# Assignment 3B: Transformer LLM — Post-Lab Report

## a. Model Description and Parameter Justification

The model is a GPT-2-style decoder-only Transformer trained on the Tiny Shakespeare dataset (~1961 kB of text). The architecture and hyperparameters are as follows:

| Parameter | Value | Justification |
|---|---|---|
| `context_length` | 256 | Provides enough context for the model to capture sentence-level and short-paragraph dependencies in Shakespeare's writing style. |
| `n_layer` | 6 | Six Transformer blocks give sufficient depth for the model to learn hierarchical language patterns without excessive training time on a single GPU. |
| `n_head` | 6 | Six attention heads allow the model to attend to different positional and semantic relationships in parallel. Divides evenly into the embedding dimension (384 / 6 = 64 per head). |
| `n_embd` | 384 | A moderate embedding size that balances representational capacity with training efficiency. Large enough to encode the character-level vocabulary of Shakespeare. |
| `dropout` | 0.2 | A standard regularization rate that helps prevent overfitting on the relatively small training corpus. |
| `batch_size` | 64 | Fits comfortably in GPU memory (NVIDIA RTX 3050 Ti) while providing stable gradient estimates. |
| `max_iters` | 2000 | Sufficient iterations to observe convergence of training and validation loss within ~30 minutes of training. |
| `learning_rate` | 1e-3 | A common starting learning rate for AdamW on Transformer models of this scale. |

The total number of trainable parameters is **10,690,625** (~10.7M), which is appropriate for the dataset size and avoids severe overfitting.

### Key Implementation Details

Three components were implemented from scratch:

1. **Scaled Dot-Product Attention**: Computes $\text{weights} = QK^T / \sqrt{d_k}$, applies a causal mask, then computes $\text{attention} = \text{dropout}(\text{softmax}(\text{weights}))V$ [3].

2. **Multi-Head Attention**: Concatenates the outputs of all attention heads and projects through a linear layer with dropout: $\text{MultiHead}(x) = \text{Dropout}(\text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O)$ [3].

3. **Sinusoidal Positional Encoding**: Precomputes position embeddings using $PE(pos, 2i) = \sin(pos / 10000^{2i/d_{model}})$ and $PE(pos, 2i+1) = \cos(pos / 10000^{2i/d_{model}})$ [3].

## b. Model Evaluation

### Training Performance

The model was trained for 2000 iterations using AdamW optimization. Training and validation losses were printed every 50 steps via `tqdm`, showing steady convergence throughout training.

### Generated Text Sample

ROMEO: I am I not a man time but the more than which I may not say well under than which she she seems not on him.

FLORIZEL: How!

POLIXENES: Lord Marshal: Go, sir, what knows your house?

FLORIZEL: Well, dispatch, consort.

FLORIZEL: So many more; though

### Qualitative Assessment

- **Overall**: The output demonstrates that the Transformer has learned meaningful patterns from the training data, including dialogue structure, character naming conventions, and Early Modern English vocabulary.

## Reflection

This lab provided hands-on experience implementing the core building blocks of a Transformer from scratch — particularly the attention mechanism, multi-head attention, and positional encoding. The most important takeaway is how the scaled dot-product attention and causal masking work together to enable autoregressive text generation.