

HYBRID QUANTUM OPTIMIZATION OF DISTILBERT FOR EFFICIENT NATURAL LANGUAGE PROCESSING

Dalydas Karanath^{1*}, Dr. Love Kumar Mittal²

^{1*}Research Scholar, Department of Computer Engineering and Applications Mangalayatan University, Aligarh, dalydask@gmail.com, ORCID ID: 0009-0004-6751-9787

²Associate Professor, Department of Computer Engineering and Applications Mangalayatan University, Aligarh, love.mittal@mangalayatan.edu.in

1. Abstract

The authors created a hybrid quantum-classical model for efficient sentiment classification based on Frozen Distilled Bidirectional Encoder Representations from Transformers (frozen dBERT), supervised dimensionality reduction, variational quantum learning, and residual integration. The binary Stanford Sentiment Treebank had a total of 67,349 training observations. Following contradiction, duplication and overlap removal, 66,971 observations were left, of which 60,274 were used for training, and 6,697 for internal validation. There were 872 observations on the official development partition that were not touched. The embeddings, which had a dimension of 768, were mean-pooled to 4, 6, or 8 dimensions. The six-dimensional bottleneck was chosen and represented in the variational circuit with six qubits. The selected circuit was composed of 12 completed trials and 14 pruned trials, and had 3 layers, 36 quantum parameters, 52 logical gates and a circuit depth of 16. The classical six-dimensional bottleneck achieved 0.8360 accuracy, 0.8427 harmonic-mean classification score, 0.9184 receiver operating characteristic area under the curve and 0.6724 Matthews correlation coefficient on the official development partition. It was able to lower the dimensionality of the representation by 99.219 percent and obtain a model-head latency of 0.000607 milliseconds per observation. The quantum-only classifier had an accuracy of 0.6663 and a harmonic-mean classification score of 0.6914, whereas the residual hybrid kept a zero-quantum contribution. The results show efficient classical compression with no quantum advantage for the simulation of six-qubit states.

2. Keywords: Quantum machine learning; transformer representation compression; sentiment classification; variational quantum circuits; hybrid quantum-classical modelling

Received: 21/07/2026

Revised: 22/08/2026

Acceptance: 31/08/2026

Publication: 09/09/2026

3. Introduction

Language models based on transformers have significantly advanced natural language processing by enabling the representation of context, which enables classification, question answering and other tasks of language understanding. Although BERT has laid a solid foundation for the subsequent applications, the computational and memory requirements can limit its use in resource-constrained environments (Devlin et al., 2019). Hence, the focus of subsequent research has been on compacting representations of the transformer and minimizing the cost of the downstream computation without compromising prediction accuracy. Large-scale tests of BERT compression have demonstrated that there is scope for meaningful efficiency improvements, though this trade-off of compactness versus task performance will vary for different applications (Ganesh et al., 2021).

Another complementary method to building compact and expressive learning systems is using quantum machine learning. It is based on the idea of using quantum-state transformations in combination with classical optimization to process encoded data via parameterized circuits (Biamonte et al., 2017; Benedetti et al., 2019). While the expressiveness of the decision boundaries in quantum neural networks can be achieved with a limited number of trainable parameters, the potential advantages are highly data-dependent, circuit-specific, optimization-stable, and compared to suitably tailored classical models (Abbas et al., 2021). Another way to convert classical inputs to representations that could be hard to replicate with classical kernels has been proposed by Havlíček et al., 2019 and is called quantum enhanced feature spaces.

A specific type of VQA, where computation is split between parameterized quantum circuits and classical optimizers, is relevant to near-term QML, and has been explored in Cerezo et al., 2021a. Noise, limited qubit availability, simulation cost, and challenging optimization landscapes can, however, limit their performance. As the circuit complexity grows, barren plateaus can cause the gradients to drop significantly, which can make architecture selection and shallow circuit design crucial for practical implementation (Cerezo et al., 2021b). The constraints suggest that the quantum components should be studied in the context of well-controlled hybrid systems and not expected to outperform classical systems.

We can apply these concepts to language representations by mapping classical language models to quantum computational models (Coecke et al., 2020). However, the direct encoding of a high-dimensional transformer embedding into quantum circuits is not feasible at the moment since the available circuits have a limited number of qubits. It is thus necessary to map pretrained language representations to a small quantum compatible space using a supervised dimensionality-reduction stage. Automated optimization is also required as circuit depth, entanglement pattern, feature re-uploading, learning rate and regularization all play a role in the performance. Structured and reproducible approaches to exploring these interacting hyperparameters exist, like Optuna (Akiba et al., 2019).

Another weakness in previous hybrid studies is that often there were no matched classical controls. The apparent quantum benefits can be obtained due to representation compression, preprocessing or the design of the classifiers, and not due to the quantum circuit. The current study aims to tackle this problem by comparing a quantum-only classifier and a protected residual hybrid one with a full-dimensional classical baseline and a classical one with the same compressed representation. The framework that was implemented had a six-dimensional quantum-compatible bottleneck, six-qubit variational circuits, hyperparameter optimization based on the complexity of the circuit, and a locked evaluation protocol.

Accordingly, the study pursued three objectives:

- To develop a hybrid quantum-classical optimization framework that integrates compressed DistilBERT representations with a variational quantum classifier.
- To assess whether the proposed framework improves computational efficiency while preserving the predictive performance of classical sentiment-classification models.
- To validate the classical, quantum-only, and residual hybrid models using benchmark sentiment data, statistical comparisons, and model-complexity and inference-efficiency measures.

2. Literature Review

The transformer-based language models have significantly advanced the field of natural language processing through their ability to learn contextual representations, which facilitate classification, question answering, and other language understanding tasks. Although BERT has opened up a promising direction for bidirectional transformer pretraining, the high memory and computing requirements may limit its use in resource-limited settings (Devlin et al., 2019). The compression of transformer representations and the minimization of the computational cost after the transformer have been compressed have therefore been a focus of subsequent research. Large scale testing of BERT compression has demonstrated that it is possible to achieve some level of efficiency without compromising the task, albeit the efficiency depends on the application (Ganesh et al., 2021).

The possibility of using quantum machine learning as an alternative strategy to build compact and expressive learning systems has arisen. It is an algorithm that uses classical optimization and quantum-state transformations to process encoded data by parameterized circuits (Biamonte et al., 2017; Benedetti et al., 2019). While the expressiveness of decision boundaries can be achieved with a relatively small number of trainable parameters, the practical usefulness of quantum neural networks will rely heavily on how the data is encoded, the architecture of the network, optimization stability, and the comparison with an appropriately matched classical model (Abbas et al., 2021). Another possible way to convert classical inputs to representations that are hard to replicate with conventional kernels is via quantum-enhanced feature spaces (Havlíček et al., 2019).

Variational quantum algorithms are especially important for near-term quantum machine learning since they split the computation between parameterized quantum circuits and classical optimizers (Cerezo et al., 2021a). But they can be constrained by noise, limited number of qubits, the cost of the simulation and complex optimization landscapes. In the case of barren plateaus, the gradients can become very small as the size of the circuits grows, thus the architecture selection and shallow circuit design become crucial for the practical implementation (Cerezo et al., 2021b). The constraints suggest that the quantum components need to be considered within a well-controlled hybrid system, not as a replacement for classical systems.

In the context of language, quantum natural language processing (QNLP) is an extension of these ideas that aims to link the classical structure of language with quantum computational models (Coecke et al., 2020). However, the direct encoding of these high-dimensional transformer embeddings into quantum circuits is not feasible at the moment as there are only limited circuits available that can be used with a small number of qubits. To convert pretrained language representations into a compact space that is compatible with quantum computers, a supervised dimensionality-reduction stage is thus necessary. Another important reason for automated optimization is that the performance depends on both the depth of the circuit, the pattern of entanglement, the re-uploading of the features, the learning rate, and the regularization. There are structured and reproducible methods for exploring such interacting hyperparameters, like frameworks such as Optuna (Akiba et al., 2019).

Another drawback of previous studies involving hybrids is that they often lack matched classical studies. The apparent quantum benefits can be obtained by the representation compression, preprocessing or design of the classifiers, not from the quantum circuit. The current study aims to tackle this question by comparing a quantum-only classifier to a full-dimensional classical baseline as well as a classical classifier using the same compressed representation. The framework that was implemented was based on a 6-dimensional quantum compatible bottleneck, 6-qubit variational circuits, complexity-aware hyperparameter optimization and a locked evaluation protocol.

3. Methodology

3.1 Research Design and Experimental Framework

The approach in this study was a controlled computational experiment to build and test a hybrid quantum-classical based binary sentiment classification model. The proposed framework comprised a hybrid model of a residual network, variational quantum classifier, supervised dimensionality reduction and frozen DistilBERT representations. The experiment tested if the architecture of the downstream system could be kept compact while preserving or enhancing predictive performance with the inclusion of a quantum component.

The experimental workflow comprised data auditing and leakage control, tokenization, DistilBERT feature extraction, classical baseline development, quantum-compatible feature compression, quantum-circuit optimization, residual hybrid modelling, locked final evaluation, and computational-efficiency assessment. All model-development decisions were based exclusively on the research-training and internal-validation partitions. The official SST-2 development partition was accessed only after the preprocessing objects, model architectures, hyperparameters, and prediction rules had been finalized.

3.2 Dataset Preparation and Experimental Partitions

The study used the binary Stanford Sentiment Treebank task, commonly known as SST-2. The dataset contains sentence-level movie-review text labelled as either negative or positive sentiment (Socher et al., 2013). Negative sentiment was coded as 0, while positive sentiment was coded as 1. The original files contained 67,349 labelled training observations, 872 labelled official-development observations, and 1,821 official-test observations. The official test partition was used only for integrity assessment because its reference labels were unavailable.

Before model development, a structured data-quality audit was conducted. Sentence text was standardized through Unicode normalization, case-normalized comparison, and whitespace cleaning. Exact duplicates, normalized-text duplicates, contradictory labels, and cross-partition overlaps were identified. Records containing identical normalized text but different sentiment labels were treated as conflicting observations.

The cleaning procedure removed 10 observations from contradictory text groups, two observations overlapping with the official test partition, and 366 repeated normalized records. The resulting leakage-controlled master dataset contained 66,971 observations. It was divided into a research-training partition of 60,274 observations and an internal-validation partition of 6,697 observations. Stratified group-based partitioning maintained the class distribution while preventing observations with identical normalized text from appearing in both partitions. The official development partition of 872 observations remained untouched during model development.

The text was tokenized using the distilbert-base-uncased tokenizer. Special tokens and attention masks were retained. All sequences were padded or truncated to a maximum length of 64 tokens. The selected sequence length covered almost all observations while limiting unnecessary padding and computational cost.

3.3 DistilBERT Representation and Classical Model Development

The pretrained DistilBERT encoder was used as a frozen contextual feature extractor. Its parameters were not updated during downstream model training. This design isolated the effects of representation compression and quantum modelling while reducing the computational requirements associated with full transformer fine-tuning.

The final hidden-state tensor was extracted for each sentence. Two sentence-level representation strategies were evaluated. The first used the hidden vector associated with the initial classification token. The second calculated an

attention-mask-aware mean across all valid token vectors. Both approaches produced a 768-dimensional sentence representation. The embeddings were stored as float32 arrays and cached separately for the research-training, internal-validation, official-development, and official-test partitions.

Three classical reference models were developed. These included a majority-class predictor, logistic regression using classification-token embeddings, and logistic regression using mean-pooled embeddings. Feature standardization was fitted only on the research-training partition and applied unchanged to the other partitions. The pooling strategy was selected through a predefined hierarchy based on validation F1 score, ROC-AUC, balanced accuracy, and inference latency. Mean pooling was retained for subsequent modelling.

A supervised bottleneck network was then used to reduce the 768-dimensional mean-pooled representation to a quantum-compatible feature space. Candidate latent dimensions of 4, 6, and 8 were assessed. The network consisted of a trainable linear projection, hyperbolic tangent activation, dropout regularization, and a two-class output layer.

Each candidate dimension was trained using random seeds 42, 52, and 62. The smallest dimension with a mean validation F1 score within 0.005 of the best-performing candidate was selected. This parsimony rule retained a six-dimensional representation. The resulting bottleneck values were bounded by the hyperbolic tangent activation and multiplied by π , mapping them to the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$ for quantum angle encoding.

3.4 Quantum and Residual Hybrid Model Development

The variational quantum classifier was implemented using PennyLane and the analytic default.qubit simulator. Six qubits were used, corresponding to the six selected bottleneck features. Each input feature was encoded through a Y-axis rotation.

Every variational layer applied trainable R_Y and R_Z rotations to each qubit, followed by CNOT entanglement. Linear and ring entanglement structures were considered. The search also examined whether the input features should be re-uploaded between variational layers. At the output stage, the expectation value of the Pauli-Z observable was measured for every qubit. These six expectation values were passed to a trainable linear layer that generated two classification logits.

Automated hyperparameter optimization was used to identify the quantum configuration. The search space included one to three variational layers, linear or ring entanglement, feature re-uploading enabled or disabled, Adam or AdamW optimization, learning rate, weight decay, and quantum-weight initialization scale.

A stratified subset of 12,000 research-training observations was used during quantum optimization to control simulation time. Every trial was evaluated on the complete internal-validation partition of 6,697 observations. Models were trained with a batch size of 128 for a maximum of eight epochs using cross-entropy loss. Gradient norms were clipped at 1.0. Early stopping was applied after two consecutive epochs without validation improvement, while median pruning terminated weak trials. Internal-validation F1 was used as the optimization objective.

The search continued until 12 valid trials had been completed. Configurations within 0.003 F1 of the best trial were treated as practically comparable. Final selection prioritized fewer quantum parameters, lower gate count, reduced circuit depth, shorter training time, and then higher ROC-AUC and F1. The selected configuration used three variational layers, linear entanglement, no feature re-uploading, and Adam optimization.

A residual hybrid architecture was subsequently developed to determine whether the quantum branch could provide a useful correction to the classical bottleneck. The combined logits were defined as:

$$z_{\text{hybrid}} = z_{\text{classical}} + \tanh\left(\frac{z_{\text{quantum}}}{\alpha}\right) z_{\text{quantum}},$$

where $z_{\text{classical}}$ represents the frozen classical logits, z_{quantum} represents the quantum logits, and α is a trainable residual-scaling parameter.

The residual scale was initialized at zero. Therefore, the hybrid model initially reproduced the classical bottleneck exactly. The quantum correction influenced predictions only when training moved the residual scale away from zero. Residual training was repeated using seeds 42, 52, and 62. Each run used a maximum of 12 epochs, a batch size of 128, gradient clipping, and early stopping patience of three epochs. The epoch-zero state remained eligible for selection to prevent deterioration relative to the classical anchor.

3.5 Model Selection and Locked Evaluation Protocol

Five models were retained for final comparison:

1. Majority-class reference
2. Full 768-dimensional logistic regression
3. Classical six-dimensional bottleneck
4. Quantum-only classifier
5. Residual hybrid classifier

The full logistic model represented the uncompressed classical baseline. The six-dimensional bottleneck served as the compact classical reference matched the quantum input space. The quantum-only classifier assessed the predictive capability of the optimized variational circuit, while the residual model tested whether the circuit could improve a protected classical prediction.

Before official evaluation, the preprocessing objects, model checkpoints, configurations, probability calculations, prediction threshold, and artifact hashes were locked. A fixed probability threshold of 0.50 was applied to all probabilistic classifiers. No feature representation, model architecture, residual scale, hyperparameter, or decision threshold was changed after the official-development evaluation began.

3.6 Performance and Statistical Evaluation

Predictive performance was assessed using accuracy, balanced accuracy, precision, recall, specificity, F1 score, ROC-AUC, Matthews correlation coefficient, and log loss. Confusion-matrix counts were also obtained for true negatives, false positives, false negatives, and true positives. Accuracy was used as the principal ranking measure during the locked official evaluation. F1 and ROC-AUC were reported as complementary measures of class-sensitive performance and discrimination.

Uncertainty was estimated through 2,000 stratified bootstrap repetitions. The bootstrap procedure preserved the class distribution of the official development partition. Percentile-based 95% confidence intervals were calculated for accuracy, F1, and ROC-AUC. Paired bootstrap distributions were additionally used to estimate metric differences between the six-dimensional classical bottleneck and the comparison models.

Exact McNemar tests were used to assess differences in paired classification outcomes. Each comparison evaluated the number of observations correctly classified only by the classical bottleneck against the number correctly classified only by the competing model. Holm adjustment was applied to account for multiple comparisons. A non-significant finding was interpreted as insufficient evidence of a performance difference and not as proof that two models were statistically equivalent.

3.7 Computational Efficiency and Reproducibility

Computational efficiency was evaluated primarily at the downstream model-head level. The assessed characteristics included representation dimension, dimensionality-reduction percentage, trainable parameter count, model artifact size, quantum parameter count, logical gate count, circuit depth, latency per observation, throughput, and process-memory consumption. The frozen DistilBERT encoder was reported separately because it was shared across all embedding-based models.

Classical model heads were benchmarked over 30 repetitions following warm-up runs. Quantum and residual model heads were benchmarked over 10 repetitions because statevector circuit simulation required greater computation. Latency was divided by the number of official-development observations to obtain milliseconds per observation. Throughput was expressed as observations processed per second. These measurements excluded tokenization and DistilBERT embedding extraction and therefore represented downstream model-head inference rather than complete raw-text processing time.

The approach in this study was a controlled computational experiment to build and test a hybrid quantum-classical based binary sentiment classification model. The proposed framework comprised a hybrid model of a residual network, variational quantum classifier, supervised dimensionality reduction and frozen DistilBERT representations. The experiment aimed to determine whether it was possible to maintain or improve predictive performance in the downstream system while still keeping the system architecture compact with the addition of a quantum component.

4. Results

4.1 Experimental Data and Evaluation Readiness

The initial SST-2 training partition contained 67,349 observations. Data-quality screening identified five contradictory normalized-text groups involving 10 observations, two records overlapping with the official test partition, and 366 repeated normalized records. After these records were removed, the leakage-controlled master dataset contained 66,971 unique observations. This dataset was divided into 60,274 research-training observations and 6,697 internal-validation observations. The official development partition containing 872 labelled observations remained untouched until the final locked evaluation. Mean-pooled DistilBERT embeddings were generated using a maximum sequence length of 64 tokens, resulting in a 768-dimensional representation for each observation. The finalized experimental partitions and configuration are summarized in Table 1.

Table 1. Dataset partitions and experimental configuration

Component	Observations/Value	Purpose
Original SST-2 training data	67,349	Initial development dataset
Leakage-controlled master data	66,971	Clean observations after screening
Research-training partition	60,274	Model fitting and optimization
Internal-validation partition	6,697	Architecture and hyperparameter selection
Official development partition	872	Locked final evaluation
Official test partition	1,821	Unlabeled external partition
Maximum sequence length	64 tokens	DistilBERT tokenization
Embedding dimension	768	Mean-pooled DistilBERT representation

Table 1 confirms that model development and final evaluation were conducted on separate partitions. The official development labels were not accessed during pooling selection, bottleneck optimization, quantum-circuit tuning, or residual-model selection.

4.2 Classical Baseline Performance and Pooling Selection

The majority-class reference achieved an internal-validation accuracy of 0.5591 and an F1 score of 0.7172. Logistic

regression using the CLS representation produced an accuracy of 0.8680, an F1 score of 0.8813, and a ROC-AUC of 0.9411. Mean pooling slightly improved accuracy to 0.8692 and F1 to 0.8823, while maintaining a ROC-AUC of 0.9407. Mean pooling was therefore selected according to the predefined hierarchy prioritizing F1, ROC-AUC, balanced accuracy, and inference latency.

4.3 Quantum-Ready Bottleneck Optimization

The 768-dimensional mean-pooled representation was compressed into candidate bottlenecks of four, six, and eight dimensions. The six- and eight-dimensional configurations met the predefined practical-equivalence criterion, whereas the four-dimensional model did not. Six dimensions were selected because this was the smallest representation within 0.005 F1 of the best candidate result. The comparative bottleneck findings are reported in Table 2.

Table 2. Internal-validation results for candidate bottleneck dimensions

Dimension	Accuracy, Mean	F1, Mean	ROC-AUC, Mean	Trainable Parameters	Practical Equivalence
4	0.8717	0.8847	0.9416	3,086	No
6	0.8771	0.8893	0.9453	4,628	Yes
8	0.8804	0.8921	0.9482	6,170	Yes

Table 2 shows that increasing the bottleneck dimension produced only modest predictive gains. The selected six-dimensional run, obtained with seed 62, achieved an accuracy of 0.8788, an F1 score of 0.8910, and a ROC-AUC of 0.9459, while reducing the representation dimension by 99.219%.

4.4 Quantum Circuit Configuration and Hyperparameter Optimization

Quantum hyperparameter optimization produced 12 valid completed trials and 14 pruned trials. Trial 5 was selected through the predefined complexity-aware procedure. The selected variational classifier used six qubits, three variational layers, linear entanglement, no feature re-uploading, and the Adam optimizer. It contained 36 quantum parameters, 52 logical gates, and a logical circuit depth of 16.

The selected quantum-only model achieved an internal-validation accuracy of 0.7009, an F1 score of 0.7389, a ROC-AUC of 0.6874, and an MCC of 0.3900. Its F1 performance retention relative to the six-dimensional classical bottleneck was 82.927%. Therefore, the optimized quantum-only classifier did not preserve the predictive quality of the compact classical model.

4.5 Residual Hybrid Model Evaluation

A residual hybrid classifier was evaluated to determine whether the quantum circuit could provide a useful correction to the frozen classical bottleneck. Three runs were conducted using seeds 42, 52, and 62. In every run, the trained quantum residual reduced validation F1 relative to the initial classical state. Consequently, epoch 0 was retained as the best checkpoint for all three seeds.

The selected residual scale was 0.0000, resulting in an accuracy of 0.8788, an F1 score of 0.8910, and a ROC-AUC of 0.9459. These values were identical to those of the classical bottleneck. The residual gate therefore rejected the quantum contribution rather than allowing it to reduce predictive performance.

4.6 Locked Official SST-2 Development Evaluation

After all configurations had been fixed, the models were evaluated on the untouched official development partition. The six-dimensional classical bottleneck obtained the highest accuracy and F1 score. The residual hybrid produced identical predictions because its selected quantum scale was zero. Full official development results are provided in Table 3.

Table 3. Locked official SST-2 development performance

Model	Accuracy (95% CI)	F1 (95% CI)	ROC-AUC (95% CI)
Classical 6D bottleneck	0.8360 (0.8096–0.8589)	0.8427 (0.8178–0.8646)	0.9184 (0.8995–0.9345)
Residual hybrid	0.8360 (0.8096–0.8589)	0.8427 (0.8178–0.8646)	0.9184 (0.8995–0.9345)
Full 768D logistic regression	0.8280 (0.8016–0.8521)	0.8315 (0.8057–0.8556)	0.9187 (0.9000–0.9353)
Quantum-only classifier	0.6663 (0.6342–0.6984)	0.6914 (0.6603–0.7214)	0.6308 (0.5918–0.6690)
Majority reference	0.5092	0.6748	0.5000

Table 3 indicates that the six-dimensional bottleneck retained strong predictive performance despite substantial dimensionality reduction. The quantum-only classifier showed considerably lower accuracy, F1, and discrimination than both classical alternatives.

The performance forest plot displays the locked estimates and stratified-bootstrap confidence intervals for all five models, allowing the predictive differences and uncertainty ranges to be examined simultaneously (Figure 1).

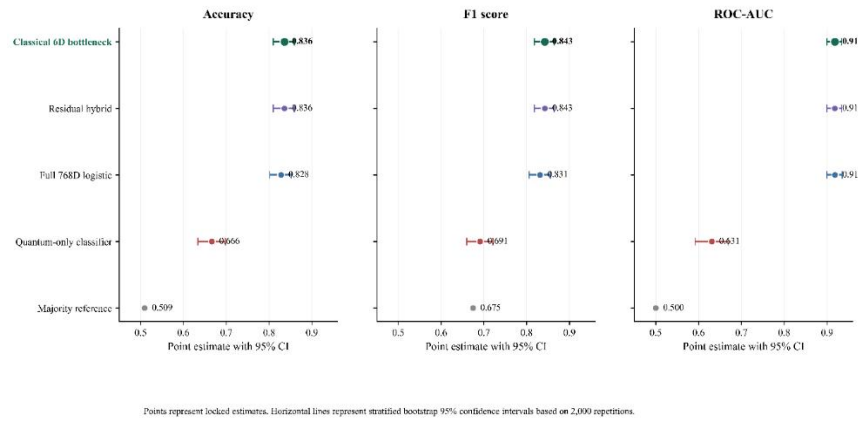


Figure 1. Locked official SST-2 development performance with 95% confidence intervals

Figure 1 shows that the classical bottleneck and residual hybrid occupy the highest positions for accuracy and F1. Their identical estimates confirm that the quantum residual made no contribution to the final predictions.

4.7 Classification Error Analysis

The recommended bottleneck correctly classified 346 negative observations and 383 positive observations. It produced 82 false-positive and 61 false-negative predictions. Sensitivity was 0.8626, specificity was 0.8084, precision was 0.8237, balanced accuracy was 0.8355, and MCC was 0.6724. The class-specific distribution of correct and incorrect predictions is presented in Figure 2.

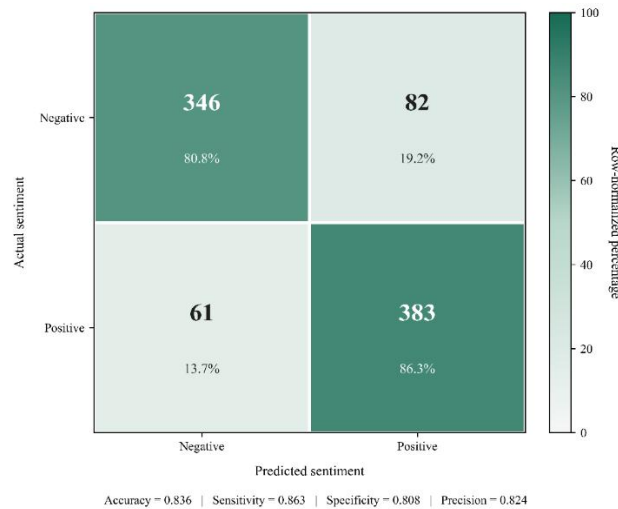


Figure 2. Confusion matrix of the recommended six-dimensional classical bottleneck

Figure 2 demonstrates that positive sentiment was recognized more accurately than negative sentiment. Nevertheless, both classes retained correct-classification rates above 80%, indicating reasonably balanced performance.

4.8 Statistical Comparison of the Models

Exact McNemar testing showed no significant difference between the six-dimensional bottleneck and full logistic regression after Holm correction. In contrast, the quantum-only classifier was significantly weaker than the classical bottleneck. The residual hybrid had no discordant observations because it produced the same predictions as the classical anchor, as shown in Table 4.

Table 4. Exact McNemar comparisons against the classical six-dimensional bottleneck

Comparison Model	Bottleneck Only Correct	Comparison Only Correct	Holm-Adjusted p	Significant
Full 768D logistic regression	28	21	.7832	No
Quantum-only classifier	193	45	< .001	Yes
Residual hybrid	0	0	1.0000	No

Table 4 indicates that the compact bottleneck did not differ significantly from the full logistic model on paired classifications. The substantially larger number of bottleneck-only correct predictions confirms the significant underperformance of the quantum-only classifier.

4.9 Computational Efficiency and Model Complexity

The six-dimensional bottleneck achieved a model-head latency of 0.000607 ms per observation and processed

approximately 1.65 million observations per second. The full logistic head required 0.007225 ms per observation, whereas the quantum-only and residual-hybrid heads required approximately 0.061 ms. These measurements exclude tokenization and DistilBERT embedding extraction. The downstream efficiency results are summarized in Table 5.

Table 5. Model-head complexity and inference efficiency

Model	Input Dimension	Trainable Parameters	Latency, ms/Observation	Throughput, Observations/s
Classical 6D bottleneck	6	4,628	0.000607	1,646,526
Residual hybrid	6	51	0.060092	16,641
Full 768D logistic regression	768	769	0.007225	138,412
Quantum-only classifier	6	50	0.060954	16,406

Table 5 shows that parameter count alone did not determine inference efficiency. Despite their smaller trainable heads, the quantum models were approximately 100 times slower than the classical bottleneck because of circuit simulation costs.

The joint accuracy–latency analysis positions the classical bottleneck as the strongest performance–efficiency option among the locked models (Figure 3).

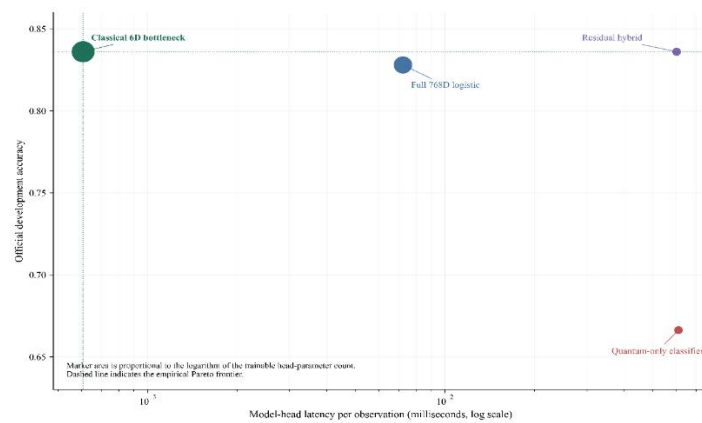


Figure 3. Accuracy–latency Pareto trade-off of the locked model heads

Figure 3 shows that the classical bottleneck combined the highest accuracy with the lowest non-trivial model-head latency. The quantum models occupied less favourable positions because they required greater inference time without improving accuracy.

The representation, parameter, and latency comparison further illustrates the contrasting computational characteristics of the classical and quantum alternatives (Figure 4).

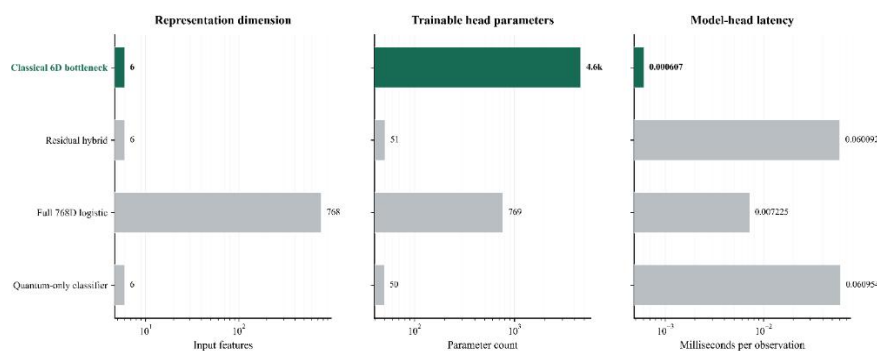


Figure 4. Representation compression, parameter complexity, and model-head latency

Figure 4 confirms that the six-dimensional bottleneck achieved substantial representation compression while retaining the most efficient downstream execution. The quantum heads used fewer trainable parameters but introduced substantially greater computational latency and no predictive benefit.

5. Discussion

The findings demonstrate that substantial compression of frozen DistilBERT representations can be achieved without

materially weakening sentiment-classification performance. The selected six-dimensional classical bottleneck reduced the original 768-dimensional embedding by 99.219% and achieved the highest official-development accuracy of 0.8360 and F1 score of 0.8427. Its performance was slightly higher than that of the full 768-dimensional logistic-regression model, although the paired difference was not statistically significant. The result should therefore be interpreted as evidence that the compact bottleneck preserved competitive predictive performance, rather than proof that it was superior to the full-dimensional baseline. The confusion matrix also showed reasonably balanced classification, with stronger sensitivity to positive sentiment than specificity to negative sentiment.

The quantum-only classifier achieved considerably lower accuracy, F1, ROC-AUC, and MCC than the matched classical bottleneck. This difference remained statistically significant after correction for multiple comparisons. The result indicates that the variational quantum circuit did not extract a more useful decision representation from the six-dimensional features under the tested configuration. The residual hybrid model provided an additional safeguard against performance deterioration. It was found that its learned residual scale was still zero, which meant that it reproduced the classical bottleneck exactly. This is informative as the framework did not mandate the quantum component to be included in the final prediction when it did not help with validation.

The results are similar to previous work that demonstrated that compressed transformer representations can maintain high downstream performance if the most informative features are kept (Sanh et al., 2019). They also enable the use of pretrained classical feature extractors as an interface for low dimensional quantum models as in the hybrid transfer-learning research (Mari et al., 2020). The poor quantum-only performance, however, is consistent with the findings of others that the performance of quantum machine learning algorithms is strongly influenced by the structure of the data and the appropriate choice of feature maps, not just the complexity of the circuit itself (Huang et al., 2021). Overall, the lack of quantum advantage highlights the current challenges in QNLP, which are rooted in the empirical studies being limited by small circuits, small number of qubits, dependence on simulators, and the availability of strong classical alternatives (Widdows et al., 2024).

The primary takeaway from the practical point of view is that supervised representation compression could be a more immediate gain than a quantum classification for sentiment analysis using transformers. The classical bottleneck was optimized for high throughput of the model-head and low latency with maintaining predictive performance. The study is also a reminder of the need for matched classical controls. The six-dimensional classical comparator was not included, and it would have been easy to mistake the contribution of the quantum circuit as being from the compressed representation. The remaining architecture also gives a general idea of how to test the quantum corrections without compromising the performance of an established classical model.

There are several caveats to be noted. Only SST-2 was used for the evaluation, so the results cannot be generalized to other NLP tasks and domains. The effect of joint or partial fine-tuning was not tested as DistilBERT was used as a frozen encoder. Noisy hardware was not used for quantum experiments, but rather an analytic statevector simulator. The six-qubit circuit was also restricted in depth and feature capacity, and the reported latency didn't include tokenization and DistilBERT encoding.

The framework should be tested on other datasets for sentiment, topic-classification and natural-language inference in the future. It should also investigate other encodings, quantum kernels, noise-aware training, and actual quantum hardware. The interaction between the two may be optimized jointly to reveal interactions which cannot be captured by sequential training of the two. There should also be a comparison of end-to-end energy usage, hardware latency, calibration and robustness to distribution shift in further work.

6. Conclusion

A hybrid framework of quantum-classical was developed and tested for sentiment classification, which includes using frozen DistilBERT representations, supervised dimensionality reduction, variational quantum learning, and residual hybrid modelling. The framework was applied in a leakage controlled and locked evaluation protocol, with the SST-2 benchmark. The results indicated that the 6-dimensional classical bottleneck configuration was the best one. It was able to compress the original 768-dimensional representation by 99.219% and obtained the accuracy of 0.8360, F1 score of 0.8427 and ROC-AUC of 0.9184 from the official test set. It performed similarly to the full-dimensional logistic-regression baseline and was significantly more efficient at the model-head level. The performance of the quantum-only classifier was significantly poorer, and the inference time was much longer than the matched classical bottleneck. The optimized residual scale was also not able to enhance the classical prediction, with the residual hybrid model still yielding a zero scale. Thus, no empirical advantage was found for quantum simulation of a statevector with the six-qubit statevector-simulation setting tested here. The key insight of the study is that supervised representation compression is capable of significant efficiency gains without compromising the predictive quality, whereas matched classical controls are crucial to assess quantum claims. Further research is needed regarding other datasets of NLP, other and larger quantum circuits, quantum kernels, multi-bottleneck circuit training, noisy simulations, and actual quantum devices. Additionally, the end-to-end latency, energy consumption, calibration and robustness to distribution shift should also be explored.

References

1. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., & Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, 1(6), 403–409. <https://doi.org/10.1038/s43588-021-00084-1>
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter

- optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2623–2631). Association for Computing Machinery.
<https://doi.org/10.1145/3292500.3330701>
3. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4), Article 043001. <https://doi.org/10.1088/2058-9565/ab4eb5>
4. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195–202. <https://doi.org/10.1038/nature23474>
5. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, Ł., & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3(9), 625–644.
<https://doi.org/10.1038/s42254-021-00348-9>
6. Cerezo, M., Sone, A., Volkoff, T., Cincio, Ł., & Coles, P. J. (2021). Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12, Article 1791. <https://doi.org/10.1038/s41467-021-21728-w>
7. Coecke, B., de Felice, G., Meichanetzidis, K., & Toumi, A. (2020). Foundations for near-term quantum natural language processing [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2012.03755>
8. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). 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) (pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
9. Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Sajjad, H., Nakov, P., Chen, D., & Winslett, M. (2021). Compressing large-scale transformer-based models: A case study on BERT. *Transactions of the Association for Computational Linguistics*, 9, 1061–1080. https://doi.org/10.1162/tacl_a_00413
10. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747), 209–212.
<https://doi.org/10.1038/s41586-019-0980-2>
11. Huang, H.-Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021). Power of data in quantum machine learning. *Nature Communications*, 12, Article 2631. <https://doi.org/10.1038/s41467-021-22539-9>
12. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 4163–4174). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.findings-emnlp.372>
13. Katsaklis, D., Fan, I., Yeung, R., Pearson, A., Lorenz, R., Toumi, A., de Felice, G., Meichanetzidis, K., Clark, S., & Coecke, B. (2021). lambeq: An efficient high-level Python library for quantum NLP [Preprint]. arXiv.
<https://doi.org/10.48550/arXiv.2110.04236>
14. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1909.11942>
15. Mari, A., Bromley, T. R., Izaac, J., Schuld, M., & Killoran, N. (2020). Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4, Article 340. <https://doi.org/10.22331/q-2020-10-09-340>
16. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9, Article 4812. <https://doi.org/10.1038/s41467-018-07090-4>
17. Meichanetzidis, K., Gogioso, S., de Felice, G., Chiappori, N., Toumi, A., & Coecke, B. (2021). Quantum natural language processing on near-term quantum computers. *Electronic Proceedings in Theoretical Computer Science*, 340, 213–229. <https://doi.org/10.4204/EPTCS.340.11>
18. Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, 98(3), Article 032309. <https://doi.org/10.1103/PhysRevA.98.032309>
19. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. (2020). Data re-uploading for a universal quantum classifier. *Quantum*, 4, Article 226. <https://doi.org/10.22331/q-2020-02-06-226>
20. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1910.01108>
21. Schuld, M., Bocharov, A., Svore, K. M., & Wiebe, N. (2020). Circuit-centric quantum classifiers. *Physical Review A*, 101(3), Article 032308. <https://doi.org/10.1103/PhysRevA.101.032308>
22. Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4), Article 040504. <https://doi.org/10.1103/PhysRevLett.122.040504>
23. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631–1642). Association for Computational Linguistics.
<https://aclanthology.org/D13-1170/>
24. Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: A compact task-agnostic BERT for resource-limited devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 2158–2170). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.195>
25. Widdows, D., Alexander, A., Zhu, D., Zimmerman, C., & Majumder, A. (2024). Near-term advances in quantum natural language processing. *Annals of Mathematics and Artificial Intelligence*, 92, 1249–1272.
<https://doi.org/10.1007/s10472-024-09940-y>