IML Exercise 5

Ahmad Ar

January 2026

1 Gaussian Mixture Model

We optimize Gaussian Mixture Models (GMMs) with different numbers of components $K \in \{1, 5, 10, 33\}$ using stochastic gradient descent. The data is normalized to zero mean and unit variance per dimension.

1.1 Sampling from the GMM

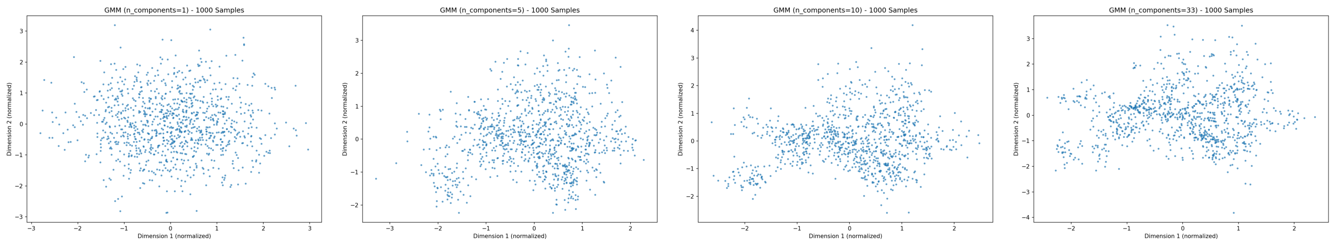

Figure 1: Samples drawn from trained GMMs with different numbers of components.

Explanation. Increasing the number of components allows the model to capture increasingly complex multi-modal structure in the data.

1.2 Conditional Sampling from Each Gaussian

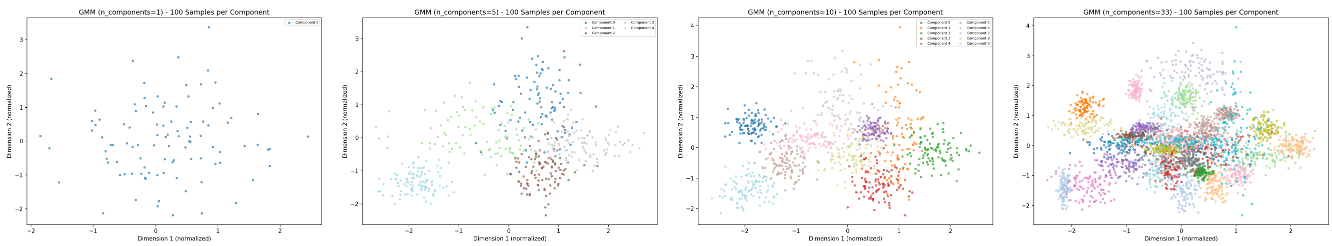

Figure 2: Conditional samples from each Gaussian component.

Explanation. Each component specializes to a localized region of the data space.

1.3 Training Dynamics with Random Initialization ($K = 33$)

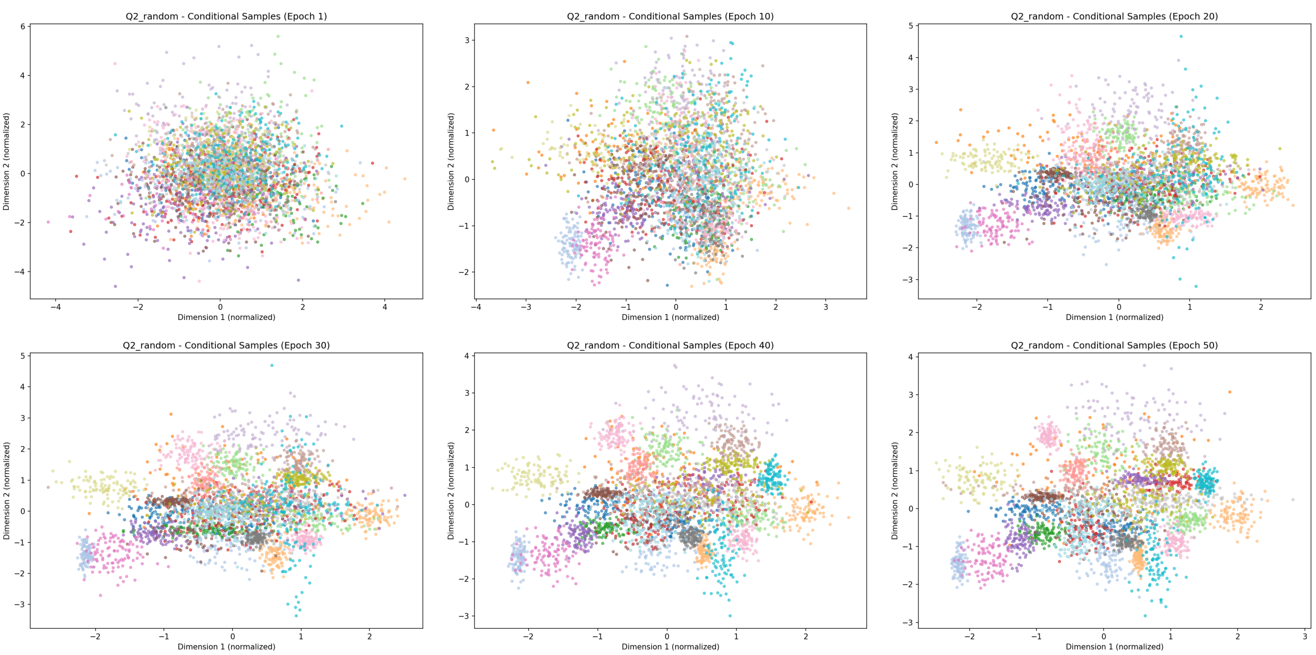

Figure 3: Conditional samples during training (random initialization).

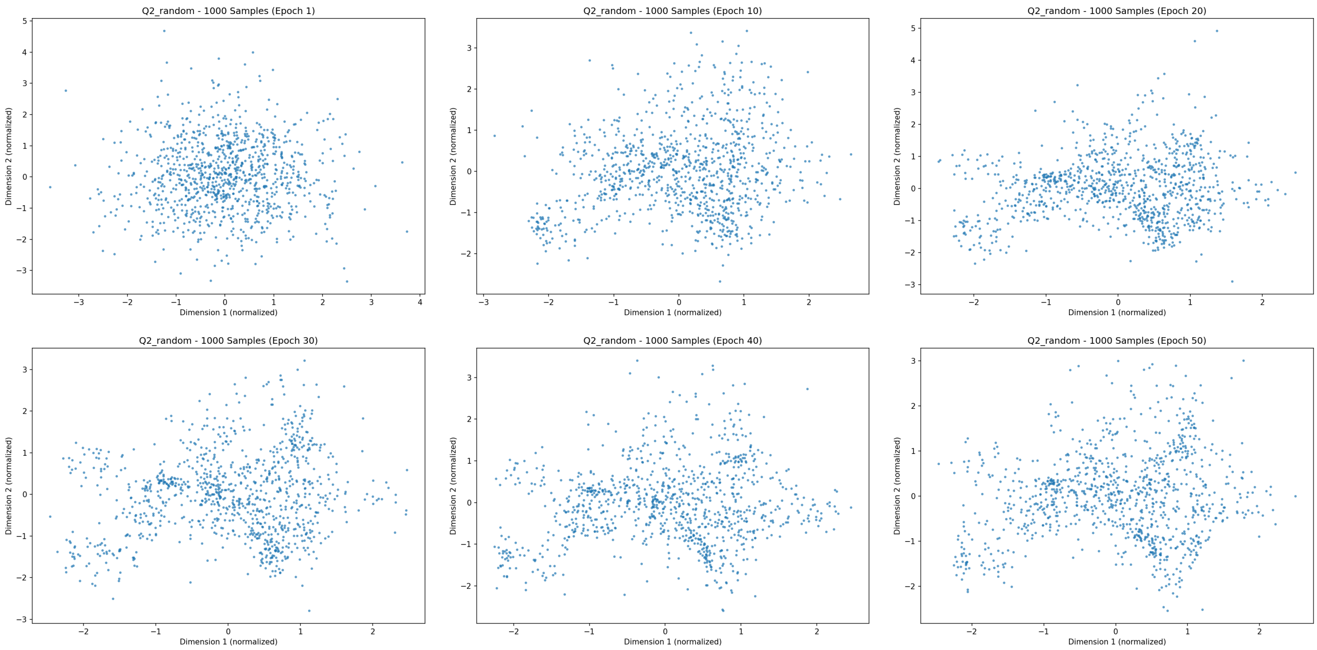

Figure 4: Unconditional samples during training.

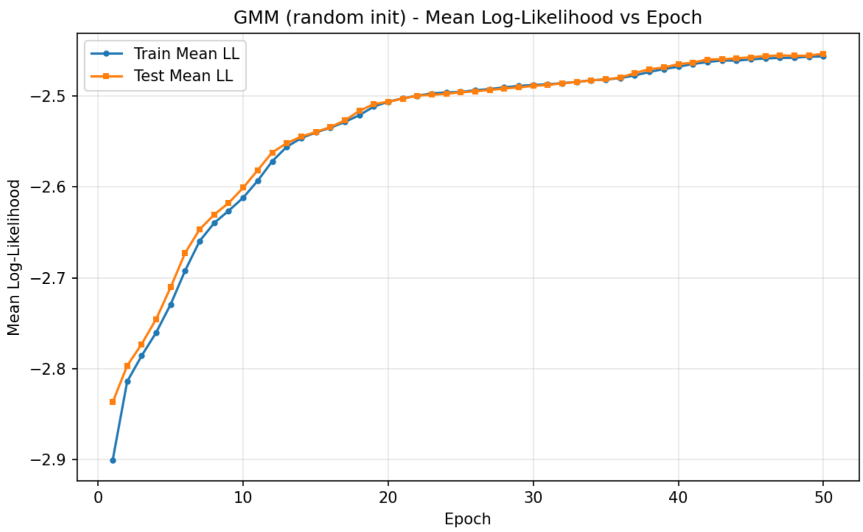

Figure 5: Mean log-likelihood vs. epoch (random initialization).

Explanation. Components gradually separate and converge to meaningful regions of the data.

1.4 Training with Mean-Based Initialization

Explanation. Mean-based initialization improves convergence speed and early structure.

Figure 6: Conditional samples with mean-based initialization.

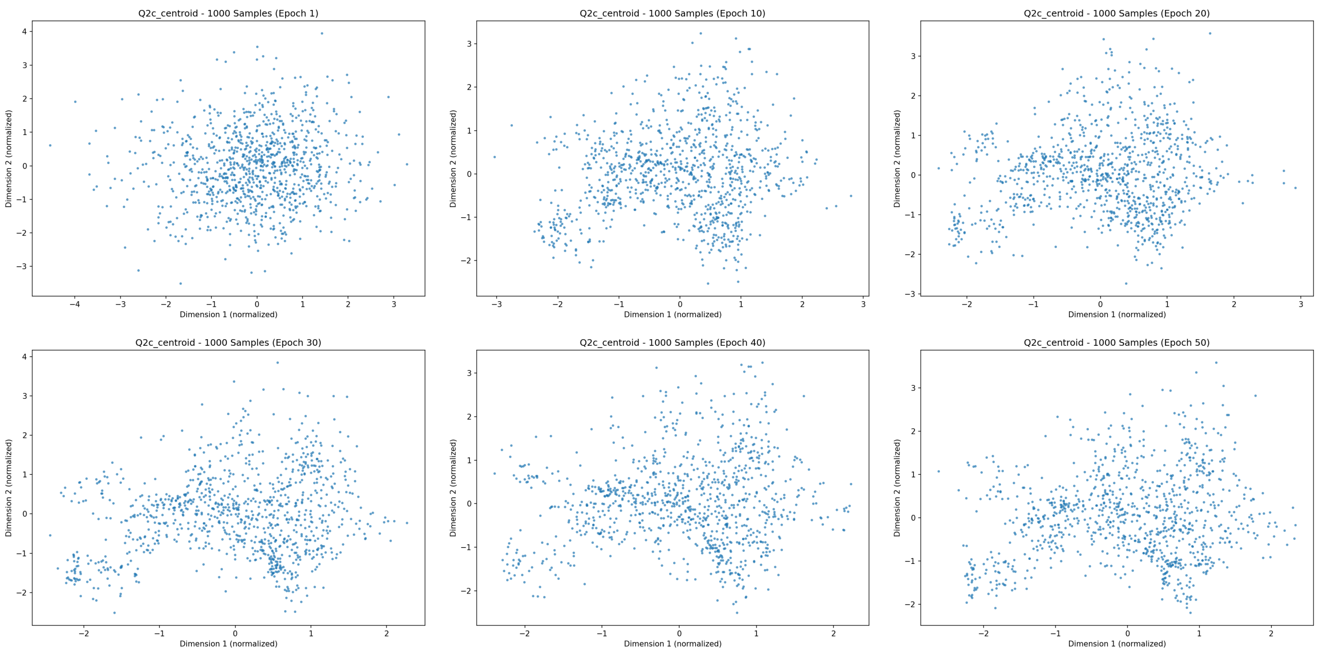

Figure 7: Unconditional samples with mean-based initialization.

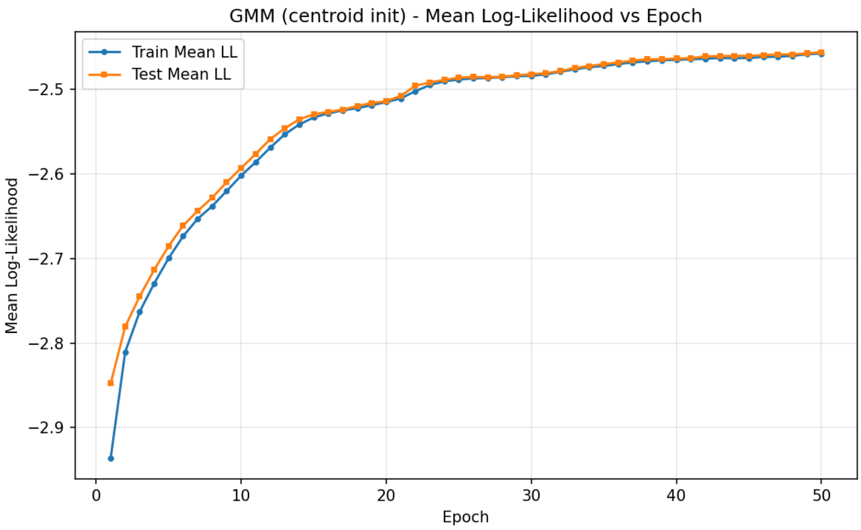

Figure 8: Mean log-likelihood vs. epoch (mean-based initialization).

2 Uniform Mixture Model

Uniform components are axis-aligned rectangles with constant density inside their support and zero probability outside.

2.1 Sampling from the Uniform Mixture Model

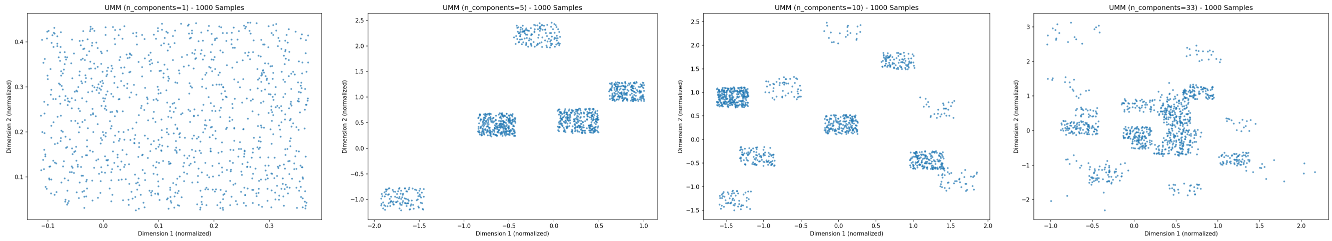

Figure 9: Unconditional samples from trained Uniform Mixture Models.

2.2 Conditional Sampling from Each Uniform Component

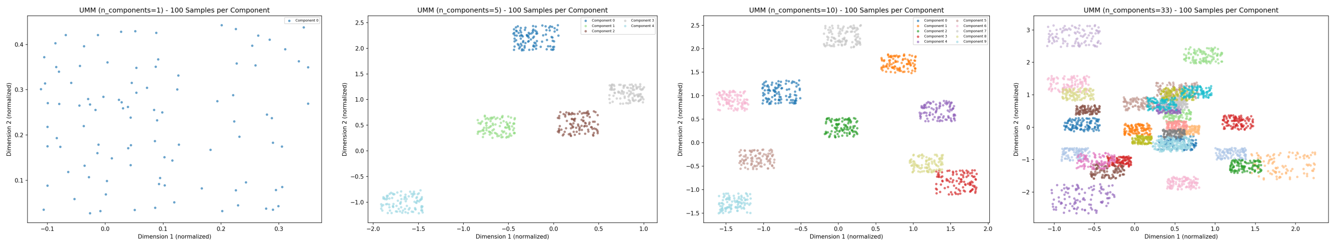

Figure 10: Conditional samples from each uniform component.

2.3 Training Dynamics with Random Initialization ($K = 33$)

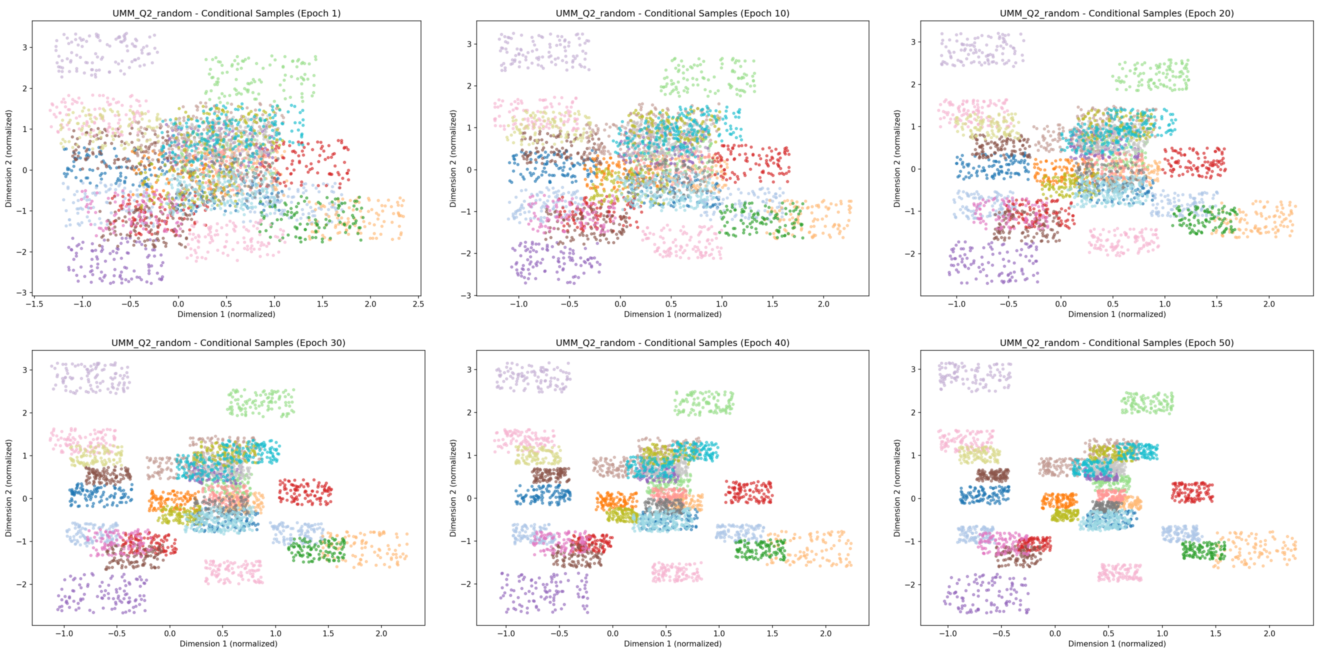

Figure 11: Conditional samples during training (random initialization).

2.4 Training with Mean-Based Initialization

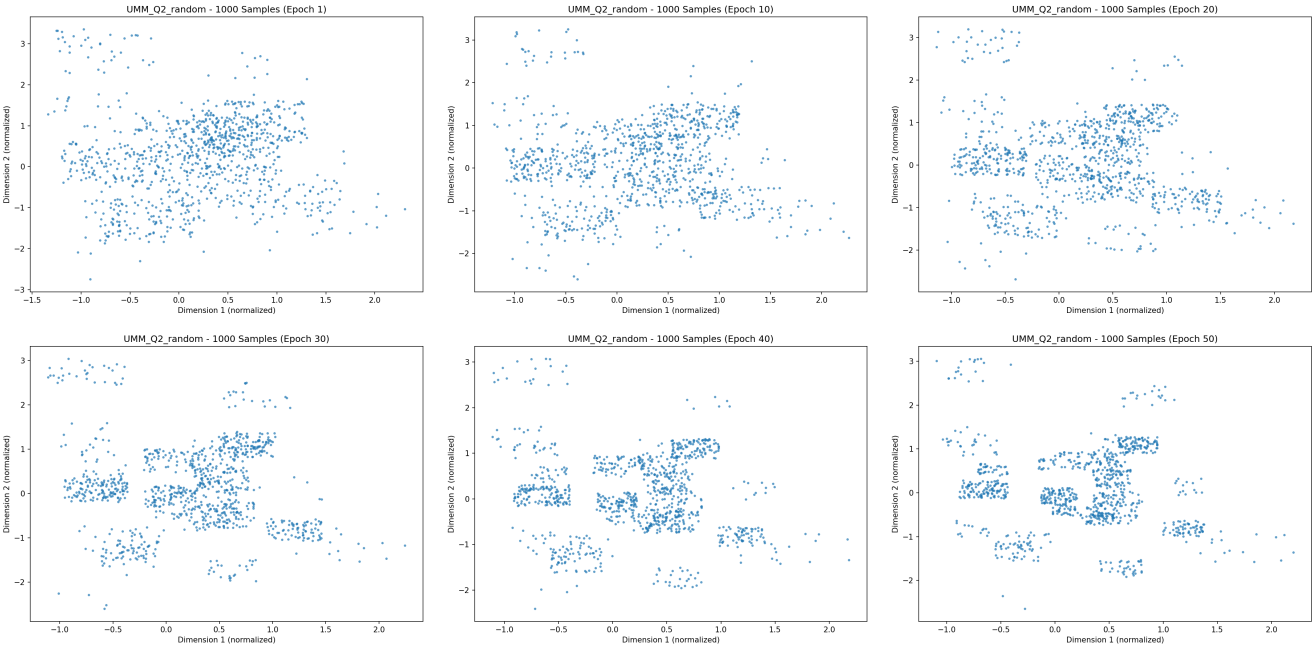

Figure 12: Unconditional samples during training.

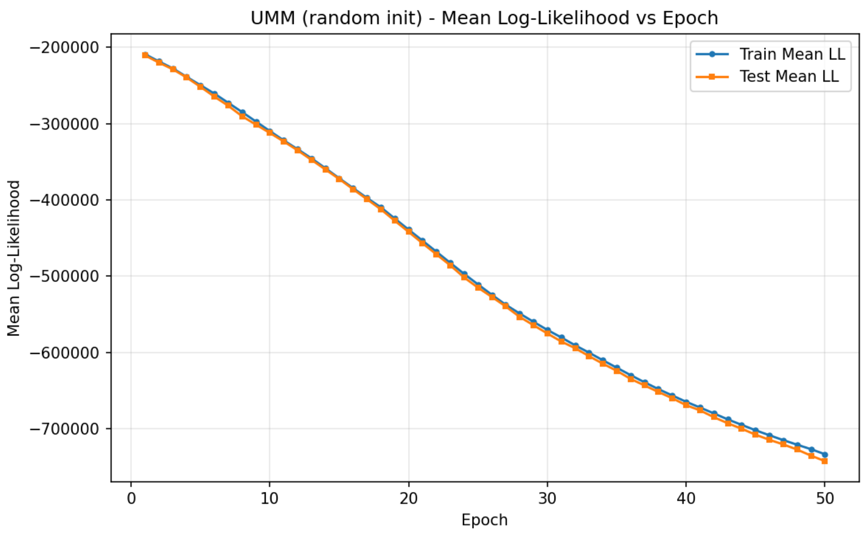

Figure 13: Mean log-likelihood vs. epoch (random initialization).

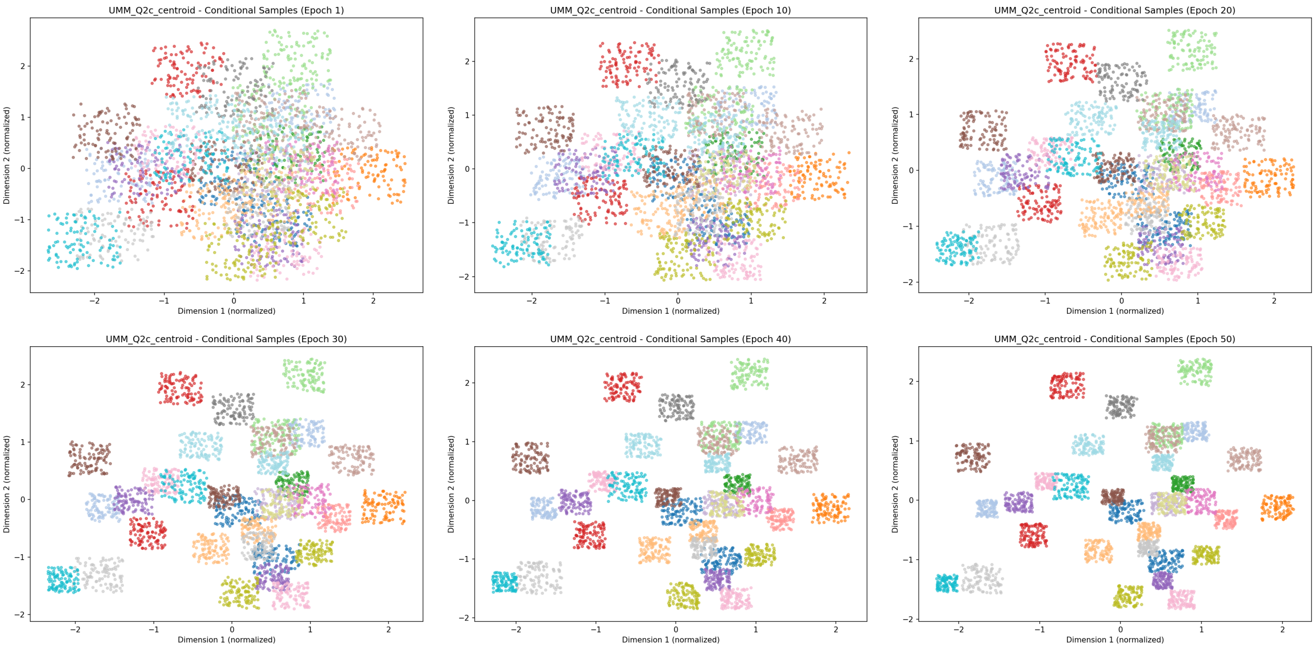

Figure 14: Conditional samples with mean-based initialization.

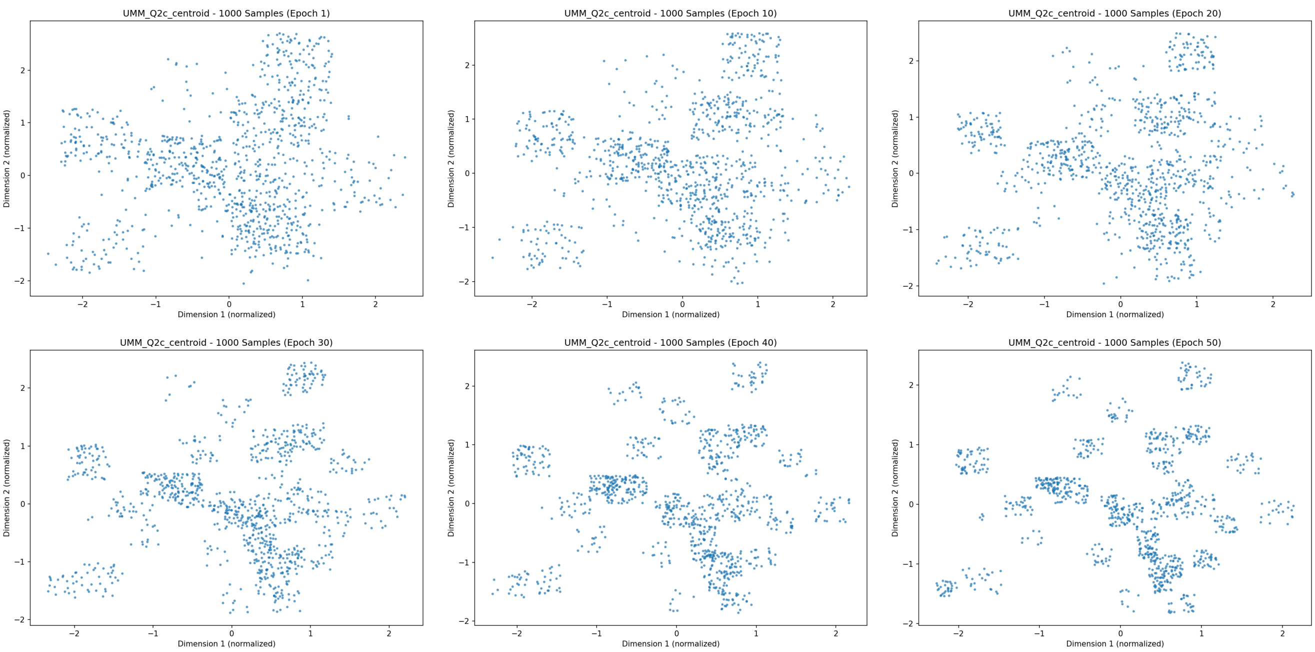

Figure 15: Unconditional samples with mean-based initialization.

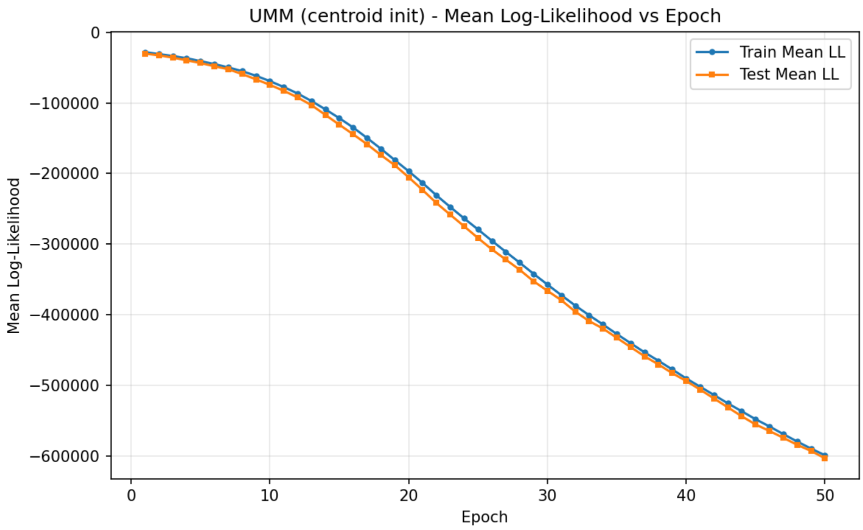

Figure 16: Mean log-likelihood vs. epoch (mean-based initialization).

3 Transformer Language Model

We train an auto-regressive Transformer model to predict the next character in a sequence using causal self-attention. The model is evaluated using cross-entropy loss and sequence-level accuracy.

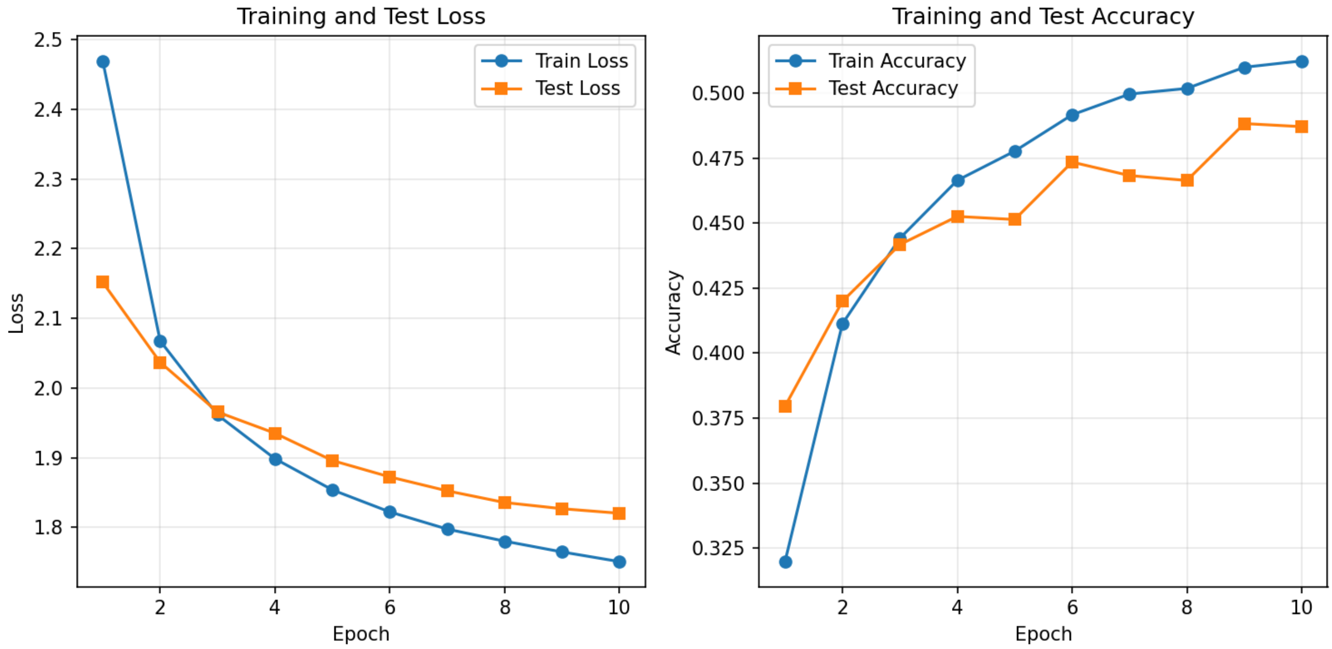

Figure 17: Training and test loss (left) and accuracy (right) across epochs.

Explanation. Both training and test loss decrease steadily, while accuracy improves consistently across epochs. The small gap between training and test curves indicates good generalization and stable optimization.

3.1 Text Generation

After each epoch, the model generates sequences in an auto-regressive manner, starting from the prefix `‘‘the ’’`. Sentences are generated using both standard sampling and top-k sampling with $k = 5$.

Observation. Top-k sampling produces more coherent and syntactically consistent sentences by suppressing unlikely character predictions, while standard sampling yields more diverse but noisier outputs.