Re-implementing the Joint Prediction Module from RigNet

Govind Joshi
CSCE 685 Research Project
Advisor: Dr. Shinjiro Sueda

April 15, 2025

1 Introduction

Mesh skeletons and skinning weights are fundamental to character animation, but creating them by hand is laborious. RigNet [1] is an end-to-end neural method that automatically predicts both a skeleton and per-vertex skinning weights from a single static mesh. In this project, I will re-implement RigNet's *joint prediction* stage in PyTorch, with a focus on modularity and future extension.

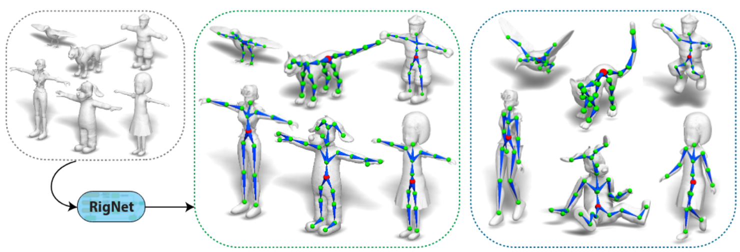

Figure 1: RigNet pipeline overview (from Xu et al. 2020).

2 RigNet Architecture Overview

RigNet decomposes the rigging problem into three stages:

1. **Joint Prediction:**
   - A Graph Neural Network (GMEdgeNet) predicts vertex displacements toward candidate joint locations.
   - A second GMEdgeNet predicts a per-vertex attention map highlighting likely joint regions.
   - An attention-weighted mean-shift clustering extracts 3D joint positions.
2. **Skeleton Connectivity:**
   - *BoneNet* predicts edge probabilities between each joint pair.
   - *RootNet* selects the root joint.
   - A Minimum Spanning Tree yields a tree-structured skeleton.

3. **Skinning Prediction:**
    - A final GMEdgeNet predicts per-vertex skinning weights given the skeleton.

This project focuses on the *Joint Prediction* module (Section 4.1 of the paper).

3 Project Plan

I divide the work into two parts:

Part 1: Understanding and Evaluation

- Study RigNet's Joint Prediction Module's architecture and training methodology in detail.
- Document strengths and failure modes.
- Present an overview of findings and architecture summary.

Part 2: Implementation

1. **Data Loading & Visualization:** Load meshes and ground-truth rigs from pickle; visualize vertex neighborhoods and joint annotations.
2. **Baseline: Single-Layer GMEdgeConv**
    - Implement one GMEdgeConv layer (topo + geodesic) to predict per-vertex displacements.
    - Implement attention-free mean-shift clustering to extract joint positions.
    - Train with Chamfer distance loss.
3. **Full Joint Prediction Module**
    - Add the vertex-attention head (second GMEdgeNet).
    - Integrate attention-weighted clustering.
    - Stack three GMEdgeConv layers, add global pooling and final MLP heads.
    - Train end-to-end on a small rigged dataset.
4. **Testing & Visualization:** Run on several online character meshes; display predicted joints overlaid on meshes.
5. **Modular Codebase:** Organize code so future stages (BoneNet, skinning) can be added seamlessly. Code will be derived and thoroughly documented in jupyter notebooks, and final scripts will be moved into python files to allow for running with different command line parameters on HPRC.

References

[1] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *ACM Transactions on Graphics (TOG)*, 39(4):1–14, 2020.