## Comparison: Logical Upgradations Summary

| What Changed | Why It Was Changed | What Effect It Had | How We Detected the Need |
|---|---|---|---|
| 1. **Added Intermediate Fully Connected Layer before Quantum Circuit** (`fc1 → relu1 → fc1_reduce`) | To prevent information collapse from large BERT embeddings (768D) directly to small qubit space (e.g., 4D) | Preserved richer feature patterns before quantum layer; helped reduce over-simplified outputs | Earlier model directly mapped from 768 → 4 (qubits), which causes underfitting |
| 2. **Introduced Non-linear Activation before Quantum Layer** (`ReLU` + `tanh`) | Non-linearity enables better representation learning before quantum mapping | Prevents linear bottleneck and enriches representation entering quantum layer | Observed flat output logits & poor F1s; classical MLPs usually benefit from non-linearity |
| 3. **Expanded Classical Layer After Quantum Output** (`fc2_expand → relu2 → fc2_out`) | Quantum outputs (n_qubits) are too compact to predict 28 emotion labels directly | Adds model capacity to interpret quantum features and make complex multi-label predictions | The model was predicting the same emotion for most inputs (low diversity of output) |
| 4. **Increased Depth in Classical Component** | Shallow nets underfit high-dimensional semantic inputs (BERT embeddings) | Helps capture higher-order interactions in classical features | Overly simplistic outputs indicated lack of depth |
| 5. **Switched to Batched BERT Embedding** | To ensure stable and fast BERT inference with memory efficiency | Optimized speed and reduced GPU memory usage | Long wait during initial run with token-by-token encoding |
| 6. **Improved Quantum Circuit Generalization (indirect)** by avoiding direct mapping from 768 → 4 | Enforced progressive dimension reduction to fit qubit space gradually | Helped stabilize training and reduce one-class collapse | F1 scores previously collapsed to a single label |

| | | | |
|---|---|---|---|
| 7. **Added Evaluation Plots (F1 per label + Confidence Histogram)** | To monitor per-label learning and overall confidence levels | Helps visualize imbalance, class-specific underfitting | Detected class dominance (e.g., love) and overconfident predictions (~1.0) |
| 8. **Added Progress Bar with `tqdm`** | For tracking model training per epoch and live loss reporting | Improved training visibility and debugging insight | Earlier model gave no feedback during long runs |

## Net Observed Outcome

| Metric | Underfit Version | Current Version |
|---|---|---|
| Macro F1 Score | ~0.00 | 0.02–0.05 |
| Micro F1 Score | ~0.00 | 0.07 |
| Predicted Labels | Mostly “love” only | “gratitude”, “surprise”, and a few others |
| Label Diversity | None | Partial |
| Training Feedback | Absent | Present (with tqdm) |
| Usability for Scaling | Weak base | Modular and extensible |

## What Still Needs to Improve (Next Steps)

| Area | Suggestion |
|---|---|
| Class Imbalance | Use `class_weight` in loss function |
| Output Thresholding | Calibrate sigmoid threshold per class |
| Data Filtering | Remove low-frequency labels or group similar ones |
| Quantum Architecture | Try deeper layers or different entanglement patterns |
| Early Stopping / LR Scheduling | To prevent overtraining on noise |