Uncertainty-Aware Transformers: Conformal Prediction for Language Models

Anonymous authors
Paper under double-blind review

Abstract

Transformers have had a profound impact on the field of artificial intelligence, especially on large language models and their variants. Unfortunately, as was the case historically with neural networks, the black-box nature of transformer architectures presents significant challenges to interpretability and trustworthiness. These challenges generally emerge in high-stakes domains, such as healthcare, robotics, and finance, where incorrect predictions can have significant negative consequences, such as misdiagnosis or failed investments. For models to be genuinely useful and trustworthy in critical applications, they must provide more than just predictions: they must supply users with a clear understanding of the reasoning that underpins their decisions. This paper presents an uncertainty quantification framework for transformer-based language models. This framework, called CONFIDE (CONformal prediction for FIne-tuned DEep language models), applies conformal prediction to the internal embeddings of encoder-only architectures, like BERT and RoBERTa, based on hyperparameters, such as distance metrics and principal component analysis. CONFIDE uses either [CLS] token embeddings or flattened hidden states to construct class-conditional nonconformity scores, enabling statistically valid prediction sets with instance-level explanations. Empirically, CONFIDE improves test accuracy by up to 4.09% on BERT-tiny and achieves greater correct efficiency (i.e., the expected size of the prediction set conditioned on it containing the true label) compared to prior methods, including NM2 and VanillaNN. We show that early and intermediate transformer layers often yield better-calibrated and more semantically meaningful representations for conformal prediction. In resource-constrained models and high-stakes tasks with ambiguous labels, CONFIDE offers robustness and interpretability where softmax-based uncertainty fails.

1 Introduction

Transformer architectures, first introduced in (Vaswani et al., 2017), revolutionized natural language processing (NLP) by introducing the self-attention mechanism, enabling models to capture complex contextual relationships. The introduction of BERT (Bidirectional Encoder Representations from Transformers) further enabled transformer progress by showcasing the effectiveness of encoder-only architectures across diverse linguistic tasks, from sentiment analysis to named entity recognition (Devlin et al., 2019).

Following BERT's success, a range of encoder-based transformer variants emerged, each optimizing different aspects like training efficiency, representational stability, and task-specific performance, including RoBERTa (Liu et al., 2019). In parallel, compact variants like BERT-tiny and DistilBERT enabled the deployment of transformer architectures in computationally-constrained environments. These lightweight models preserve much of the semantic power of their larger counterparts while offering faster inference and lower memory footprints (Devlin et al., 2019). Crucially, the open-source nature of many of these models has democratized access to cutting-edge NLP capabilities.

Conformal prediction (CP) has emerged as a rigorous, distribution-free approach for quantifying uncertainty and improving interpretability in machine learning models. Unlike traditional uncertainty estimation techniques that rely heavily on model-specific internals, CP provides a straightforward yet powerful guarantee: it

constructs *prediction sets* that contain the true output with a predefined probability, such as 95%, independent of the model or the underlying data distribution (Angelopoulos & Bates, 2021). This property makes CP especially appealing for deep neural networks and transformer architectures as it provides reliable calibration (Messoudi et al., 2020). Furthermore, CP can adapt to various learning tasks—classification, regression, anomaly detection—through careful design of the nonconformity function, which maps a model prediction and its context to a scalar that reflects uncertainty (Angelopoulos & Bates, 2021). The CONFINE (CONFormal INterpretable Explanation) algorithm (Huang et al., 2025) extends CP to neural networks, preserving its theoretical rigor while enhancing its interpretability and robustness.

Building upon CONFINE, this paper introduces **CONFIDE** (**CON**formal prediction for **FI**ne-tuned **DE**ep language models), extending conformal prediction techniques to encoder-based transformer architectures. CONFIDE addresses unique transformer challenges, such as layered attention complexities and sequential dependencies, thereby enhancing interpretability and reliability in transformer predictions.

The paper makes the following contributions.

- It proposes **CONFIDE**, a conformal prediction framework tailored to fine-tuned language-based transformer models, enabling layer-wise uncertainty calibration and interpretable prediction sets.
- It shows that CONFIDE achieves up to 4.09% absolute accuracy improvement and up to 5.40% higher correct efficiency over softmax-based confidence baselines across GLUE and SuperGLUE benchmarks.
- It demonstrates that CONFIDE outperforms prior uncertainty quantification and interpretability methods—such as NM2 and VanillaNN—on tasks where standard predictors suffer from undercoverage or skewed confidence.

The paper is organized as follows. Section 2 dives further into background and related works on model architecture, explainable AI, and conformal prediction, including the CONFINE algorithm. Section 3 presents datasets, models, and metrics, followed by the paper's methodology in Section 4. Section 5 presents experimental results, while Section 6 discusses limitations. Section 7 summarizes CONFIDE's findings and use.

2 Background and Related Work

In this section, we provide background material and discuss related work.

2.1 Transformer Architectures and Self-Attention

Unlike previous-generation NLP architectures, including recurrent neural network, long short-term memory (LSTM), and gated recurrent unit, transformers do not process data sequentially. These prior models had limitations that posed computational bottlenecks, reducing scalability and limiting performance on complex linguistic tasks. Instead, transformers handle entire sequences simultaneously, allowing them to effectively capture long-range contextual dependencies that were historically challenging to compute (Vaswani et al., 2017).

Formally, self-attention computes a contextualized representation for each token by employing a scaled dot-product attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k denotes the dimension of the keys. Through this method, each token's embedding becomes richly contextualized through its relationships with every other token in the sequence (Vaswani et al., 2017).

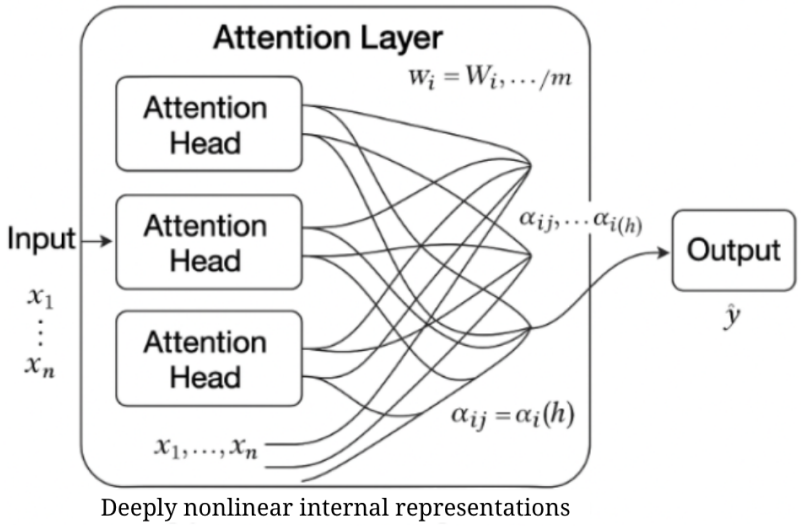


Figure 1: The complexity of transformer attention layers. Multi-head attention layers aggregate token interactions via learned weights (α_{ij}), forming opaque internal representations.

BERT first introduced the use of a bidirectional training strategy, which simultaneously conditions each token's representation on both preceding and subsequent tokens within a sequence (Bosley et al., 2023; Sbei et al., 2024). BERT had sophisticated internal semantic and contextual representations, resulting in superior performance across diverse NLP tasks, including text classification, sentiment analysis, and question answering (Devlin et al., 2019). However, the architectural complexity of BERT, which typically comprises multiple layers of self-attention blocks and millions of parameters, results in significant interpretability challenges (Gao et al., 2021). In addition, recent research has highlighted the susceptibility of transformers to exploiting spurious correlations or superficial features within datasets, such as sentence-end markers or punctuation, instead of deeper semantic relationships (Talebi et al., 2024).

2.1.1 Interpretability and Explainability in Transformers

Interpretability and explainability are interconnected concepts that are essential for ensuring responsible and trustworthy deployment of transformer-based models. While often used interchangeably, they have slightly different meanings: interpretability refers to the degree to which a human can understand the cause of a model's internal logic or decision-making, while explainability involves providing contextual justifications for predictions (Leblanc & Germain, 2024).

Traditionally, simpler models, like decision trees, achieve interpretability through intuitive visualizations, such as feature importance plots, enabling users to observe the factors that influence each prediction (Quinlan, 1986). However, neural networks and transformers consist of numerous layers with intricate interactions among thousands or millions of parameters, with nonlinear frameworks that do not naturally lend themselves to visualization or intuitive interpretation (Lipton, 2017; Gao et al., 2021).

As shown in Fig. 1, the hidden-layer representations, especially due to the entangled dynamics of multi-head attention, make it nearly impossible to comprehensively trace how inputs map to outputs. Ideally, one would want to be able to understand how a specific input leads to a specific prediction.

Attempts to use built-in decision metrics, such as softmax scores, have also failed as they do not guarantee reliability because they are not calibrated to reflect the true likelihood of correctness: they only provide confidence based on a model's internal logits (Pearce et al., 2021). These scores can sometimes be used as a proxy for confidence, but often cannot be generalized (Ozbulak et al., 2018).

To address the above issue, several interpretability techniques have been proposed. Attention visualization, for instance, is a technique where attention weights within transformer layers are examined to infer token importance and relationships (Chefer et al., 2021). Studies have demonstrated that attention weights may provide intuitive insights into linguistic structure and the semantic rationale behind predictions (Vig, 2019). Recent research is mixed on the reliability of attention weights to indicate true token importance, with some works finding that attention patterns can be redundant or misleading and are not always directly correlated with prediction outcomes (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019).

In addition, gradient-based attribution methods, such as integrated gradients (Sundararajan et al., 2017), provide explanations by attributing predictions directly back to input tokens or embeddings based on their gradients. These methods quantify the sensitivity of model predictions to individual input components, enabling improved interpretation of model behavior. Similarly, layer-wise relevance propagation has been adapted to transformers, distributing prediction relevance backward through the model's layers to highlight critical input tokens and patterns that drive particular predictions (Chefer et al., 2021). Most approaches remain computationally demanding, especially for larger transformer architectures, and may not always yield easily interpretable, actionable insights.

Other methodologies include Bayesian neural networks, which model parameters as probability distributions to propagate uncertainty; ensemble methods, which combine predictions from multiple models to estimate variability; and Monte Carlo Dropout, which approximates Bayesian inference using dropout during inference (Kwon et al., 2020). Recent innovations, like spectral-normalized neural Gaussian processes, further integrate neural networks and transformers with Gaussian processes to produce uncertainty estimates (Liu et al., 2020). However, computational costs for all of these methods can be high as they require multiple forward passes. Furthermore, ensuring calibration remains an ongoing difficulty, with poorly calibrated models often producing overconfident predictions that erode trust in their outputs (Cardenas et al., 2023). This issue is particularly pronounced when models are confronted with out-of-distribution data, where inputs diverge significantly from the training distribution.

2.2 Conformal Prediction

One method that has had some success is CP. It is a distribution-free framework for uncertainty quantification that offers formal statistical guarantees. By constructing *prediction sets* that contain the true label with a user-defined probability (e.g., 95%), CP ensures that the model's uncertainty estimates are valid regardless of the underlying data distribution or model architecture (Angelopoulos & Bates, 2021). Central to CP is the use of a *nonconformity measure*, which quantifies how unusual or "nonconforming" a test example is relative to a set of calibration examples (Angelopoulos & Bates, 2021). Prediction set sizes can be adjusted based on how "confident" a model is in its predictions.

2.2.1 Nonconformity Measures

The flexibility of CP arises from its ability to incorporate various nonconformity measures, which map model predictions to scalar values that indicate how atypical a prediction is. Common choices include:

- **Distance-based measures**, such as cosine or Euclidean distance to same-class vs. different-class neighbors.
- **Softmax-based measures**, including margin scores between the top-1 and second-highest probabilities, or scores based on adversarial robustness (Huang et al., 2025).
- **Feature-space metrics**, computed from fixed-layer embeddings to reduce noise and cost.

Distance metrics can have vastly different performance and computation results. 1-nearest neighbor (1-NN), which measures distances between test points and calibration data in the raw input space, is computationally efficient, but struggles to capture the semantic richness of modern high-dimensional representations. As a result, it tends to produce overly conservative prediction sets that lack informativeness (Papernot & McDaniel, 2018). Conversely, deep k-nearest neighbors (DkNN) uses hidden-layer activations to compare

test points to their closest training counterparts, thus improving semantic alignment (Huang et al., 2025). However, DkNN is computationally intensive, as it often computes nearest neighbors across multiple layers and stores substantial internal representations. Compare this to softmax-based approaches, which are more efficient, but can be vulnerable to overconfidence and adversarial inputs (Cardenas et al., 2023).

2.3 CONFINE: CP for Neural Networks

CONFINE (Huang et al., 2025) introduces a novel feature-based nonconformity score that compares the test example to its top-k nearest neighbors in an intermediate representation space (e.g., from a specific network layer). It compares distances to same- vs. different-class neighbors in the embedding space, enabling p-value estimation (discussed later) based on semantic similarity. Thus, CONFINE leverages the internal structure of neural networks — specifically, representations from a fixed intermediate layer — to compute nonconformity scores that more accurately reflect semantic similarity (Huang et al., 2025).

For each candidate class label y, CONFINE calculates the average distance between the test input's embedding and its k nearest neighbors in a calibration set B. This yields the following nonconformity score:

$$A_k(B, x, y) = \frac{\min\left\{\frac{1}{k}\sum_{i=1}^{k} \text{CosDist}(f(x), f(x_i)) \mid y_i = y\right\}}{\min\left\{\frac{1}{k}\sum_{i=1}^{k} \text{CosDist}(f(x), f(x_i)) \mid y_i \neq y\right\}}, \tag{1}$$

where $f(x)$ denotes the embedding from a selected layer of the neural network, $(x_i, y_i) \in B$, and $\text{CosDist}(a, b) = 1 - \frac{a \cdot b}{\|a\|\|b\|}$ is the cosine distance.

By using embeddings from a single pre-defined layer (chosen via hyperparameter grid search), CONFINE avoids the high memory and computational costs associated with DkNN approaches that rely on multiple layers. This simplification enables real-time applicability while retaining a meaningful structure for interpretability. In addition, CONFINE supports class-conditional p-value computation, offering improved coverage and label-wise reliability in settings with imbalanced data (Huang et al., 2025).

CONFIDE builds on the strengths of CONFINE while adapting its framework to better suit transformer language architectures.

2.4 Conformal Prediction for Transformers

Most current CP approaches for transformers suffer from one or more of the following issues.

First, they predominantly rely on embeddings derived exclusively from the transformer's final layer. While final-layer representations encapsulate a condensed semantic summary, neglecting intermediate layers discards the hierarchical and nuanced linguistic information inherently captured throughout the model's architecture (Huang et al., 2025).

Second, existing CP implementations are limited to large-scale transformer models, such as full-scale BERT or RoBERTa variants. Little research has been conducted on using CP on resource-limited models.

Third, simplistic nonconformity measures, usually direct softmax probabilities, dominate the current landscape. Such simplistic metrics may not always capture the complexity and semantic nuance within transformer embedding spaces, reducing the interpretability and efficacy of prediction sets (Sikar et al., 2025).

Several recent studies have started to address some of these issues. Giovannotti et al. integrate CP with transformer architectures for paraphrase detection tasks, using raw transformer output scores instead of softmax probabilities and introducing variants, such as Mondrian conformal predictors (Giovannotti et al., 2021). Dey et al. propose inductive conformal prediction methods for text infilling and part-of-speech tagging, demonstrating finite-sample control of type-I error with transformer and BiLSTM embeddings (Dey et al., 2021). Lee et al. use transformer decoders to produce quantile-based conformal intervals for time series predictions, effectively capturing temporal dependencies (Lee et al., 2024).

2.4.1 Research Gaps in Conformal Transformers

The above analysis reveals three substantial gaps that future research must explicitly address to advance conformal prediction for transformers.

Underexplored lightweight transformer models: Lightweight models remain under-investigated in CP applications. Their computational efficiency makes them especially valuable for edge-device deployment, yet their capability to support reliable uncertainty quantification through CP remains largely unexplored.

Limited experimentation with diverse nonconformity measures: The effectiveness of conformal prediction strongly depends on the selected nonconformity measure. Alternative metrics, such as Mahalanobis distance (capturing correlations between embedding dimensions), offer potentially improved interpretability, precision, and reliability, yet remain largely ignored.

Lack of evaluation on challenging benchmarks: Current evaluations predominantly occur within straightforward NLP tasks or specialized domains (e.g., paraphrase detection). Rigorous benchmarks that demand advanced linguistic reasoning and comprehensive understanding, such as SUPERGLUE, are underexplored.

2.4.2 Positioning CONFIDE

CONFIDE integrates intermediate-layer representations from transformer models to harness richer semantic information, thereby enhancing prediction robustness and reliability. By focusing both on lightweight architectures, such as BERT-tiny and BERT-small, CONFIDE also evaluates the feasibility and benefits of conformal prediction in resource-constrained, real-time, and edge-computing applications, a significant step toward democratizing reliable uncertainty estimation.

CONFIDE also investigates advanced, embedding-sensitive nonconformity measures, such as Mahalanobis distance, and application of principal component analysis (PCA). This thorough exploration provides deeper insights into how transformer-specific nonconformity measures can enhance prediction intervals and sets.

Lastly, CONFIDE is rigorously evaluated on challenging and comprehensive NLP benchmarks, notably three SUPERGLUE datasets, providing robust empirical validation of the method's capability to handle sophisticated reasoning and linguistic complexity.

3 Datasets and Models

This section introduces our experimental setup to provide grounding for many of CONFIDE's design choices. We discuss datasets used, model choice, and model fine-tuning.

3.1 GLUE and SuperGLUE Benchmarks

To evaluate CONFIDE, we focus on tasks from two of the most widely used natural language understanding benchmarks: the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) and its successor, the SuperGLUE benchmark (Wang et al., 2020). Together, these benchmarks span a diverse array of classification tasks, ranging from binary question answering to multi-class entailment detection, providing a rigorous testing ground for uncertainty quantification methods.

We restrict our evaluation to tasks that are labeled as classification problems. Accordingly, we exclude tasks involving regression or span-level predictions (e.g., STS-B). Therefore, the chosen tasks are robust and varied. For example, BoolQ is a binary question-answering task drawn from naturally occurring queries, often featuring ambiguous phrasing and noisy labels. In contrast, CB (CommitmentBank) is a three-class natural language inference task with very limited training data, making it a valuable benchmark for assessing calibration under data scarcity.

A full list of included tasks and their characteristics is provided in Appendix A.

3.2 Encoder-Based Transformer Models

We evaluate CONFIDE using encoder-only transformer architectures, focusing specifically on the BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) model families. These models are well aligned with our goals of developing a scalable, interpretable, and reproducible CP framework for several key reasons. We focus on language transformers, in particular, due to their ubiquity across modern NLP applications, the abundance of open-source pretrained models, and the broad relevance of text classification tasks in both academic and industrial domains.

First, encoder-only models are architecturally optimized for classification tasks. Unlike decoder-only or encoder-decoder architectures (e.g., GPT or T5), which are tailored to generation and sequence-to-sequence modeling, encoder-based transformers are designed to produce rich, contextualized sentence-level representations suitable for downstream classification (Raffel et al., 2023; Devlin et al., 2019). This makes them naturally compatible with CONFIDE's prediction set formulation and avoids the added complexity of autoregressive decoding or encoder-decoder alignment.

Second, encoder architectures expose intermediate hidden states across multiple layers, enabling us to probe semantic representations at various depths. This is particularly valuable for CP, where the choice of embedding layer directly impacts the quality and stability of nonconformity scores. Layer-level flexibility also supports CONFIDE's core principle of reusing internal representations without model retraining.

Third, we use open-source, pretrained models from the HuggingFace Transformers library to ensure transparency and reproducibility. We include BERT-variant models to validate CONFIDE's generalization beyond its original scope and to avoid overfitting our method to a single backbone. This diversity strengthens the validity of our results.

We evaluate four pretrained transformer variants:

- `BERT-tiny`: 2 transformer layers, ∼4M parameters
- `BERT-small`: 4 transformer layers, ∼29M parameters
- `RoBERTa-base`: 12 transformer layers, ∼125M parameters
- `RoBERTa-large`: 24 transformer layers, ∼355M parameters

3.3 Fine-tuning Protocol

To ensure consistency across all datasets and model variants, we apply a standardized fine-tuning protocol using the HuggingFace `Trainer` framework, adapted for stability and performance. Each model is fine-tuned independently per dataset–task pair, with the following procedures:

- **Model initialization**: We load pretrained model checkpoint from HuggingFace and replace the classification head to match the number of target labels.
- **Tokenization strategy:** We tokenize the inputs using task-aware preprocessing. Sentence-pair tasks, such as BoolQ, RTE, and CB, use paired input formatting, while datasets like MultiRC are treated as binary classification problems over each (question, answer) pair.
- **Input processing**: We pad all inputs and truncate them to a maximum sequence length of 512 tokens.
- **Optimization:** We use the AdamW optimizer with a learning rate of 5×10^{-5}, a linear warm-up over the first 500 steps, and weight decay of 0.01. We train models for up to 10 epochs with early stopping (usually around 3-5 epochs) based on validation accuracy to reduce overfitting and unnecessary computation (Howard & Ruder, 2018).
- **Evaluation metrics**: We track accuracy, precision, recall, and F1-score during training. We use macro-averaging for multi-class tasks (e.g., CB), and binary-averaging for two-class tasks.

- **Model selection:** We choose the final model checkpoint based on the highest validation accuracy. All downstream CONFIDE evaluations use this fixed checkpoint to ensure a consistent prediction backbone.

This protocol enables a fair comparison across models and tasks while minimizing confounding factors unrelated to the CONFIDE framework.

4 Methodology

This section introduces CONFIDE, built atop the CONFINE framework, for transformer-based language models. Applying CONFINE directly to transformer models presents several unique challenges, especially in terms of identifying meaningful embedding layers and maintaining computational efficiency. Our work tailors CONFINE to language encoder-based transformer models. We begin by outlining the original CONFINE algorithm, then describe our transformer-specific modifications, and finally detail the design decisions underlying CONFIDE.

4.1 The CONFINE Algorithm

The CONFINE algorithm provides prediction sets that are calibrated under the exchangeability assumption and leverages the internal structure of neural networks to compute semantically meaningful nonconformity scores.

4.1.1 Key Assumptions and Definitions

- **Exchangeability assumption**: Let $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$ be a sequence of labeled examples. The data are assumed to be *exchangeable* if their joint distribution is invariant under permutations. That is, for any permutation π of $\{1, \ldots, n\}$,

$$\Pr[(x_1, y_1), \ldots, (x_n, y_n)] = \Pr[(x_{\pi(1)}, y_{\pi(1)}), \ldots, (x_{\pi(n)}, y_{\pi(n)})]. \tag{2}$$

- **P-values and prediction sets**: For a test input x_{l+1}, we compute the p-value of each candidate label y_j as:

$$p(y_j) = \frac{\#\{i = 1, \ldots, l : \alpha_i \geq \alpha_{l+1}\} + 1}{l + 1}, \tag{3}$$

  where α_i's are the nonconformity scores from the calibration set, and $\alpha_{l+1} = A(B, x_{l+1}, y_j)$. The conformal prediction set is:

$$\Gamma^{\varepsilon}(x) = \{y_j \in \mathcal{Y} \mid p(y_j) > \varepsilon\}. \tag{4}$$

- **Credibility and confidence**:

$$\text{Credibility} = \max_{y_j \in \mathcal{Y}} p(y_j), \quad \text{Confidence} = 1 - \max_{y_j \neq y^*} p(y_j), \tag{5}$$

  where $y^* = \arg\max p(y_j)$. Credibility reflects how well the most probable class conforms; confidence measures its separation from the rest.

- **Marginal coverage guarantee**: Conformal prediction guarantees that, under exchangeability,

$$\Pr[y \in \Gamma^{\varepsilon}(x)] \geq 1 - \varepsilon. \tag{6}$$

  This is the standard coverage guarantee: the true label appears in the prediction set with probability at least $1 - \varepsilon$, averaged over the test distribution.

- **Class-conditional coverage guarantee**: A stronger guarantee holds if the coverage condition is satisfied *within each class*:

$$\Pr[y_i \in \Gamma_i^{\varepsilon} \mid y_i = Y_j] \geq 1 - \varepsilon, \quad \forall Y_j \in \mathcal{C}. \tag{7}$$

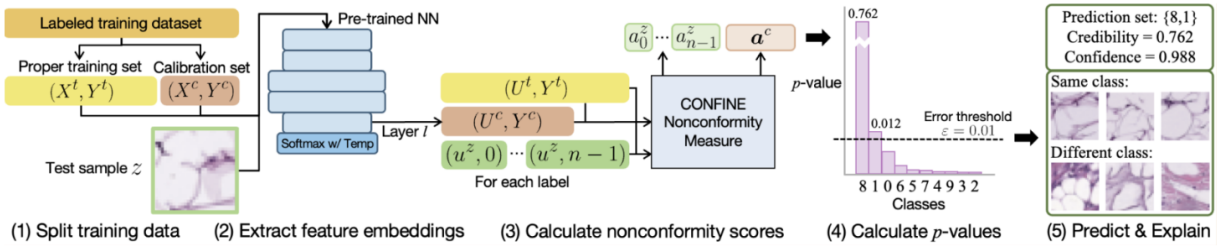


Figure 2: Overview of the CONFINE algorithm adapted from (Huang et al., 2025). Feature embeddings are extracted from a fixed layer and class-conditional nearest neighbors are used to compute nonconformity scores.

This ensures that every class is covered fairly, not just on average. This guarantee is achieved by computing p-values using only calibration points from the same class, yielding coverage control per class label.

- **Efficiency and correct efficiency**: The average size of the prediction set,

$$\mathbb{E}[|\Gamma^{\varepsilon}(x)|], \tag{8}$$

measures the method's efficiency, where smaller sets are more informative. However, this metric can be manipulated by outputting small sets that frequently omit the correct label. Therefore, we also consider *correct efficiency*, the expected size of the prediction set conditioned on it containing the true label (Huang et al., 2025).

4.1.2 Algorithm Tracing

We implement the process illustrated in Fig. 2.

- Given a test input x, CP computes a prediction set $\Gamma^{\varepsilon}(x) \subseteq \mathcal{Y}$ such that the true label y lies inside it with probability at least $1 - \varepsilon$.
- The method requires a nonconformity measure $A(B, x, y)$, which quantifies how atypical a candidate label y is for input x, relative to a calibration set B.
- The corresponding p-value for a candidate label y_j is computed as:

$$p(y_j) = \frac{1}{|\mathcal{C}| + 1} \left(\# \{ i : \alpha_i \geq \alpha_{l+1} \} + 1 \right), \tag{9}$$

where α_i's are the nonconformity scores from the calibration set, and $\alpha_{l+1} = A(B, x, y_j)$ is the score for the test input.

- The prediction set is then defined as:

$$\Gamma^{\varepsilon}(x) = \{ y_j \in \mathcal{Y} \mid p(y_j) > \varepsilon \}. \tag{10}$$

Note that the original CONFINE framework was developed for convolutional neural networks (CNNs) and multilayer perceptrons, where internal representations are single, global feature vectors (e.g., after spatial pooling or dense layers). For CONFIDE, we need to take into account some architectural differences in transformer models, like BERT:

- No fixed global representation: Transformers produce one embedding per token.
- Layer variability: Unlike CNNs, deeper layers in transformers do not necessarily yield more abstract or stable representations.
- High-dimensional representations: Flattening token-level outputs results in very high-dimensional vectors.

4.2 The CONFIDE Algorithm

CONFIDE extends the CONFINE framework to transformer models by introducing two key adaptations: (1) a representation mode selector to aggregate token-level embeddings and (2) an optional dimensionality reduction step and distance-metric modifier for handling high-dimensional transformer outputs. The full pipeline is detailed in Algorithm 1.

From a specified transformer layer ℓ, CONFIDE extracts representations using either of two modes: **Flattened**, which reshapes the entire token matrix into a vector, or **Attention**, which selects the [CLS] token embedding. Only correctly predicted training samples are retained for reference, improving the reliability of k-NN comparisons.

Step 1: Representation Extraction

For each labeled training example (x_i, y_i), we extract hidden state representations, which are extracted by flattening the hidden state matrix, from a fixed encoder layer ℓ of the transformer. The method of extraction depends on the selected mode: Flattened, which reshapes the entire token matrix into a vector, or Attention, which selects the [CLS] token embedding. Only correctly predicted training samples, those where $\hat{y}_i = y_i$, are retained for reference.

Note that while this paper shows results for both the flattened and attention-based variation, we emphasize that CONFIDE-Flattened is significantly more computationally expensive due to the complexity of flattening rich embeddings. Therefore, CONFIDE-Attention is used as the primary approach throughout our experiments.

Algorithm 1 The CONFIDE Algorithm

Input: Labeled training dataset (X, Y); model $\mathcal{M}$; significance level ϵ; test input z; layer ℓ; number of neighbors k; representation mode (`Flattened` or `Attention`); PCA flag; distance metric

Step 1: Extract Representations from Training Data

foreach *training example* (x_i, y_i) **do**

  Run $\mathcal{M}(x_i)$ to extract H_i from layer ℓ **if** *mode* == ***Flattened*** **then**

    $h_i \leftarrow \texttt{reshape}(H_i, [1, \text{seq\_len} \cdot \text{hidden\_dim}])$; `// Flatten full matrix`

  else if *mode* == ***Attention*** **then**

    $h_i \leftarrow H_i[\texttt{CLS}]$; `// Use [CLS] token`

  if $\hat{y}_i = y_i$ **then**

    Store h_i in pool for class y_i

Step 2: Apply PCA (Optional)

if *PCA enabled* **then**

  Fit PCA on $\{h_i\}$ to retain 95% variance and transform all vectors

Step 3: Fit k-NN Models per Class

foreach *class* $c \in \mathcal{Y}$ **do**

  Fit k-NN on $\{h_i \mid y_i = c\}$ using selected distance metric

Step 4: Calibration Nonconformity Scoring

foreach *calibration example* (x, y) **do**

  Extract $h(x)$ as above; Compute $A_k(x, y) = \frac{\text{avg dist to class } y}{\text{avg dist to other classes}}$; Store score in α

Step 5: Compute Test Prediction Sets

foreach *test input* x **do**

  foreach *label* $y \in \mathcal{Y}$ **do**

    Compute $A_k(x, y)$ and p-value: $p_y(x) = \frac{1 + \#\{\alpha_i \geq A_k(x,y)\}}{1 + n_{\text{cal}}}$

  Output prediction set: $\Gamma^\epsilon(x) = \{\, y \in \mathcal{Y} \mid p_y(x) > \epsilon \}$

Step 2: Apply PCA (Optional)

To mitigate the curse of dimensionality and improve distance metric robustness, PCA may be optionally applied. If enabled, PCA reduces the dimensionality of all h_i vectors while retaining at least 95% of the variance. This transformation is also applied to calibration and test embeddings.

Step 3: Fit Class-Conditional k-NN Models

For each class $c \in \mathcal{Y}$, we train a K-nearest neighbor model using the representations $\{h_i \mid y_i = c\}$, based on the user-specific distance metric.

Step 4: Calibration Nonconformity Scoring

For each calibration point (x, y), a nonconformity score $A_k(x, y)$ is computed using the same embedding extraction and PCA pipeline. These scores quantify how typical a point x is for its true class y. They are stored and later used to derive p-values for each candidate class during prediction.

Step 5: Test-time Prediction Set

At test time, for each input x, a p-value is computed for every possible class $y \in \mathcal{Y}$, as well as a prediction set. This set guarantees marginal coverage $1 - \epsilon$ under the exchangeability assumption. The size and accuracy of this set depend on the quality of extracted embeddings and chosen distance metric.

4.2.1 Evaluation Metrics and Baselines

We evaluate CONFIDE variants using three core metrics:

- **Accuracy**: Standard top-1 classification accuracy.
- **Correct efficiency**: Expected size of prediction sets *conditioned* on including the correct label (Huang et al., 2025).
- **Coverage**: Fraction of predictions for which the prediction set contains the ground-truth label.

We compare CONFIDE against the following baselines:

- VanillaNN: 1-nearest neighbor using raw embeddings and cosine distance (Papadopoulos et al., 2007).
- NM1/NM2: Softmax-based nonconformity scoring methods, originally proposed in (Vovk et al., 2005).

4.3 Justifying Design Choices

Building on the CONFIDE variants described in Algorithm 1, we detail the key design choices that significantly impact performance. Our implementation supports a wide range of configurations and we systematically evaluate each axis of variation to identify robust, high-performance settings.

Embedding selection. A critical component of CONFIDE is how fixed-dimensional representations are extracted from transformer models. We considered two primary approaches:

- Flattened: Activations from internal transformer layers (as specified by the user) are intercepted via forward hooks and flattened into a single vector per input.
- Attention: As a lighter-weight alternative, we extract only the [CLS] token embedding from a specified layer without registering custom hooks.

In empirical evaluations, both versions produce fairly similar results, although, due to memory and running time issues, this paper emphasizes the attention method.

Choice of distance metric. We implement support for both cosine and Mahalanobis distances when querying the k-nearest neighbor models:

- **Cosine distance:** For two vectors u and v, cosine distance is defined as:

$$\text{Dist}_{\cos}(u, v) = 1 - \frac{u^\top v}{\|u\| \, \|v\|}$$

- **Mahalanobis distance:** For a feature vector x and a class-specific distribution with covariance matrix Σ, the Mahalanobis distance is computed as:

$$\text{Dist}_{\text{Mah}}(x, \mu) = \sqrt{(x - \mu)^\top \Sigma^{-1} (x - \mu)}$$

  where μ is the class mean. This metric captures feature correlations but is sensitive to ill-conditioned or low-rank Σ when few training examples are available.

Why Mahalanobis distance? The original CONFINE framework employed cosine distance for computing similarity in the embedding space due to its scale-invariant properties and robustness in high-dimensional representations. In CONFIDE, we extend this design by incorporating Mahalanobis distance as an alternative metric. This choice is motivated by its ability to capture feature correlations, which can be particularly valuable in transformer embeddings where dimensions are not independent.

Unlike Euclidean or Manhattan distances, Mahalanobis distance considers the data covariance structure. This allows the k-NN model to account for ellipsoidal structures in the feature space, improving discrimination in cases where cosine similarity may fail to capture variance along non-principal directions (Moutafis et al., 2017).

Dimensionality reduction. To mitigate high-dimensionality effects and improve numerical conditioning, we optionally apply PCA to the training activations before fitting k-NN models. PCA is computed as follows:

$$Z = XW, \quad \text{where } W \text{ retains components explaining } \geq 95\% \text{ variance}$$

where X is the centered activation matrix and W is the eigenvector matrix of the covariance of X. PCA is applied post-embedding extraction and before distance computation during inference phases, including before calibration and testing. We retain 95% of the variance when applying PCA, following common practice in machine learning to preserve essential data structure while discarding noise (Jolliffe & Cadima, 2016). This threshold strikes a balance between dimensionality reduction and performance.

Hyperparameter selection. Three key hyperparameters are tuned for each configuration:

- *Layer*: the internal or softmax layer selected.
- k: the number of nearest neighbors used in nonconformity scoring.
- T: the temperature scaling factor for logits when softmax-based predictions are required,

We conduct a grid search over appropriate ranges for each hyperparameter within every (model, dataset) configuration, ensuring that each CONFIDE variant is tuned for both predictive accuracy and valid coverage. The values for k and T follow those used in the original CONFINE framework (Huang et al., 2025), facilitating a direct comparison. For the *Layer* parameter, we select a diverse set of layers spanning the early, middle, and late stages of the encoder. These layers are chosen to reflect different architectural roles, ranging from multi-head attention blocks to linear feedforward components, to capture the progression of semantic abstraction across the model. The full list of layers chosen can be found in Appendix B.2.

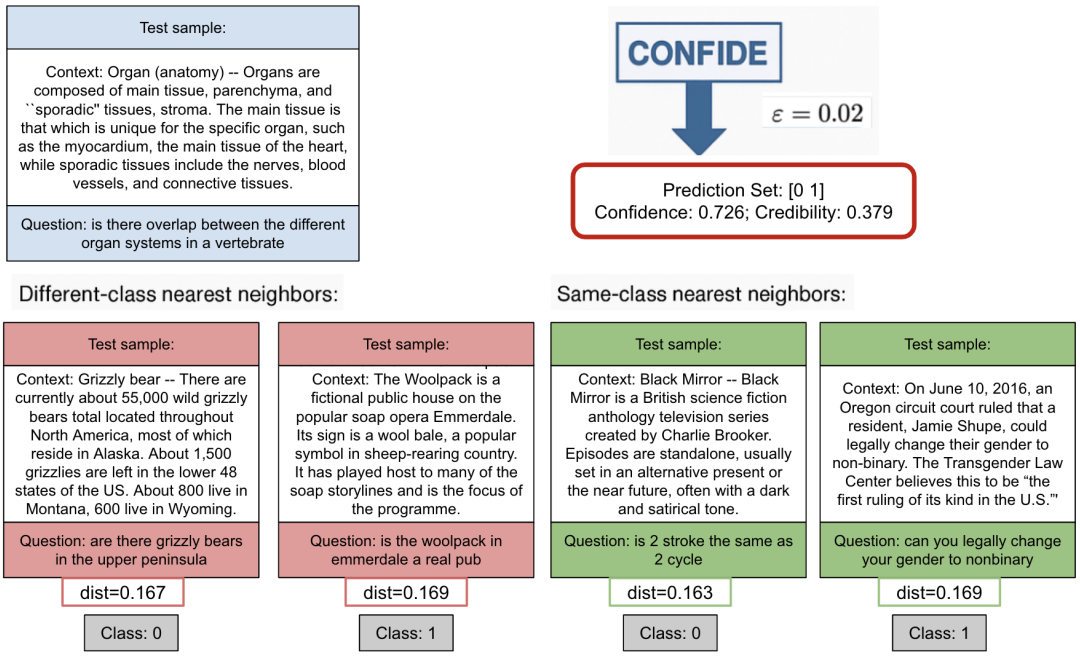


Figure 3: Failure case on BoolQ (BERT-tiny). CONFIDE fails to distinguish between same-class and different-class neighbors, resulting in a high-uncertainty prediction.

Combinatorial design evaluation. To exhaustively evaluate CONFIDE, we explore all combinations of:

- Distance metric (Cosine vs. Mahalanobis).
- Dimensionality reduction (PCA applied vs. not applied).
- Classwise (Classwise calibrated or not).

Select attempts of attention vs. flattening embedding variants are also considered. These design axes are evaluated across all four transformer models and eleven tasks. Each configuration undergoes comprehensive hyperparameter tuning to ensure fair and reproducible comparisons. Our results, presented in the following section, highlight the impact of these design choices on both prediction set coverage and efficiency.

5 Experimental Results

Next, we present the experimental results.

5.1 CONFIDE Provides Interpretability

CONFIDE offers interpretable insights on individual test cases. Figs. 3 and 4 demonstrate how an end-user can inspect predictions by visualizing the predicted set, associated confidence metrics, and the structure of nearest neighbors. We note that in contrast to image-based models, where neighbor similarity reflects visual resemblance, transformer-based language models encode similarity based on shared semantic structures, syntactic roles, or contextual usage. Thus, nearby neighbors in CONFIDE often reflect questions or passages with comparable phrasing, topic domains, or logical structure. Sometimes, however, interpretability can be less intuitive than in vision tasks. While image-based models yield visually similar examples, linguistic similarity often reflects subtler semantic or syntactic patterns that may be harder for users to recognize without context.

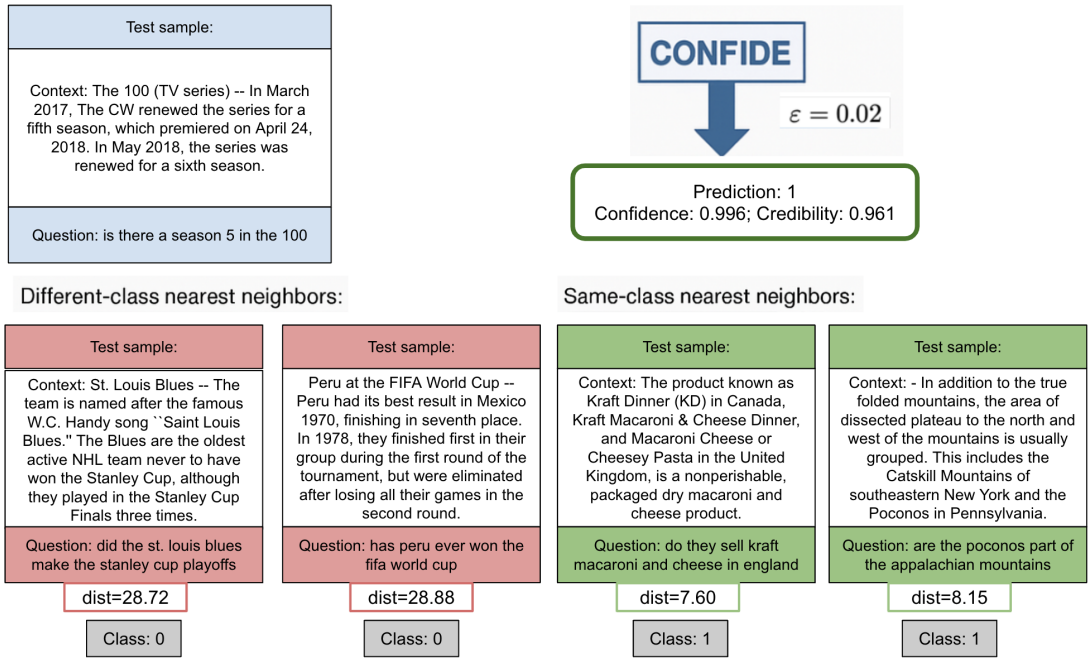


Figure 4: Success case on BoolQ (BERT-tiny). CONFIDE separates neighbors by class, yielding a confident and credible singleton prediction.

Fig. 3 presents a failure case on the BoolQ dataset. The model outputs a full prediction set $\{0, 1\}$, with an associated *credibility* of only 0.379, signaling substantial uncertainty: the model is unable to confidently distinguish between the two classes. Importantly, the average distances to same-class and different-class neighbors are nearly identical (0.16–0.17), indicating that the embedding space lacks meaningful class separation for this example. Fittingly, the test sample is a medically themed question; an end user would see that the model has low representational certainty and could defer to a domain expert, such as a doctor, for validation.

In contrast, Fig. 4 shows a successful CONFIDE prediction. The model returns a singleton prediction $\{1\}$ with high *credibility* (0.961) and *confidence* (0.996). Here, the average distance to same-class neighbors is significantly smaller ($\sim$7.87) than to different-class neighbors (28.80), validating the low nonconformity score for class 1. This separation suggests the test point lies well within a class-coherent region of the embedding space.

These examples highlight CONFIDE's utility as a tool for interpretability and model auditing. By identifying similar and distant neighbors, users gain visibility into the stability and discriminative structure of a model's internal representations.

5.2 Performance on Resource-Constrained Transformer Models

We first evaluate CONFIDE on compact transformer models: BERT-tiny and BERT-small. These models offer constrained capacity, enabling us to assess CONFIDE's calibration in low-resource regimes. Full results are reported in Tables 1–6. CONFIDE-A targets accuracy and CONFIDE-C targets correct efficiency.

Reliable calibration with modest capacity. Despite their reduced depth and size, both models benefit significantly from CONFIDE's nonconformity-based calibration. Notably, CONFIDE improves upon NM1/NM2 baselines, which both rely solely on softmax logits (final layer of the model) for their nonconformity scores. In contrast, CONFIDE flexibly selects semantically rich internal layers, enabling it to outperform strong baselines across several benchmarks.

Table 1: Performance on GLUE benchmarks using BERT-tiny: CoLA, MNLI, MRPC, and QNLI. Hyperparameters used can be found in Appendix B.1

Method	CoLA		MNLI		MRPC		QNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.5858	–	0.6949	–	0.7010	–	0.7895	–
1-Nearest Neighbor	0.5791	0.5762	0.3403	0.3254	0.6103	0.6103	0.5067	0.5050
NM (2)	0.5858	0.5724	0.6949	**0.6936**	0.7010	0.7010	0.7895	0.7878
CONFIDE-A (ours)	**0.6903**	**0.6903**	**0.6982**	0.6907	**0.7304**	0.7181	**0.8157**	0.8104
CONFIDE-C (ours)	**0.6903**	**0.6903**	0.6975	0.6932	0.7279	**0.7279**	0.8131	**0.8129**

Table 2: Performance on GLUE benchmarks using BERT-tiny: QQP, RTE, SST-2, and WNLI.

Method	QQP		RTE		SST-2		WNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.8116	–	0.5668	–	0.8303	–	0.5634	–
1-Nearest Neighbor	0.6423	0.6391	0.4693	0.4549	0.5011	0.4794	0.2394	0.2394
NM (2)	0.8214	0.8073	0.5668	0.5668	0.8303	0.8280	0.5634	0.5634
CONFIDE-A (ours)	**0.8566**	**0.8530**	**0.5957**	0.5848	**0.8372**	0.8326	**0.8360**	0.8245
CONFIDE-C (ours)	**0.8566**	**0.8530**	0.5884	**0.5884**	0.8349	**0.8349**	0.8326	**0.8280**

For example, on SST-2, CONFIDE-C boosts top-1 correct efficiency to **0.8911** on BERT-small and **0.8349** on BERT-tiny—compared to just **0.8830** and **0.8280**, respectively, under NM1/NM2. On QQP, BERT-small improves from **0.8739** (NM) to **0.8846** (CONFIDE-C), while BERT-tiny improves from **0.8073** to **0.8530**. MRPC shows similar trends: on BERT-small, CONFIDE-C achieves **0.7574** correct efficiency, compared to **0.7034** with a previous NM; BERT-tiny moves from **0.7010** to **0.7279**. These results demonstrate that CONFIDE provides substantial improvements beyond what is achievable with output-layer-only methods.

Layer-level design matters: Intermediate layers outperform softmax. An important trend across both models is that CONFIDE achieves its strongest performance not from the final softmax layer but from intermediate transformer layers, typically in the first half of the model. Based on our grid search (see Appendix B.2), optimal configurations often use mid-layer embeddings (Appendix B.1):

- For BERT-tiny, top-performing runs often used **layer 16**, corresponding to the first encoder layer when flattened.
- For BERT-small, strong results are frequently found at **layers 27 and 34**, again early in or middle of the stack.

These findings suggest that early-layer representations capture richer, less overconfident features that are especially amenable to class-conditional distance metrics. Specifically, because these embeddings retain a nuanced semantic structure without being overly aligned to the model's final output, they allow CONFIDE to more effectively compare a test input's proximity to examples from the same class versus different classes. However, this advantage comes at a cost: using early hidden states increases compute and memory overhead during inference due to the need for large flattened vectors and distance comparisons. We discuss this tradeoff in greater detail in Section 6 and Appendix C.1.

A strong model matters: BERT-small > BERT-tiny. Across nearly every task, BERT-small consistently outperforms BERT-tiny in both test accuracy and correct efficiency. On MRPC, QNLI, and QQP, the performance gap is clear: BERT-small with CONFIDE-C achieves correct efficiencies of **0.7574**, **0.8697**,

Table 3: Performance on SuperGLUE benchmarks using BERT-tiny: BoolQ, CB, and MultiRC.

Method	BoolQ		CB		MultiRC	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.6596	–	0.5357	–	0.6219	–
1-Nearest Neighbor	0.5602	0.5590	0.4286	0.4286	0.5192	0.4792
NM (2)	0.6596	0.6529	0.5357	0.5000	0.6219	0.6209
CONFIDE-A (ours)	**0.6636**	**0.6630**	**0.7321**	**0.7143**	**0.6275**	0.5918
CONFIDE-C (ours)	**0.6636**	**0.6630**	**0.7321**	**0.7143**	0.6244	**0.6229**

Table 4: Performance on GLUE benchmarks using BERT-small: CoLA, MNLI, MRPC, and QNLI.

Method	CoLA		MNLI		MRPC		QNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.7421	–	0.7914	0.7805	0.7230	–	0.8691	–
1-Nearest Neighbor	0.5791	0.5762	0.3403	0.3254	0.6103	0.6103	0.5067	0.5050
NM1/NM2	0.7421	0.7392	**0.7914**	0.7805	0.7230	0.7034	0.8691	0.8671
CONFIDE-A (ours)	**0.7546**	**0.7546**	0.7904	**0.7865**	**0.7672**	0.7525	**0.8704**	0.8669
CONFIDE-C (ours)	**0.7546**	**0.7546**	0.7904	**0.7865**	0.7598	**0.7574**	0.8699	**0.8697**

and **0.8846** respectively, whereas BERT-tiny peaks at **0.7279**, **0.8129**, and **0.8530**. Even modest increases in model depth (from 2 layers to 4, although the increase is larger once flattened) enable CONFIDE to form tighter, more accurate prediction sets.

Difficulties on hard and ambiguous datasets. Despite CONFIDE's consistent improvements, both models still struggle on several particularly difficult tasks — especially **CB**, **WNLI**, **BoolQ**, and **MultiRC** — where ambiguity, low supervision, or multi-label structure present major challenges. In these cases, even the best CONFIDE configurations offer only modest gains, as the base model itself often fails to learn a strong decision boundary.

- On **CB**, BERT-tiny achieves only **0.5357** test accuracy and BERT-small performs only slightly better at **0.6250**.
- **MultiRC** presents a unique challenge due to its partial correctness metric and longer input contexts. With BERT-tiny, CONFIDE-C reaches **0.6229** top-1 correct efficiency, and BERT-small reaches **0.6582**.

Overall, the underlying limitations of the models remain the primary constraint. In the most favorable configurations, CONFIDE improves test accuracy by up to 4.09% for BERT-tiny and 2.69% for BERT-small over the best baseline methods. These results highlight that while conformal calibration can meaningfully improve reliability, overall performance remains bottlenecked by base model capacity.

5.3 Performance on Larger Models

We now turn to high-capacity transformer models: RoBERTa-base and RoBERTa-large. Results are summarized in Tables 7–10.

CONFIDE maintains accuracy while consistently improving calibration. Across both RoBERTa variants, CONFIDE consistently improves top-1 correct efficiency relative to softmax-based baselines like NM1 and NM2, while preserving or slightly improving classification accuracy. For example, on MRPC, CONFIDE-C lifts correct efficiency from **0.8627** to **0.8848** on RoBERTa-base, and from **0.8578** to **0.8652** on RoBERTa-large (Tables 7, 10). Similar 1–2 percentage point gains are observed on RTE, SST-2, and CoLA: on RTE, CONFIDE-A and CONFIDE-C increase top-1 efficiency from **0.7112** (NM2) to **0.7256**

Table 5: Performance on GLUE benchmarks using BERT-small: QQP, RTE, SST-2, and WNLI.

Method	QQP		RTE		SST-2		WNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.8893	–	0.5884	–	0.8911	–	0.4648	–
1-Nearest Neighbor	0.6423	0.6391	0.4693	0.4549	0.5011	0.4794	0.2394	0.2394
NM1/NM2	0.8893	0.8739	0.5884	0.5884	0.8911	0.8830	0.4648	0.4648
CONFIDE-A (ours)	**0.8933**	0.8820	**0.6245**	**0.6173**	**0.8933**	0.8647	**0.5915**	0.5493
CONFIDE-C (ours)	0.8901	**0.8846**	**0.6245**	**0.6173**	0.8899	**0.8911**	0.5634	**0.5634**

Table 6: Performance on SuperGLUE benchmarks using BERT-small: BoolQ, CB, and MultiRC.

Method	BoolQ		CB		MultiRC	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.7275	–	0.6250	–	**0.6640**	–
1-Nearest Neighbor	0.5602	0.5590	0.4286	0.4286	0.5192	0.4792
NM1/NM2	0.7275	0.6269	0.6250	0.6250	**0.6640**	**0.6621**
CONFIDE-A (ours)	**0.7297**	0.6104	**0.7143**	0.6786	0.6625	0.6557
CONFIDE-C (ours)	0.7272	**0.7245**	**0.7143**	**0.7143**	0.6599	0.6582

and **0.7292**, respectively (Table 8), while CoLA improves from **0.8370** to **0.8495** (Table 7). Even on high-performing tasks like MNLI, CONFIDE-C yields a marginal improvement from **0.8497** to **0.8512**. Crucially, these calibration benefits are not achieved at the cost of accuracy. On RTE, CONFIDE-C increases accuracy from **0.7220** (NM2) to **0.7329**, with CONFIDE-A reaching **0.7365** — the highest among all variants (Table 8).

Larger models enable more confident prediction sets. RoBERTa-large exhibits a consistently narrower gap between test accuracy and correct efficiency, even before applying CONFIDE. However, CONFIDE still meaningfully refines this calibration. On MNLI and RTE, RoBERTa-large under CONFIDE-A achieves **0.8437** efficiency on CoLA and **0.7978** on RTE.

Beyond raw accuracy and efficiency, larger models also enable smaller, more compact prediction sets. For example, on the RTE benchmark, the average prediction set size among correct predictions is **1.84** for RoBERTa-base, compared to **1.99** for BERT-tiny. This indicates that the higher-capacity model is more confident in its predictions.

Scalability and resource constraints. Despite these improvements, CONFIDE faces scalability challenges when applied to large models. On long-context datasets, such as BoolQ and MultiRC, CONFIDE runs on RoBERTa-large exceeded available GPU memory during distance matrix construction and flattened-layer extraction. As a result, evaluation on all datasets was not possible. Details of memory usage, training time, and dataset-specific runtime characteristics are provided in Appendix A. All successfully completed results are included in Table 10.

5.4 Classwise and Aggregate Validity of CONFIDE on BERT-tiny

A central goal of conformal prediction is to guarantee that the true label is included in the model's prediction set with high probability—typically at least $1 - \varepsilon$, where ε is the target error rate. This is known as *marginal validity* and is achieved when the average coverage of the model's prediction sets exceeds this threshold. However, as emphasized in CONFINE (Huang et al., 2025), marginal validity alone does not ensure equitable performance across all classes. In many applications, particularly high-stakes ones like healthcare, what matters more is *class-conditional validity*: the guarantee that each individual class receives its own calibrated coverage. We analyze both marginal and classwise coverage on BERT-tiny using CONFIDE,

Table 7: Performance on GLUE benchmarks using RoBERTa-base: CoLA, MNLI, MRPC, and QNLI.

Method	CoLA		MNLI		MRPC		QNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.8408	–	–	–	0.8824	–	0.9231	–
1-Nearest Neighbor	0.6069	0.6021	–	–	0.5564	0.5490	–	–
NM1/NM2	0.8408	0.8370	**0.8608**	0.8497	0.8824	0.8627	0.9231	0.9098
CONFIDE-A (ours)	**0.8495**	0.8092	**0.8608**	0.8497	**0.8848**	0.8725	**0.9235**	0.9087
CONFIDE-C (ours)	**0.8495**	**0.8495**	0.8588	**0.8512**	**0.8848**	**0.8848**	**0.9235**	**0.9217**

Table 8: Performance on GLUE benchmarks using RoBERTa-base: QQP, RTE, SST-2, and WNLI.

Method	QQP		RTE		SST-2		WNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.9069	–	0.7220	–	0.9323	–	0.4366	–
1-Nearest Neighbor	–	–	0.4693	0.4657	0.5092	0.5034	0.1972	0.1831
NM1/NM2	0.9069	**0.8961**	0.7220	0.7112	0.9323	0.9255	0.4366	0.4366
CONFIDE-A (ours)	**0.9075**	0.8955	**0.7365**	0.7256	**0.9346**	0.9289	–	–
CONFIDE-C (ours)	0.9070	**0.8961**	0.7329	**0.7292**	0.9323	**0.9323**	–	–

focusing on three representative datasets: CoLA, MNLI, and BoolQ. All models are evaluated using attention representations, with all other metrics varied.

5.4.1 Comparative Results

CONFIDE's classwise behavior reveals both strengths and limitations in calibration quality across tasks. CoLA presents a failure case. As shown in Fig. 5, the aggregate coverage curve slightly underperforms, falling below the $1 - \varepsilon$ diagonal, with correct efficiency peaking briefly before declining. Classwise analysis reveals the issue: while the "acceptable" class is overcovered, the "unacceptable" class is severely undercovered, with coverage dropping to zero for all $\varepsilon > 0.25$.

In contrast, MNLI demonstrates strong calibration. Both the overall and correct efficiency curves align well with the ideal diagonal (Fig. 6), and all three classes achieve classwise validity. This likely reflects MNLI's clear class boundaries and low label noise, enabling even small models to learn well-separated representations and produce reliable prediction sets.

BoolQ initially appears valid at the aggregate level, but classwise curves tell a different story (Fig. 7). The model heavily overpredicts the "true" class, assigning it to 906 of 1,237 "false" examples. As a result, the "false" class suffers from poor accuracy (26.8%) and low coverage, while "true" examples enjoy inflated metrics. Confidence and credibility are also skewed, averaging 0.894 vs. 0.865 (confidence) and 0.645 vs. 0.576 (credibility) for "true" and "false," respectively. This reflects a representational collapse: inputs from both classes are embedded too similarly, leading to overconfident, incorrect predictions.

These results underscore a critical limitation of marginal coverage: improvements in accuracy and correct efficiency do not guarantee per-class calibration. When one class dominates model predictions, marginal metrics may appear strong even as minority classes are systematically misrepresented.

5.4.2 Classwise Conformal Prediction Improves, but Does Not Solve, Failures of Conformal Validity in BERT-tiny

Classwise adjustment can be valuable when the base model exhibits strong class-specific bias, as it attempts to re-balance coverage in a targeted manner.

Table 9: Performance on SuperGLUE benchmarks using RoBERTa-base: BoolQ, CB, and MultiRC.

Method	BoolQ		CB		MultiRC	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.8092	–	0.6786	–	0.7473	–
1-Nearest Neighbor	–	–	0.4464	0.4286	–	–
NM1/NM2	0.8092	0.7654	0.6786	0.6607	**0.7473**	**0.7471**
CONFIDE-A (ours)	**0.8095**	0.5483	**0.7500**	**0.7500**	0.4299	0.4292
CONFIDE-C (ours)	0.8089	**0.8089**	**0.7500**	**0.7500**	0.4299	0.4292

Table 10: Performance on GLUE and SuperGLUE benchmarks using RoBERTa-large: CoLA, MRPC, RTE, and WNLI.

Method	CoLA		MRPC		RTE		WNLI	
	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff	Test Acc	Top Corr Eff
Original NN	0.8418	–	0.8578	–	0.8087	–	**0.5634**	–
1-Nearest Neighbor	0.6069	0.6021	0.5564	0.5490	0.4693	0.4657	0.1972	0.1831
NM1/NM2	0.8418	0.8293	0.8578	0.8578	0.8087	**0.8087**	**0.5634**	**0.5493**
CONFIDE-A (ours)	**0.8466**	**0.8437**	**0.8710**	0.8578	**0.8123**	0.7978	**0.5634**	**0.5493**
CONFIDE-C (ours)	**0.8466**	**0.8437**	0.8652	**0.8652**	0.8051	0.8051	0.5493	**0.5493**

In BoolQ, applying classwise calibration substantially improves coverage for the minority "false" class, and both classes begin to track more closely along the ideal coverage line (Fig. 8). However, some under-coverage remains, especially under the ideal line for low ε, suggesting residual bias. This suggests that classwise calibration is a useful but partial solution. We also share CoLA graphs (Fig. 9), which show similar improvements but lingering asymmetry. Fully resolving these failures may require further adjustments, such as class reweighting, further temperature scaling, or more expressive representations.

BERT-small: Larger model, same pitfalls. Interestingly, increasing model capacity does not resolve the violation. Both coverage and classwise coverage curves are similar, exhibiting abrupt drops in performance at low ϵ, as shown, for example, in Fig. 10. The persistent coverage gap reflects that BERT-small, though better optimized, still inherits inductive biases from the dataset and training protocol. Results graphically are similar for BERT-small model/dataset combinations. Hence, we next move on to larger models and share additional results in the Appendix.

5.5 Calibration Robustness in RoBERTa Models

We similarly analyze CONFIDE's performance under both non-classwise and classwise calibration across CoLA, BoolQ, and CB for the larger models, with additional results shared in Appendix D. These experiments reveal that larger models can produce more stable and exchangeable embeddings but that calibration quality still varies by dataset and class distribution.

RoBERTa-base: Poor baseline calibration, strong classwise gains. On CoLA, RoBERTa-base again has poor calibration without classwise correction. As shown in Fig. 11, the aggregate coverage curve is below the $1 - \varepsilon$ diagonal, and the correct efficiency is not tight, indicating that most prediction sets are large and contain incorrect classes. Here, however, enabling classwise calibration on CoLA yields strong improvements, even above the diagonal curve. Fig. 12 shows these trends, with classwise coverage providing near-uniform performance between classes. This reveals how with RoBERTa-base, when the embeddings are already well-structured, classwise correction can become a *fine-tuning tool*. We also note the continued poor performance of correct efficiency, indicating that CoLA remains a difficult dataset for even the larger models to correctly predict.

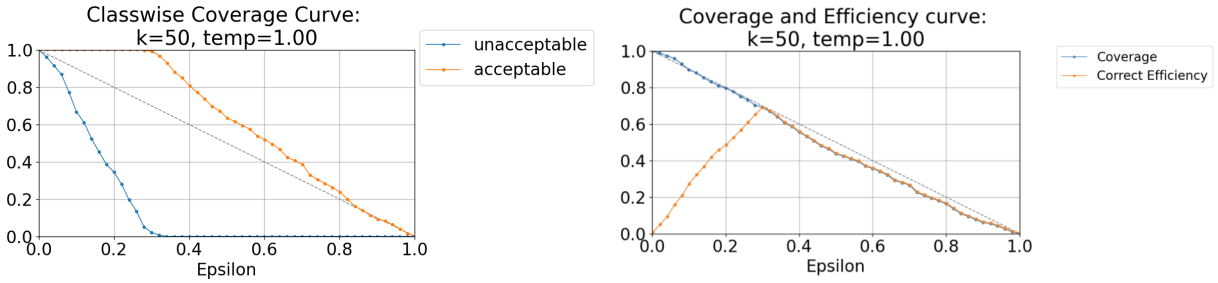


Figure 5: Classwise and aggregate coverage curves for BERT-tiny on CoLA. The false class severely undercovers, violating conformal validity.

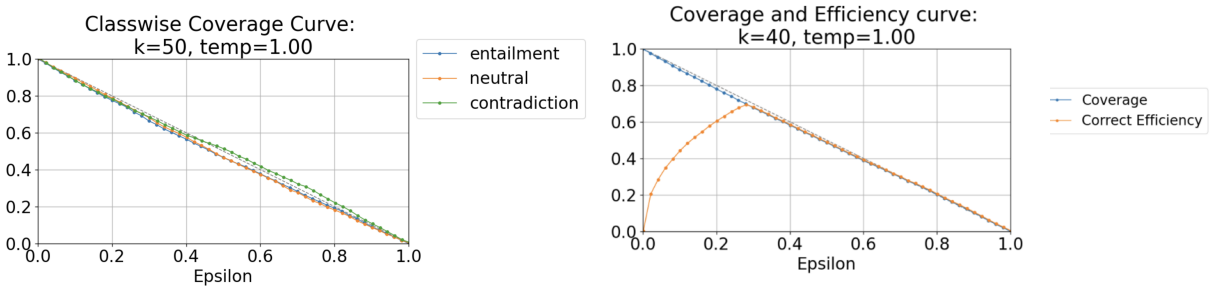


Figure 6: Classwise and aggregate coverage curves for BERT-tiny on MNLI. All classes are well calibrated and closely follow the diagonal, indicating robust conformal validity.

On BoolQ (Fig. 13), RoBERTa-base demonstrates improved calibration relative to smaller models. The aggregate coverage curve closely tracks the $1 - \varepsilon$ line, and the correct efficiency curve is stable and high until $\varepsilon \approx 0.6$, after which performance drops sharply. Further results for comparative classwise metrics are presented in Appendix D.

This behavior likely reflects instability in the nonconformity score distribution for ambiguous examples. At higher ε thresholds, CONFIDE is expected to tolerate more errors by assigning lower nonconformity scores to uncertain examples. However, in tasks like BoolQ, where many questions may not be easily distinguishable as "true" or "false" without fine-grained reasoning, the model may have difficulty ranking examples consistently. This leads to abrupt shifts in which examples are included in the prediction set as ε increases, resulting in jagged drops in coverage and efficiency. In addition, these cliffs suggest a breakdown in the pseudo-exchangeability assumption: for inputs near the decision boundary, nearest neighbors may be semantically dissimilar, making conformal thresholds unreliable. Despite these drawbacks, performance remains substantially better than with smaller models, with more symmetric classwise behavior and better.

RoBERTa-large: Improvements with limits. RoBERTa-large similarly can show strong calibration on structured datasets, but struggles under label ambiguity. On the CB dataset (Fig. 14), entailment and contradiction maintain reasonable classwise coverage curves, but the "neutral" class fails, remaining below 0.5 across all ε. This is likely due to two factors: the semantic vagueness of the neutral class and its very low representation in the dataset. Even with high-capacity models, these challenges cannot be resolved purely by scale. The failure of RoBERTa-large on CB highlights the fact that large model size alone is insufficient for tackling calibrated uncertainty under semantic ambiguity and class sparsity. More examples of RoBERTa-large performance are presented in Appendix D.

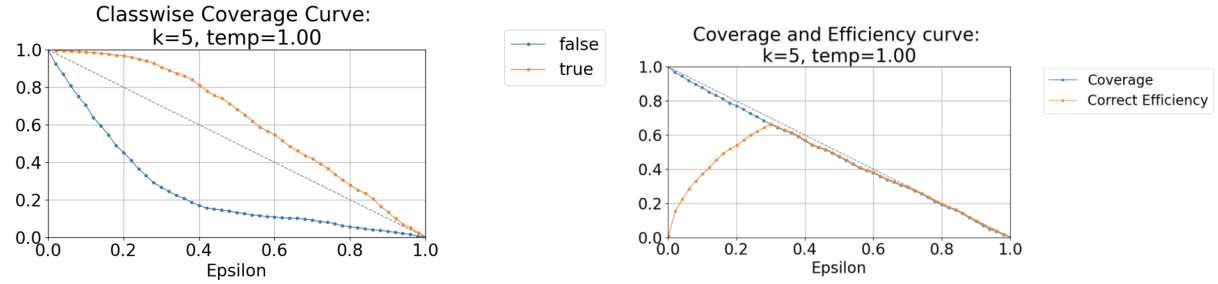


Figure 7: Classwise and aggregate coverage curves for BERT-tiny on BoolQ (non-classwise). The false class shows massive undercoverage due to biased calibration, despite almost diagonal aggregated statistics.

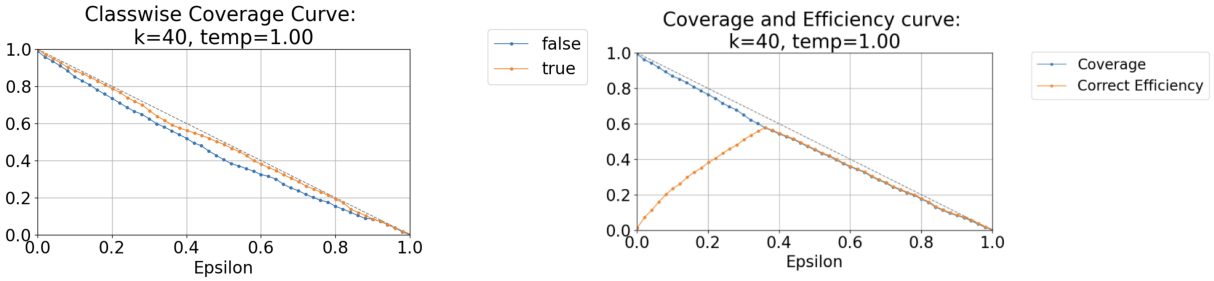


Figure 8: Coverage curves on BoolQ using classwise calibration. The classwise method reduces undercoverage but does not fully resolve it.

5.6 Prior Method Comparisons

We also present marginal coverage graphs on the CoLA and BoolQ datasets using prior methods. Figs. 15a–16b show their classwise coverage behavior across model and method combinations.

On CoLA, NM2 similarly exhibits significant calibration issues across both BERT-tiny and RoBERTa-base backbones. As can be seen from Fig. 15, the "unacceptable" class suffers from consistent undercoverage throughout the ε range, with coverage falling far below the $1 - \varepsilon$ diagonal. This trend persists in Fig. 15b, where even the stronger RoBERTa-base model fails to improve the validity of the "unacceptable" class. These results mirror the classwise failures of CONFIDE on CoLA and indicate that the inherent ambiguity of the task and label noise challenge all distance-based methods.

On BoolQ, both NM2 and VanillaNN fail to provide reliable classwise coverage for BERT-tiny. In Fig. 16a, NM2's performance is poor for the "false" class, with coverage dropping steadily below the diagonal. The "true" class fares slightly better but still fails to meet the theoretical guarantee. VanillaNN, shown in Fig. 16b, demonstrates similarly imbalanced behavior — while both classes begin near full coverage at low ε,

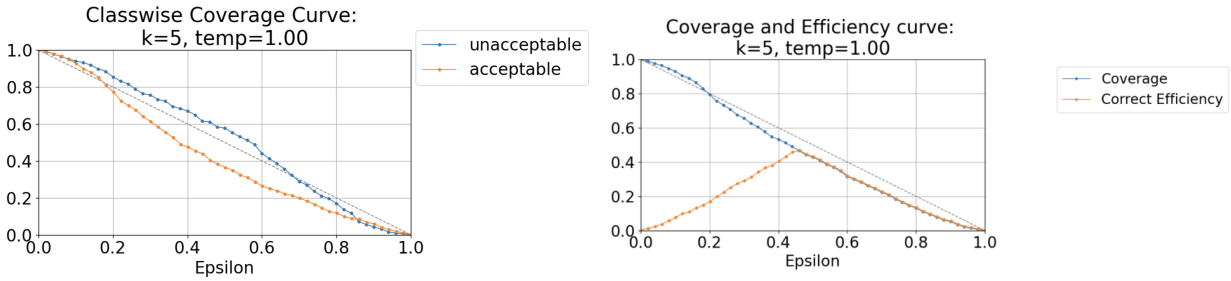


Figure 9: Coverage curves on CoLA using classwise calibration.

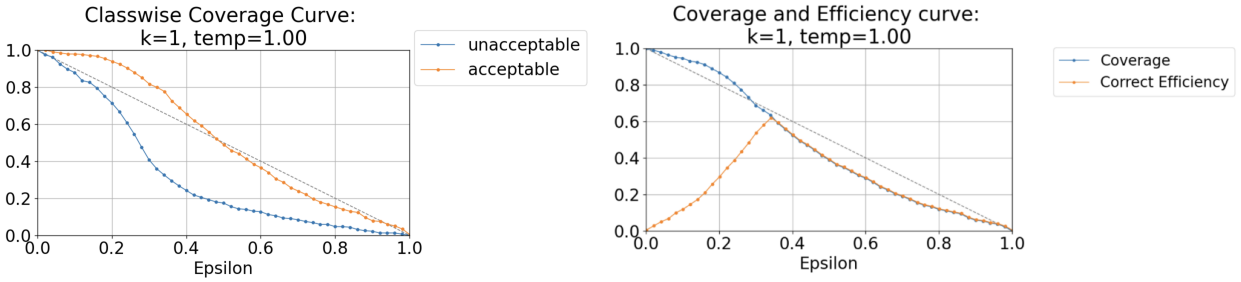


Figure 10: Coverage curves on CoLA using BERT-small models. Undercoverage and classwise calibration perform as poorly as BERT-tiny models.

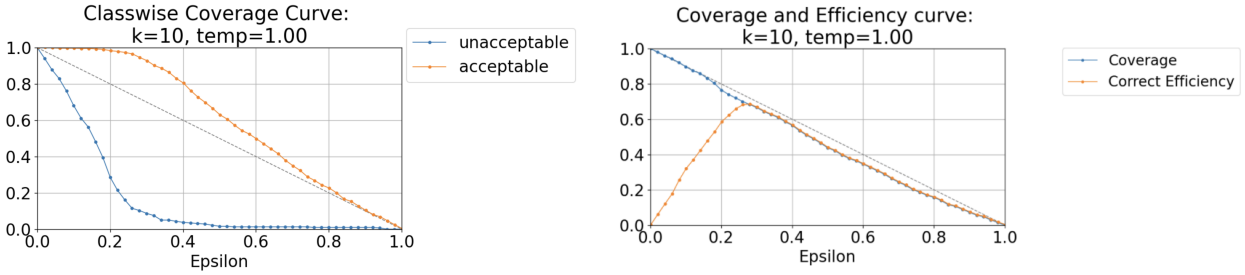


Figure 11: Coverage curves on CoLA using RoBERTa models. Undercoverage and classwise calibration perform marginally better than smaller models but still poorly.

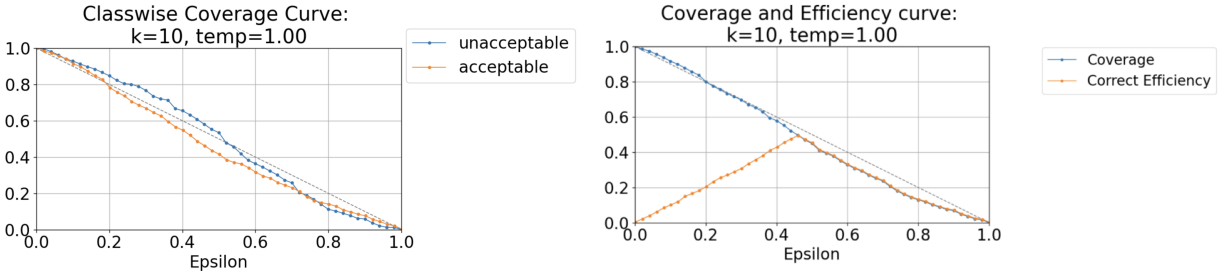


Figure 12: Improvement performances for CoLA with classwise calibration.

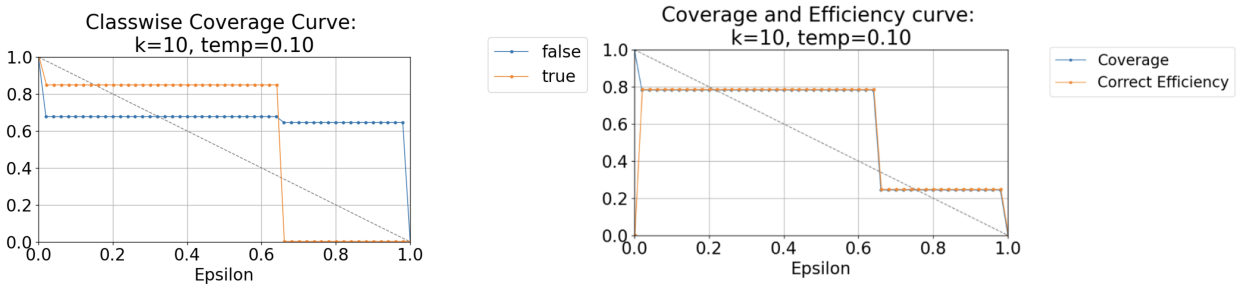


Figure 13: Improvement performances for BoolQ compared to smaller models

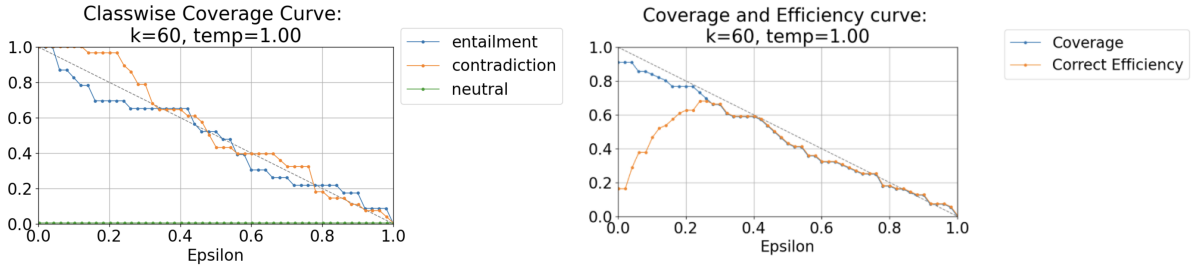


Figure 14: RoBERTa-large on CB (classwise): `neutral` class coverage fails entirely for neutral labels despite overall decent accuracy.

performance rapidly degrades. The "false" class undercovers more severely, likely due to the method's lack of calibrated similarity scaling.

Together, these results highlight how both prior conformal methods have similar classwise calibration failures on tasks with label imbalance or semantic ambiguity. Across methods, the most severe coverage violations occur for minority or hard-to-learn classes, indicating that the primary bottleneck may not lie in the conformal framework but in the quality and richness of underlying embeddings.

5.7 Comparing Attention vs. Flattened CONFIDE Variants

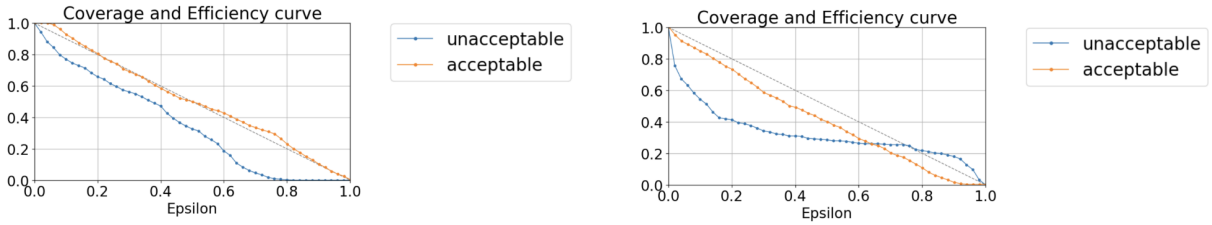
In prior sections, all presented results use the *attention-based* variant of CONFIDE, in which nonconformity scores are computed solely using the `[CLS]` token representation. Next, we compare it to the *flattened* variant, which instead uses the full hidden state matrix from the selected transformer layer, flattened across the sequence dimension to create a dense, high-dimensional representation. Results are limited due to the high computational power required to flatten a layer.

We share brief results comparing top-1 accuracy and correct efficiency across two representative tasks: MNLI using BERT-tiny and SST-2 using BERT-small in Table 11.

Almost across the board, the flattened variant of CONFIDE slightly outperforms the attention-based version. In MNLI, while accuracies are roughly comparable, the flattened variant achieves noticeably more stable and occasionally higher correct efficiency.

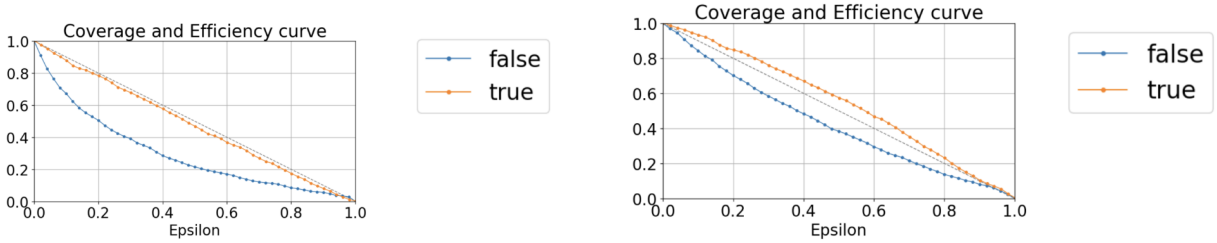
Why does flattened outperform attention. These gains are expected for several reasons:

- **Higher information content.** Flattened embeddings incorporate features from all tokens in the input sequence, capturing nuanced syntactic and semantic relationships that are lost when compressing to a single `[CLS]` vector.
- **Better class separability.** The larger representation space enables better discrimination between conforming and non-conforming examples, leading to sharper nonconformity scores.



(a) BERT-tiny with NM2 (b) RoBERTa-base with NM2

Figure 15: Classwise coverage curves for CoLA using the NM2 prior method. Both BERT-tiny and RoBERTa-base show persistent undercoverage, particularly for the "unacceptable" class.



(a) NM2 Prior Method (b) Vanilla NN Prior Method

Figure 16: Classwise coverage curves for BoolQ using BERT-tiny with two prior methods. Both NM2 and Vanilla NN fail to maintain valid coverage, especially for the "false" class.

- **Reduced overfitting to [CLS].** Attention-based approaches rely on the `[CLS]` token, which is often tuned task-specifically. Flattened embeddings, by contrast, preserve broader contextual information and generalize more.

Overall, when memory constraints permit, flattened CONFIDE emerges as a preferable variant, offering improvements in both accuracy and reliability of the prediction set construction.

6 Discussions and Limitation

Despite its strengths, CONFIDE suffers from persistent calibration failures in real-world datasets and can be computationally burdensome. As can be seen from our comprehensive results, CONFIDE often suffers from significant *coverage violations*, often falling below the expected $(1-\epsilon)$ threshold, especially for minority classes, such as `neutral` in MNLI or the `false` class in BoolQ. However, as previously noted, prior baselines, such as NM2 or VANILLANN, demonstrate even more erratic or collapsed coverage curves, thus validating CONFIDE's superiority. These violations highlight a key limitation of CP in practice: the assumption of exchangeability is often not met, particularly in datasets with inherent class imbalance or input distribution drift. This limitation suggests that in safety-critical domains, conformal methods must be combined with explicit checks on calibration behavior per class or region of the input space. Otherwise, falsely assuming valid coverage can lead to underrepresented groups being frequently misclassified.

From a computational standpoint, a key limitation of CONFIDE is that its best-performing configurations often occur not at the final softmax layer but within early or intermediate layers of the transformer. For instance, in RoBERTa-base on CoLA (Fig. 11), the optimal performance arises at $k = 60$, $t = 1.0$, and `layer 190`, which is within the first half of the network. While earlier layers yield richer contextual embeddings for semantic similarity, this benefit comes at a cost. As shown in Appendix Table 13, shifting from layer 38 to 41 (i.e., deeper into the model) reduces calibration and test time by over 5 and 0.8 seconds, respectively, averaged

Table 11: Comparison of CONFIDE-A and CONFIDE-C using Attention and Flattened representations for MNLI (BERT-tiny) and SST-2 (BERT-small). Bolded values highlight the best result for each task and metric.

Task / Variant	**Attention**		**Flattened**	
	Acc.	Corr. Eff.	Acc.	Corr. Eff.
MNLI – BERT-TINY				
CONFIDE-A	0.6982	**0.6907**	**0.6993**	0.6497
CONFIDE-C	0.6975	0.6932	**0.6980**	**0.6971**
SST-2 – BERT-SMALL				
CONFIDE-A	0.8933	0.8647	**0.8968**	**0.8853**
CONFIDE-C	0.8899	0.8911	**0.8933**	**0.8933**

across datasets for BERT-tiny. In contrast, using earlier layers (e.g., layers 9 to 16) can increase computation time, particularly for the calibration phase. This is due to the larger hidden dimensions and more complex internal representations at earlier stages, which inflate the cost of k-nearest neighbor lookups. In addition, CONFIDE's higher test-time cost makes it less suitable for real-time deployment unless additional efficiency strategies are incorporated.

This trade-off between interpretability and efficiency is magnified in the `flattened` variant of CONFIDE, which extracts token-level embeddings across the full sequence length. As demonstrated in Section 5.7, flattened representations can outperform attention-based ones. They, however, amplify memory usage and make pre-processing expensive due to the sequence-level granularity. This is due to the need to store and compare high-dimensional token-level vectors across long input sequences. Moreover, they frequently run into GPU memory limits on RoBERTa models and long-context tasks like BoolQ.

These limitations could be addressed by (i) designing memory-efficient approximations for flattened embeddings, (ii) exploring hybrid-layer representations that balance cost and informativeness, and (iii) benchmarking early-layer embeddings with realistic latency, memory, and throughput constraints to identify tradeoff-optimal layers.

7 Conclusions and Future Work

This paper introduced CONFIDE, a CP framework designed to bring principled uncertainty quantification and interpretable prediction sets to transformer-based language models. Built upon the CONFINE algorithm, CONFIDE adapts these techniques to the unique architecture and representational structure of transformers, particularly BERT and RoBERTa variants. CONFIDE enables users to generate prediction sets that are transparent and explainable through nearest-neighbor retrievals.

Across a suite of GLUE and SuperGLUE tasks, CONFIDE outperforms traditional uncertainty baselines in both accuracy and correct efficiency. On most models, CONFIDE improves prediction set quality without sacrificing top-1 accuracy. Our empirical results show that CONFIDE variants with [CLS] embeddings and class-conditional nonconformity measures outperform NM2 and VanillaNN in both aggregate and classwise coverage. These gains are especially pronounced on tasks like BoolQ and RTE. Every method suffers from undercoverage for underrepresented classes.

CONFIDE also exposes failure modes in low-capacity models by demonstrating poor coverage for multiple domains. Class-conditional analysis reveals that both BERT-tiny and BERT-small frequently overfit dominant classes, producing skewed prediction sets. CONFIDE, however, attempts to make these imbalances legible through classwise coverage curves and credibility metrics, surfacing issues that remain hidden in aggregate statistics.

A key insight from our study is that the most effective CONFIDE configurations frequently rely on intermediate or early layers of the model. For instance, optimal configurations for tasks like CoLA and MNLI

often emerge from layers 9 and 16 in BERT-tiny, well before the final classification/softmax layer. These internal layers contain more generalized linguistic structure, which enhances nearest-neighbor comparisons. However, this also introduces computational overhead: compared to softmax-based predictors, CONFIDE requires significant time for embedding extraction and k-NN distance computation, particularly in high-dimensional spaces, such as RoBERTa, which had dimensions upwards of 768. Table 13 in Appendix C quantifies this tradeoff, showing increased calibration and test time when using richer embeddings.

The broader value of conformal prediction lies in its potential to deliver trustworthy and introspective AI. In high-stakes domains like healthcare, conformal methods can output reliable differential diagnoses with guaranteed coverage, enabling practitioners to weigh multiple outcomes without over-reliance on a single prediction. One example is a model that outputs "pneumonia, bronchitis" with 95% confidence based on a series of symptoms, better than simply offering just "pneumonia" at 98% probability. CONFIDE enhances trust by providing these instance-level explanations: potentially the most similar past patients or cases in the calibration data.

Robotics offers another compelling application area. In real-world systems, such as autonomous vehicles, safety depends not only on the accuracy of decisions but also on the model's ability to know when *not* to act. CONFIDE enables safety-aware behavior by using prediction set size as a proxy for certainty: if the prediction set is large or ambiguous, the robot may default to a safer fallback action. This is important for applications like object classification in unstructured environments or decision-making under partial observability, where taking the wrong action can have irreversible consequences.

The promise of CP in language models extends even further. Future work can extend CONFIDE's ideas to generative settings (e.g., conformal decoding), multimodal tasks (e.g., image-text retrieval), and structured outputs (e.g., entity spans or logical forms), thereby expanding the reach of conformal reasoning into the core of next-generation AI systems.

In sum, CONFIDE takes a significant step toward interpretable, robust, and theoretically grounded NLP systems. It equips users with prediction sets they can trust and explanations they can audit without needing to modify the underlying model architecture. While computational costs remain a barrier to real-time deployment, especially at earlier layers, this tradeoff is often justified in applications where transparency and reliability are paramount. By bridging deep language models and conformal inference, CONFIDE contributes a step toward introspective, explainable, and reliable AI.

References

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Mitchell Bosley, Musashi Jacobs-Harukawa, Hauke Licht, and Alexander Hoyle. Do we still need BERT in the age of GPT? Comparing the benefits of domain-adaptation and in-context-learning approaches to using LLMs for Political Science Research. Manuscript, April 14, 2023. URL `https://mbosley.github.io/papers/bosley_harukawa_licht_hoyle_mpsa2023.pdf`.

I. C. Cardenas, T. Aven, and R. Flage. Addressing challenges in uncertainty quantification: The case of geohazard assessments. *Geoscientific Model Development*, 16:1601–1615, 2023.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. *arXiv preprint arXiv:2012.09838*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

Neil Dey, Jing Ding, Jack Ferrell, Carolina Kapper, Maxwell Lovig, Emiliano Planchon, and Jonathan P Williams. Conformal prediction for text infilling and part-of-speech prediction. *arXiv preprint arXiv:2111.02592*, 2021.

Shang Gao, Mohammed Alawad, M. Todd Young, John Gounley, Noah Schaefferkoetter, Hong Jun Yoon, Xiao-Cheng Wu, Eric B. Durbin, Jennifer Doherty, Antoinette Stroup, Linda Coyle, and Georgia Tourassi. Limitations of transformers on clinical text classification. *IEEE Journal of Biomedical and Health Informatics*, 25(9):3596–3607, 2021.

Enrico Giovannotti, Massimiliano Fontana, and Marco Vichi. Validity guarantees for approximate conformal prediction. In Osbert Bastani, Gautam Kumar, and Michael Carl Tschantz (eds.), *Proceedings of the First Conference on Certifiable Artificial Intelligence (CertAI)*, volume 152 of *Proceedings of Machine Learning Research*, pp. 1–25, 2021.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Linhui Huang, Sayeri Lala, and Niraj K. Jha. CONFINE: Conformal prediction for interpretable neural networks. *arXiv preprint arXiv:2406.00539v2*, 2025.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

I. T. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150202, Apr. 2016.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

Benjamin Leblanc and Pascal Germain. On the relationship between interpretability and explainability in machine learning. *arXiv preprint arXiv:2311.11491*, 2024.

Junghwan Lee, Chen Xu, and Yao Xie. Transformer conformal prediction for time series. *arXiv preprint arXiv:2406.05332*, 2024.

Zachary C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2017.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *arXiv preprint arXiv:2006.10108*, 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Samir Messoudi, Sylvie Rousseau, and Sébastien Destercke. Deep conformal prediction for robust models. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2020)*, volume 1237 of *Communications in Computer and Information Science*, pp. 528–540. Springer, 2020.

Panagiotis Moutafis, Mengjun Leng, and Ioannis A. Kakadiaris. An overview and empirical comparison of distance metric learning methods. *IEEE Transactions on Cybernetics*, 47(3):612–625, 2017.

Utku Ozbulak, Wesley De Neve, and Arnout Van Messem. How the softmax output is misleading for evaluating the strength of adversarial examples. *arXiv preprint arXiv:1811.08577*, 2018.

Harris Papadopoulos, Volodya Vovk, and Alex Gammerman. Conformal prediction with neural networks. In *Proceedings of the Nineteenth IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pp. 388–395, 2007.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Tim Pearce, Alexandra Brintrup, and Jun Zhu. Understanding softmax confidence and uncertainty. *arXiv preprint arXiv:2106.04972*, 2021.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2023.

Arafet Sbei, Khaoula ElBedoui, and Walid Barhoumi. Assessing the efficiency of transformer models with varying sizes for text classification: A study of rule-based annotation with distilBERT and other transformers. *Vietnam Journal of Computer Science*, 2024.

Daniel Sikar, Artur d'Avila Garcez, and Tillman Weyde. Explorations of the softmax space: Knowing when the neural network doesn't know... *arXiv preprint arXiv:2502.00456*, 2025.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

S. Talebi, E. Tong, A. Li, et al. Exploring the performance and explainability of fine-tuned BERT models for neuroradiology protocol assignment. *BMC Medical Informatics and Decision Making*, 24(1):40, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Jesse Vig. A multiscale visualization of attention in the transformer model. In Marta R. Costa-jussà and Enrique Alfonseca (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, Florence, Italy, July 2019. Association for Computational Linguistics.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2019.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2020.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

Appendix

A Model and Dataset Details

Benchmark selection. We evaluate CONFIDE on a suite of sentence- and passage-level classification tasks drawn from the GLUE and SuperGLUE benchmarks. These datasets are widely used in NLP research and present varied challenges in terms of linguistic complexity, dataset size, number of classes, and label ambiguity. All tasks selected are classification-based, ensuring compatibility with CP methods that require discrete label spaces. By testing CONFIDE on this diverse set, we validate its robustness, interpretability, and generalizability across real-world NLP scenarios.

GLUE datasets (Wang et al., 2019)

- **SST-2 (Stanford Sentiment Treebank v2):** Binary sentiment classification of movie reviews. Contains approximately 67,000 training examples. Well-curated and widely used for benchmarking sentence-level sentiment classification.
- **MNLI (Multi-Genre Natural Language Inference):** Three-class classification (entailment, neutral, contradiction) with over 392,000 examples. Features premise–hypothesis pairs from multiple domains (e.g., fiction, government), testing general NLI capabilities.
- **CoLA (Corpus of Linguistic Acceptability):** Binary acceptability judgment task using expert-labeled grammatical sentences. Relatively small with roughly 8,500 examples, challenging due to linguistic nuance.
- **MRPC (Microsoft Research Paraphrase Corpus):** Binary classification of whether sentence pairs are semantically equivalent. Contains around 3,700 examples and introduces noise due to label ambiguity.
- **QQP (Quora Question Pairs):** Large-scale binary paraphrase classification with over 400,000 sentence pairs. Includes noisy user-generated data, simulating web-scale text inference tasks.
- **QNLI (Question Natural Language Inference):** Binary classification derived from question–answering. Reformulated from SQuAD as sentence-pair entailment. Mid-sized with approximately 105,000 examples.
- **RTE (Recognizing Textual Entailment):** Binary entailment detection with only 2,500 examples. Known for being difficult due to label imbalance and low-resource setting.
- **WNLI (Winograd NLI):** Binary coreference reasoning task derived from the Winograd Schema Challenge. Extremely small (635 samples), difficult to model due to subtle pronoun resolution required.

SuperGLUE Datasets (Wang et al., 2020)

- **BoolQ (Boolean Questions):** Binary QA task requiring yes/no answers to naturally occurring queries paired with Wikipedia passages. Over 9,000 training examples, but highly noisy and linguistically ambiguous.
- **CB (CommitmentBank):** Three-class NLI task with only $\sim$250 training examples. Requires fine-grained reasoning over implicatures in short discourse contexts. Serves as a stress test for uncertainty calibration in low-data settings.
- **MultiRC (Multi-Sentence Reading Comprehension):** Framed as a binary classification task on question–answer–context triples. Multi-label format with complex, often contradictory annotations. Approximately 9,000 QA pairs from 300 paragraphs.

B Hyperparameter Testing Results

B.1 Hyperparameter Search

We performed grid search over the CONFIDE hyperparameters: *layer*, *k*, *distance metric*, *PCA*, and *temperature*. For each (model, dataset) pair, we selected the configuration that maximized either top-1 accuracy (CONFIDE-A) or correct efficiency (CONFIDE-C). Full results are visualized as heatmaps, but first, we share top-accuracy and top-correct-efficiency configurations across datasets and methods. Table 12 summarizes the key settings.

Table 12: Best CONFIDE hyperparameters across datasets and models.

Dataset	Model	Method	Hyperparameters
glue_cola	BERT-small	CONFIDE-A	$l = 81,\ k = 10,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 81,\ k = 10,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 16,\ k = 50,$ flattened, mahalanobis, PCA : Yes
		CONFIDE-C	$l = 16,\ k = 50,$ flattened, mahalanobis, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 190,\ k = 20,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 1.0,\ k = 1,$ flattened, cosine, PCA : Yes
	RoBERTa-large	CONFIDE-A	$l = 340,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 340,\ k = 1,$ flattened, cosine, PCA : No
glue_mnli	BERT-small	CONFIDE-A	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l =$ softmax, $T = 1.0,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 10.0,\ k = 60,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 225,\ k = 10,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 225,\ k = 1,$ flattened, cosine, PCA : No
glue_mrpc	BERT-small	CONFIDE-A	$l = 52,\ k = 20,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 70,\ k = 20,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 34,\ k = 50,$ flattened, mahalanobis, PCA : No
		CONFIDE-C	$l = 16,\ k = 5,$ flattened, mahalanobis, PCA : No
	RoBERTa-base	CONFIDE-A	$l = 189,\ k = 50,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 191,\ k = 5,$ flattened, mahalanobis, PCA : Yes
	RoBERTa-large	CONFIDE-A	$l =$ softmax, $T = 20.0,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 40.0,\ k = 1,$ flattened, cosine, PCA : No
glue_qnli	BERT-small	CONFIDE-A	$l = 85,\ k = 40,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 82,\ k = 1,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 27,\ k = 40,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 27,\ k = 1,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 225,\ k = 5,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 5,$ flattened, cosine, PCA : No
glue_qqp	BERT-small	CONFIDE-A	$l = 81,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 40,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 27,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 27,\ k = 1,$ flattened, cosine, PCA : No
	RoBERTa-base	CONFIDE-A	$l = 225,\ k = 20,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 225,\ k = 1,$ flattened, cosine, PCA : No
glue_rte	BERT-small	CONFIDE-A	$l = 38,\ k = 60,$ flattened, mahalanobis, PCA : Yes
		CONFIDE-C	$l = 38,\ k = 60,$ flattened, mahalanobis, PCA : Yes
	BERT-tiny	CONFIDE-A	$l = 27,\ k = 50,$ flattened, cosine, PCA : Yes
		CONFIDE-C	$l = 38,\ k = 50,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 190,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 189,\ k = 40,$ flattened, cosine, PCA : Yes
	RoBERTa-large	CONFIDE-A	$l = 340,\ k = 20,$ flattened, mahalanobis, PCA : Yes
		CONFIDE-C	$l = 340,\ k = 5,$ flattened, mahalanobis, PCA : Yes
glue_sst2	BERT-small	CONFIDE-A	$l = 82,\ k = 1,$ flattened, cosine, PCA : Yes
		CONFIDE-C	$l =$ softmax, $T = 10.0,\ k = 1,$ flattened, cosine, PCA : Yes
	BERT-tiny	CONFIDE-A	$l = 27,\ k = 5,$ flattened, mahalanobis, PCA : No
		CONFIDE-C	$l = 34,\ k = 50,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l =$ softmax, $T = 10.0,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : Yes
glue_wnli	BERT-small	CONFIDE-A	$l = 16,\ k = 60,$ flattened, cosine, PCA : Yes
		CONFIDE-C	$l = 16,\ k = 50,$ flattened, cosine, PCA : Yes
	BERT-tiny	CONFIDE-A	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : No
	RoBERTa-large	CONFIDE-A	$l = 45,\ k = 1,$ flattened, mahalanobis, PCA : Yes
		CONFIDE-C	$l = 45,\ k = 1,$ flattened, cosine, PCA : No
superglue_boolq	BERT-small	CONFIDE-A	$l =$ softmax, $T = 10.0,\ k = 1,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 5,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 16,\ k = 5,$ flattened, mahalanobis, PCA : No
		CONFIDE-C	$l = 16,\ k = 5,$ flattened, mahalanobis, PCA : No
	RoBERTa-base	CONFIDE-A	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : Yes
		CONFIDE-C	$l =$ softmax, $T = 0.01,\ k = 1,$ flattened, cosine, PCA : Yes
superglue_cb	BERT-small	CONFIDE-A	$l = 9,\ k = 5,$ flattened, cosine, PCA : Yes

Continued on next page

Dataset	Model	Method	Hyperparameters
		CONFIDE-C	$l = 27,\ k = 1,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 16,\ k = 5,$ flattened, cosine, PCA : Yes
		CONFIDE-C	$l = 16,\ k = 10,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 225,\ k = 1,$ flattened, mahalanobis, PCA : No
		CONFIDE-C	$l = 225,\ k = 1,$ flattened, mahalanobis, PCA : No
	RoBERTa-large	CONFIDE-A	$l = 106,\ k = 60,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 106,\ k = 60,$ flattened, cosine, PCA : No
superglue_multirc	BERT-small	CONFIDE-A	$l = 85,\ k = 50,$ flattened, cosine, PCA : No
		CONFIDE-C	$l =$ softmax, $T = 0.1,\ k = 40,$ flattened, cosine, PCA : No
	BERT-tiny	CONFIDE-A	$l = 27,\ k = 5,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 38,\ k = 5,$ flattened, cosine, PCA : Yes
	RoBERTa-base	CONFIDE-A	$l = 45,\ k = 60,$ flattened, cosine, PCA : No
		CONFIDE-C	$l = 45,\ k = 60,$ flattened, cosine, PCA : No

B.2 Heatmaps

We include in Figs. 17-22 a subset of experimental plots organized by dataset and model. For each grid parameter of k and l, we present the best accuracy and best top correct efficiency, found at *any* combination of the other hyperparameters, including flattened vs. attention, cosine vs. mahalanobis, or PCA true vs. false. Similarly, for softmax layers, we do the same for k vs T.

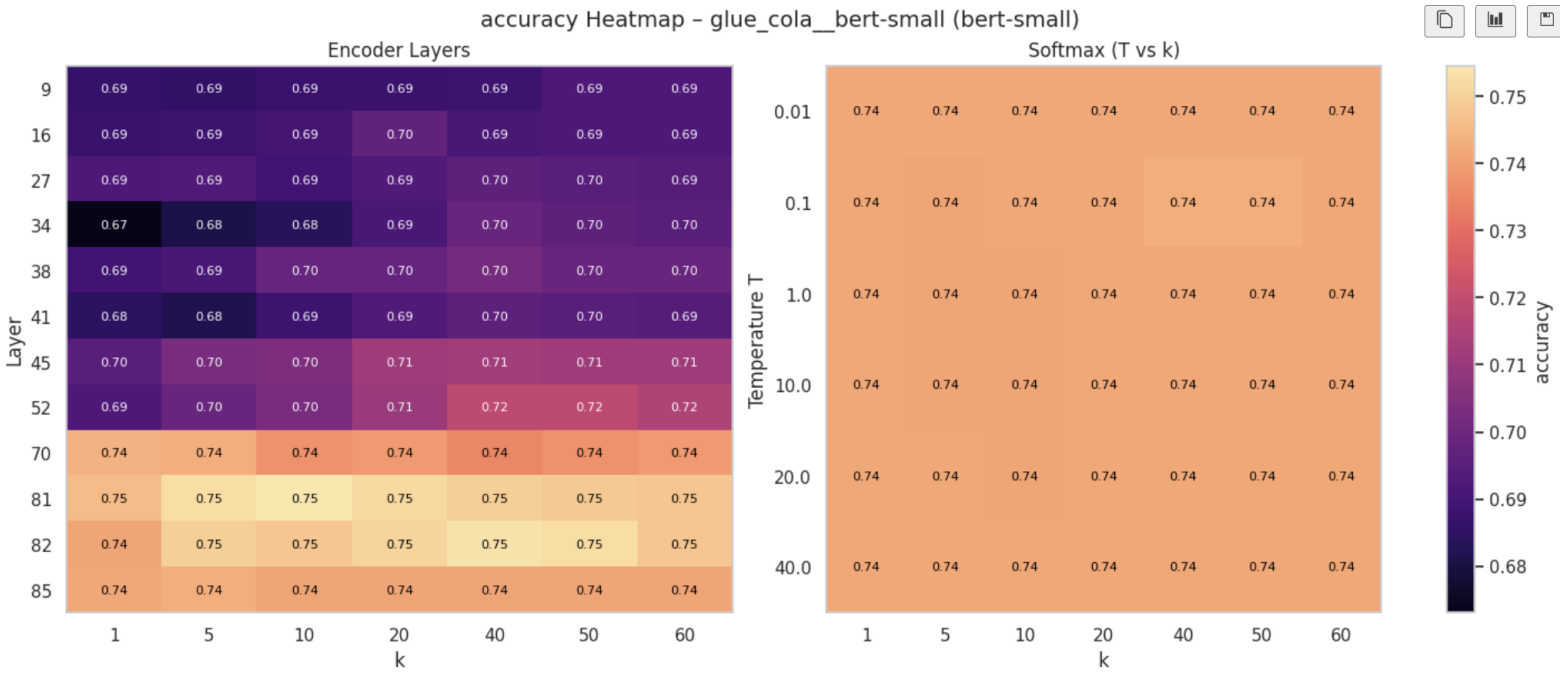


Figure 17: BERT-small hyperparameter testing on the CoLA dataset - accuracy

C Timing and Computational Cost

C.1 Timing Implications of Layer Choice

Table 13 analyzes the change in CPU time across CONFIDE's core stages—`train`, `calib`, and `test`—when shifting embedding extraction from an earlier transformer layer to a later one. A key hypothesis is that *later layers should exhibit lower computational cost*, as their embeddings tend to be more distilled, lower-variance, and less semantically rich. These properties could reduce both the storage overhead and the cost of nearest-neighbor distance calculations.

Overall, the data support this trend. The transition from layer 16.0 to 27.0 yields clear speedups across all stages, with calibration time dropping by over 0.5 seconds. A further shift to layer 34.0 continues this

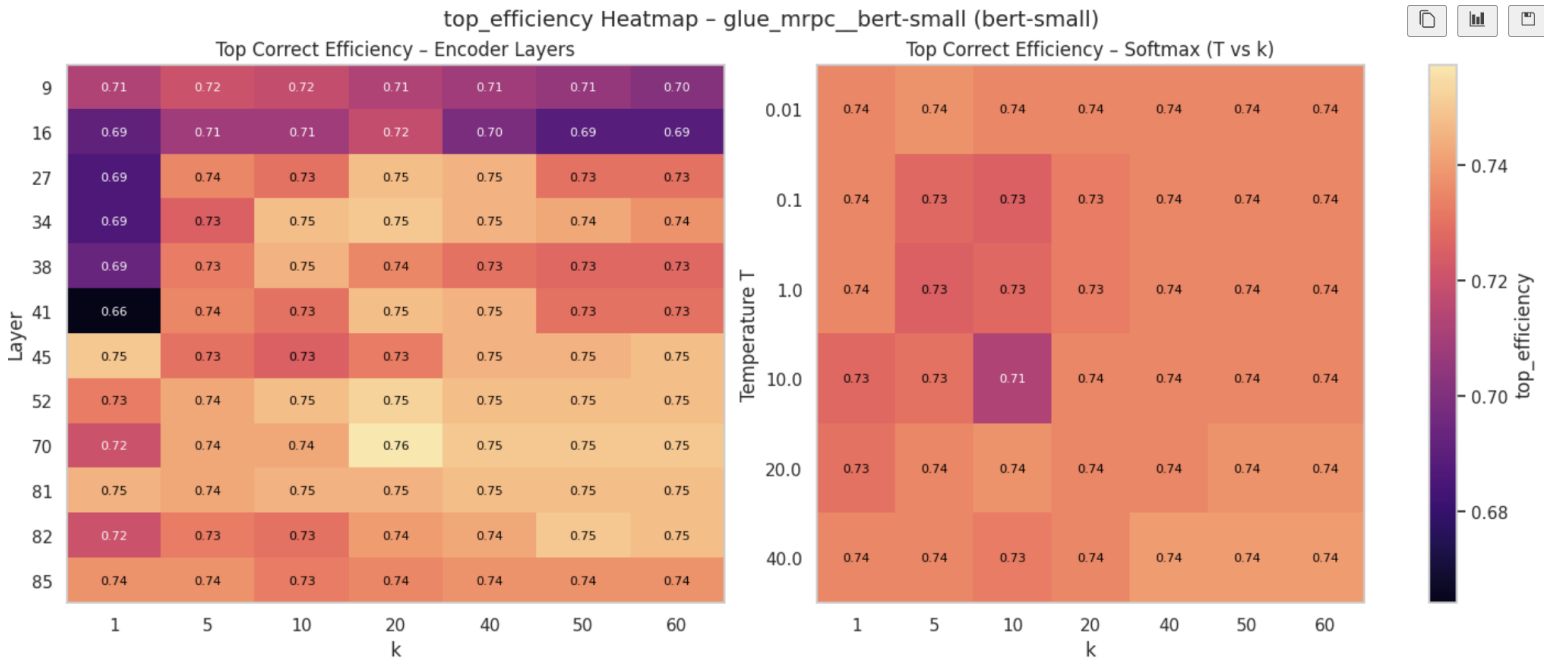


Figure 18: BERT-small hyperparameter testing on the MRPC dataset - correct efficiency

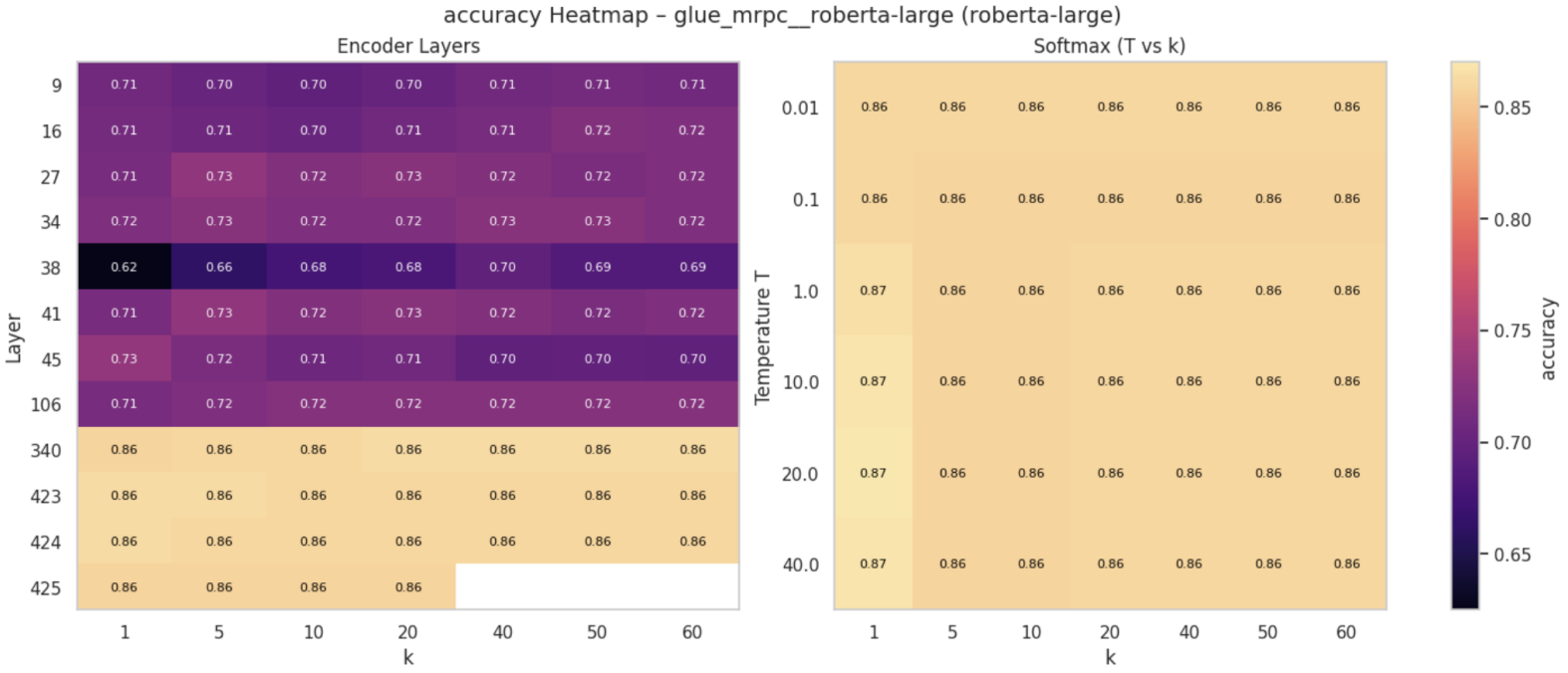


Figure 19: RoBERTa-large hyperparameter testing on the MRPC dataset - accuracy

downward trend. These reductions are consistent with the notion that deeper transformer layers produce more compact and separable representations, thereby streamlining the conformal prediction pipeline. However, not all transitions are monotonic. The shift from layer 34.0 to 38.0 causes calibration time to spike by over 5 seconds—an anomaly that likely reflects residual attention complexity or an unusually high activation dimensionality at that depth.

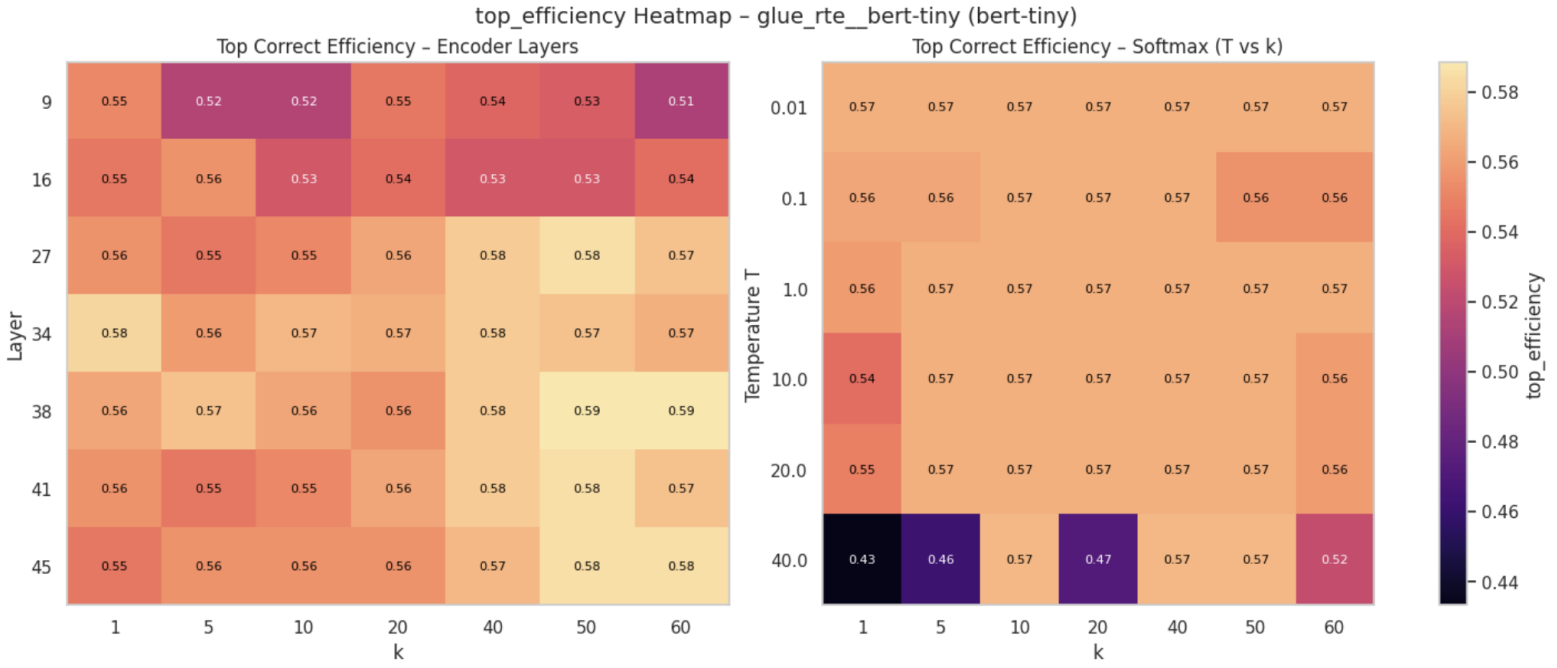


Figure 20: BERT-tiny hyperparameter testing on the RTE dataset - correct efficiency

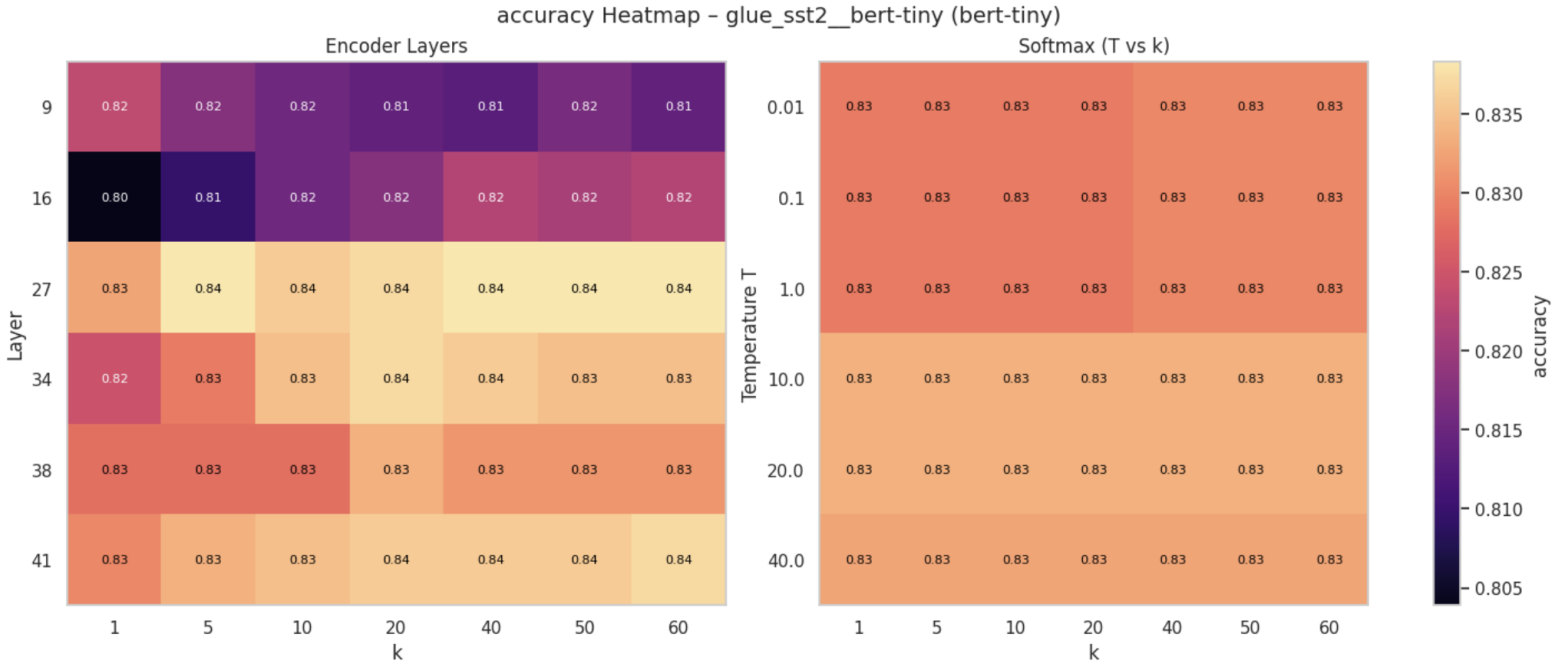


Figure 21: BERT-tiny hyperparameter testing on the SST2 dataset - accuracy

These findings suggest that while the relationship between layer depth and efficiency is not strictly linear, CONFIDE configurations that operate on later layers, particularly near the output, tend to offer superior computational efficiency. This trend reinforces the importance of jointly optimizing for both interpretability and runtime performance when selecting embedding layers for conformal prediction.

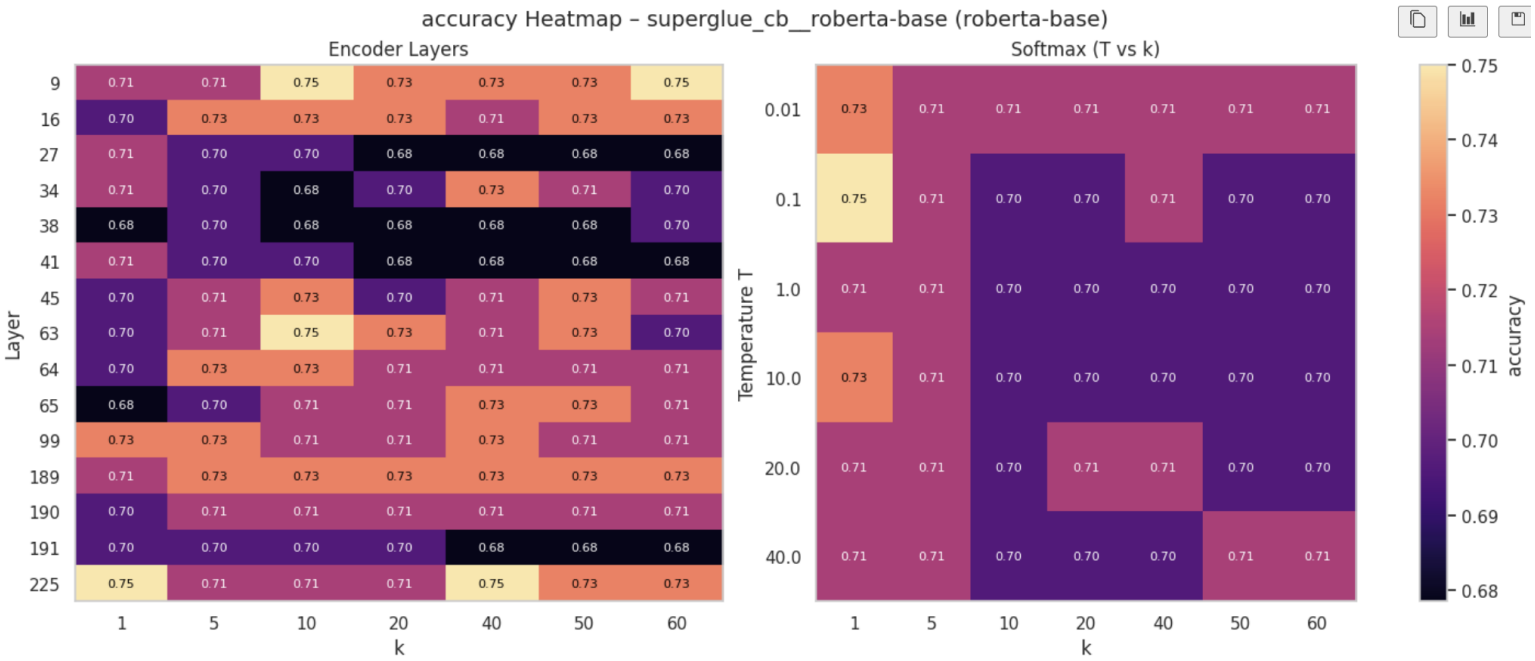


Figure 22: RoBERTa-base hyperparameter testing on the CB dataset - accuracy

Table 13: Change in CPU time (seconds) when using a later layer instead of an earlier one for CONFIDE's `train`, `calib`, and `test` stages. Generated using BERT-tiny, averaged across every dataset.

Layer Range	Train	Calib	Test
9.0 → 16.0	0.1724	-0.0008	0.0913
16.0 → 27.0	-0.4169	-0.5246	-0.1123
27.0 → 34.0	-0.1213	-0.0732	0.0128
34.0 → 38.0	0.0779	5.3893	0.7544
38.0 → 41.0	-0.1793	-5.2908	-0.8349

C.2 Computational Complexity Analysis

Our computational complexity analysis is fully based on the original CONFINE framework (Huang et al., 2025). The pre-processing phase requires every instance in the training and calibration sets to undergo a forward pass through the neural network for feature extraction. Let d denote the dimensionality of the extracted feature vector, N_t the size of the proper training set, and N_c the size of the calibration set. If T_f is the time taken for a single forward pass, then the total pre-processing time is given by $O((N_t + N_c)T_f + N_c d N_t)$. This includes both network inference and class-conditional nearest-neighbor index construction. Once feature vectors are extracted, they are stored in $O(dN_t)$ space. During inference, evaluating CONFINE on a single test point requires one additional forward pass and one nearest-neighbor lookup. This results in a total runtime of $O(T_f + dN_t)$ per test example. Compared to standard neural inference, which only incurs $O(T_f)$ time, CONFINE adds a cost of $O(dN_t)$ to enable interpretability.

D Additional Results

In Figs. 23-44, we present between 1-3 model configurations, both with and without classwise split, of each dataset. Graphs are sourced from the best hyperparameter combination, found in Section B.1. Boolq results are not repeated as both best parameter combinations are already presented in Section 5.

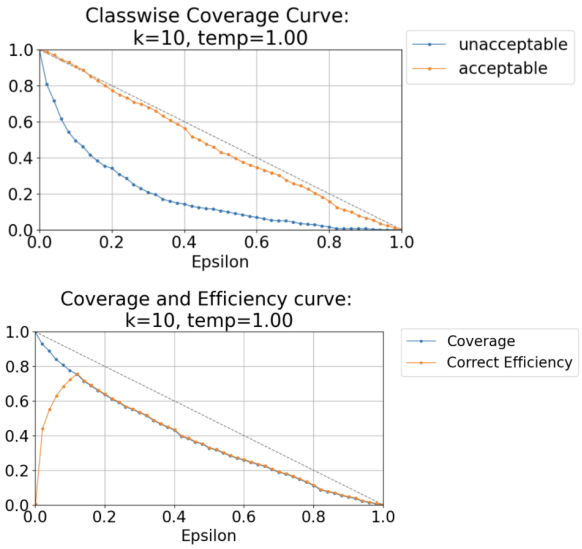


Figure 23: BERT-small CoLA classwise-false

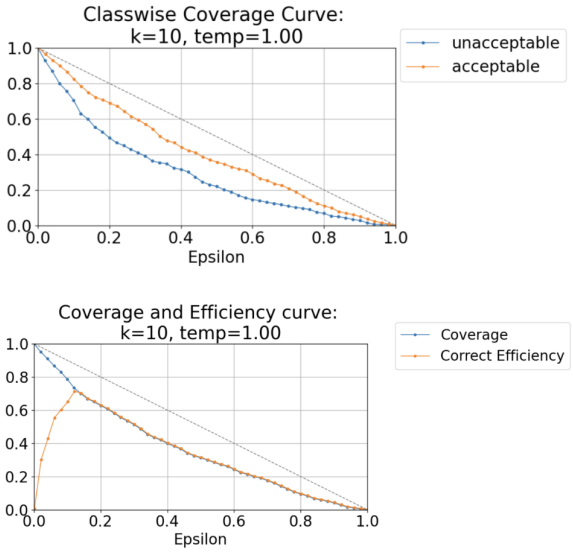


Figure 24: BERT-small CoLA classwise-true

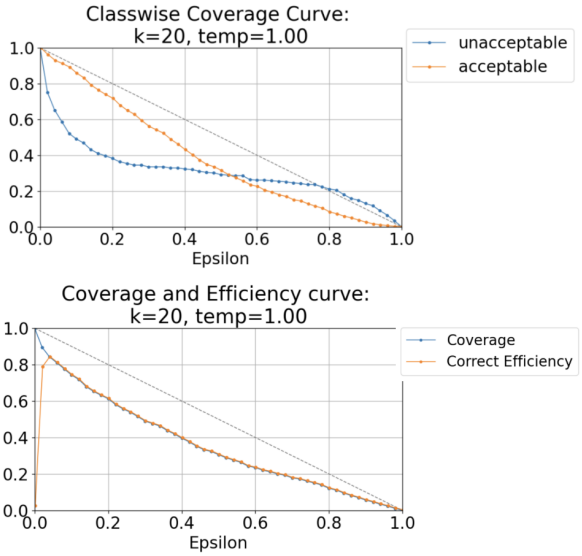


Figure 25: RoBERTa-base CoLA classwise-false

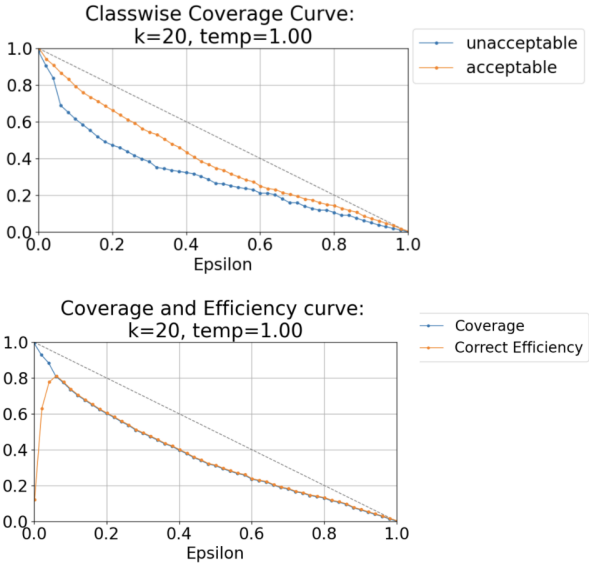


Figure 26: RoBERTa-base CoLA classwise-true

We observe a strong match between coverage and correct efficiency across most ϵ values in the QNLI task using BERT-tiny, especially when classwise calibrated. This suggests that even with a limited model capacity, the learned representations in QNLI provide sufficient semantic separation between the "entailment" and "not_entailment" classes. QNLI tends to involve syntactically structured queries with informative lexical anchors. This consistency likely helps CONFIDE identify meaningful neighborhoods in the embedding space.

Then, despite WNLI being a notoriously noisy dataset with adversarial structure, we see surprisingly effective calibration here. The slight smoothness and separation in classwise curves imply that the selected layer (and hyperparameters) successfully isolate decision boundaries in a way that suppresses overfitting to artifacts.

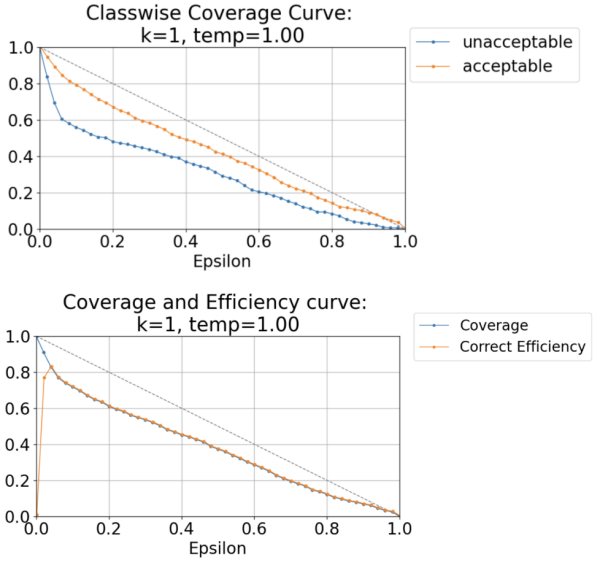


Figure 27: RoBERTa-large CoLA classwise-false

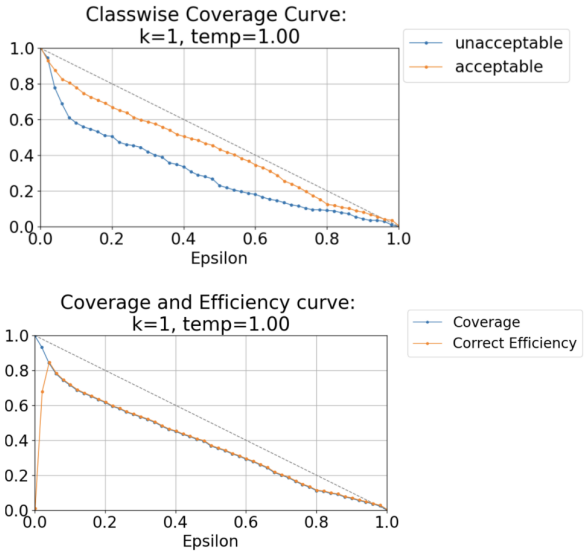


Figure 28: RoBERTa-large CoLA classwise-true

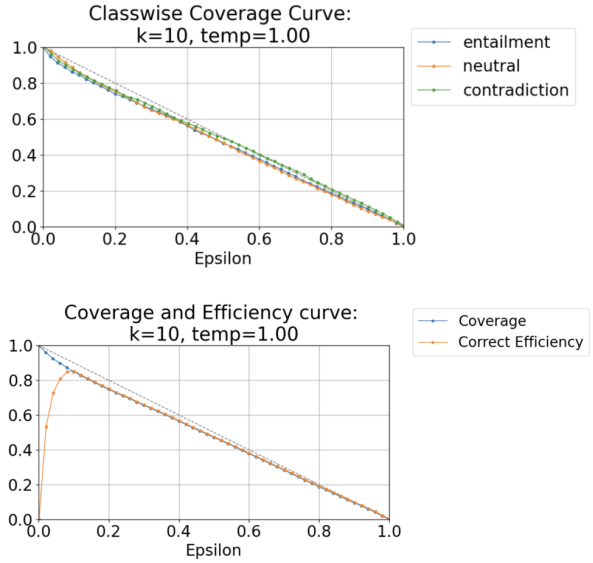


Figure 29: RoBERTa-base MNLI classwise-true

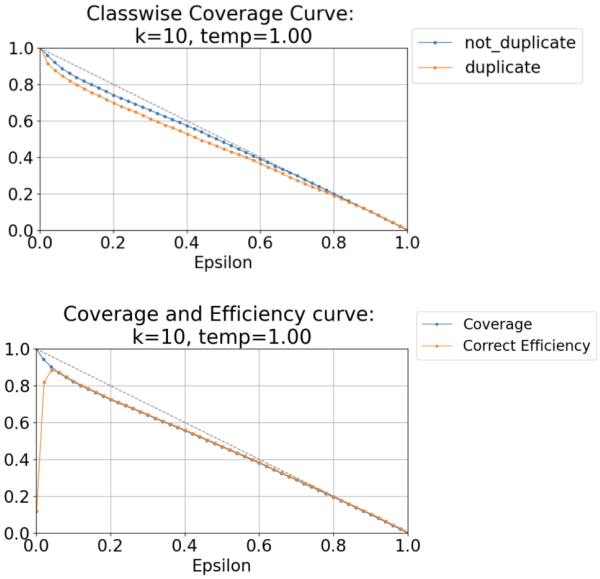


Figure 30: BERT-small QQP classwise-true

Meanwhile, CONFIDE struggles with MultiRC in generating meaningful classwise separation or maintaining correct efficiency. Coverage drops off rapidly while correct efficiency remains flat, indicating that the internal representations fail to capture the task's complexity.

Ultimately, these examples illustrate that CONFIDE is highly adaptable across datasets and transformer architectures. However, its success is dependent on internal factors such as the semantic coherence of class labels, the quality of intermediate layer representations, and dataset-specific noise or adversarial traits. Careful calibration and model-aware design choices are critical to unlocking CONFIDE's full potential.

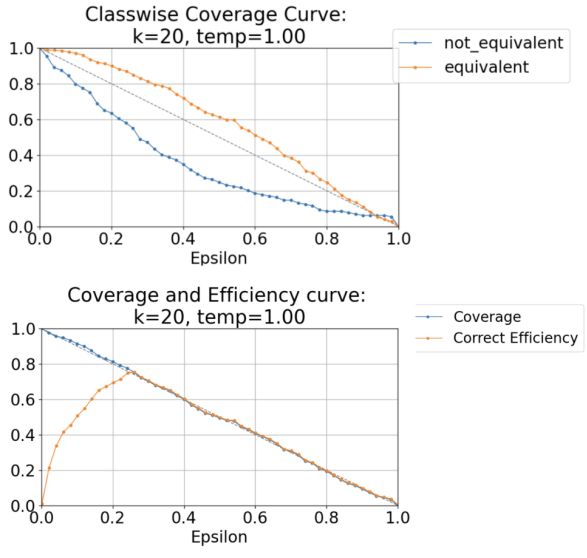


Figure 31: BERT-small MRPC classwise-false

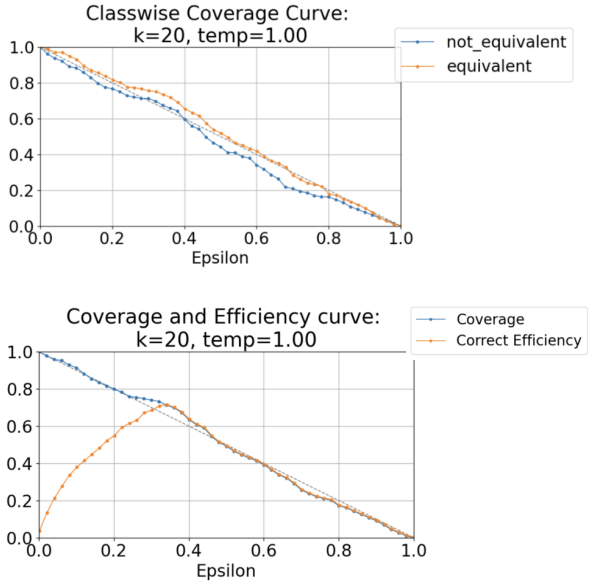


Figure 32: BERT-small MRPC classwise-true

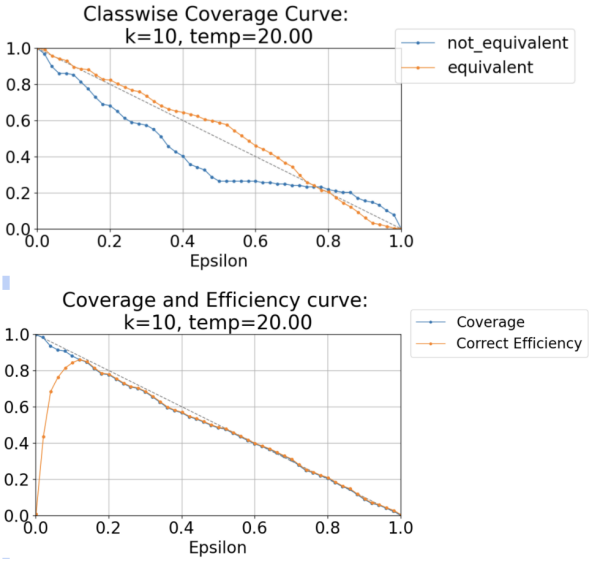


Figure 33: RoBERTa-large MRPC classwise-false

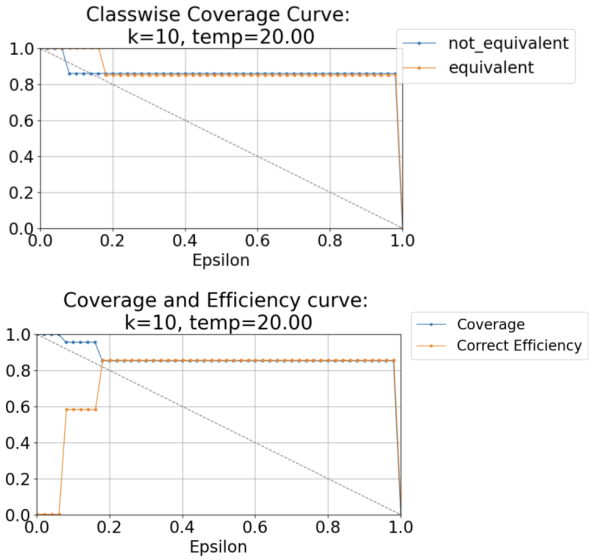


Figure 34: RoBERTa-large MRPC classwise-true

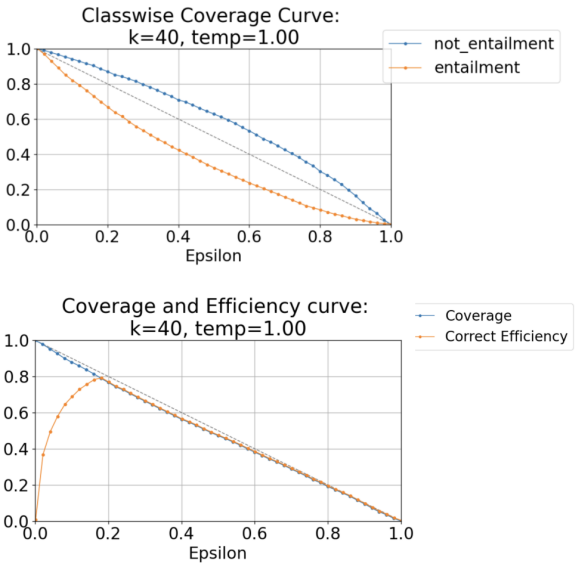


Figure 35: BERT-tiny QNLI classwise-false

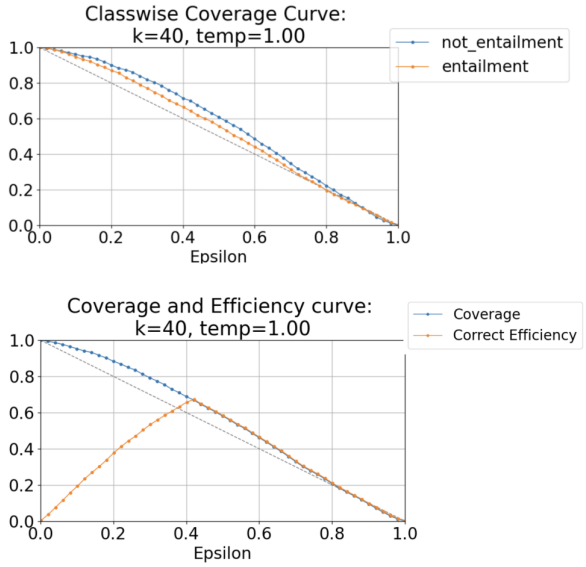


Figure 36: BERT-tiny QNLI classwise-true

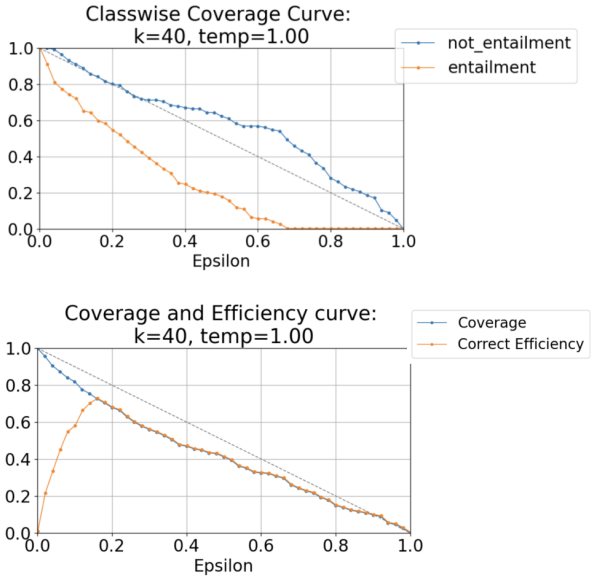


Figure 37: RoBERTa-base RTE classwise-false

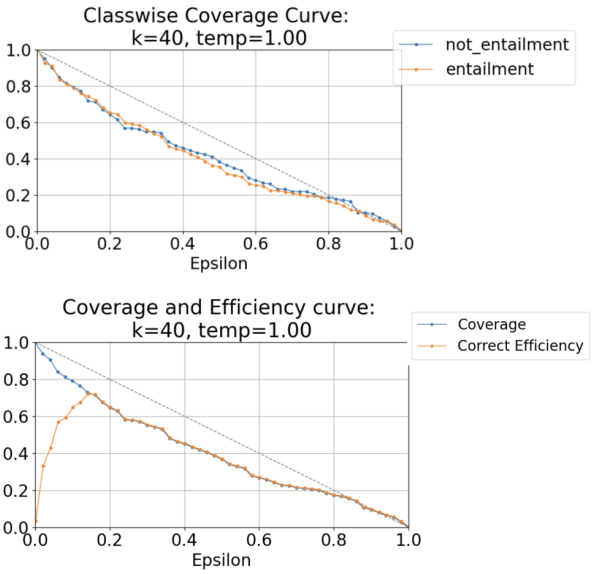


Figure 38: RoBERTa-base RTE classwise-true

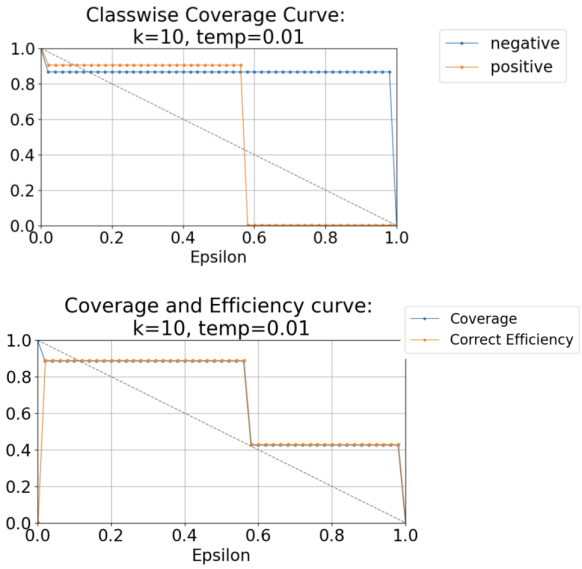


Figure 39: BERT-small SST2 classwise-false

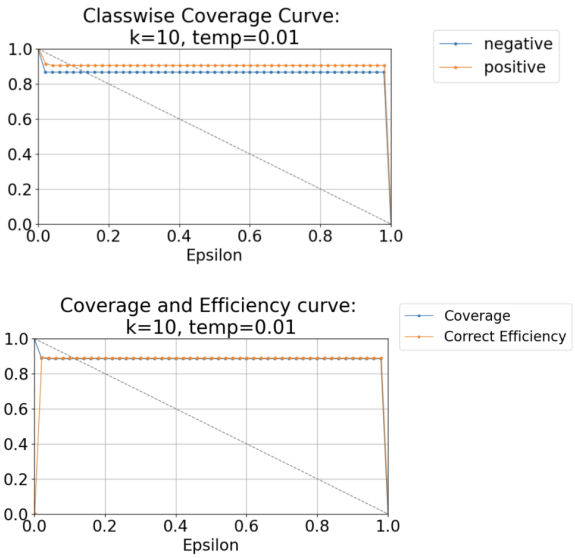


Figure 40: BERT-small SST2 classwise-true

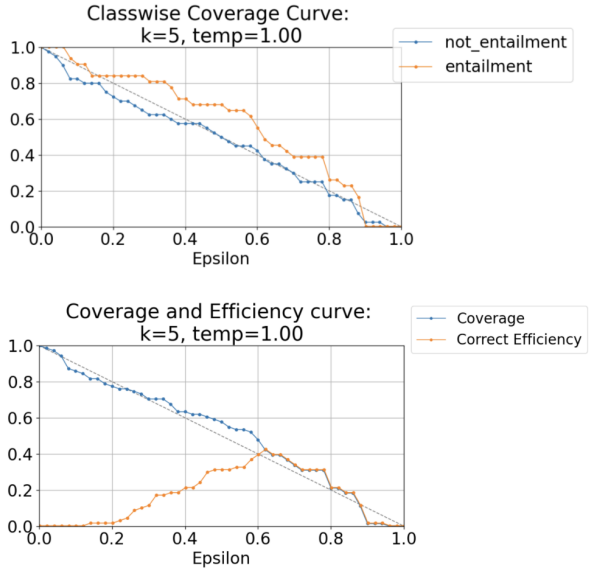


Figure 41: BERT-small WNLI classwise-false

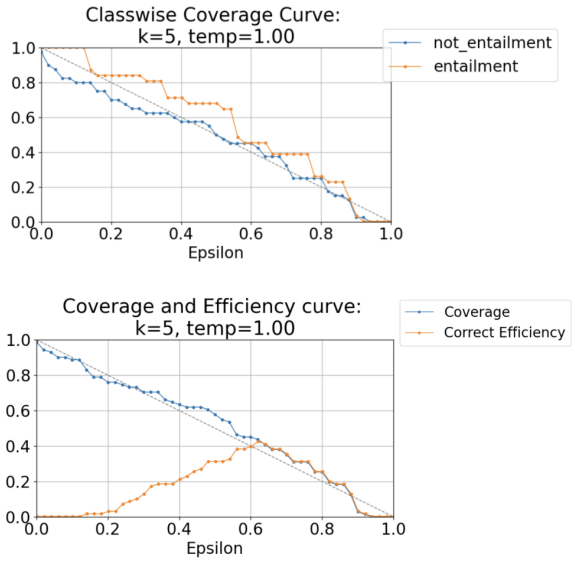


Figure 42: BERT-small WNLI classwise-true

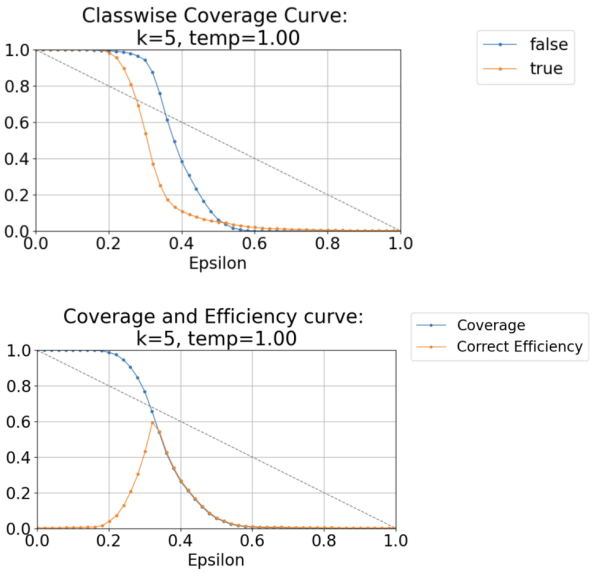


Figure 43: BERT-tiny MultiRC classwise-false

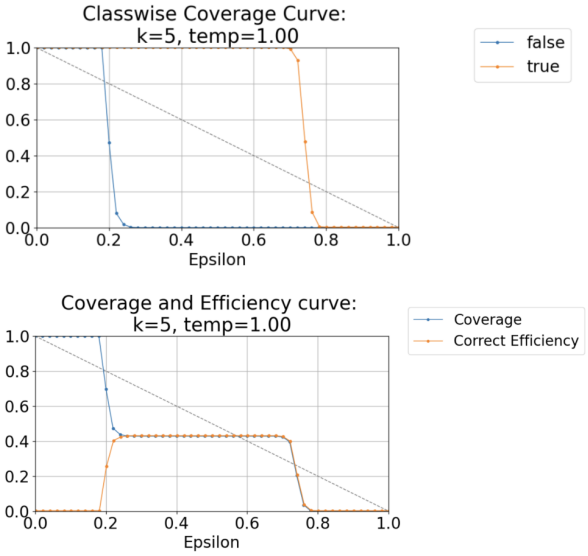


Figure 44: BERT-tiny MultiRC classwise-true