Scalable Early Detection of Mental Health Signals in Social Media Text

Emmanuel G. Maminta[*]

Artificial Intelligence Graduate Program at the University of the Philippines

Early detection of mental health signals in social media text can enable timely interventions and reduce the risk of crises. This paper presents a systematic study of encoder adaptation strategies for automated screening of mental-health indicators in social media posts, with a primary focus on BERT-based encoders. Using a curated dataset labeled across seven clinically informed categories, we compare frozen-encoder classifiers, full encoder fine-tuning, and parameter-efficient approaches including low-rank adaptation (LoRA). Through extensive hyperparameter sweeps and controlled ablations we evaluate layer-wise unfreezing schedules, encoder/classifier head learning-rate splits, and regularization schemes. We show that full encoder fine-tuning achieves the highest classification performance with a peak F1-score of 0.83, substantially outperforming frozen and low-rank alternatives while maintaining strong generalization and balanced class sensitivity. Partial fine-tuning and LoRA offer notable compute and memory savings but require careful parameterization to avoid instability at large LoRA ranks. Overall, the findings demonstrate that a fine-tuned BERT encoder can serve as a reliable foundation for early diagnostic support, provided its deployment remains grounded in ethical design and clinical responsibility.

Date: October 10, 2025
Correspondence: emmanuel.maminta@eee.upd.edu.ph

1 Introduction

Mental health disorders affect hundreds of millions of individuals worldwide, yet many cases remain undiagnosed until symptoms reach critical severity. Early identification is vital, enabling timely intervention that can prevent crises and mitigate the progression of serious conditions. Textual data offers an underexplored but powerful diagnostic signal: social media posts, online forums, therapy transcripts, and personal writings often encode linguistic patterns reflective of psychological states. Leveraging these linguistic cues through natural language processing (NLP) provides a pathway toward proactive, data-driven mental health support.

Among contemporary NLP architectures, BERT (Bidirectional Encoder Representations from Transformers) [1] stands as a foundation model. Built upon the Transformer encoder and driven by self-attention mechanisms, BERT captures bidirectional contextual dependencies, enabling fine-grained semantic understanding. Its capacity to represent subtle linguistic nuances makes it particularly suited for identifying early indicators of mental distress, including depression, anxiety, or suicidal ideation.

This work implements and fine-tunes a BERT-based model for multi-class mental health classification across seven diagnostic categories: normal, depression, suicidal, anxiety, stress, bipolar, and personality disorder. We reconstruct BERT's architecture from first principles, align it with pre-trained parameters, and adapt it to a curated corpus of mental health discourse [2]. Beyond empirical performance, we examine the broader implications of deploying BERT in clinical and online settings as depicted in Figure 1, where interpretability, fairness, and ethical responsibility are as crucial as predictive accuracy. The main contributions are as follows:

- Implement and fine-tune a BERT-based classifier for seven-way mental health diagnosis, evaluating full encoder adaptation against frozen and low-rank (LoRA) fine-tuning schemes.
- Conduct systematic ablations on learning rate schedules, batch size, and dropout regularization to quantify their influence on optimization stability and generalization.

[*] Academic work conducted for a class; not a formal publication.

- Demonstrate that full encoder fine-tuning outperforms parameter-efficient methods by a large margin, establishing it as a strong and interpretable baseline for early mental health detection in text.

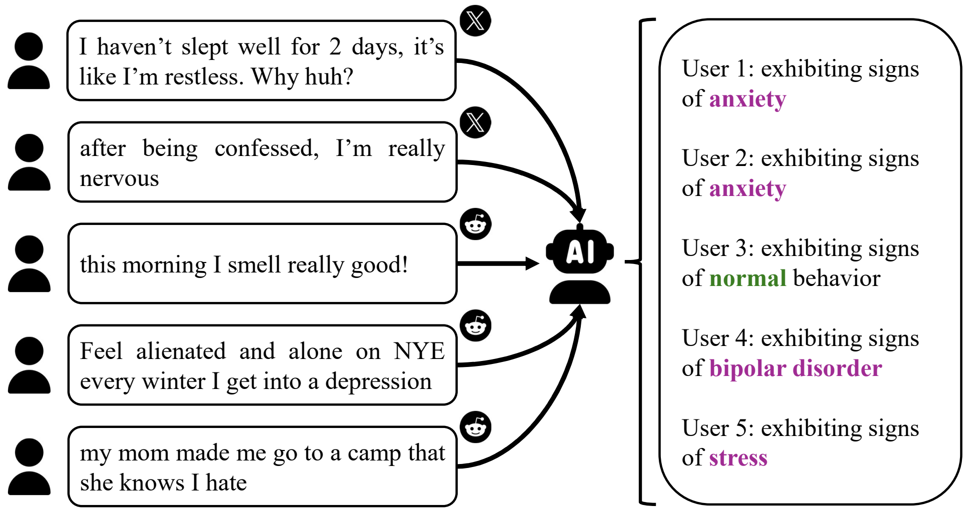
I haven't slept well for 2 days, it's like I'm restless. Why huh?
after being confessed, I'm really nervous
this morning I smell really good!
Feel alienated and alone on NYE every winter I get into a depression
my mom made me go to a camp that she knows I hate
AI
User 1: exhibiting signs of anxiety
User 2: exhibiting signs of anxiety
User 3: exhibiting signs of normal behavior
User 4: exhibiting signs of bipolar disorder
User 5: exhibiting signs of stress

Figure 1 End-to-end system for screening mental health signs in online discourse. A fine-tuned BERT analyzes consented user posts scraped from social media, such as X (formerly Twitter) and Reddit, to identify patterns suggesting early signs of mental illness, facilitating potential professional intervention.

2 Methods

2.1 Theoretical overview of BERT

BERT was introduced by Devlin et al. [1] as the first deeply bidirectional, fine-tuning-based language representation model. BERT conditions on both left and right context at every layer. This design enables it to capture richer semantics crucial for tasks requiring nuanced understanding, including mental health text classification as depicted in Figure 2. The empirical strength of BERT lies in its ability to generalize across tasks without heavy task-specific architectures. Its broad linguistic pre-training provides general knowledge of semantics, while fine-tuning adapts these representations to sensitive domains such as online mental health discourse. This dual capacity makes BERT particularly effective for detecting early warning signals in text, where meaning often depends on subtle variations in tone, phrasing, and context. At its core, BERT is a multi-layer Transformer encoder [3], with self-attention as the central operation. Given an input sequence of n tokens $\{x_1, \dots, x_n\}$, each token is mapped into a hidden vector, forming the input matrix $H^{(0)} \in \mathbb{R}^{n \times H}$ to the first encoder block. For a single attention head, the output is

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right) V, \tag{1}$$

where the query, key, and value projections are

$$Q = H^{(0)} W^Q, \quad K = H^{(0)} W^K, \quad V = H^{(0)} W^V,$$

with $W^Q, W^K, W^V \in \mathbb{R}^{H \times d_h}$ and $d_h = H/h$ the per-head dimension. Multi-head attention applies h such projections in parallel, concatenates the resulting head outputs, and projects them back to $\mathbb{R}^H$, enabling the model to capture dependencies across multiple representational subspaces. Each encoder block then combines multi-head self-attention with a position-wise feed-forward network, joined via residual connections and layer normalization.

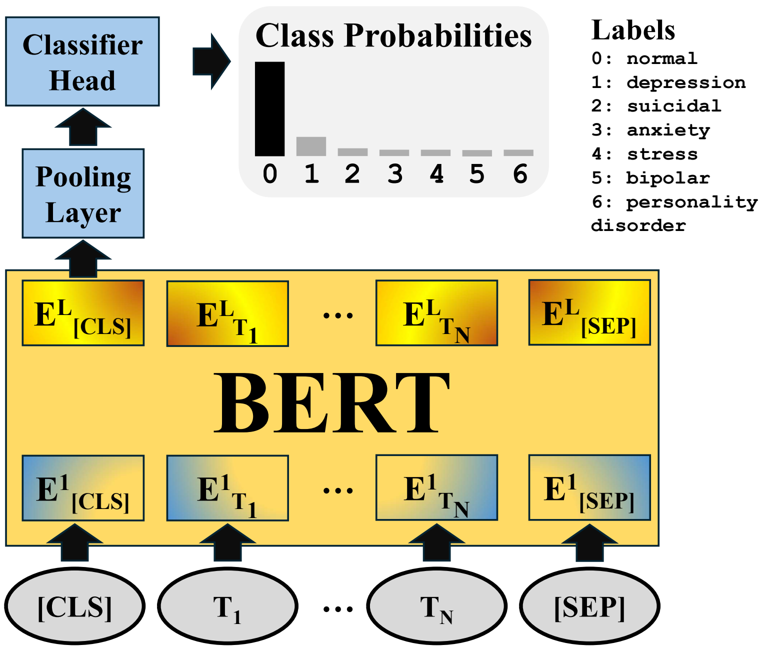

Figure 2 Fine-tuning BERT for classification. The model is initialized with pre-trained parameters, processes input tokens $\{T_1, \ldots, T_n\}$ with special tokens `[CLS]` and `[SEP]`, and extracts the final hidden state $\mathbf{E}^L_{[CLS]}$. A pooling layer maps $\mathbf{E}^L_{[CLS]}$ into the pooled representation c_i, which is then passed to the classifier head for task-specific prediction.

BERT uses WordPiece embeddings with a 30K-token vocabulary [1]. Input representation is formed by summing token, segment, and positional embeddings as shown in Figure 3. The special tokens `[CLS]` and `[SEP]` serve as classification anchors and segment delimiters, respectively. For classification tasks, the final hidden state corresponding to `[CLS]` (denoted $C \in \mathbb{R}^H$) is fed into a task-specific output layer, where H is the size of the contextualized embeddings output by the last BERT encoder layer.

Pre-training objectives. The original BERT framework is trained on two unsupervised tasks. The first is masked language modeling (MLM), inspired by the Cloze task [4], where 15% of tokens are masked and the model predicts the original token identity. This forces the encoder to build deep bidirectional representations. The second is next sentence prediction (NSP), where the model predicts whether sentence B follows sentence A in the corpus. Together, MLM and NSP encourage both token-level understanding and inter-sentence coherence.

Fine-tuning for mental health detection. Once pre-trained on large corpora, specifically BooksCorpus (800M words) [5] and English Wikipedia (2.5B words), BERT is fine-tuned for downstream tasks with minimal modifications. For early detection of mental illness, the final hidden state of the `[CLS]` token, $\mathbf{E}^L_{[CLS]} = H^{(L)}_{i,\texttt{[CLS]}}$, is passed through a pooler layer to obtain $c_i \in \mathbb{R}^H$, which serves as the sequence representation. This pooled embedding is then fed to a classifier head that maps it into diagnostic categories (e.g., depression, anxiety, suicidal), as illustrated in Figure 2. The classifier head computes logits $z_i \in \mathbb{R}^K$ for K classes, followed by softmax probabilities:

$$p(y_i = k \mid c_i) = \frac{\exp(z_{i,k})}{\sum_{j=1}^{K} \exp(z_{i,j})}. \tag{2}$$

and is trained with cross-entropy loss:

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^{B} \log p_{i,y_i}, \tag{3}$$

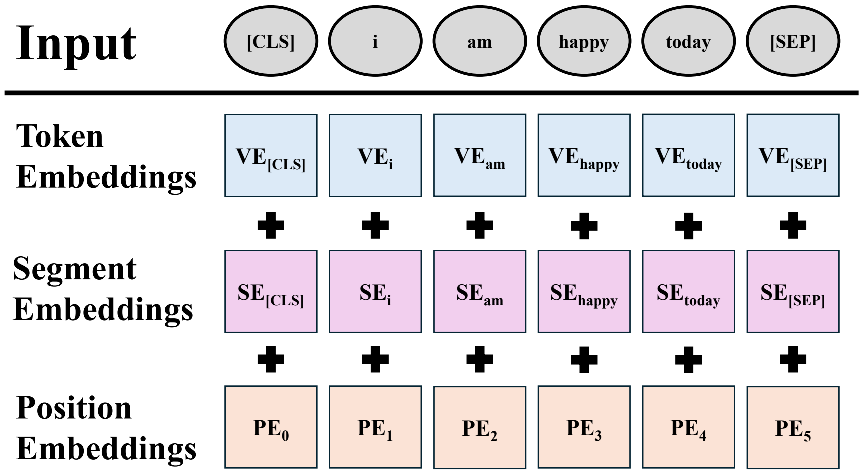

Figure 3 BERT input embeddings, computed as the sum of its component vectors: $\mathbf{E} = \mathbf{VE} + \mathbf{SE} + \mathbf{PE}$. Here, $\mathbf{VE}$ is the token embedding, $\mathbf{SE}$ is the segment embedding, and $\mathbf{PE}$ is the absolute position embedding.

where B is the batch size, y_i is the gold label for instance i, and p_{i,y_i} is the probability assigned by the model to the correct class. The bidirectional self-attention mechanism is particularly important in this domain: figurative expressions such as *"I am drowning in work"* must be distinguished from genuine indications of distress, while symptom-specific phrasing like *"I cannot get out of bed anymore"* directly signals clinical depression. By conditioning on the entire sequence, BERT leverages full contextual information to disambiguate figurative language from clinically relevant symptom descriptions.

2.2 Mental health dataset

We conduct our experiments on the *Sentiment Analysis for Mental Health* dataset available on Kaggle, a curated collection of user-generated text annotated for mental health classification [2]. The dataset aggregates and cleans raw data from multiple sources, including Reddit posts, Twitter discussions, chatbot conversation datasets [6, 7, 8, 9, 10, 11, 12, 13, 14], and prior Kaggle corpora on depression, stress, anxiety, bipolar disorder, and suicidal ideation. By consolidating these sources into a unified resource, the dataset provides a broad and diverse representation of language associated with different psychological states. Sampled statements along with corresponding labels are shown in Table 1. Figure 4 reveals a right-skewed distribution, where the majority of textual statements across training, validation, and test splits contain fewer than 200 tokens. Each statement is paired with one of seven labels: `normal`, `depression`, `suicidal`, `anxiety`, `stress`, `bipolar`, and `personality disorder`. The dataset composed of 52,681 online texts was divided into training, validation, and test sets (70%, 10%, 20%) to facilitate fair assessment of predictive performance.

Table 1 Sampled statements for each label from the dataset.

Statement	Label
`I'm feeling happy today`	`normal`
`I recently went through a breakup and she said she ...`	`depression`
`I have so many stressors in my life, all major things ...`	`suicidal`
`Have you ever felt nervous but didn't know why?`	`anxiety`
`Head noise, intrusive thoughts, obsessive ...`	`bipolar`
`Is it normal to feel a gurgling in your chest ...`	`stress`
`I feel like I've missed out on my teenage year...`	`personality disorder`

The statements are informal and conversational, reflecting the fragmented, colloquial style typical of online discourse rather than long-form narratives. This makes the dataset particularly valuable for evaluating

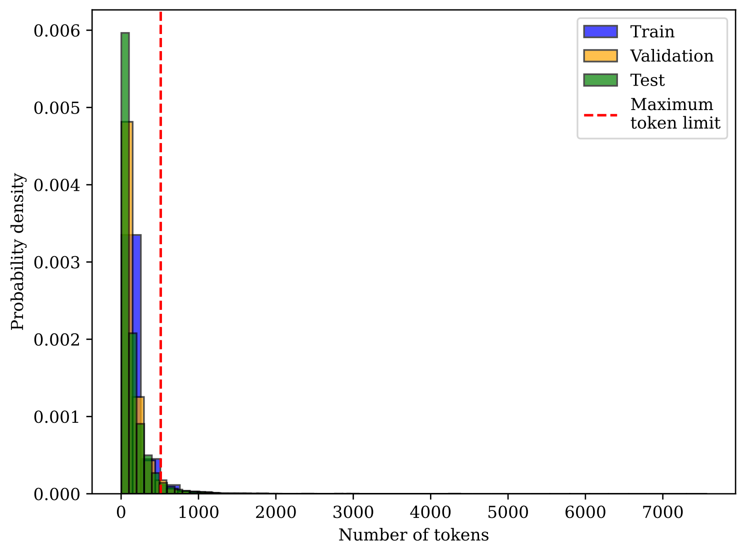

Figure 4 Token count distribution across the training, validation, and test splits. Most samples are short, clustering below 200 tokens, with a long tail extending toward the 512-token maximum limit. The vertical line denotes the truncation boundary used during pre-processing to fit the BERT tokenizer's sequence length constraint.

Transformer-based models, which must capture subtle variations in tone, phrasing, and context. However, the dataset is inherently imbalanced, with classes such as `normal` and `depression` appearing more frequently than rarer categories like `bipolar` or `personality disorder`. The small fraction of samples approaching the 512-token boundary suggests that truncation minimally affects data coverage, preserving most semantic content. The alignment of training, validation, and test curves further confirms consistent token length distributions across splits, reducing the risk of length-induced bias during training. Overall, these characteristics validate the chosen pre-processing pipeline and justify using a fixed sequence length of 512 for efficient batching without sacrificing representational coverage.

2.3 Fine-tuning BERT for downstream transfer

We adapt a pre-trained BERT model to the mental health classification task by attaching a lightweight classifier head and updating (most or all) encoder parameters using supervised signals. This subsection details the exact computational pathway from raw text to loss, decomposing every module involved in the forward and training passes. Our goal is to make fine-tuning transparent and reproducible.[1]

Input processing. Each raw statement is tokenized with WordPiece, originally proposed by Schuster and Nakajima [15], via `bert-base-uncased` tokenizer yielding a sequence of token IDs:

$$[\texttt{[CLS]}, x_1, x_2, \ldots, x_n, \texttt{[SEP]}],$$

truncated or padded to fixed length $n \le n_{\max}$. We set $n = 512$ based on the $n_{\max}$ from the original BERT tokenizer configuration. We build: `input_ids` (token indices), `token_type_ids` (all zeros for single-sequence classification), and `attention_mask` (`1` for real tokens, `0` for padding). These feed the embedding stack.

Embedding layer. The embedding module produces the initial continuous representation

$$E_i = W^{(\mathrm{tok})} x_i + W^{(\mathrm{pos})} i + W^{(\mathrm{seg})} s_i, \quad i = 0, \ldots, n-1, \tag{4}$$

where $\mathbf{VE} = W^{(\mathrm{tok})} \in \mathbb{R}^{V \times H}$, $\mathbf{PE} = W^{(\mathrm{pos})} \in \mathbb{R}^{n_{\max} \times H}$, $\mathbf{SE} = W^{(\mathrm{seg})} \in \mathbb{R}^{2 \times H}$, and H is the hidden size. A LayerNorm-Dropout pair normalizes and regularizes:

$$\tilde{E}_i = \mathrm{Dropout}(\mathrm{LayerNorm}(E_i)) \,. \tag{5}$$

[1]Experiments run on a single NVIDIA A100 GPU; seed used is `42`.

Transformer encoder stack. BERT applies L identical encoder blocks. For block $\ell \in \{1, \dots, L\}$ and token position i:

$$\begin{aligned}
&\text{(i) Multi-head attention: } Z_i^{(\ell)} = \mathrm{MHA}\big(H^{(\ell-1)}; \mathrm{mask}\big),\\
&\text{(ii) Residual + LayerNorm: } \hat{H}_i^{(\ell)} = \mathrm{LayerNorm}\Big(H_i^{(\ell-1)} + \mathrm{Dropout}(Z_i^{(\ell)})\Big),\\
&\text{(iii) Feed-forward (GELU): } F_i^{(\ell)} = \phi\Big(\hat{H}_i^{(\ell)} W_1 + b_1\Big) W_2 + b_2,\\
&\text{(iv) Residual + LayerNorm: } H_i^{(\ell)} = \mathrm{LayerNorm}\Big(\hat{H}_i^{(\ell)} + \mathrm{Dropout}(F_i^{(\ell)})\Big),
\end{aligned} \tag{6}$$

where ϕ is GELU. Each head computes:

$$\mathrm{Head}_k = \mathrm{Softmax}\left(\frac{(HW_k^Q)(HW_k^K)^\top + M}{\sqrt{d_h}}\right)(HW_k^V), \tag{7}$$

with $W_k^Q, W_k^K, W_k^V \in \mathbb{R}^{H \times d_h}$, $d_h = H/h$, and M the additive mask ($-\infty$ for padding). Heads are concatenated and projected:

$$\mathrm{MHA}(H) = \big[\mathrm{Head}_1 \mid \cdots \mid \mathrm{Head}_h\big] W^O, \tag{8}$$

where the concatenation yields a tensor in $\mathbb{R}^{n \times (h \cdot d_h)}$, and $W^O \in \mathbb{R}^{(h \cdot d_h) \times H}$ projects it back into the model's hidden dimension H.

Pooler. Although the original BERT paper does not explicitly mention a pooling step, common implementations (e.g., Hugging Face [16]) insert a small projection before the classifier head. The pooler takes the final hidden state of the `[CLS]` token and applies a linear transformation followed by a tanh nonlinearity:

$$c_i = \tanh\big(H_{i,\texttt{[CLS]}}^{(L)} W^{(\mathrm{pool})} + b^{(\mathrm{pool})}\big), \tag{9}$$

where $\mathbf{E}_{[CLS]}^{L} = H_{i,\texttt{[CLS]}}^{(L)} \in \mathbb{R}^H$ is the last-layer `[CLS]` embedding of instance i, and $W^{(\mathrm{pool})}, b^{(\mathrm{pool})}$ are learned parameters with $W^{(\mathrm{pool})} \in \mathbb{R}^{H \times H}$. The pooled representation c_i is then passed to the classifier head.

Classifier head. From the `[CLS]` embedding $c_i \in \mathbb{R}^H$ of instance i, the classifier head computes

$$z_i = c_i W^{(\mathrm{cls})} + b^{(\mathrm{cls})} \in \mathbb{R}^K, \tag{10}$$

$$p_i = \mathrm{Softmax}(z_i), \tag{11}$$

$$\mathcal{L}_w = -\frac{1}{B}\sum_{i=1}^{B} \alpha_{y_i} \log p_{i,y_i}, \tag{12}$$

where α_{y_i} denotes the weight assigned to the gold label y_i. We introduce the per-class weights α_c for $c \in C$ (so $\alpha_{y_i} = \alpha_c$ with $c = y_i$); the α_c are computed from class relative frequencies $\mathrm{portion}_c$ via inverse-frequency normalization:

$$\alpha_c = \frac{\dfrac{1}{\mathrm{portion}_c}}{\displaystyle\sum_{j=1}^{|C|} \frac{1}{\mathrm{portion}_j}} \qquad (c \in C), \tag{13}$$

which ensures $\sum_{c \in C} \alpha_c = 1$. For clarity, the relative frequency $\mathrm{portion}_c$ is

$$\mathrm{portion}_c = \frac{n_c}{\sum_{j=1}^{|C|} n_j}, \tag{14}$$

where n_c is the number of examples of class c. As shown in Table 2, the computed weights explicitly enforce this imbalance correction, ensuring that the rarest class (`personality disorder`) receives the strongest influence during training. By construction, the corresponding weights satisfy the inverse ordering:

$$\alpha_{\texttt{normal}} < \alpha_{\texttt{depression}} < \alpha_{\texttt{suicidal}} < \alpha_{\texttt{anxiety}} < \alpha_{\texttt{bipolar}} < \alpha_{\texttt{stress}} < \alpha_{\texttt{personality disorder}}.$$

Table 2 Class distribution sorted from least to most frequent, with normalized inverse-frequency weights. Lower portions (↓) correspond to higher weights (↑), highlighting the imbalance correction.

Class	**Portion (↓)**	**Weight α (↑)**
`personality disorder`	**0.02**	**0.43**
`stress`	0.05	0.17
`bipolar`	0.05	0.18
`anxiety`	0.07	0.04
`suicidal`	0.20	0.03
`depression`	0.29	0.03
`normal`	0.31	0.12

Freezing vs. full fine-tuning. We consider two regimes. In the frozen encoder setting, all Transformer parameters remain fixed and only the classifier head $(W_{\text{cls}}, b_{\text{cls}})$, and optionally the final LayerNorm scales and biases, are updated. This probes the sufficiency of generic pretrained semantic priors without domain-specific adaptation. In contrast, full fine-tuning updates the entire parameter set, allowing token- and phrase-level representations to adjust to subtle domain cues critical for mental health detection.

Optimization and parameter grouping. We use the AdamW optimizer [17], which augments Adam [18] with decoupled weight decay, ensuring stable convergence in fine-tuning. Parameters are partitioned into two groups with distinct learning rates: the encoder group (all Transformer weights) with $\eta_{\text{enc}} \in \{9.3\times10^{-6},\ 1.3\times10^{-5},\ 2.7\times10^{-5}\}$, and the classifier head with elevated $\eta_{\text{head}} \in \{1.3\times10^{-3},\ 2.7\times10^{-3},\ 9.6\times10^{-3}\}$. This "two-tier" scheme accelerates adaptation of the randomly initialized head while preserving pretrained linguistic structure in the encoder. We adopt AdamW with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$, and decoupled weight decay $\lambda = 0.01$ applied only to non-bias and non-normalization parameters:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \eta \cdot \lambda\theta_t, \tag{15}$$

where $\hat{m}_t, \hat{v}_t$ are Adam's bias-corrected first- and second-moment estimates. Gradient clipping (global norm $= 1.0$) is additionally employed to prevent rare exploding updates.

Learning rate (LR) schedule. We employ a linear warmup for the first $T_{\text{warm}} = 500$ steps, followed by cosine decay until $T_{\text{decay}} = 30{,}000$ steps, with a floor of $\eta_{\min} = 10^{-7}$. For step index t, the schedule is given by

$$\eta(t) = \begin{cases} \eta_0 \cdot \dfrac{t}{T_{\text{warm}}}, & 0 \le t < T_{\text{warm}}, \\ \eta_{\min} + \frac{1}{2}\big(1 + \cos\big(\pi \cdot \frac{t - T_{\text{warm}}}{T_{\text{decay}} - T_{\text{warm}}}\big)\big) \cdot (\eta_0 - \eta_{\min}), & T_{\text{warm}} \le t \le T_{\text{decay}}, \\ \eta_{\min}, & t > T_{\text{decay}}. \end{cases} \tag{16}$$

Here η_0 is the peak LR for each parameter group (encoder η_{enc}, classifier head η_{head}). Both groups share the same warmup and decay schedule. Warmup mitigates early training instability when gradients from a randomly initialized head backpropagate into pre-trained layers, while cosine decay prevents over-shooting and provides a smooth annealing toward $\eta_{\min}$.

Regularization. Dropout is the primary stochastic regularizer with default dropout rate $p_{\text{dropout}} = 0.1$ (ablations at $p_{\text{dropout}} = \{0.2, 0.3\}$). Increasing p_{dropout} modestly reduces overfitting but slows convergence by attenuating signal flow in attention and feed-forward layers. No label smoothing was applied; class distributions remain one-hot, preserving sharp decision boundaries that aid minority-class recall.

Training loop pseudocode. Algorithm 1 outlines the fine-tuning procedure. For each epoch, batches of tokenized inputs $(\texttt{input\_ids}, \texttt{type\_ids}, \texttt{mask}, y)$ are embedded and passed through L encoder blocks, each following the update rules in Equation 6. Within these, multi-head attention (Equations 7–8) contextualizes

Algorithm 1 BERT supervised fine-tuning (SFT)

```
1:  for step $t = 1$ to $T$ do
2:      for batch $(\texttt{input\_ids}, \texttt{type\_ids}, \texttt{mask}, y)$ do
3:          $H \leftarrow \text{Encoder}(\texttt{input\_ids}, \texttt{type\_ids}, \texttt{mask})$
4:          $c \leftarrow H[\texttt{[CLS]}]$
5:          $z \leftarrow cW_{\text{cls}} + b_{\text{cls}}$
6:          $p \leftarrow \text{softmax}(z)$
7:          $\mathcal{L} \leftarrow -\frac{1}{B} \sum_i \log p_{i,y_i}$
8:          Backpropagate $\nabla \mathcal{L}$
9:          Update encoder params with LR $\eta_{\text{enc}}(t)$ (if unfrozen)
10:         Update head params with LR $\eta_{\text{head}}(t)$
11:     end for
12:     Evaluate on validation set; save best checkpoint by F1-score
13: end for
```

tokens bidirectionally. After the final layer, the pooled [CLS] embedding c is obtained and transformed into logits using the classifier head in Equation 10. Softmax then yields class probabilities (Equation 11), and the loss is computed as cross-entropy (Equation 12). Gradients are backpropagated; encoder parameters are updated with learning rate $\eta_{\text{enc}}(t)$ if unfrozen, while the classifier head uses $\eta_{\text{head}}(t)$. Following each epoch, validation performance is monitored, and the best checkpoint is saved according to F1-score.

3 Results and discussion

Tables 3–5 present the performance of our BERT-based classifier under different learning rates, batch sizes, and fine-tuning strategies. We evaluate accuracy, precision, recall, and F1-score for both frozen and fully fine-tuned BERT models.

Table 3 Performance with BERT learning rate 9.3×10^{-6} and classifier head learning rate 9.6×10^{-3}.

Batch size	Epochs	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)	Full FT
8	2	0.5149	0.5625	0.4417	0.4948	✗
8	3	0.5402	0.5785	0.4689	0.5180	✗
8	4	0.5183	0.6173	0.3419	0.4400	✗
8	2	0.8254	0.7873	0.8344	0.8102	✓
8	3	0.8210	0.7710	0.8385	0.8033	✓
8	4	0.8225	0.7984	0.8435	**0.8203**	✓
16	2	0.5624	0.5065	0.4815	0.4937	✗
16	3	0.5522	0.5051	0.5337	0.5190	✗
16	4	0.5699	0.5133	0.5175	0.5154	✗
16	2	0.8159	0.7620	0.8316	0.7953	✓
16	3	0.8213	0.7867	0.8420	0.8134	✓
16	4	0.8184	0.7956	0.8351	0.8149	✓
32	2	0.5520	0.5142	0.4890	0.5013	✗
32	3	0.6035	0.5421	0.5201	0.5309	✗
32	4	0.6321	0.5756	0.5275	0.5505	✗
32	2	0.7905	0.7149	0.8153	0.7618	✓
32	3	0.8107	0.7687	0.8229	0.7949	✓
32	4	0.8201	0.8085	0.8094	0.8089	✓

Fine-tuning drives performance. Tables 3–5 demonstrate the sharp contrast between frozen BERT encoders and full fine-tuning. With a frozen backbone, F1-scores plateau between 0.44 and 0.55 across all batch sizes

Table 4 Performance with BERT learning rate 2.7×10^{-5} and classifier head learning rate 2.7×10^{-3}.

Batch size	Epochs	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)	Full FT
8	2	0.6263	0.5432	0.4943	0.5176	✗
8	3	0.5875	0.4853	0.5514	0.5162	✗
8	4	0.6202	0.5363	0.5626	0.5491	✗
8	2	0.8250	0.7980	0.8220	0.8098	✓
8	3	0.8277	0.8006	0.8345	0.8172	✓
8	4	0.8383	0.8146	0.8432	**0.8287**	✓
16	2	0.5993	0.5266	0.5354	0.5310	✗
16	3	0.6262	0.5275	0.5579	0.5423	✗
16	4	0.6195	0.5100	0.5645	0.5359	✗
16	2	0.8239	0.7702	0.8232	0.7958	✓
16	3	0.8305	0.7895	0.8428	0.8152	✓
16	4	0.8328	0.7918	0.8387	0.8146	✓
32	2	0.5481	0.4832	0.4820	0.4826	✗
32	3	0.5781	0.5022	0.5597	0.5294	✗
32	4	0.5813	0.5323	0.5420	0.5371	✗
32	2	0.7728	0.7521	0.8181	0.7837	✓
32	3	0.8146	0.7761	0.8379	0.8058	✓
32	4	0.7965	0.7725	0.8302	0.8004	✓

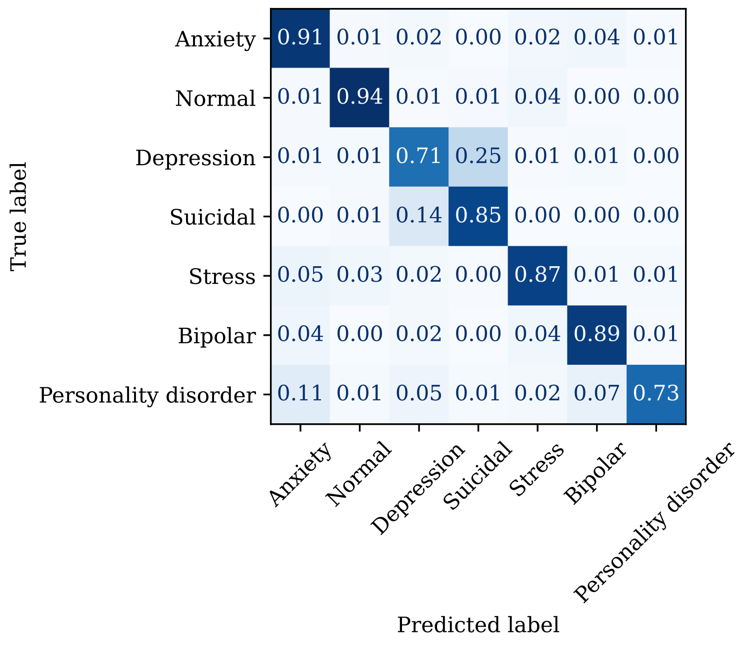

Figure 5 Normalized confusion matrix of the best-performing model checkpoint under full fine-tuning. Values represent class-wise prediction proportions normalized by true label counts. The model achieves strong discrimination across most categories, with highest recall for `normal` and `suicidal` classes, and minor confusion between semantically related conditions such as `depression` and `suicidal`, and between `anxiety` and `personality disorder`.

Table 5 Performance with BERT learning rate 1.3×10^{-5} and classifier head learning rate 1.3×10^{-3}.

Batch size	Epochs	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)	Full FT
8	2	0.6224	0.5483	0.4720	0.5073	✗
8	3	0.6036	0.5323	0.5196	0.5259	✗
8	4	0.6517	0.5443	0.5413	0.5428	✗
8	2	0.8223	0.7781	0.8415	0.8085	✓
8	3	0.8402	0.8187	0.8417	**0.8301**	✓
8	4	0.8368	0.8137	0.8453	0.8292	✓
16	2	0.5350	0.5133	0.4935	0.5032	✗
16	3	0.5890	0.5060	0.4776	0.4914	✗
16	4	0.6369	0.5211	0.5705	0.5447	✗
16	2	0.8001	0.7666	0.8076	0.7866	✓
16	3	0.8257	0.7651	0.8284	0.7955	✓
16	4	0.8394	0.8115	0.8491	0.8299	✓
32	2	0.5618	0.5027	0.5078	0.5053	✗
32	3	0.6113	0.5505	0.5011	0.5247	✗
32	4	0.6107	0.5029	0.5753	0.5366	✗
32	2	0.8036	0.7725	0.8259	0.7983	✓
32	3	0.8294	0.7934	0.8413	0.8166	✓
32	4	0.8200	0.7809	0.8465	0.8124	✓

and epochs. In contrast, full fine-tuning consistently surpasses 0.80, even under modest hyperparameter settings. For example, at batch size 32 and four epochs in Table 3, freezing BERT yields an F1-score of 0.5505, while full fine-tuning reaches 0.8089, an absolute gain of more than 25 points. This large gap reflects the inability of static embeddings to reallocate representational emphasis toward domain-specific lexical cues, such as idiomatic expressions of ideation. Fine-tuning allows attention heads to retune inter-token salience and feed-forward filters to sharpen nonlinear decision boundaries, whereas the shallow classifier head alone cannot compensate for a misaligned latent geometry. This is because a frozen encoder supplies generic embeddings trained on BERT's pre-training data which capture broad semantics but fail to adapt to the distinct discourse of mental health text from online forums. Figurative language such as *"I feel like I'm drowning in my own thoughts"* and symptom-specific phrasing like *"I haven't slept in three nights"* require encoder-level adaptation to map these domain cues to clinically relevant categories. The decisive gains from fine-tuning confirm that domain adaptation of the encoder is indispensable for reliable early detection.

The normalized confusion matrix of the fully fine-tuned BERT model shown in Figure 5 reveals strong per-class separability, confirming that full encoder adaptation enables precise differentiation of nuanced mental health expressions. The model achieves near-perfect discrimination for the `normal` class and robust performance on clinically significant labels such as `suicidal` and `bipolar`. Residual misclassifications primarily occur between conceptually and linguistically related categories. `depression` and `suicidal` posts share overlapping emotional tone and lexical cues (e.g., hopelessness, self-harm intent), while `anxiety` and `personality disorder` exhibit common affective markers like panic, instability, or self-doubt. These boundary confusions are expected in real-world text, where mental health narratives often exhibit comorbid or ambiguous patterns. Overall, the structure of the confusion matrix confirms that full fine-tuning allows BERT to capture fine-grained distinctions across disorders while maintaining balanced sensitivity and specificity.

Learning rate and convergence. We also tested three learning rate configurations to evaluate how optimization dynamics differ between frozen and fully fine-tuned models. When BERT is frozen, higher learning rates for the classifier head accelerate convergence, because the classifier head alone must adapt to static embeddings. For instance, with BERT (backbone) learning rate fixed at 2.7×10^{-5} and the classifier head trained at 2.7×10^{-3} (Table 4), the model quickly attains F1-scores above 0.80 under full fine-tuning but stalls near 0.52 when the encoder is frozen.

For fully fine-tuned models, peak performance arises with moderate learning rates. Excessively high rates destabilize encoder weights and degrade the rich linguistic priors captured during pre-training, a phenomenon

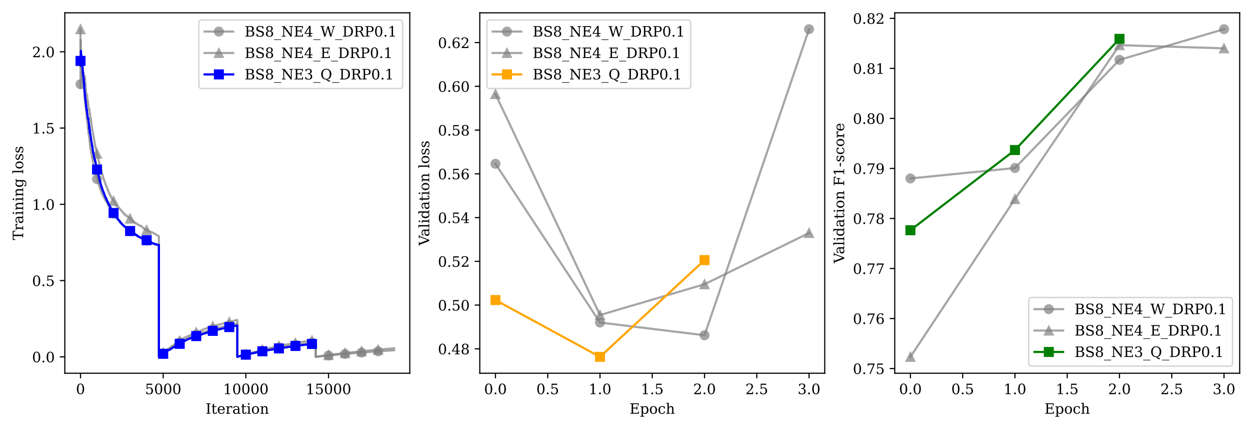

Figure 6 Training and validation dynamics for the best-performing configurations across three learning rate regimes: Q ($\eta_{\text{BERT}} = 1.3 \times 10^{-5}$, $\eta_{\text{head}} = 1.3 \times 10^{-3}$), W ($\eta_{\text{BERT}} = 2.7 \times 10^{-5}$, $\eta_{\text{head}} = 2.7 \times 10^{-3}$), and E ($\eta_{\text{BERT}} = 9.3 \times 10^{-6}$, $\eta_{\text{head}} = 9.6 \times 10^{-3}$), all with dropout 0.1. Highlighted curves mark the best checkpoint's performance. Each subplot shows (left) training loss versus iteration, (middle) validation loss versus epoch, and (right) validation F1-score versus epoch. BS: batch size, NE: number of training epochs.

akin to catastrophic forgetting, while excessively low rates slow adaptation. As shown in Figure 6, the configuration labeled Q ($\eta_{\text{BERT}} = 1.3 \times 10^{-5}$, $\eta_{\text{head}} = 1.3 \times 10^{-3}$) achieves the most stable convergence, with steadily decreasing loss and consistently improving validation F1-score. In contrast, W ($\eta_{\text{BERT}} = 2.7 \times 10^{-5}$, $\eta_{\text{head}} = 2.7 \times 10^{-3}$) shows oscillations indicative of mild overfitting, while E ($\eta_{\text{BERT}} = 9.3 \times 10^{-6}$, $\eta_{\text{head}} = 9.6 \times 10^{-3}$) converges slowly, suggesting underfitting due to insufficient encoder updates.

The optimal setting, corresponding to configuration Q and Table 5, yields the best F1-score of 0.8301 at batch size 8 and three epochs. This balance allows the encoder to adjust gradually while the classifier head tracks domain-specific decision boundaries. Distinct learning rates for head and encoder therefore play a critical role: they prevent catastrophic forgetting, stabilize optimization, and ensure convergence toward domain-adapted representations.

Batch size effects. Batch size further shapes the trade-off between convergence speed and stability. Smaller batches, such as size 8, produce faster initial improvements due to more frequent weight updates. This stochasticity injects beneficial gradient noise that helps escape sharp local minima under skewed distributions. Larger batches, such as size 32, yield steadier convergence across epochs and reduce variance in validation performance. This pattern is evident in Tables 3–5: batch size 8 reaches F1-scores above 0.82 within three epochs, while batch size 32 converges more gradually but stabilizes around 0.81-0.82. Thus, smaller batches act as implicit regularization through noisy gradients, while larger ones promote smoother but less exploratory updates.

Class imbalance and sensitivity. The dataset also inherits the imbalance common in internet-derived mental health forums, where certain categories such as general stress appear more frequently than severe conditions like suicidal ideation. Without full fine-tuning, the model tends to collapse onto majority classes, as reflected in frozen backbone recall scores that drop as low as 0.34 (Table 3). Once fine-tuned, recall improves markedly, rising above 0.83 in the best runs. This improvement arises because encoder adaptation redistributes representational capacity to minority categories. Rare but critical expressions, such as *"I bought a rope yesterday and don't know why"*, are unlikely to be frequent in the training set; static embeddings struggle to elevate such sequences into meaningful decision boundaries. Fine-tuning allows attention weights to shift toward these signals, producing balanced precision and recall even under imbalance. In high-stakes applications, recall is particularly important, since missing a positive case can have severe consequences, whereas false positives can often be resolved through human review.

Impact of dropout. Table 6 and Figure 7 jointly examine the effect of varying dropout rates on training stability and generalization. The results reveal a decisive conclusion: full fine-tuning consistently surpasses frozen backbone training across all dropout settings. At $p_{\text{dropout}} = 0.1$, the model achieves the most stable convergence and the highest F1-score of 0.8301, representing a gain of over 30 absolute points relative to the frozen encoder baseline. Accuracy, precision, and recall all improve markedly, with recall showing the greatest increase indicating that moderate regularization enhances minority-class sensitivity without disrupting learned attention patterns.

Table 6 Performance with batch size 8, no. of training epochs 3, BERT learning rate 1.3×10^{-5}, classifier head learning rate 1.3×10^{-3}, and varying dropout rates ($p_{\text{dropout}} \in \{0.1, 0.2, 0.3\}$).

p_{dropout}	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)	Full FT
0.1	0.6036	0.5323	0.5196	0.5259	✗
0.2	0.5812	0.5286	0.5029	0.5154	✗
0.3	0.5825	0.4865	0.4998	0.4930	✗
0.1	0.8402	0.8187	0.8417	**0.8301**	✓
0.2	0.8277	0.7884	0.8487	0.8175	✓
0.3	0.8355	0.7993	0.8430	0.8205	✓

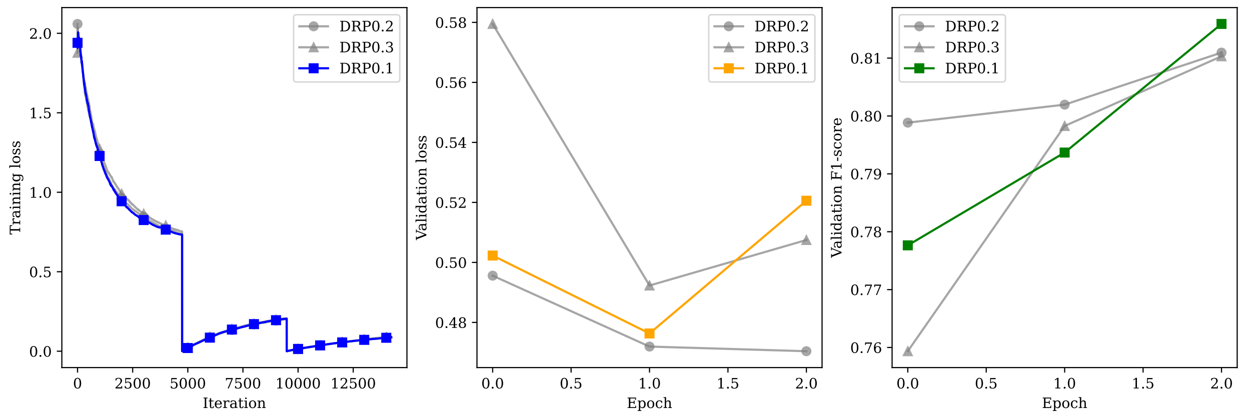

Figure 7 Training and validation loss/F1-score curves for different dropout rates ($p_{\text{dropout}} \in \{0.1, 0.2, 0.3\}$) under full fine-tuning. The model with $p_{\text{dropout}} = 0.1$ achieves the most stable convergence and highest F1-score, while higher dropout rates lead to slower recovery and greater variance, indicating excessive regularization.

The training curves further confirm that higher dropout rates ($p_{\text{dropout}} \in \{0.2, 0.3\}$) lead to slower recovery and greater variance between training and validation F1-scores, a sign of underutilized capacity and over-regularization. In contrast, $p_{\text{dropout}} = 0.1$ produces a smooth, monotonic decline in loss and a stable rise in F1-score, signifying an optimal balance between regularization and gradient flow. When the encoder is frozen, dropout variation has negligible effect, with F1-score fluctuating narrowly between 0.49 and 0.53. Thus, once encoder adaptation is enabled, dropout tuning becomes secondary, with optimal performance plateauing near $p_{\text{dropout}} = 0.1$.

Low-rank adaptation (LoRA) fine-tuning. Table 7 reports the impact of LoRA rank and injection points on model performance. LoRA on the `QKV` projection layers consistently offered the best trade-off between accuracy and parameter efficiency. At rank 64, `QKV`-LoRA achieved 0.747 accuracy and 0.718 F1-score while updating only 3.14% of parameters, suggesting that attention-level adaptation is more effective than modifying the `pooler` or `dense` feedforward layers. Here, `dense` refers exclusively to all feedforward layers, excluding QKV projections, while `all` covers all linear layers.

LoRA on `dense` or `all` layers caused substantial performance drops despite higher trainable parameter counts, likely due to overparameterization and interference with pre-trained weights. Interestingly, increasing the rank from 8 to 512 did not improve downstream performance; in fact, high-rank LoRA often collapsed accuracy to near random-chance. This counterintuitive behavior indicates that excessive adaptation capacity can destabilize optimization, particularly under small-batch and limited-data regimes, and that merely increasing rank does not guarantee better feature learning. Overall, attention-level LoRA, particularly on `QKV`, achieves efficient specialization with minimal overhead, while `pooler`-level injection offers moderate gains and `dense`- or `all`-layer LoRA proves largely ineffective. Future work could explore hybrid or sparse strategies to balance adaptability and stability.

Table 7 Performance of LoRA fine-tuning with batch size 8, 3 training epochs, BERT learning rate 1.3×10^{-5}, classifier head learning rate 1.3×10^{-5}, and LoRA scaling factor 32. The "**Applied to**" column indicates which layers the LoRA weights are injected: `all` = all dense layers, `pooler` = pooler dense layer only, `QKV` = attention projection matrices, `dense` = all feedforward dense layers excluding QKV projections and pooler dense layer.

Rank	Applied to	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)	Trainable params.
8	`pooler`	0.6325	0.5089	0.541	0.5245	17,671 (0.02%)
8	`QKV`	0.7288	0.6794	0.7415	0.7091	447,751 (0.41%)
8	`dense`	0.7281	0.6792	0.6967	0.6878	742,663 (0.67%)
8	`all`	0.3102	0.0443	0.1429	0.0677	1,344,775 (1.21%)
64	`pooler`	0.6361	0.5344	0.5884	0.5601	103,687 (0.09%)
64	`QKV`	0.7474	0.7012	0.7353	**0.7178**	3,544,327 (3.14%)
64	`dense`	0.4541	0.1271	0.2465	0.1677	5,903,623 (5.12%)
64	`all`	0.3102	0.0443	0.1429	0.0677	10,720,519 (8.92%)
512	`pooler`	0.706	0.6047	0.6792	0.6398	791,815 (0.72%)
512	`QKV`	0.3102	0.0443	0.1429	0.0677	28,316,935 (20.55%)
512	`dense`	0.3102	0.0443	0.1429	0.0677	47,191,303 (30.12%)
512	`all`	0.3102	0.0443	0.1429	0.0677	85,726,471 (43.92%)

Increasing the batch size from 8 to 32 led to more stable gradient updates and mitigated the performance collapse seen at higher ranks. As shown in Table 8, the `QKV` and `dense` configurations both achieved substantial gains (F1-scores of 0.76 and 0.75, respectively) at rank 64, while `pooler`-only adaptation improved moderately. However, `all`-layer injection continued to fail, suggesting persistent instability in overparameterized LoRA setups even under smoother optimization.

Table 8 Performance of LoRA fine-tuning with batch size 32 across different injection points and ranks.

Rank	Applied to	Accuracy (↑)	Precision (↑)	Recall (↑)	F1-score (↑)
64	`pooler`	0.6187	0.5190	0.5689	0.5428
64	`QKV`	0.7838	0.7258	0.7979	**0.7601**
64	`dense`	0.7770	0.7033	0.7926	0.7453
64	`all`	0.2022	0.0289	0.1429	0.0481
512	`pooler`	0.6410	0.5550	0.6403	0.5946
512	`QKV`	0.7531	0.6948	0.7344	0.7141
512	`dense`	0.2022	0.0289	0.1429	0.0481
512	`all`	0.3102	0.0443	0.1429	0.0677

4 Conclusion

Full fine-tuning of BERT proved essential for accurate detection of early mental health signals in online text, yielding an F1-score of 0.83 through balanced encoder-head learning rates and moderate dropout. Encoder adaptation emerged as the key determinant of domain transfer, enabling robust contextual understanding of psychological cues. In contrast, LoRA-based adaptation underperformed due to overparameterization and interference between injected and pre-trained weights, highlighting stability limits in low-rank adaptation.

These results affirm the importance of principled fine-tuning for sensitive, domain-specific language tasks. Future deployment of such models must prioritize fairness, transparency, and privacy to ensure ethical and socially responsible application in mental health contexts.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[2] S. Sarkar, "Sentiment Analysis for Mental Health," 2023. [Online]. Available: https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health/data

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] W. L. Taylor, "Cloze procedure: A new tool for measuring readability," *Journalism quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[5] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[6] K. Rajani, "3K Conversations Dataset for ChatBot," 2022. [Online]. Available: https://www.kaggle.com/datasets/kreeshrajani/3k-conversations-dataset-for-chatbot

[7] ——, "Human Stress Prediction," 2022. [Online]. Available: https://www.kaggle.com/datasets/kreeshrajani/human-stress-prediction

[8] S. Saha, "Students anxiety and depression dataset," 2022. [Online]. Available: https://www.kaggle.com/datasets/sahasourav17/students-anxiety-and-depression-dataset

[9] V. Shinde, "Depression: Reddit Dataset (Cleaned)," 2022. [Online]. Available: https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned

[10] S. A. Mahmud and N. Islam, "Suicidal Tweet Detection Dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/aunanya875/suicidal-tweet-detection-dataset

[11] N. Ghoshal, "Reddit Mental Health Data," 2023. [Online]. Available: https://www.kaggle.com/datasets/neelghoshal/reddit-mental-health-data

[12] A. Kandhari, "Suicidal Mental Health Dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/aradhakkandhari/suicidal-mental-health-dataset

[13] M. V. Patricia, "Mental Health Data (Bipolar)," 2023. [Online]. Available: https://www.kaggle.com/datasets/michellevp/mental-health-dataset-bipolar

[14] ——, "Mental Health Data (Anxiety)," 2023. [Online]. Available: https://www.kaggle.com/datasets/michellevp/predicting-anxiety-in-mental-health-data

[15] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 5149–5152.

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[18] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.