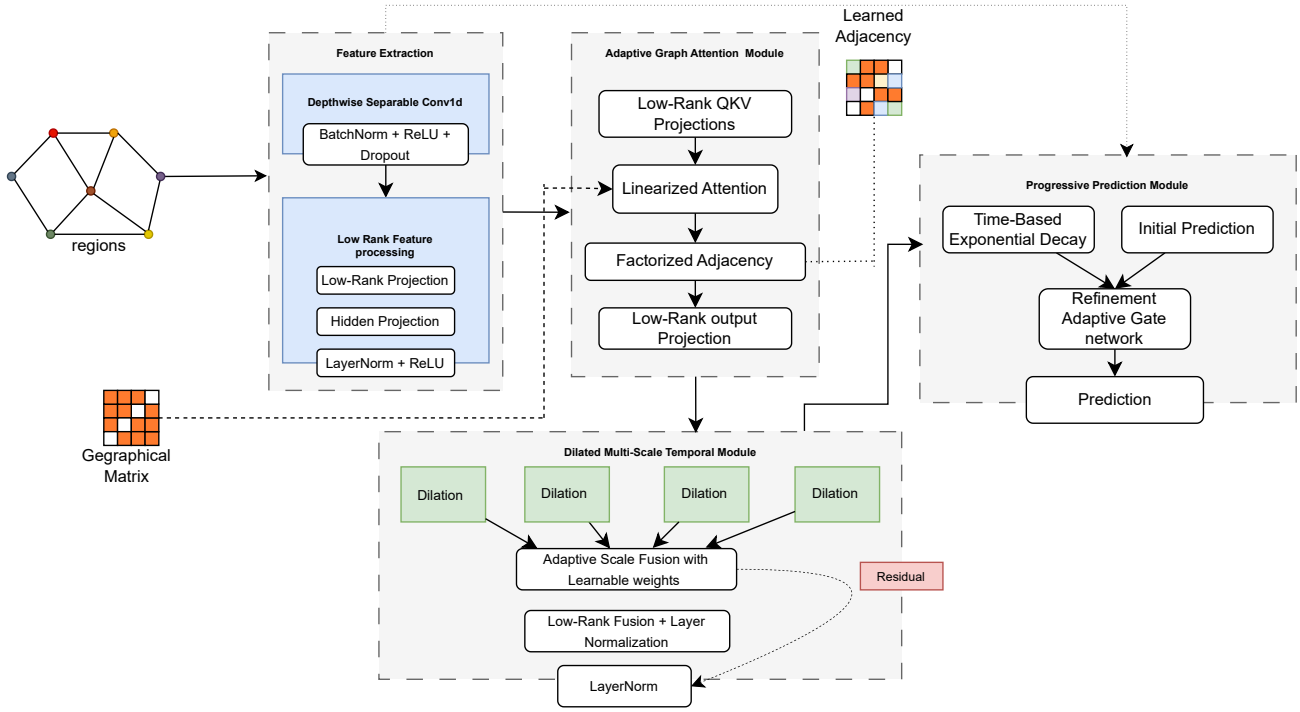


Fig. 1: The proposed MSAGAT-Net architecture, comprising four specialised modules: Feature Extraction, Adaptive Graph Attention (AGAM), Multi-scale Fusion (MTFM), and Progressive Multi-step Prediction Refinement (PPRM).

efficiently. This approach significantly reduces the computational complexity and number of parameters whilst maintaining expressive power. The depthwise separable convolution consists of two stages:

For each region's time-series $\mathbf{x}^i \in \mathbb{R}^w$, the depthwise convolution applies a separate filter to each input channel (in this case, we treat the time-series as a single-channel input):

$$\mathbf{z}^i_{\text{depth}} = \text{Conv1D}_{\text{depth}}(\mathbf{x}^i; \Theta_{\text{depth}}) \tag{3}$$

where $\mathbf{z}^i_{\text{depth}} \in \mathbb{R}^{w \times d_{\text{mid}}}$ represents the intermediate features after depthwise convolution, and d_{mid} is the number of intermediate feature channels.

Following the depthwise convolution, a pointwise convolution (implemented as a 1×1 convolution) is applied to combine the features across channels:

$$\mathbf{z}^i_{\text{point}} = \text{Conv1D}_{\text{point}}(\mathbf{z}^i_{\text{depth}}; \Theta_{\text{point}}) \tag{4}$$

where $\mathbf{z}^i_{\text{point}} \in \mathbb{R}^{w \times d_{\text{feat}}}$ represents the features after pointwise convolution, and d_{feat} is the number of channels of output feature.

This decomposition significantly reduces the computational complexity and number of parameters compared to standard convolutions whilst maintaining similar expressive power. Specifically, for a standard convolution with kernel size k, input channels c_{in}, and output channels c_{out}, the parameter count is $k \times c_{\text{in}} \times c_{\text{out}}$. In contrast, the separable convolution in depth requires only $k \times c_{\text{in}} + c_{\text{in}} \times c_{\text{out}}$ parameters.

After each convolutional operation, we apply batch normalisation followed by ReLU activation to enhance training stability and introduce non-linearity:

$$\mathbf{z}^i_{\text{norm}} = \text{ReLU}(\text{BatchNorm}(\mathbf{z}^i_{\text{point}})) \tag{5}$$

This normalisation step helps mitigate internal covariate shift during training, whilst the non-linear activation enables the model to capture complex temporal patterns in the epidemic data.

2) Low-Rank Feature Projection: After extracting features using depthwise separable convolutions, we applied a low-rank projection to further reduce dimensionality and capture the most salient features. This projection consists of two linear transformations with a bottleneck in between:

$$\mathbf{F}^i_{\text{low}} = \text{Linear}_{\text{low}}(\text{Flatten}(\mathbf{z}^i_{\text{norm}})) \tag{6}$$

$$\mathbf{F}^i = \text{Linear}_{\text{high}}(\mathbf{F}^i_{\text{low}}) \tag{7}$$

where $\mathbf{F}^i_{\text{low}} \in \mathbb{R}^{d_{\text{bottle}}}$ is the bottleneck representation with dimension d_{bottle}, and $\mathbf{F}^i \in \mathbb{R}^{d_{\text{hidden}}}$ is the final representation of characteristics for region i with dimension d_{hidden}.

The flattening operation converts the convolutional features $\mathbf{z}^i_{\text{norm}} \in \mathbb{R}^{w \times d_{\text{feat}}}$ into a vector of dimension $w \times d_{\text{feat}}$. This is then projected to the bottleneck dimension and subsequently to the hidden dimension.

After applying the low-rank projection to each region's convolutional features, we obtain a comprehensive feature matrix $\mathbf{F} \in \mathbb{R}^{N \times d_{\text{hidden}}}$, where each row $\mathbf{F}^i \in \mathbb{R}^{d_{\text{hidden}}}$ represents

the temporal feature embedding for region i. This matrix encapsulates the essential temporal dynamics across all N regions in a compact, information-dense representation suitable for subsequent spatial modelling.

The low-rank bottleneck projection ($w \times d_{\text{feat}} \to d_{\text{bottle}} \to d_{\text{hidden}}$) serves multiple critical functions within the architecture:

1) By compressing information through a dimension bottleneck $d_{\text{bottle}} \ll w \times d_{\text{feat}}$, we reduce computational complexity from $\mathcal{O}(N \times w \times d_{\text{feat}} \times d_{\text{hidden}})$ to $\mathcal{O}(N \times (d_{\text{bottle}} \times d_{\text{hidden}} + w \times d_{\text{feat}} \times d_{\text{bottle}}))$, enabling efficient processing of large-scale spatio-temporal datasets.
2) The bottleneck architecture creates an information bottleneck that forces the model to distill the most salient temporal patterns. This constrains the effective capacity of the network, preventing overfitting particularly when training data are limited relative to the high-dimensional input space.
3) By forcing information through a lower-dimensional manifold, the projection encourages separation of relevant signals from noise, allowing subsequent layers to focus on truly predictive temporal patterns rather than spurious correlations.
4) The shared projection parameters across all regions create a common latent space, facilitating meaningful comparison and interaction between region features in subsequent graph attention layers.

To stabilise training and enhance feature quality, we apply layer normalisation followed by a non-linear activation:

$$\mathbf{F} = \text{ReLU}(\text{LayerNorm}(\mathbf{F})) \tag{8}$$

The layer normalisation operates across the feature dimension, normalising each region's feature vector independently. This addresses internal covariate shift, enabling faster convergence during training whilst making the model robust to variations in feature scale across different regions. The ReLU activation introduces non-linearity essential for modelling complex temporal patterns whilst preserving sparse activation, a property particularly valuable for epidemic time-series that often exhibit punctuated patterns of activity against background stability.

This processed feature matrix $\mathbf{F}$ now encodes the essential temporal characteristics of each region's epidemic time-series in a form optimised for the subsequent Adaptive Graph Attention Module (AGAM), which will model dynamic spatial dependencies between regions based on these temporal feature representations. The feature extraction pipeline is illustrated in Figure 2.

In our implementation, we set the default feature channel dimension d_{feat} to 16, the bottleneck dimension d_{bottle} to 8, and the hidden dimension d_{hidden} to 32. These values were determined through ablation studies to balance model expressiveness with computational efficiency. For the convolutional operations, we use a kernel size of 3 with appropriate padding to maintain the temporal dimension. This combination of parameters enables our model to effectively capture temporal patterns whilst keeping the parameter count manageable, a crucial consideration for deployment in resource-constrained epidemic monitoring scenarios.

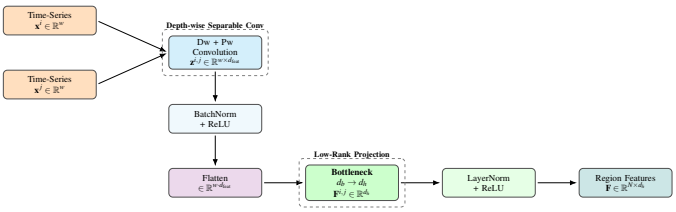


Fig. 2: Feature-extraction pipeline. Independent regional time-series $\mathbf{x}^i$ and $\mathbf{x}^j$ are processed in parallel by depth-wise and point-wise convolutions, normalised, flattened, passed through a bottleneck projection ($d_b \to d_h$), and normalised again to yield region-level feature vectors $\mathbf{F}$.

C. Adaptive Graph Attention with Low-Rank Decomposition

The second core component of our MSAGAT-Net architecture is the Adaptive Graph Attention Module (AGAM). Traditional approaches to spatial modelling often rely on fixed adjacency matrices based on geographical proximity or administrative boundaries, which do not capture the evolving nature of epidemic spread influenced by factors such as population mobility, healthcare referral patterns, and socioeconomic connections. Based on the principles of graph attention networks velickovic2017graph, our AGAM adaptively learns the relationships between regions based on their feature representations, rather than being constrained by a predefined graph structure. This adaptive approach allows the model to discover and leverage spatial dependencies that may not be immediately apparent from geographical proximity alone, and to adjust these dependencies as the epidemic evolves.

A significant challenge in implementing graph attention mechanisms for large-scale epidemic or forecasting problems is computational complexity. Standard attention mechanisms in graph neural networks (GNNs) typically incur quadratic complexity with respect to the number of nodes, making them prohibitively expensive for large graphs. Additionally, these methods often suffer from over-smoothing when modelling long-range dependencies, where node representations become increasingly similar after multiple message-passing iterations.

Recent advances in efficient attention mechanisms have shown that low-rank decomposition techniques can substantially reduce computational complexity whilst maintaining expressive power. Several influential works have explored this direction. Researchers such as puny2020global propose a low-rank global attention (LRGA), an adaptive module that replaces the total attention of the dot product in GNN with a decomposed low-rank form. kong2023low present the Global Representation Key (GRK) attention layer, where the attention scores of each node are calculated using a shared projection of the features of its neighbours. A learnt adaptive low-rank matrix captures the most salient structural information, mitigating over-smoothing and improving performance on graphs. While yang2023self embeds an adaptive low-rank decomposition step in each propagation layer within each ego network to concentrate message passing on the most

prominent low-dimensional subspaces. This lets the model adaptively focus on the most informative subspace per node, improving robustness without labels. These studies collectively demonstrate that low-rank factorisation offers an efficient, scalable and expressive alternative to full-rank attention in graph architectures, and motivate the design of this module in our framework.

Motivated by these advances, AGAM employs a novel attention mechanism that combines low-rank decomposition with a learnable graph structure. Rather than computing the full attention matrix between all pairs of regions (which would incur $\mathcal{O}(N^2)$ complexity), we decompose the attention computation into more efficient operations. Specifically, we used a low-rank approximation of the attention matrix, which allows us to capture the most salient relationships between regions without incurring the full computational cost. This is achieved by projecting the feature representations into a lower-dimensional space before computing attention scores, effectively reducing the number of parameters and operations required.

This allows the model to adaptively learn the strength of connections between regions based on their feature representations rather than relying on a fixed adjacency matrix. The AGAM module consists of several key components: (1) low-rank feature projections, (2) multi-head attention computation, (3) enhanced attention stability, (4) learnable graph structure bias, and (5) attention regularisation.

1) Bottleneck Projection: Given the feature matrix $\mathbf{F} \in \mathbb{R}^{N \times d_{\text{hidden}}}$ from the feature extraction module, where N is the number of regions and d_{hidden} is the hidden dimension, we first project these features into query, key, and value representations through an efficient bottleneck projection:

$$\mathbf{Q}_{\text{low}}, \mathbf{K}_{\text{low}}, \mathbf{V}_{\text{low}} = \text{Split}(\text{Linear}_{\text{low}}(\mathbf{F}), 3) \tag{9}$$

where $\text{Linear}_{\text{low}} : \mathbb{R}^{d_{\text{hidden}}} \to \mathbb{R}^{3 \times d_{\text{bottle}}}$ projects the features into a lower-dimensional space and Split divides the output into three separate tensors of dimension $\mathbb{R}^{N \times d_{\text{bottle}}}$.

These low-dimensional projections are then expanded back to the full hidden dimension:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \text{Split}(\text{Linear}_{\text{high}}([\mathbf{Q}_{\text{low}}; \mathbf{K}_{\text{low}}; \mathbf{V}_{\text{low}}]), 3) \tag{10}$$

where $\text{Linear}_{\text{high}} : \mathbb{R}^{3 \times d_{\text{bottle}}} \to \mathbb{R}^{3 \times d_{\text{hidden}}}$ and each of $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d_{\text{hidden}}}$.

This bottleneck projection significantly reduces the parameter count from $\mathcal{O}(3 \times d_{\text{hidden}}^2)$ to $\mathcal{O}(3 \times d_{\text{hidden}} \times d_{\text{bottle}})$, where $d_{\text{bottle}} \ll d_{\text{hidden}}$.

2) Multi-Head Attention Mechanism: To enhance the model's capacity to capture different types of inter-regional relationships, we implement a multi-head attention mechanism where the hidden representations are split into h heads, each with dimension $d_{\text{head}} = d_{\text{hidden}}/h$:

$$\mathbf{Q}^{(i)}, \mathbf{K}^{(i)}, \mathbf{V}^{(i)} \in \mathbb{R}^{N \times d_{\text{head}}}, \quad i \in \{1, 2, \ldots, h\} \tag{11}$$

For efficient computation, we reshape these tensors to explicitly represent the multiple heads:

$$\mathbf{Q}_h = \text{Reshape}(\mathbf{Q}, [N, h, d_{\text{head}}]) \tag{12}$$

$$\mathbf{K}_h = \text{Reshape}(\mathbf{K}, [N, h, d_{\text{head}}]) \tag{13}$$

$$\mathbf{V}_h = \text{Reshape}(\mathbf{V}, [N, h, d_{\text{head}}]) \tag{14}$$

We then transpose the first two dimensions to facilitate batch-wise processing across attention heads:

$$\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h = \text{Transpose}(\mathbf{Q}_h, 0, 1), \text{Transpose}(\mathbf{K}_h, 0, 1), \text{Transpose}(\mathbf{V}_h, 0, 1) \tag{15}$$

resulting in tensors of shape $[h, N, d_{\text{head}}]$.

A key innovation in our approach is the specific attention computation mechanism employed within each head. Rather than relying on standard scaled dot-product attention with softmax, we employ an enhanced mechanism with better numerical stability and more nuanced relationship modelling.

First, we apply the Exponential Linear Unit (ELU) activation function followed by adding a constant value of 1 to both query and key representations:

$$\hat{\mathbf{Q}}_h = \text{ELU}(\mathbf{Q}_h) + 1 \tag{16}$$

$$\hat{\mathbf{K}}_h = \text{ELU}(\mathbf{K}_h) + 1 \tag{17}$$

This transformation ensures that all attention inputs are positive, improving gradient stability during training whilst allowing for more flexible attention patterns than the standard dot-product attention.

Next, we compute the key-value product for each attention head:

$$\mathbf{KV}_h = \hat{\mathbf{K}}_h^T \mathbf{V}_h \tag{18}$$

where $\mathbf{KV}_h \in \mathbb{R}^{h \times d_{\text{head}} \times d_{\text{head}}}$. This operation captures the relationships between keys and values, allowing the model to learn how to weight the features of different regions based on their similarity.

To ensure stable normalisation, we calculate a normalisation factor based on the sum of keys:

$$\mathbf{z} = \frac{1}{\hat{\mathbf{K}}_h \cdot \mathbf{1} + \varepsilon} \tag{19}$$

where $\mathbf{1}$ is a vector of ones and ε is a small constant (10^{-8} in our implementation) to prevent division by zero. This operation ensures stable normalisation across the attention heads, allowing for effective learning of inter-regional relationships.

The final attention output for each head is computed as:

$$\mathbf{O}_h = \hat{\mathbf{Q}}_h \mathbf{KV}_h \mathbf{z} \tag{20}$$

where $\mathbf{O}_h \in \mathbb{R}^{h \times N \times d_{\text{head}}}$ represents the attended features across all heads.

3) Learnable Graph Structure: An important feature of our AGAM is the incorporation of a learnable graph structure bias. Unlike traditional graph attention networks that rely solely on node features for computing attention, we include a learnable bias term that captures persistent structural relationships between regions that may not be evident from the node features alone.

This bias is implemented as a low-rank decomposition for parameter efficiency:

$$\mathbf{A}_{\text{bias}} = \mathbf{U}\mathbf{V} \tag{21}$$

where $\mathbf{U} \in \mathbb{R}^{h \times N \times d_{\text{bias}}}$ and $\mathbf{V} \in \mathbb{R}^{h \times d_{\text{bias}} \times N}$ are learnable parameters and $d_{\text{bias}} \ll N$ is the bottleneck dimension of the bias term.

The bias is added to the computed attention scores before applying softmax:

$$\mathbf{A} = \text{softmax}\left(\frac{\hat{\mathbf{Q}}_h \hat{\mathbf{K}}_h^T}{\sqrt{d_{\text{head}}}} + \mathbf{A}_{\text{bias}}\right) \tag{22}$$

This formulation allows the model to learn and encode persistent spatial dependencies between regions, such as geographical proximity or administrative hierarchies, whilst still adapting to dynamic relationships that emerge from the data.

4) Attention Regularisation: To promote sparse and interpretable attention patterns, we apply L1 regularisation to the attention weights:

$$\mathscr{L}_{\text{attn}} = \lambda \|\mathbf{A}\|_1 \tag{23}$$

where λ is the regularisation weight. This encourages the model to focus on the most relevant connections between regions, improving both interpretability and generalisation performance. The value of λ (denoted as attention_regularization_weight in our implementation) is treated as a hyperparameter and adjusted during model development.

After computing the attended values for each head, we combine them and project back to the original feature dimension:

$$\mathbf{O} = \text{Reshape}(\text{Transpose}(\mathbf{O}_h, 0, 1), [N, d_{\text{hidden}}]) \tag{24}$$

Similarly to the input projection, we employ a low-rank output projection for efficiency:

$$\mathbf{O}_{\text{low}} = \text{Linear}_{\text{out\_low}}(\mathbf{O}) \tag{25}$$

$$\mathbf{O}_{\text{final}} = \text{Linear}_{\text{out\_high}}(\mathbf{O}_{\text{low}}) \tag{26}$$

where $\mathbf{O}_{\text{low}} \in \mathbb{R}^{N \times d_{\text{bottle}}}$ and $\mathbf{O}_{\text{final}} \in \mathbb{R}^{N \times d_{\text{hidden}}}$.

The output of the AGAM, $\mathbf{O}_{\text{final}}$, represents the features of the region after incorporating spatial dependencies. This output, along with the attention regularisation loss $\mathscr{L}_{\text{attn}}$, is passed to the subsequent multi-scale Fusion Module for further processing.

In our implementation, we set the number of attention heads $h = 4$ and the bottleneck dimension $d_{\text{bottle}} = 8$ as default values. These were determined through ablation studies to provide an optimal balance between model expressiveness and computational efficiency. The weight of attention regularisation λ is set to 10^{-5}, which we found effectively promotes sparse attention patterns without overly constraining the model. The Figure 3 image below represents the flow of data in the

Figure 3 illustrates the flow of data through the AGAM module. The input feature matrix $\mathbf{F}$ is processed through low-rank projections to obtain query, key, and value representations. These representations are then reshaped for multi-head attention computation, where the adaptive graph attention mechanism is applied. The learnable graph structure bias is incorporated into the attention scores, and L1 regularisation is applied to promote sparse attention patterns. Finally, the output features are obtained through high-rank projections, ready for further processing in the multi-scale Fusion Module.

D. Multi-scale Fusion Module

The third major component of our MSAGAT-Net architecture is the multi-scale Fusion Module (MTFM), which addresses a fundamental challenge in epidemic forecasting: capturing temporal patterns that operate at different time scales simultaneously. Epidemics often show intricate temporal dynamics with various scales, including short-term changes (such as weekend effects), medium-term patterns (like incubation periods), and long-term trends (e.g., seasonal variations). Accurate forecasting depends heavily on effectively modelling these multi-scale dynamics.

dengColaGNNCrosslocationAttention2020a introduce the idea of multi-scale dilated convolutional with the same filter and stride sides but different dilation rate, which xie2022epignn improved on by making use of the multi-scale convolution to capture features. Building on this, the MTFM employs parallel dilated convolutional layers to efficiently capture temporal dependencies across multiple scales using the output from the AGAM. This approach enables the model to maintain an awareness of both immediate and distant temporal relationships whilst controlling parameter count and computational complexity.

1) Dilated Convolutions for Multi-scale Processing: The core of our MTFM is a set of parallel convolutional branches operating at different dilation rates. For a given input feature tensor $\mathbf{G} \in \mathbb{R}^{B \times N \times d_{\text{hidden}}}$ (where B is the batch size, N is the number of regions, and d_{hidden} is the hidden dimension), we first transpose the tensor to prepare for 1D convolutions along the temporal dimension:

$$\mathbf{G}_{\text{conv}} = \text{Transpose}(\mathbf{G}, 1, 2) \tag{27}$$

resulting in a tensor of shape $[B, d_{\text{hidden}}, N]$. We then process this tensor through S parallel branches, each consisting of a dilated convolutional layer with a specific dilation rate, followed by batch normalisation, ReLU activation and dropout:

$$\mathbf{H}^{(i)} = \text{Dropout}(\text{ReLU}(\text{BatchNorm}(\text{Conv1D}(\mathbf{G}_{\text{conv}}; k, d^{(i)})))) \tag{28}$$

where $i \in \{1, 2, \ldots, S\}$ indexes the scale, k is the kernel size (set to 3 by default), and $d^{(i)} = 2^{i-1}$ is the dilation

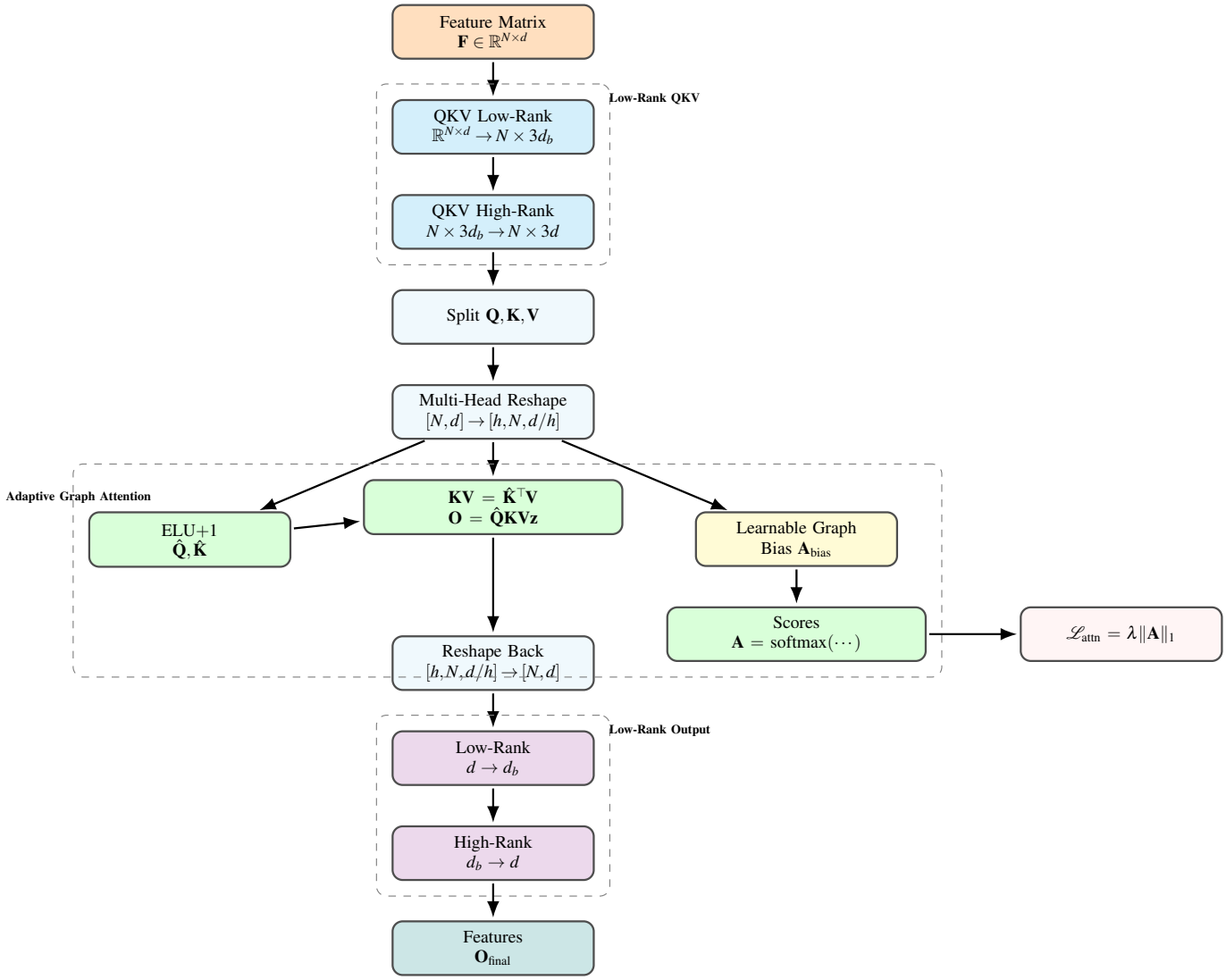


Fig. 3: Flow of data and structure in the AGAM module.

rate for the scale i. Each branch produces an output tensor $\mathbf{H}^{(i)} \in \mathbb{R}^{B \times d_{\text{hidden}} \times N}$.

The increasing dilation rates create an exponentially expanding receptive field across the branches:

- Scale 1 ($d^{(1)} = 1$): Captures immediate temporal dependencies with a receptive field of k time steps
- Scale 2 ($d^{(2)} = 2$): Captures medium-range dependencies with a receptive field of $k + (k-1)$ time steps
- Scale 3 ($d^{(3)} = 4$): Captures longer-range dependencies with a receptive field of $k + 3(k-1)$ time steps
- And so on for higher scales

This multi-scale approach allows the model to efficiently capture a wide range of temporal dependencies without requiring deep sequential processing, which is particularly advantageous for epidemic time-series that often exhibit both rapid changes and gradual trends.

2) Adaptive Scale Fusion: Rather than simply concatenating or averaging the outputs from different scales, we implement an adaptive fusion mechanism that allows the model to learn the relative importance of each temporal scale. This is achieved through learnable fusion weights:

$$\alpha = \text{softmax}(\mathbf{w}) \tag{29}$$

where $\mathbf{w} \in \mathbb{R}^S$ is a learnable parameter vector and $\alpha \in \mathbb{R}^S$ represents the normalised importance weights for each scale.

The multi-scale features are then fused using these weights:

$$\mathbf{H}_{\text{fused}} = \sum_{i=1}^{S} \alpha_i \mathbf{H}^{(i)} \tag{30}$$

where $\mathbf{H}_{\text{fused}} \in \mathbb{R}^{B \times d_{\text{hidden}} \times N}$ is the scale-fused feature representation.

This adaptive fusion mechanism offers several advantages. It allows the model to automatically adjust the importance of different temporal scales based on the data, can adapt to different regions that might exhibit varying temporal characteristics, and provides interpretable insights into which temporal scales are most relevant for forecasting.

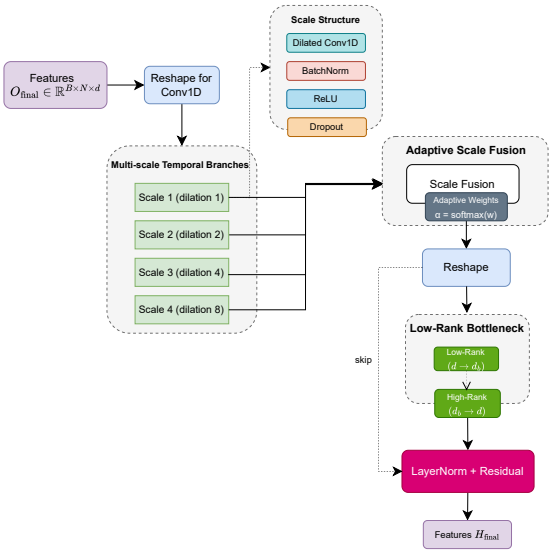


Fig. 4: Flow of data in the MTFM architecture.

3) Bottleneck Projection and Residual Connection: To enhance training stability and allow for more effective feature transformation, we apply a low-rank bottleneck projection to the fused features:

$$\mathbf{H}_{\text{low}} = \text{Linear}_{\text{fusion\_low}}(\text{Transpose}(\mathbf{H}_{\text{fused}}, 1, 2)) \tag{31}$$

$$\mathbf{H}_{\text{proj}} = \text{Linear}_{\text{fusion\_high}}(\mathbf{H}_{\text{low}}) \tag{32}$$

where $\mathbf{H}_{\text{low}} \in \mathbb{R}^{B\times N\times d_{\text{bottle}}}$ is the bottleneck representation with dimension $d_{\text{bottle}} \ll d_{\text{hidden}}$, and $\mathbf{H}_{\text{proj}} \in \mathbb{R}^{B\times N\times d_{\text{hidden}}}$ is the projected representation.

We then apply layer normalisation and a residual connection to facilitate gradient flow during training:

$$\mathbf{H}_{\text{final}} = \text{LayerNorm}(\text{Transpose}(\mathbf{H}_{\text{fused}}, 1, 2) + \mathbf{H}_{\text{proj}}) \tag{33}$$

where $\mathbf{H}_{\text{final}} \in \mathbb{R}^{B\times N\times d_{\text{hidden}}}$ is the final output of the MTFM.

In our implementation, the number of temporal scales S is set to 4 by default, with dilation rates of $\{1, 2, 4, 8\}$. This configuration allows the model to capture dependencies ranging from immediate neighbours to relationships spanning up to 17 time steps ($3 + 7 \times 2 = 17$ with kernel size 3), which is sufficient for most epidemic forecasting applications given our sliding-window approach.

The kernel size k is set to 3, providing a good balance between capturing local patterns and maintaining computational efficiency. The hidden dimension d_{hidden} is maintained throughout the module to preserve the information capacity, while the bottleneck dimension d_{bottle} is set to $d_{\text{hidden}}/4$ to reduce the parameters in the projection layers.

To avoid overfitting, we apply dropout with probability 0.355 after each convolutional layer and ReLU activation. This relatively high dropout rate was determined by cross-validation to be effective for epidemic data, which can be noisy and prone to overfitting. The Figure 4 presents a detailed representation of the architecture of this module.

E. Progressive Multi-step Prediction Refinement

The final component of our MSAGAT-Net architecture is the Progressive Refinement Multi-step Prediction Module (PPRM), which addresses the critical challenge of generating accurate forecasts across multiple future time steps. Whilst the preceding modules excel at extracting spatio-temporal features, converting these features into reliable predictions, particularly for longer horizons, requires additional consideration of how prediction errors can compound over time and how recent observations might inform future trajectory adjustments.

The PPRM addresses the critical issue of error accumulation inherent in multi-horizon epidemiological forecasting. Through an extensive analysis of multistep forecasting failures, it became evident that prediction errors compound exponentially with increasing prediction horizons BENTAIEB20127067, chandra2021evaluation. This module incorporates concepts from residual error correction, ensemble learning, and adaptive gating mechanisms common in recurrent neural network architectures hochreiter1997long.

The PPRM takes the rich spatio-temporal representations from previous modules and transforms them into horizon-specific predictions, incorporating an adaptive refinement mechanism that balances model-based forecasts with data-driven trends. This design is motivated by epidemiological observations that disease progression often follows certain patterns based on a recent trajectory, even as it responds to various complex factors captured by our neural network components.

1) Low-Rank Prediction Projection: Given the spatio-temporal feature tensor $\mathbf{H}_{\text{final}} \in \mathbb{R}^{B\times N\times d_{\text{hidden}}}$ from the multi-scale Fusion Module, where B is the batch size, N is the number of regions, and d_{hidden} is the hidden dimension, we first apply a bottleneck projection to distil the most forecast-relevant information:

$$\mathbf{P}_{\text{low}} = \text{Linear}_{\text{pred\_low}}(\mathbf{H}_{\text{final}}) \tag{34}$$

where $\mathbf{P}_{\text{low}} \in \mathbb{R}^{B\times N\times d_{\text{bottle}}}$ is the bottleneck representation with dimension $d_{\text{bottle}} \ll d_{\text{hidden}}$.

This bottleneck projection serves multiple purposes:

1) It reduces the parameter count in the subsequent prediction layers
2) It forces the model to identify the most salient features for forecasting
3) It acts as an implicit regulariser, helping to prevent overfitting

We then apply layer normalisation, ReLU activation, and dropout to the bottleneck representation:

$$\mathbf{P}_{\text{mid}} = \text{Dropout}(\text{ReLU}(\text{LayerNorm}(\mathbf{P}_{\text{low}}))) \tag{35}$$

This intermediate processing enhances training stability and introduces non-linearity necessary for modelling complex forecast patterns.

2) Horizon-Specific Prediction: From the processed bottleneck representation, we generate initial predictions for all forecast horizons using a linear projection:

$$\mathbf{P}_{\text{initial}} = \text{Linear}_{\text{pred\_high}}(\mathbf{P}_{\text{mid}}) \tag{36}$$

where $\mathbf{P}_{\text{initial}} \in \mathbb{R}^{B\times N\times h}$ represents the raw model predictions for each region in all forecast horizons h.

This approach generates predictions for all horizons simultaneously rather than autoregressively, which offers several advantages:

1) It avoids the compounding error problem of sequential prediction
2) It allows the model to learn horizon-specific patterns directly
3) It enables more efficient training and inference

However, we recognise that different forecast horizons may require different prediction strategies—near-term forecasts might benefit more from recent observations, whilst longer-term forecasts might rely more heavily on learnt patterns. Our architecture addresses this through an adaptive refinement mechanism.

3) Adaptive Refinement Mechanism: A key innovation in our PPRM is the adaptive refinement gate, which dynamically balances model-based predictions with trend-based extrapolations conditioned on the most recent observations. This mechanism is particularly valuable in the prediction of epidemics, where the recent trajectory often provides strong signals about the short-term future progression.

We first compute an adaptive gate based on the spatio-temporal features:

$$\mathbf{G} = \sigma(\text{Linear}_{\text{gate\_high}}(\text{ReLU}(\text{Linear}_{\text{gate\_low}}(\mathbf{H}_{\text{final}})))) \tag{37}$$

where $\mathbf{G} \in \mathbb{R}^{B\times N\times h}$ represents gate values between 0 and 1 for each region and forecast horizon, and σ denotes the sigmoid activation function.

Currently, we use the most recent observation $\mathbf{x}_{\text{last}} \in \mathbb{R}^{B\times N}$ to generate a trend-based forecast using an exponential decay projection:

$$\mathbf{T} = \mathbf{x}_{\text{last}} \odot \exp(-\gamma \cdot \mathbf{d}) \tag{38}$$

where $\mathbf{x}_{\text{last}}$ is expanded to the shape $[B,N,h]$, $\mathbf{d} \in \mathbb{R}^h$ is a vector of increasing horizon indices $[1,2,\ldots,h]$, γ is a decay factor (set to 0.1 in our implementation), and $\odot$ represents element-wise multiplication.

This exponential decay formulation is inspired by epidemiological models that exhibit exponential growth or decay patterns, providing a simple yet effective baseline that captures the natural progression tendencies of epidemic time-series.

The final predictions are then computed as a weighted combination of the model-based predictions and the trend-based projections:

$$\mathbf{P}_{\text{final}} = \mathbf{G} \odot \mathbf{P}_{\text{initial}} + (1-\mathbf{G}) \odot \mathbf{T} \tag{39}$$

where $\mathbf{P}_{\text{final}} \in \mathbb{R}^{B\times N\times h}$ represents the refined predictions for each region across all forecast horizons.

This adaptive gating mechanism offers several important advantages for epidemic forecasting. By dynamically determining the balance between model-based predictions and trend-based projections, the gate enables the model to rely more heavily on recent trends for near-term forecasts while increasingly leveraging learnt patterns for longer-term predictions. This approach provides inherent robustness against model errors by incorporating a simple, interpretable baseline that can compensate when the deep learning component encounters unfamiliar patterns. Furthermore, the mechanism adapts dynamically to different regions and temporal contexts, recognising that the optimal balance between model predictions and trend extrapolation may vary based on local epidemiological characteristics and data quality. Perhaps most importantly, this design improves forecast stability by creating a smooth transition between recent observations and model predictions, avoiding the discontinuities that often plague multi-step forecasting approaches and enhancing the practical utility of the predictions for healthcare resource planning.

In our implementation, the bottleneck dimension d_{bottle} is set to 8, which we determined through ablation studies to provide an optimal balance between parameter efficiency and predictive performance. The horizon length h is configurable based on the specific forecasting task requirements; in our experiments, we mainly use $h = 5$ to predict forecasts 5 days in advance, though the architecture supports arbitrary horizon lengths.

The decay factor γ in the exponential projection is set to 0.1, which provides a moderate decay rate appropriate for the typical progression of the epidemic. This value was selected based on empirical analysis of epidemic curves in our datasets and can be adjusted based on the specific characteristics of the target epidemic.

A dropout rate of 0.355 is applied in the prediction pathway to prevent overfitting, which is particularly important for the final prediction layers that directly influence the model output. This relatively high dropout rate was determined by cross-validation to be effective for the noisy and often irregular nature of epidemic data.

The figure 5 illustrates the flow of data through the PPRM module. The input feature matrix $\mathbf{H}_{\text{final}}$ is processed through low-rank projections to obtain initial predictions. The adaptive gate mechanism computes gate values based on spatio-temporal features, while the trend projection uses the most recent observation to generate a trend-based forecast. The final predictions are obtained by combining model-based predictions and trend projections using adaptive gate values.

IV. Experiments and Analysis

This section presents a comprehensive evaluation of our proposed MSAGAT-Net model across multiple epidemic datasets with varying characteristics. We compare MSAGAT-Net against state-of-the-art baseline models to assess its effectiveness in capturing complex spatio-temporal dynamics and generating accurate multi-horizon forecasts. The evaluation encompasses both traditional influenza datasets and more recent COVID-19 datasets, enabling us to test the model's

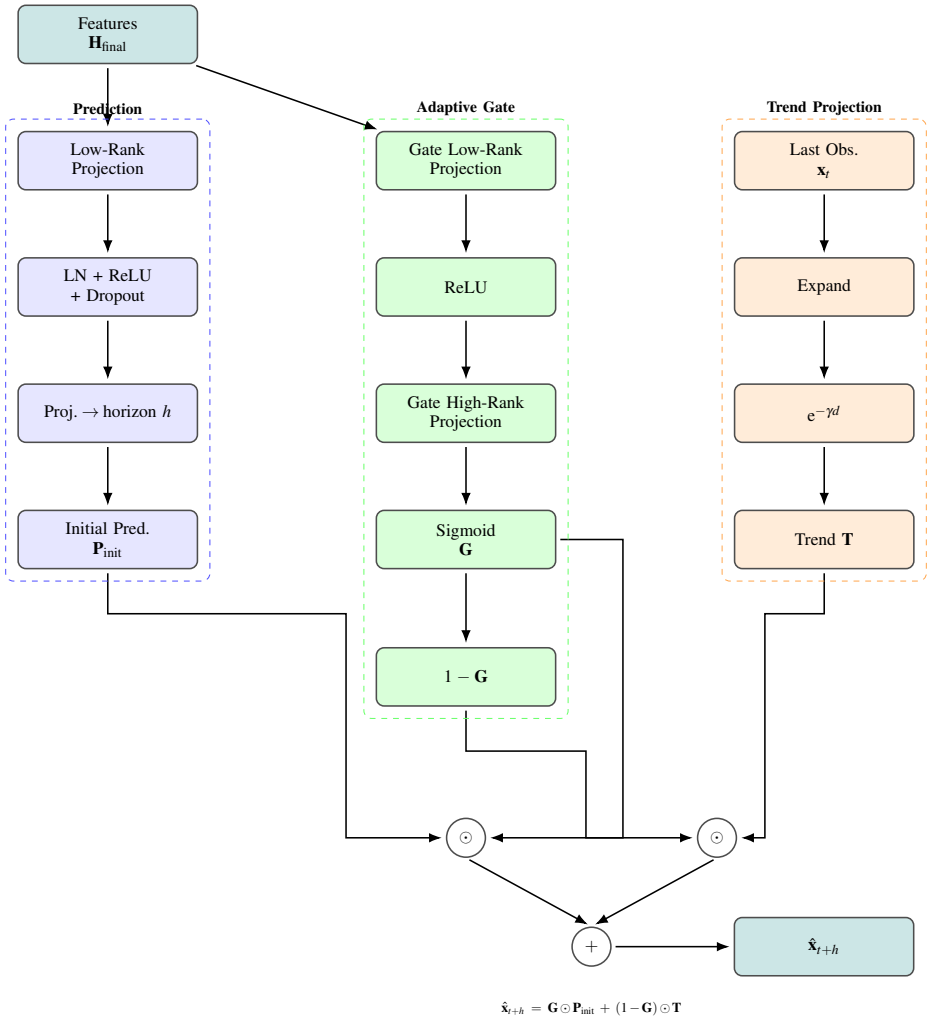


Fig. 5: The flow of data in the PPRM architecture

versatility and generalisation capabilities across different epidemic scenarios. We evaluated the models using multiple metrics, including Root Mean Square Error (RMSE) and Pearson Correlation Coefficient (PCC), Mean Absolute Error (MAE) and R^2 score, to provide a comprehensive assessment of their performance. In the results tables presented, the best performance is typically highlighted in **bold**, while the second-best performance is underlined.

A. Experimental Setup

All experiments were conducted on the same high performance computing (HPC) cluster equipped with NVIDIA RTX 8000 GPUs to ensure consistent hardware conditions in all model evaluations. This controlled environment allows for a fair comparison between different approaches and eliminates potential variations due to hardware differences.

B. Datasets

To comprehensively evaluate the performance and generalisability of our proposed MSAGAT-Net framework, we performed experiments on several real-world epidemic datasets spanning various geographical regions, time periods, and disease types. This approach enables a thorough assessment of the model's versatility and robustness across varying spatio-temporal characteristics and epidemic scenarios.

Our experimental evaluation encompasses seven distinct datasets, each offering unique challenges and characteristics for epidemic forecasting. These datasets represent different geographical scales (from local authorities to national regions), temporal resolutions (daily and weekly measurements), and disease contexts (seasonal influenza and COVID-19). Table I provides a statistical overview of these datasets, summarising their key characteristics and numerical properties.

1) Influenza Datasets: We utilised three established influenza datasets from different regions to evaluate our model's performance on seasonal patterns:

- **Japan-Prefecture Dataset:** This dataset is derived from the Infectious Disease Weekly Report (IDWR) published by the Japanese government[1]. It comprises weekly statistics of Influenza-Like Illness (ILI) cases from August 2012 to March 2019 in all 47 prefectures in Japan.

[1]https://tinyurl.com/y5dt7stm

TABLE I: Overview of the epidemic datasets used in our experimental evaluation. "Granularity" indicates the temporal resolution of the epidemic data, whilst "Size" represents the product of the number of locations and the number of time steps.

Dataset	Size	Min	Max	Mean	Granularity
Japan-Prefecture	348×47	0	26,635	655	Weekly
US-Region	785×10	0	16,526	1,009	Weekly
US-State	360×49	0	9,716	223	Weekly
Spain-COVID	122×35	0	4,623	38	Daily
Australia-COVID	556×8	0	9,987	539	Daily
LTLA-COVID	839×372	0	4,170	85	Daily
NHS-ICUBeds	895×7	0	1,215	102	Daily

- **US-Region Dataset:** Extracted from the ILINet surveillance system maintained by the US Health and Human Services (US-HHS)[2], this dataset includes weekly influenza activity levels in ten HHS regions across the continental United States from 2002 to 2017.
- **US-State Dataset:** Obtained from the Centres for Disease Control and Prevention (CDC), this dataset consists of weekly numbers of visits to healthcare providers with influenza-like illnesses from 2010 to 2017 for 49 states in the US (one state was excluded due to incomplete data).

2) COVID-19 Datasets: To assess the adaptability of our model to new epidemic scenarios, we incorporated four COVID-19 datasets that span different countries and healthcare metrics:

- **Spain-COVID Dataset:** This dataset encompasses daily COVID-19 case data from 20 February 2020 to 20 June 2020 for 35 administrative NUTS3 regions in Spain significantly affected by the first wave of the pandemic.
- **Australia-COVID Dataset:** Compiled from the Johns Hopkins University Centre for Systems Science and Engineering (JHU-CSSE) repository, this dataset contains daily new confirmed cases of COVID-19 from 27 January 2020 to 4 August 2021 across all eight Australian jurisdictions (six states and two territories).
- **LTLA-COVID Dataset:** Derived from the UK Health Security Agency[3], this dataset contains daily data from COVID-19 cases from March 2020 to February 2022 for 372 Lower-Tier Local Authority districts in England.
- **NHS-ICUBeds Dataset:** Obtained from the National Health Service (NHS) EnglandNHS2024HospitalActivity, this dataset provides daily counts of occupied mechanical ventilator beds in seven regions of the NHS from March 2020 to February 2022. Unlike the other datasets that focus on case counts, this dataset offers an opportunity to evaluate the model's capability to predict healthcare resource utilisation, which is critical for effective epidemic response and management.

A critical aspect of our approach involves the construction of an appropriate graph structure to represent spatial relationships between regions. For our implementation, we constructed the adjacency matrix based on geographic proximity, using the Haversine formula to calculate the great circle distance between regions. Two regions are considered connected if the distance between them falls below a threshold $d_{\text{threshold}}$ (set to 150 km in our experiments):

$$a_{ij} = \begin{cases} 1, & \text{if Haversine}(\text{region}_i, \text{region}_j) \leq d_{\text{threshold}} \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

This threshold-based connectivity captures the intuition that epidemic spread is influenced by the movement of people between nearby regions. Whilst more sophisticated connectivity measures could be employed, this approach provides a straightforward and interpretable baseline for spatial relationship modelling. The noise in the dataset was smoothed using the rolling mean of 7 days established in previous studies ajao2023deep, oluwasakin2023data, zeroual2020deep, Kamalov2022ReviewDL, and normalisation was performed to ensure that the data are on a similar scale in different regions.

The diverse nature of these datasets, spanning different geographic regions, temporal resolutions, and epidemic contexts, allows us to comprehensively evaluate the performance and generalisability of our proposed MSAGAT-Net model across a range of epidemic forecasting scenarios.

C. Model Optimisation

The MSAGAT-Net model is trained using the AdamW optimiser with a learning rate of 1×10^{-3}, which was determined by cross-validation to provide optimal convergence speed and stability. The model is trained for a maximum of 1500 epochs, with early stopping criteria based on validation loss to prevent overfitting. The training process is monitored using a patience parameter of 100 epochs, which means that if the validation loss does not improve for 100 consecutive epochs, the training will be stopped.

The loss function for the MSAGAT-Net model is a combination of prediction error and regularisation terms:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{pred}} + \lambda_{\text{attn}} \mathscr{L}_{\text{attn}} + \lambda_{\text{l2}} \|\Theta\|_2 \tag{41}$$

where:

- $\mathscr{L}_{\text{pred}}$ is the mean squared error (MSE) measuring discrepancies between the model predictions and the observed data.
- $\mathscr{L}_{\text{attn}}$ represents the attention regularisation term that enforces sparsity and interpretability in spatial relationships.
- The hyperparameters $\lambda_{\text{attn}} = 10^{-4}$ and $\lambda_{\text{l2}} = 5 \times 10^{-4}$ control the strength of regularisation.

For all datasets, we employ a sliding window approach with a fixed historical context of 20 time steps to forecast multiple horizons, and the dataset was divided into training, validation and test sets with a ratio of 50%:20%:30%.

The training algorithm for the MSAGAT-Net model is formalised in Algorithm 1, which outlines the step-by-step procedure to optimise model parameters through backpropagation and early stopping.

[2]https://tinyurl.com/y39tog3h

[3]https://ukhsa-dashboard.data.gov.uk/respiratory-viruses/covid-19

Algorithm 1: MSAGAT-Net Training Algorithm

Input: Data $\mathbf{X} \in \mathbb{R}^{N \times T}$, adjacency $\mathbf{A} \in \mathbb{R}^{N \times N}$, hyperparameters
Output: Trained parameters $\blacksquare^*$
Initialize model parameters, optimizer, scheduler;
$L^*_{\text{val}} \leftarrow \infty$, $p \leftarrow 0$;
for *epoch* $e = 1$ **to** $E_{\max}$ **do**
  foreach *batch* $(\mathbf{X}_b, \mathbf{y}_b)$ *in training set* **do**
    $\mathbf{F} \leftarrow \emptyset$;
    for *region* $i \in \{1, \ldots, N\}$ **do**
      Extract features via depthwise separable convolutions;
      $\mathbf{F} \leftarrow \mathbf{F} \cup \{\mathbf{F}^i\}$;
    $\mathbf{F} \leftarrow \text{ReLU}(\text{LayerNorm}(\mathbf{F}))$;
    $\mathbf{O}, L_{\text{attn}} \leftarrow \text{AGAM}(\mathbf{F}, \mathbf{A})$;
    $\mathbf{H} \leftarrow \text{MTFM}(\mathbf{O})$;
    $\hat{\mathbf{y}}_b \leftarrow \text{PPRM}(\mathbf{H}, \mathbf{X}_b[:, -1, :])$;
    $L_b \leftarrow \text{MSE}(\hat{\mathbf{y}}_b, \mathbf{y}_b) + \lambda_{\text{attn}} \cdot L_{\text{attn}}$;
    Update parameters via backpropagation;
  Compute validation loss L_{val};
  Update scheduler;
  if $L_{val} < L^*_{val}$ **then**
    $\blacksquare^* \leftarrow \blacksquare$, $L^*_{\text{val}} \leftarrow L_{\text{val}}$, $p \leftarrow 0$;
  else if $p + 1 \geq P$ **then**
    break;
  else
    $p \leftarrow p + 1$;
return $\blacksquare^*$

D. Baseline Models

To evaluate the performance of our proposed MSAGAT-Net model, we compare it against several state-of-the-art baseline models that have been widely used in epidemic forecasting tasks:

- **DCRNN** li2017diffusion: A diffusion convolution recurrent neural network that integrates graph convolutions with recurrent neural networks in an encoder-decoder architecture to capture both spatial dependencies and temporal dynamics. It models spatial dependencies using a diffusion process on graphs and temporal dependencies through recurrent units.
- **LSTNet** lai2018modeling: A model that combines convolutional neural networks and recurrent neural networks to extract short-term local dependency patterns and discover long-term patterns for time-series trends. It employs a convolutional component to extract local dependency patterns and a recurrent component to capture long-term temporal dependencies.
- **CNNRNN-Res** wu2018deep: A deep learning framework that combines convolutional neural networks, recurrent neural networks, and residual connections to solve epidemiological prediction problems. It uses CNNs to extract spatial features, RNNs to capture temporal dependencies, and residual connections to enhance gradient flow during training.
- **Cola-GNN** dengColaGNNCrosslocationAttention2020a: A graph neural network model that leverages cross-location attention mechanisms to capture dynamic spatial relationships between regions. It employs location-aware attention to model the impact of each region on others, allowing for adaptive and context-dependent spatial dependency learning.
- **EpiGNN** xie2022epignn: A model based on graph neural networks specifically designed for epidemic forecasting. It incorporates a transmission risk encoding module to characterise local and global spatial effects, and features a Region-Aware Graph Learner (RAGL) that considers transmission risk, geographical dependencies, and temporal information to explore spatio-temporal dependencies.

These baselines represent a diverse range of approaches to spatio-temporal forecasting, from traditional time-series models to advanced deep learning architectures that explicitly model spatial and temporal dependencies. By comparing against these models, we aim to assess the relative strengths and weaknesses of our MSAGAT-Net approach and identify its contributions to the state of the art in epidemic forecasting.

E. Results and Discussion

Table **??** presents a comprehensive comparison of our proposed MSAGAT-Net model against state-of-the-art baseline approaches across three influenza datasets (Japan-Prefectures, US-Regions, and US-States) and four forecast horizons (3, 5, 10, and 15 days ahead). Furthermore, Table **??** shows the performance comparison on four COVID-19 datasets (Australia-COVID, LTLA-TimeSeries, NHS-TimeSeries and Spain-COVID) for horizons of 3, 7, and 14 days ahead.

MSAGAT-Net demonstrates consistent and superior performance across the three influenza datasets, particularly for short- and medium-term forecasting horizons. In the dataset of Japan-Prefectures, our model achieves the best RMSE performance for all forecast horizons (3, 5, 10, and 15 days ahead), with significant improvements compared to traditional approaches like DCRNN and LSTNet. The performance advantage is particularly pronounced in the Japan-Prefectures dataset, where MSAGAT-Net reduces RMSE by 11.2% compared to the second best model (Cola-GNN) for 3-day forecasts and by 11.2% compared to the second-best model (EpiGNN) for 15-day forecasts.

In the US-Regions dataset, MSAGAT-Net achieves the best RMSE performance for 10-day forecasts with a value of 999, improving on EpiGNN's 1098 by 9.0%. However, for 3-day, 5-day and 15-day forecasts, EpiGNN shows better performance with RMSE values of 622, 779, and 1076, respectively. This could be attributed to EpiGNN's explicit modelling of transmission risk, which might be particularly effective for the spatial characteristics of the US-Regions dataset. However, MSAGAT-Net achieves the highest PCC values for 3-day and 10-day forecasts (0.911 and 0.763, respectively), indicating its strong ability to capture correlation patterns at these horizons.

For the US-States dataset, EpiGNN outperforms all other models for 3-day, 5-day, and 15-day forecasts, with RMSE

values of 166, 203, and 136, respectively. Cola-GNN shows the best performance for 10-day forecasts with an RMSE of 248, closely followed by MSAGAT-Net at 250. Although MSAGAT-Net does not achieve the lowest RMSE for most horizons in this dataset, it does attain the highest PCC for 3-day forecasts (0.931), demonstrating strong correlation accuracy for short-term predictions. For longer horizons, Cola-GNN shows superior PCC performance for 10-day and 15-day forecasts (0.874 and 0.872, respectively).

In terms of PCC, which measures the correlation between predicted and actual values, MSAGAT-Net shows strong performance in most scenarios, particularly for the dataset from Japan-Prefectures, where it achieves the highest PCC for all forecast horizons (0.885, 0.884, 0.827, and 0.778). This indicates a superior ability to capture trends and patterns across different time scales for this particular dataset.

The performance of MSAGAT-Net on COVID-19 datasets shows more varied results compared to influenza datasets. In the LTLA-Timeseries dataset, MSAGAT-Net achieves the best RMSE for short- and medium-term forecasts (3 days and 7 days), with values of 106 and 163, respectively. For 14-day forecasts, EpiGNN performs slightly better with an RMSE of 184 compared to MSAGAT-Net's 196.

In the Spain-COVID dataset, MSAGAT-Net and Cola-GNN are tied for the best RMSE for 3-day forecasts (135). However, DCRNN significantly outperforms all models for 7-day and 14-day forecasts with RMSE values of 99 and 106, respectively. This suggests that for the specific patterns in the Spain-COVID dataset, the diffusion-based approach of DCRNN might be particularly effective for medium to long-term forecasting.

In the Australia-COVID dataset, MSAGAT-Net shows less competitive performance, with LSTNet achieving significantly better results across all horizons (137, 229, and 294 for 3-day, 7-day and 14-day forecasts, respectively). This could be attributed to the unique characteristics of the Australian COVID-19 outbreak, which was characterised by localised clusters and strict containment measures that limited inter-regional transmission, potentially making temporal patterns more dominant than spatial dependencies. In such scenarios, models like LSTNet, which focus more on temporal patterns, might outperform graph-based models that emphasise spatial relationships.

In the NHS-Timeseries dataset, which represents ICU bed occupancy, Cola-GNN achieves the best performance for short-term forecasts with an RMSE of 4 for 3-day forecasts, followed by MSAGAT-Net with an RMSE of 6. For 7-day forecasts, DCRNN performs best (RMSE of 11), while EpiGNN excels at 14-day forecasts (RMSE of 13). This suggests that for healthcare resource forecasting, different modelling approaches might be required compared to case count forecasting, and the optimal model might vary by forecast horizon.

Comparison of MSAGAT-Net with the two strongest baselines, Cola-GNN and EpiGNN, reveals interesting patterns. In the dataset of Japan-Prefectures, MSAGAT-Net consistently outperforms both models across all metrics and horizons. This suggests that MSAGAT-Net's multi-scale module and adaptive graph attention mechanism provide significant advantages for this particular dataset, which features complex spatio-temporal patterns across Japan's diverse prefectures.

The comparison on the US-Regions and US-States datasets is more nuanced. Although MSAGAT-Net shows superior performance for 10-day forecasts on the US-Regions dataset and competitive performance on the US-States dataset, EpiGNN often performs better for 3-day, 5-day, and 15-day forecasts. This could be attributed to EpiGNN's explicit modelling of transmission risk and its region-aware graph learning approach, which might be particularly effective for capturing both short-term and long-term dependencies in the US epidemic context.

In the COVID-19 datasets, where the epidemic dynamics was highly volatile and influenced by policy interventions, MSAGAT-Net shows competitive but more varied performance compared to baselines. It excels for shorter horizons on datasets like LTLA-Timeseries but is outperformed by other models on datasets like Australia-COVID. This highlights the challenge of generalising epidemic forecasting models across different disease contexts and the need for adaptive modelling approaches that can handle diverse epidemic scenarios.

A critical observation from the results is the robustness of MSAGAT-Net's ability to forecast the horizon extension, especially on the Japan-Prefectures dataset. Although all models exhibit performance degradation as the forecast horizon increases (a common challenge in spatio-temporal forecasting), MSAGAT-Net demonstrates superior resilience to this degradation on this dataset.

For instance, on the Japan-Prefectures dataset, MSAGAT-Net's RMSE increases by only 28.0% from 3-day to 15-day forecasts (from 1045 to 1338), compared to Cola-GNN's 50.5% increase (from 1177 to 1771). This enhanced stability across forecast horizons can be attributed to MSAGAT-Net's Progressive Prediction Module, which employs an adaptive fusion of model predictions and exponential decay to mitigate error accumulation in long-horizon forecasts.

The performance advantage of MSAGAT-Net is most consistent in the Japan-Prefectures dataset, where it outperforms all other models in all metrics and horizons. Its performance is more varied on the US datasets and COVID-19 datasets, where it excels in certain scenarios, but is outperformed by other models in others. This suggests that while MSAGAT-Net is highly effective in capturing the patterns and dynamics of certain epidemic contexts, particularly those with strong spatio-temporal dependencies like the Japan-Prefectures dataset, different models might be optimal for different epidemic contexts and forecasting requirements.

The figure 6 illustrates the attention matrices learnt by MSAGAT-Net on the Japan-Prefectures dataset for a 5-day forecast horizon. The attention weights reflect the model's focus on different regions and their relationships, providing insights into how the model captures spatial dependencies in the data. The attention matrices reveal that the model learns to focus on nearby prefectures, indicating that local transmission dynamics play a significant role in epidemic spread.

The observed variations in performance across datasets highlight the complexity of epidemic forecasting and un-

derscore the importance of model selection based on the specific characteristics of the epidemic and the forecasting requirements. They also point to potential directions for future research, such as developing more adaptive modelling approaches that can dynamically adapt to changing epidemic dynamics and incorporate exogenous factors such as policy interventions and behavioural changes.

F. Ablation Study of Model Components

We performed in-depth ablation studies across various datasets and forecast horizons to systematically assess each architectural component's contribution to the overall performance of MSAGAT-Net. This rigorous analysis provides valuable information on how each component affects the accuracy of forecasting in different epidemic contexts and time scales, helping to validate our architectural design choices and identify opportunities for context-specific optimisations.

We focused our primary experiments on two representative datasets: the Japan-Prefectures dataset (seasonal influenza) and the LTLA-Timeseries dataset (COVID-19), using a fixed historical context window of 20 time steps while varying the forecast horizon.

The ablation experiments were designed to isolate the effects of three key architectural components of MSAGAT-Net. We created three variants of the model for comparison:

- **Without AGAM**: Replacing the Efficient Adaptive Graph Attention Module with a standard graph convolutional layer that lacks the adaptive attention mechanisms.
- **Without MTFM**: Replacing the Dilated Multi-scale Module with a single-scale temporal convolutional layer with fixed kernel size and dilation rate.
- **Without PPRM**: Replacing the Progressive Prediction Module with direct multi-step prediction without the adaptive refinement mechanism.

All model variants were trained using identical hyperparameters, optimisation procedures, and data splits to ensure fair comparison. We evaluated performance using four complementary metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson Correlation Coefficient (PCC) and the coefficient of determination (R^2).

Tables IV and V present the comprehensive results of our ablation experiments. The values in parentheses indicate the percentage change relative to the full model, with positive values for error metrics (MAE, RMSE) indicating performance degradation and negative values indicating performance improvement. For correlation metrics (PCC, R^2), positive values indicate improvement and negative values indicate degradation.

Our ablation study reveals striking differences in the importance of components between the influenza data sets (Japan-Prefectures) and COVID-19 (LTLA-Timeseries), as well as between different forecast horizons, as visualised in Figure 7.

1) Impact of AGAM Removal: The Adaptive Graph Attention Module shows dramatically different patterns of importance across the two datasets:

For the dataset of Japan-Prefectures, the removal of AGAM consistently degrades the performance of RMSE in the short term (3 days: +5. 28%) and medium term forecasts (5 days: +2. 48%), with a particularly dramatic impact for longer term forecasts (10 days: +23. 12% RMSE, -22. 54% PCC, -38. 14% R2), as clearly visible in Figure 7c. This escalating pattern suggests that spatial dependencies become increasingly crucial for longer-term influenza forecasting, likely reflecting the complex spatial transmission dynamics of seasonal influenza across Japan's diverse prefectures.

In stark contrast, for the LTLA-Timeseries dataset (COVID-19), the removal of AGAM unexpectedly improves the performance for short- and medium-term forecasts (3-day: -1.98% RMSE; 7-day: -1.20% RMSE), while only becoming detrimental for long-term predictions (14 days: +19. 81% RMSE, -191. 99% R2). This catastrophic collapse in the R^2 value for 14-day forecasts (from positive to -0.170) indicates that without adaptive spatial modelling, the model loses all ability to explain the variance in long-term COVID-19 patterns. This suggests that while simpler spatial modelling may be sufficient for short-term COVID-19 forecasting in the UK context, adaptive attention becomes absolutely essential for longer horizons, possibly due to the emergence of complex spatial patterns in COVID-19 transmission over extended time periods.

The figure 8 illustrates the attention matrices learnt by MSAGAT-Net on the Japan-Prefectures dataset for a 5-day forecast horizon without the AGAM module. Also figure 9 presents a comparison of RMSE values across different model variants (including the full MSAGAT-Net and its ablated versions) for selected datasets, illustrating how prediction error changes with increasing forecast horizons.

These contrasting patterns highlight the disease-specific nature of spatial dependency modelling requirements and challenge the assumption that more complex spatial modelling always leads to better forecasts.

2) Impact of MTFM Removal: The multi-scale Fusion Module shows surprisingly consistent effects across both datasets, with its contribution being minimal or even slightly negative in most scenarios:

For the dataset of Japan-Prefectures, the removal of MTFM leads to only slight changes in RMSE across all horizons (3-day: +1.57%; 5-day: -0.87%; 10-day: +0.68%), with negligible impacts on correlation metrics, as evidenced in all panels of Figure 7. More notably, removal of MTFM actually improves MAE across multiple horizons (3-day: -2.70%; 5-day: -6.94%), suggesting that simpler temporal modelling might reduce absolute error in some cases. For the 15-day horizon (Figure 7d), removing MTFM appears to noticeably improve performance, suggesting that simpler temporal processing may be advantageous for very long-term forecasts.

Similarly, for the LTLA-Timeseries dataset, MTFM removal consistently results in slight RMSE improvements (3-day: -0.65%; 7-day: -1.03%; 14-day: -0.62%), with notable gains in correlation metrics for longer horizons (14-day: +10.85% PCC, +5.48% R^2).

These consistent findings across two very different epidemiological contexts suggest that complex multi-scale processing may be less critical than initially hypothesised for epidemic forecasting. The relatively stable temporal patterns of disease

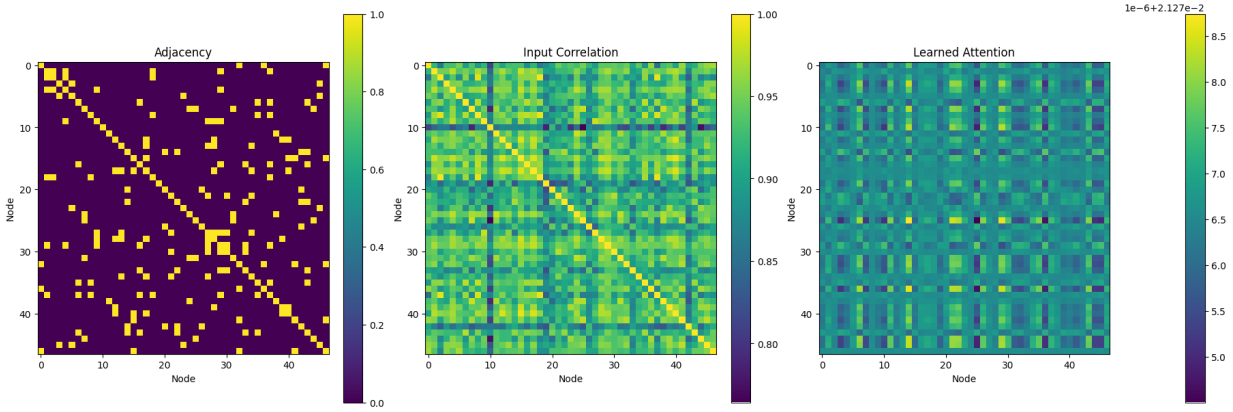


Fig. 6: Adjacencies and learnt attention matrices of MSAGAT-Net on the Japan-Prefectures dataset for 5-day horizon.

TABLE IV: Ablation study results on Japan-Prefectures dataset with window size 20 across different forecast horizons. Values in parentheses indicate percentage change relative to the full model.

Model Variant	Metric	3-day Horizon		5-day Horizon		10-day Horizon	
		Value	% Change	Value	% Change	Value	% Change
Without AGAM	MAE	328.57	(+1.30%)	388.46	(-0.86%)	613.53	(+32.77%)
	RMSE	1100.40	(+5.28%)	1114.42	(+2.48%)	1647.20	(+23.12%)
	PCC	0.876	(-1.03%)	0.874	(-1.21%)	0.641	(-22.54%)
	R^2	0.712	(-3.80%)	0.705	(-1.96%)	0.356	(-38.14%)
Without MTFM	MAE	315.59	(-2.70%)	364.63	(-6.94%)	470.36	(+1.79%)
	RMSE	1061.58	(+1.57%)	1078.05	(-0.87%)	1347.03	(+0.68%)
	PCC	0.890	(+0.51%)	0.885	(-0.02%)	0.818	(-1.06%)
	R^2	0.732	(-1.11%)	0.724	(+0.67%)	0.569	(-1.01%)
Without PPRM	MAE	348.91	(+7.57%)	339.21	(-13.43%)	445.68	(-3.55%)
	RMSE	1074.59	(+2.81%)	1076.47	(-1.01%)	1289.66	(-3.60%)
	PCC	0.904	(+2.15%)	0.898	(+1.43%)	0.851	(+2.95%)
	R^2	0.726	(-2.00%)	0.725	(+0.79%)	0.605	(+5.23%)

transmission might be adequately captured by simpler temporal models, and the additional complexity of multi-scale processing might introduce noise or lead to overfitting in some cases.

3) Impact of PPRM Removal: The Progressive Prediction Refinement Module reveals the most dramatic and divergent effects between the two datasets:

PPRM removal degrades performance for short-term forecasts (3-day: +2.81% RMSE, +7.57% MAE), as shown in Figure 7a, but surprisingly improves performance for medium- and longer-term horizons (5-day: -1.01% RMSE, -13.43% MAE; 10-day: -3.60% RMSE, -3.55% MAE), which is visible in Figures 7b and 7c. In particular, PPRM removal consistently improves correlation metrics across all horizons (PCC: +2.15%, +1.43%, +2.95%), suggesting that direct prediction better captures the underlying trends even when it occasionally increases the absolute error. This pattern indicates that progressive refinement helps with immediate influenza forecasts, but direct prediction becomes more effective for longer horizons in this context for the Japan datatset.

For the LTLA-Timeseries dataset, PPRM removal catastrophically degrades performance across all horizons, with particularly severe impacts on shorter forecasts (3-day: +67.23% RMSE, -56.52% R^2; 7-day: +18.02% RMSE, -51.30% R^2; 14-day: +5.03% RMSE, -45.50% R^2). This dramatic contrast with the Japan results indicates that progressive prediction is absolutely essential for COVID-19 forecasting in the UK context. The consistent pattern of diminishing importance of PPRM as the horizon increases (67. 23% 18. 02% 5. 03% for RMSE degradation) suggests that although progressive prediction remains beneficial for all COVID-19 forecasts, its relative importance decreases for longer horizons, potentially due to the inherent unpredictability of long-term COVID-19 patterns regardless of prediction strategy.

This striking divergence in the importance of PPRM probably reflects fundamental differences in the predictability and volatility of the influenza versus COVID-19 transmission dynamics. Seasonal influenza follows more established patterns that might be amenable to direct prediction, especially for medium and longer horizons, while COVID-19 transmission

TABLE V: Ablation study results on LTLA-Timeseries dataset (COVID-19) with window size 20 across different forecast horizons. Values in parentheses indicate percentage change relative to the full model.

Model Variant	Metric	3-day Horizon		7-day Horizon		14-day Horizon	
		Value	% Change	Value	% Change	Value	% Change
Without AGAM	MAE	48.10	(+1.02%)	79.22	(-4.73%)	136.12	(+27.91%)
	RMSE	104.14	(-1.98%)	161.47	(-1.20%)	234.92	(+19.81%)
	PCC	0.910	(+0.14%)	0.760	(+4.34%)	0.564	(+9.20%)
	R^2	0.770	(+1.23%)	0.447	(+3.13%)	-0.170	(-191.99%)
Without MTFM	MAE	47.34	(-0.59%)	81.21	(-2.34%)	100.30	(-5.75%)
	RMSE	105.55	(-0.65%)	161.76	(-1.03%)	194.86	(-0.62%)
	PCC	0.910	(+0.14%)	0.742	(+1.81%)	0.573	(+10.85%)
	R^2	0.764	(+0.41%)	0.445	(+2.67%)	0.195	(+5.48%)
Without PPRM	MAE	82.02	(+72.25%)	97.20	(+16.90%)	112.40	(+5.62%)
	RMSE	177.67	(+67.23%)	192.89	(+18.02%)	205.95	(+5.03%)
	PCC	0.731	(-19.57%)	0.643	(-11.78%)	0.480	(-7.15%)
	R^2	0.331	(-56.52%)	0.211	(-51.30%)	0.101	(-45.50%)

during the study period was characterised by rapid changes due to policy interventions and emerging variants, necessitating a more adaptive progressive refinement approach.

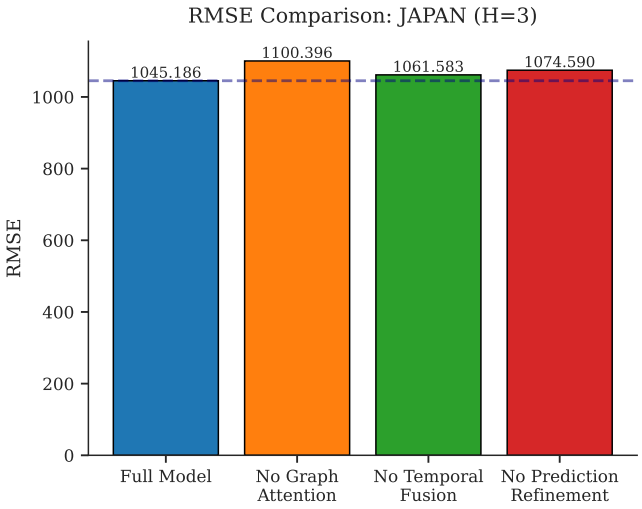
A particularly intriguing finding from our ablation study is the non-linear relationship between component importance and forecast horizon. Rather than the importance of the components increasing or decreasing monotonically with the length of the horizon, we observe complex patterns that challenge simplistic assumptions, as clearly illustrated in the different panels of Figure 7. The figure

For the dataset of Japan-Prefectures, the importance of AGAM increases dramatically from 5-day (+2.48% RMSE when removed) to 10-day (+23.12%), suggesting a critical threshold where spatial modelling becomes essential. However, data from the 15-day horizon (Figure 7d) reveal that the importance of AGAM unexpectedly decreases again (+0.98% RMSE), indicating a non-monotonic relationship. Similarly, MTFM removal becomes increasingly beneficial as the horizon extends to 15 days (-3.70% RMSE), contrary to the typical assumption that temporal modelling becomes more important for longer horizons.

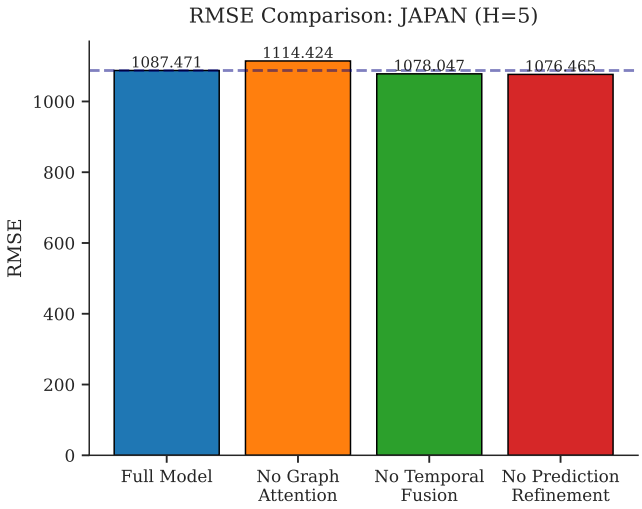
For the LTLA-Timeseries dataset, PPRM importance follows a clear diminishing pattern with horizon length, while AGAM importance follows the opposite trajectory, becoming dramatically more important for longer horizons. This suggests a potential "crossover point" where the relative importance of prediction strategy versus spatial modelling reverses as the forecast horizon extends.

The component importance heatmap in Figure 13 summarises the relative importance of each component in all datasets and forecast horizons. The heatmap shows the percentage importance of the component ablation study of the japan influeza dataset.

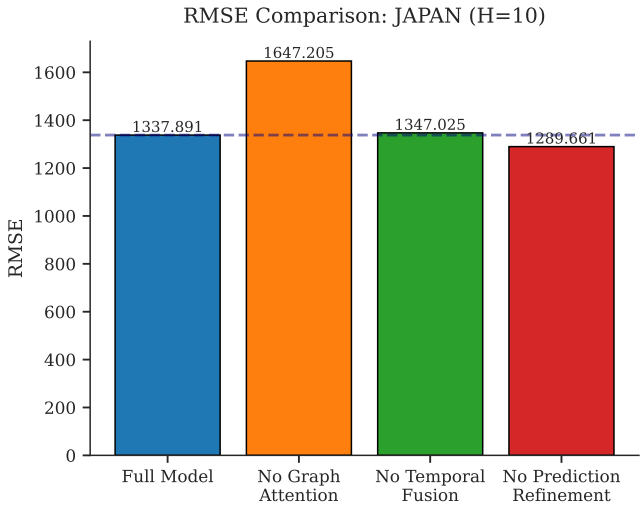
REFERENCES



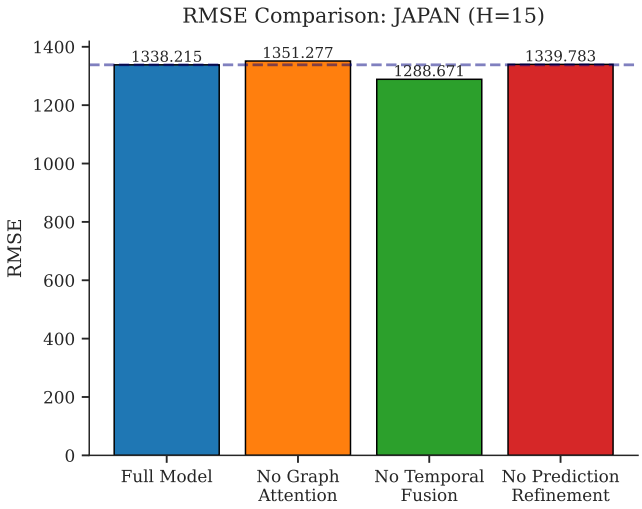
(a) RMSE comparison across model variants for the Japan dataset horizon 3



(b) RMSE comparison across model variants for the Japan dataset horizon 5



(c) RMSE comparison across model variants for the Japan dataset horizon 10



(d) RMSE comparison across model variants for the Japan dataset horizon 15

Fig. 7: Comparison of RMSE across different model variants for the Japan dataset.

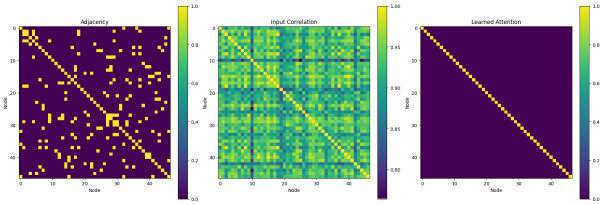


Fig. 8: Adjacencies and learnt attention matrices of MSAGAT-Net on the Japan-Prefectures dataset for 5-day horizon with no agam module

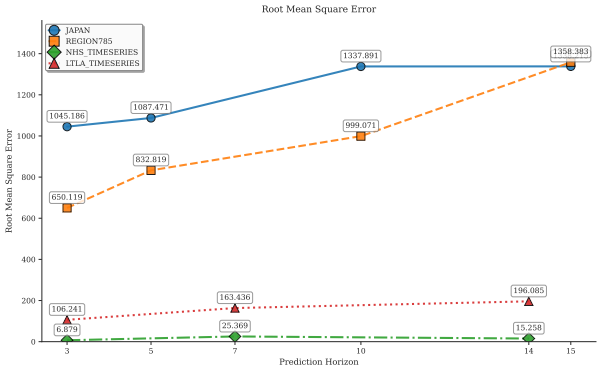


Fig. 9: Comparison of RMSE across different model variants for some choosen dataset

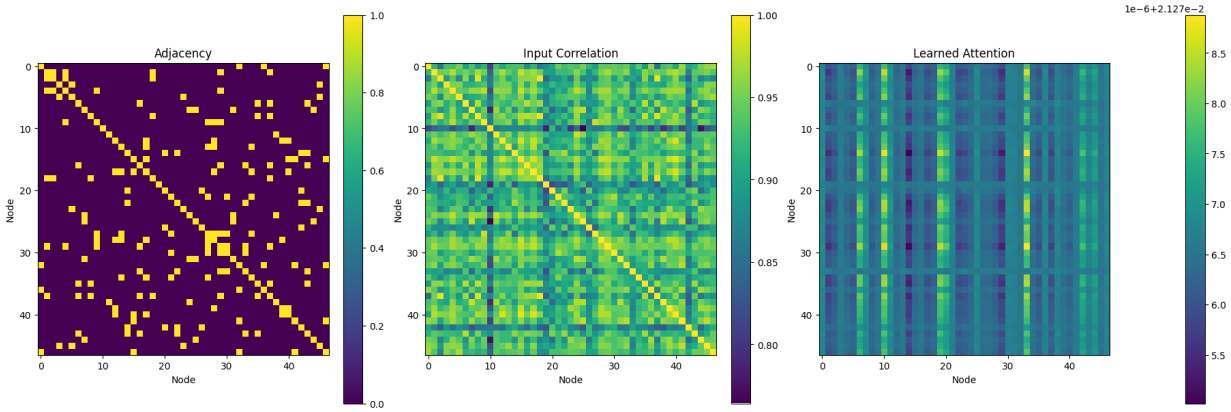


Fig. 10: Adjacencies and learnt attention matrices of MSAGAT-Net on the Japan-Prefectures dataset for 5-day horizon with no mtfm module

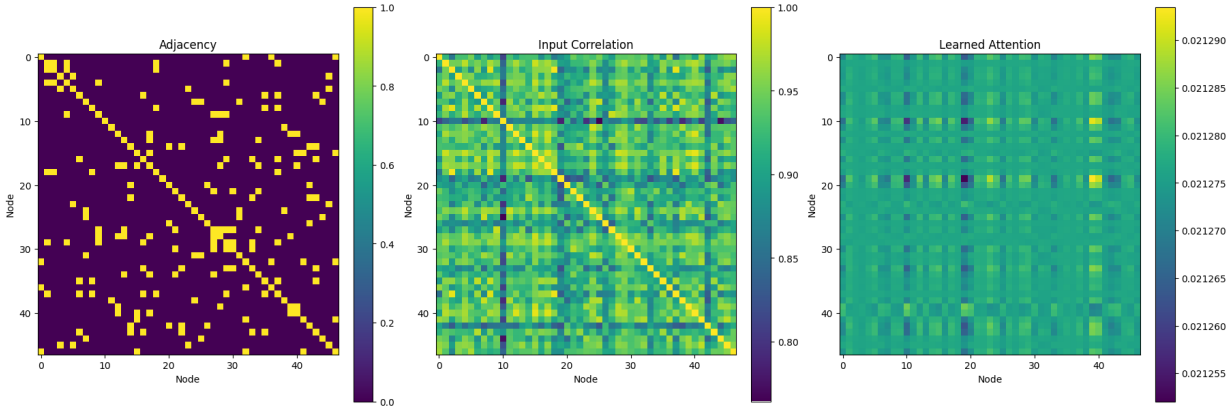
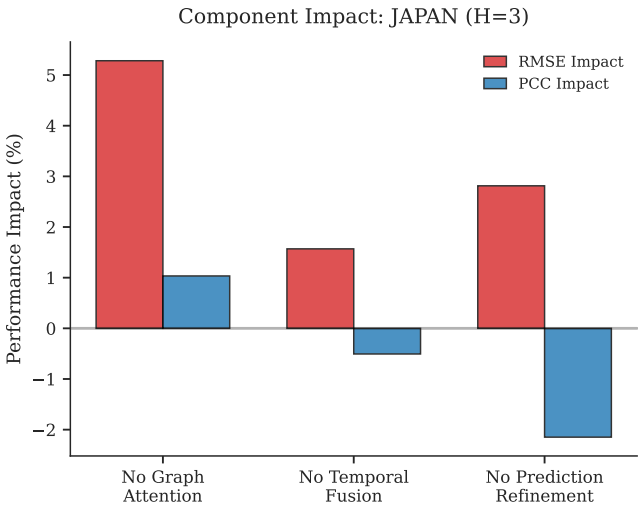
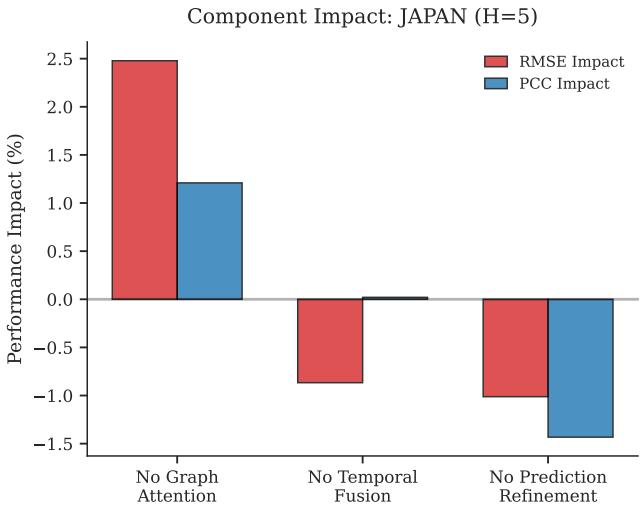


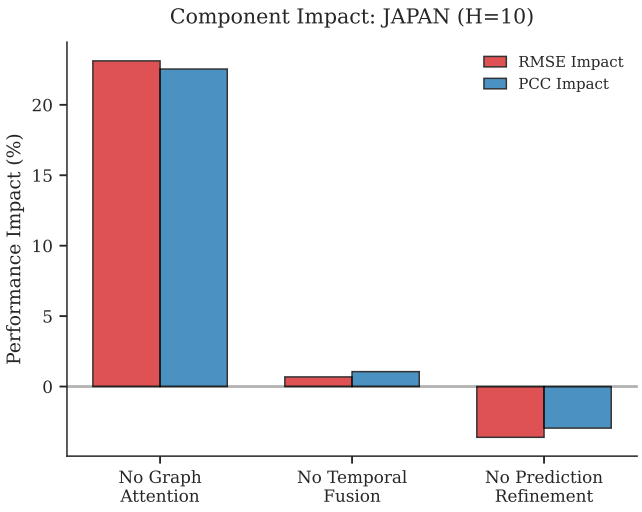
Fig. 11: Adjacencies and learnt attention matrices of MSAGAT-Net on the Japan-Prefectures dataset for 5-day horizon with no pprm module



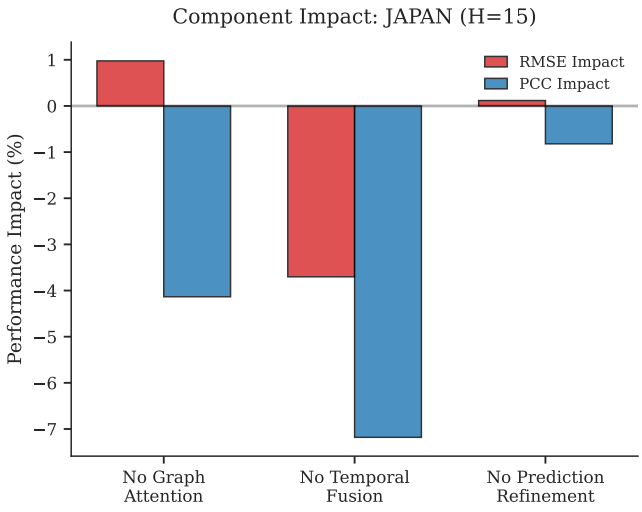
(a) component impact for Japan dataset horizon 3



(b) component impact for Japan dataset horizon 5



(c) component impact for Japan dataset horizon 10



(d) component impact for Japan dataset horizon 15

Fig. 12: Comparison of component impact across different model variants for the Japan dataset.

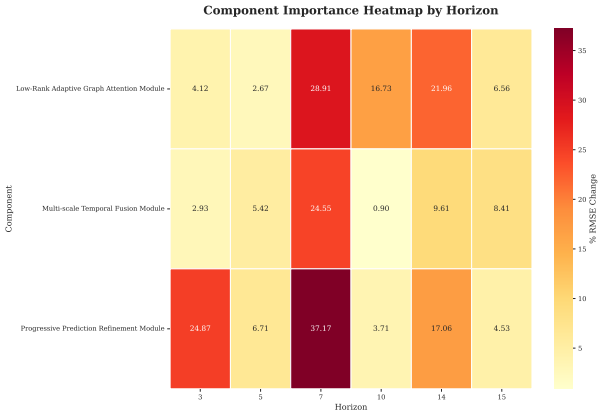


Fig. 13: Component importance heatmap across different datasets and forecast horizons.