# Paper 5: Virtual Network Function Placement Using Differentiated Weight Graph Convolutional Neural Network and Maximal Weight Matching

## 1. Introduction

In modern 5G networks, efficiently placing Virtual Network Functions (VNFs) is crucial for optimizing performance and resource utilization. Traditional methods often struggle with the complexity and dynamic nature of these networks. This paper introduces a novel approach that combines a Differentiated Weight Graph Convolutional Neural Network (DWGCN) with Maximal Weight Matching (MWM) to enhance VNF placement strategies.

## 2. Key Concepts

### 2.1 Graph Convolutional Neural Networks (GCNs)

📌 Definition: GCNs are neural networks designed to operate on graph-structured data, capturing relationships between nodes through message passing.

📌 Application in Networking: GCNs can model complex network topologies, making them suitable for tasks like VNF placement.

### 2.2 Differentiated Weight Graph Convolutional Neural Network (DWGCN)

📌 Innovation: Unlike standard GCNs that treat all nodes and edges uniformly, DWGCN assigns different weights to various types of entities and relationships within the network.

📌 Benefit: This differentiation allows the model to capture the heterogeneity of network components more effectively.

### 2.3 Maximal Weight Matching (MWM)

📌 Definition: A combinatorial optimization technique that pairs elements of a set to maximize the sum of weights associated with the pairs.

📌 Role in VNF Placement: MWM is used to determine the optimal assignment of VNFs to physical nodes, ensuring efficient resource utilization.

## 3. Proposed Methodology

### 3.1 Model Architecture

📌 Input Representation: The network is modeled as a graph where nodes represent physical servers, and edges denote communication links.

📌 Feature Encoding: Each node and edge is assigned features such as computational capacity, bandwidth, and latency.

📌 DWGCN Application: The DWGCN processes this graph, considering the differentiated weights to produce embeddings that capture the network's nuanced characteristics.

### 3.2 Integration with Maximal Weight Matching

📌 Objective: To map VNFs onto physical nodes in a manner that maximizes performance and resource efficiency.

📌 Process:

1️⃣ Graph Construction: Create a bipartite graph where one set represents VNFs and the other represents potential host nodes.

2️⃣ Weight Assignment: Assign weights based on the embeddings generated by the DWGCN, reflecting the suitability of each VNF-node pair.

3️⃣ MWM Application: Apply the MWM algorithm to find the optimal matching that maximizes the total weight, leading to an efficient VNF placement.

## 4. Numerical Example

Let's illustrate the proposed methodology with a simplified numerical example.

### 📌 Scenario Setup

• \*\*Physical Network:\*\* Consists of 3 physical nodes (P1, P2, P3) with varying computational capacities.  
• \*\*VNFs to be Placed:\*\* 3 VNFs (V1, V2, V3) each with specific resource requirements.  
• \*\*Objective:\*\* Place each VNF onto a physical node such that the overall resource utilization is optimized.

### 📌 Step 1: Graph Representation

• \*\*Nodes:\*\* Represent both physical nodes and VNFs.  
• \*\*Edges:\*\* Connect each VNF to all potential physical nodes where it can be placed.  
• \*\*Features:\*\* Include computational capacity for physical nodes and resource requirements for VNFs.

### 📌 Step 2: Applying DWGCN

• \*\*Input:\*\* The graph with nodes and edges as defined above.  
• \*\*Processing:\*\* The DWGCN processes the graph, taking into account the differentiated weights (e.g., higher weights for edges connecting VNFs to nodes with sufficient resources).  
• \*\*Output:\*\* Node embeddings that capture the suitability of each VNF-node pair.

### 📌 Step 3: Constructing the Bipartite Graph

• \*\*Bipartite Graph:\*\* Create a bipartite graph where one set consists of VNFs (V1, V2, V3) and the other set consists of physical nodes (P1, P2, P3).

• \*\*Weights:\*\* Assign weights to the edges between VNFs and physical nodes based on the embeddings from the DWGCN. For instance:

- Edge (V1, P1): Weight = 0.8  
 - Edge (V1, P2): Weight = 0.6  
 - Edge (V1, P3): Weight = 0.4

### 📌 Step 4: Applying Maximal Weight Matching

• \*\*Objective:\*\* Find the matching between VNFs and physical nodes that maximizes the total weight.  
• \*\*Algorithm:\*\* Use the MWM algorithm to determine the optimal placement.  
• \*\*Result:\*\* An optimal matching, for example:  
 - V1 → P1  
 - V2 → P3  
 - V3 → P2

✅ This matching ensures that VNFs are placed on physical nodes in a manner that optimizes resource utilization and performance.

## 5. Advantages of the Proposed Approach

✅ \*\*Enhanced Accuracy:\*\* By incorporating differentiated weights, the model captures the unique characteristics of network components, leading to more precise VNF placements.

✅ \*\*Scalability:\*\* The combination of DWGCN and MWM allows the approach to scale effectively with large and complex network topologies.

✅ \*\*Adaptability:\*\* The model can adapt to dynamic network conditions, making it suitable for real-time VNF placement decisions.