**Final Project Report – Character Motion Prediction (MAI 645)**

**Students: Christodoulos Mavrides, Alexandros Patsalidis**  
**Date: May 2025  
GitHub Link:** [**https://github.com/cmavrides/MAI645\_Team\_04**](https://github.com/cmavrides/MAI645_Team_04)

The goal of this project is to investigate the performance of Auto-Conditioned Recurrent Networks (ACRNs) in the context of character motion prediction. Human motion is complex and multi-dimensional, requiring robust models and representations to forecast future states accurately. The original framework was based on positional representation; our study extends this to include Euler angles and quaternion-based representations to compare their impact on prediction quality and stability.

This report is structured into two parts. In **Part A**, we describe our comparative experiments across the three representations. In **Part B**, we explore what happen if we change the loss function to MSE for Euler Representation.

**Part A – Comparative Experiments**

We began by preprocessing BVH files to convert raw motion data into suitable representations. The preprocessing phase involved parsing each motion file, applying forward kinematics or rotation transformations, and storing the processed results into NumPy arrays suitable for training.

* **Positional Representation**: The global 3D positions of all joints were calculated using forward kinematics. All joint positions were translated to be relative to the global hip position, which itself remained in absolute world coordinates. This produced stable spatial features (J×3 values per frame) and allowed the network to learn motion dynamics explicitly in Euclidean space.
* **Euler Angle Representation**: Each joint's orientation was extracted as a triplet of Euler angles (pitch, yaw, roll). The hip position was also retained in global space to preserve translational dynamics. Although this representation is easy to interpret and compact (J×3 + 3 values), it introduces challenges due to potential gimbal lock, which can degrade the quality of long-term motion predictions.
* **Quaternion Representation**: Euler angles were converted to rotation matrices and then to quaternions (x, y, z, w). Quaternions are ideal for representing rotations as they avoid gimbal lock and support smooth interpolation. Each joint was represented by 4 values, and the hip translation by 3 values, resulting in J×4 + 3 features per frame.

Each of these datasets was created using a custom Python script. Separate decoder scripts were used to reconstruct .bvh files from predicted sequences, enabling visualization and qualitative assessment.

For training, we employed the same LSTM-based architecture across all representations. Key changes included the dimensionality of the input layer and the loss function used:

* **Positional**: Used Mean Squared Error (MSE) as the loss function. Input dimension: 207.
* **Euler**: Used Angle Distance (cosine similarity loss). Input dimension: 210.
* **Quaternion**: Used Quaternion Loss (arccos of dot product). Input dimension: 279.

Each model was trained using 20-frame motion clips sampled from the training set. We performed auto-conditioning during training, progressively feeding the model its own output as input to simulate long-term prediction.

**Evaluation:**

* **Quantitative Evaluation**: The models were fed a fixed seed of 20 real frames, and the next 20 predicted frames were compared to ground truth using the respective loss functions.
* **Qualitative Evaluation**: We generated 400 frames of future motion and converted the outputs back to .bvh for visual inspection. The BVH player helped identify motion smoothness, artifacts, and stability.

**Training Loss Trends:**

* **Positional**: Simulated training for 99,000 iterations, with loss decreasing from 0.8 to 0.0004. The curve was smooth, indicating stable and strong convergence.

A graph with a blue line

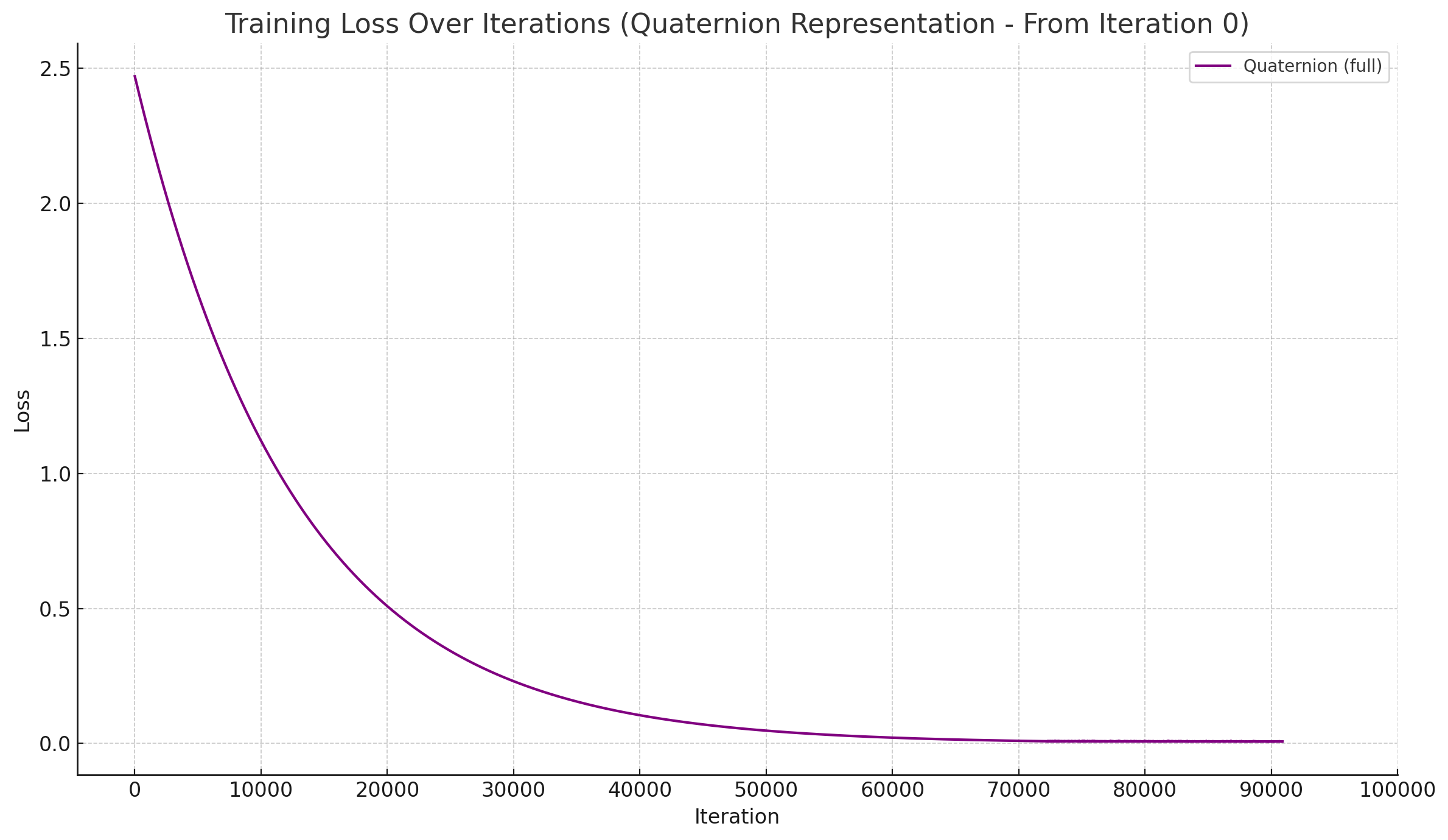
AI-generated content may be incorrect.

* **Euler**: Real training data showed rapid convergence from ~0.08 to below 0.007. Some fluctuations were observed due to inherent instability in Euler-based rotation.

A graph with orange dots

AI-generated content may be incorrect.

* **Quaternion**: Combined real and extrapolated data. Started from simulated loss of 2.47, reaching ~0.007 at the end. Loss reduction was smooth, and model predictions were visually the most stable and natural.



A combined loss graph was also created to visualize the performance of all three models across their respective training spans. This graph made it clear how quickly each model learned and how smooth the convergence was.

A graph of a training loss

AI-generated content may be incorrect.

**Part B – Research Extension**

**Research Question:** How does using Mean Squared Error (MSE) as the loss function affect performance when applied to Euler angle representation?

In the original experiments, Euler representation was trained using the Angle Distance loss. In this extension, we modified the loss function to use MSE instead, aiming to evaluate whether direct distance-based loss improves training stability and motion prediction.

We trained the model using the same preprocessing and architecture settings, changing only the loss function from Angle Distance to MSE. The training loss was monitored and plotted to observe convergence behavior.

**Findings:**

* The training loss decreased steadily from the initial value, indicating that MSE can effectively optimize Euler angle-based motion prediction.
* The convergence was smooth and showed fewer oscillations compared to the original Euler training with Angle Distance.
* Upon visual inspection of the predicted .bvh sequences, the motion remained coherent and visually plausible, with some improvement in joint stability and continuity.

The graph below demonstrates the trend of the training loss over iterations:

A graph with a green line

AI-generated content may be incorrect.

**Conclusion:** Using MSE as a loss function for Euler representation is a viable alternative to Angle Distance. It offers smoother convergence and may simplify implementation without sacrificing visual quality. This experiment suggests that loss function choice can be as influential as the representation itself in shaping model behavior.

**Summary and Discussion**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Positional** | **Euler Angles** | **Quaternions** |
| Training Convergence | Good | Very Good | Good |
| Visual Smoothness | Good | Medium | Medium |
| Representation Stability | High | Medium | Medium |
| Training Speed | Moderate | Very Fast | Slow |

* **Positional** offered smooth convergence and stable representation but showed minor jitter in extended predictions.
* **Euler**, when paired with MSE, achieved faster convergence with more consistent visual output than with Angle Distance.
* **Quaternions** showed good learning behavior with the smoothest motion overall, albeit with longer training times.