Baseline Joint Prediction after Mean–Shift Clustering

Govind Joshi
CSCE 685, Dr. Shinjiro Sueda

April 24, 2025

1 Introduction

Having implemented and tested the mean-shift clustering module for joint localization, I designed a *simple baseline* to verify end-to-end training before adding attention or deeper architectures. The goal is to confirm that even a minimal pipeline—one graph edge-convolutional layer plus clustering—can learn to collapse vertices toward ground-truth joints.

2 Dataset

- **Size:** 25 meshes total.
- **Split:** 20 training, 5 validation.
- Each mesh comes with annotated joint coordinates.

3 Baseline Architecture

1. **GMEdgeConv Feature Extractor:** A single GMEdgeConv layer maps each vertex's 3D position to a 64-dimensional embedding.
2. **Displacement Head:** A 2-layer MLP takes the 64-dim embedding and predicts a 3D displacement vector for each vertex.
3. **Clustering:** Displaced vertices $q_i = v_i + \Delta v_i$ are all assigned equal attention ($a_i \equiv 1$). We run mean-shift until convergence and extract modes as joint predictions.

4. **Loss:** We minimize the symmetric Chamfer distance between predicted joints $\{t_k\}$ and ground-truth $\{\hat{t}_\ell\}$:

$$\mathcal{L} = \sum_k \min_\ell \|t_k - \hat{t}_\ell\| + \sum_\ell \min_k \|\hat{t}_\ell - t_k\|.$$

4 Results

4.1 Training Loss

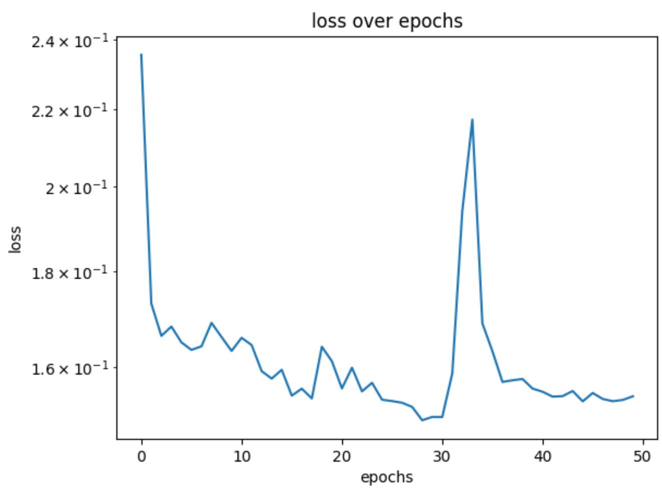

Figure 1: Semilogy plot of Chamfer loss over 50 epochs (20 training examples, learning rate $= 1 \times 10^{-3}$).

Over 50 epochs, the validation Chamfer loss drops from the untrained level to a significantly lower value, demonstrating that the simple baseline can indeed learn.

4.2 Visualizations

In the following visualization, I visualize the ground truth joints as blue spheres, and the predicted joints as purple spheres. There are far more purple spheres than blue, which hints at the fact that the model is not learning to collapse vertices to joint locations well yet.

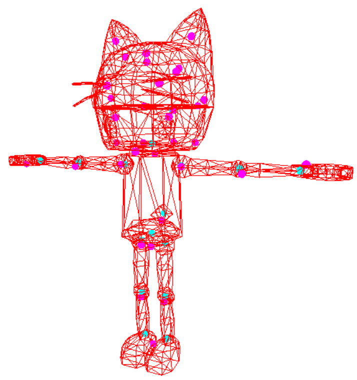

Figure 2: Joint predictions on a training example after 50 epochs: displaced vertices collapse toward the true mesh joints.

5 Conclusion

This stripped-down GMEdgeConv + mean-shift pipeline already demonstrates the ability to learn joint localization: even without attention, vertices collapse toward true joints under Chamfer supervision. This baseline provides a solid foundation for integrating the full attention module, deeper stacks, and connectivity prediction in the next project stages.

6 Next Steps

- Build out the full GMEdgeNet architecture with multiple stacked GMEdgeConv layers as in the paper.
- Create the final *joint prediction network* that includes a GMEdgeNet for learning attention weights for vertices.

Update: I found that the attention module requires an additional supervision signal. The paper finds that pre-training the attention module with an additional signal can improve results. Next, I will

- create boolean attention masks set to 1 for vertices lying close to the true joint locations
- pre-train the attention network via cross-entropy against these masks.

(a) Untrained Model
Validation Loss: 3.3388×10^{-1}

(b) Trained Model
Validation Loss: 1.3608×10^{-1}

Figure 3: Comparison of joint predictions on a validation mesh. The trained model displaces vertices much closer to the annotated joints.