### ****Task 4: Training Loop Implementation (MTL)****

#### **Key Decisions & Explanations**

1. **Freezing the Transformer Backbone:**

The BERT model is frozen (requires\_grad=False) to retain its pre-trained knowledge and reduce computational load. This way, we leverage its language understanding while focusing training on the task-specific heads, avoiding unnecessary updates and speeding up training.

1. **Task-Specific Training Flexibility:**

The training loop allows selective training of either the classification head, the NER head, or both. Flags (train\_classification and train\_ner) control this behavior, making it adaptable for different scenarios—whether one task needs more fine-tuning or data is imbalanced across tasks.

1. **Selective Parameter Updates:**

Only the parameters of the active task-specific heads are updated. This prevents wasting resources on parts of the model that don’t need training, keeping memory usage low and optimization focused.

1. **Dynamic Forward Pass:**

The model only computes outputs for the active tasks, skipping unnecessary calculations. This keeps training efficient, avoiding redundant work when only classification or NER is required.

1. **Loss Calculation & Backpropagation:**

Loss is calculated only for the tasks being trained. If both tasks are active, their losses are summed before backpropagation. This ensures balanced updates without unnecessary computations.