SecPE: Secure Prompt Ensembling for Private and Robust Large Language Models

Anonymous ACL submission

Abstract

With the growing popularity of LLMs among the general public users, privacy-preserving and adversarial robustness have become two pressing demands for LLM-based services, which have largely been pursued separately but rarely jointly. In this paper, to the best of our knowledge, we are among the first attempts towards robust and private LLM inference by tightly integrating two disconnected fields : private inference and prompt ensembling. The former protects users' privacy by encrypting inference data transmitted and processed by LLMs, while the latter enhances adversarial robustness by yielding an aggregated output from multiple prompted LLM responses. Although widely recognized as effective individually, private inference for prompt ensembling together entails new challenges that render the naive combination of existing techniques inefficient.

To overcome the hurdles, we propose SecPE, which designs efficient fully homomorphic encryption (FHE) counterparts for the core algorithmic building blocks of prompt ensembling. We conduct extensive experiments on 8 tasks to evaluate the accuracy, robustness, and efficiency of SecPE. The results show that SecPE maintains high clean accuracy and offers better robustness at the expense of merely 2.5% efficiency overhead compared to baseline private inference methods, indicating a satisfactory "accuracy-robustness-efficiency" tradeoff. For the efficiency of the encrypted Argmax operation that incurs major slowdown for prompt ensembling, SecPE is 20.8 times faster than the state-of-the-art peers, which can be of independent interest beyond this work.

1 Introduction

Large language models (LLMs) have garnered a meteoric rise in popularity among general public users due to their remarkable performance across myriad natural language processing (NLP) tasks (Xu et al., 2019; Yang et al., 2019a). LLMs are oftentimes deployed by service providers in the form of Machine Learning as a Service (MLaaS) (Yang et al., 2019b; Raffel et al., 2020), whereby users can conveniently exploit the full potential of LLM by submitting their inference data, prepended by specific prompts from prompt learning techniques (Li et al., 2023c,a; Xu et al., 2024), to obtain high-performing LLM outputs tailored to their downstream tasks. Accompanying this widespread adoption, there arise privacy and robustness concerns for LLMs (Gilad-Bachrach et al., 2016; Juvekar et al., 2018; Liu et al., 2017; Brutzkus et al., 2019; Chou et al., 2018; Lou and Jiang, 2019).

Privacy concerns and private inference. On the privacy aspect, users' inference data can inadvertently reveal sensitive information if transmitted and processed by the LLM service provider in plaintext (Yang et al., 2019b; Raffel et al., 2020), risking identification and privacy breaches. Additionally, the user-submitted prompts can be valuable intellectual property and also raise privacy concerns. As a result, both inference data and user-side prompts demand privacy-preserving measures (Gilad-Bachrach et al., 2016; Juvekar et al., 2018; Liu et al., 2017; Brutzkus et al., 2019; Chou et al., 2018; Lou and Jiang, 2019). Among the many attempts to avoid submitting raw data for LLM inference, private inference offers very strict privacy protection by allowing inference to be conducted on encrypted data. For instance, Fully Homomorphic Encryption (FHE) allows rich computations (covering most operations needed in LLM inference) on encrypted data without exposing sensitive information (Gentry, 2009). By encrypting inputs using FHE, only encrypted predictions are sent to the server, ensuring privacy throughout the process. As legal and societal pressures mount, the adoption of such privacy-preserving technologies by service providers has received increasing research attention (Barua, 2021; Masters et al., 2019).

Robustness concern and prompt ensembling. On the robustness aspect, it is well-recognized that the output of LLMs can be manipulated by subtle yet deliberate changes in the inference sample or the prompt (Wang et al., 2024). There has been a growing focus on enhancing the robustness of LLMs, especially in safety-critical downstream application areas. Various methods have been proposed, ranging from more advanced (and sophisticated) (Vu et al., 2021; Asai et al., 2022) to simple methods (Dvornik et al., 2019; Liu et al., 2020). One representative method from the latter category follows the idea of prompt ensembling (Schick and Schütze, 2020; Lester et al., 2021), which involves making multiple inferences for a single inference data and providing the aggregated result as the final prediction.

This study. The current research efforts on safeguarding privacy and robustness during LLM inference are largely explored separately. Driven by the simultaneous demands from both privacy and robustness aspects, we envision that these two aspects should be pursued jointly. Among the first attempts toward mitigating both concerns of LLMs jointly, we investigate the potential to achieve private and robust LLM inference through tight integration of private inference and prompt ensemble. We focus on these two techniques due to their effectiveness in addressing their respective concerns. In particular, we note that while there may be more advanced techniques for enhancing robustness than prompt ensembling, achieving a balance between robustness and efficiency within the private inference workflow of the simpler prompt ensembling method already poses significant challenges. That is, naive application of existing private inference methods for prompt ensembling entails great efficiency overhead. The crux of efficient private inference for prompt ensembling is that the aggregation operation introduced by prompt ensembling, albeit simple and efficient in plaintext computation, requires prohibitive computation in ciphertext.

To overcome the inefficiency challenges, we propose `SecPE` : a new secure prompt ensembling method for private and robust LLM inference. As illustrated in Figure 1, `SecPE` allows user to encrypt their inference data and prompts before transmitting to the LLM server for inference. The inference results from the LLM server are aggregated from multiple prompted responses and transmitted back to the user in ciphertext format, which can be de-

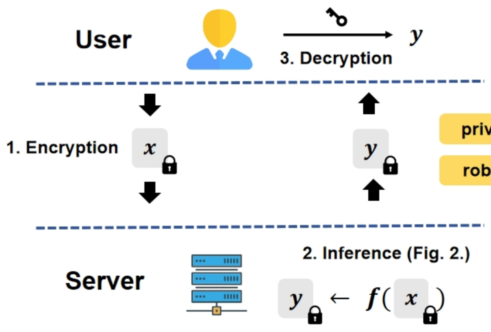

FIGURE 1 – A high-level overview of `SecPE` for private and robust LLM inference in FHE-based MLaaS.

crypted only by the user's private key. The encrypted aggregation operation heavily relies on efficient computation of `Argmax`, which is unfortunately not readily supported by the common homomorphic primitives like the RNS-CKKS FHE scheme (Lee et al., 2022). Lying at the design core of `SecPE` is a new efficient private aggregation algorithm to be presented in Algorithm 1, which resorts to an efficient approximation of `Argmax` to circumvent this efficiency bottleneck. We conduct extensive experiments to test the accuracy, robustness, and efficiency of `SecPE` across 14 tasks from GLUE, Adv-GLUE, and mathematical reasoning data sets. The results show that `SecPE` is capable of maintaining both high utility and robustness while providing privacy protection.

The main contributions of this paper are summarized as follows :

— To the best of our knowledge, we are among the first to jointly study the privacy and robustness concerns of LLM inference, which become increasingly pressing considering the growing deployment of LLM-based services.

— We propose `SecPE` to achieve private and robust LLM inference, which devises new secure primitives tailor-made for prompt ensembling to strike a satisfactory "accuracy-robustness-efficiency" tradeoff.

— We conduct extensive experiments on 8 tasks from 3 popular benchmarks to corroborate the superior performance of `SecPE` against baseline methods.

2 Background

2.1 Privacy Issues of LLMs

LLMs such as the GPT have revolutionized natural language processing and understanding with human-level proficiency (Kenton and Toutanova, 2019; Brown et al., 2020). However, with their increasing deployment in MLaaS by service providers and growing popularity among the general public users, there arise aggravating privacy concerns. In the typical MLaaS serving setting, users submit inference data to the remote server hosting a proprietary model and receive predictions in return. Users therefore have privacy concerns about their inference data that, despite being sensitive or even confidential, are transmitted and processed in plaintext by the MLaaS service provider (Shen et al., 2007; Christoph et al., 2015). This issue has even led to ChatGPT being temporarily banned in Italy (Mauran, 2023; Natasha Lomas, 2023; Cecily Mauran, 2023). Recognizing this pressing privacy concern, existing works introduce various means to avoid direct transmission and processing inference data in plain text form.

Private inference emerges as a viable solution, promising to reconcile the need for high-performant inference data processing with strict privacy requirements (Srinivasan et al., 2019; Hao et al., 2022; Pang et al., 2024). Private inference provides a way to guarantee the privacy and confidentiality of both the inference data and the proprietary LLM. It ensures that data is not transmitted or processed in plaintext but as ciphertext, thereby safeguarding sensitive details about the server's model weights and the User's inputs from disclosure. While private inference has significant applications in computer vision and image processing (Arnab et al., 2021; Wang et al., 2022b; Zeng et al., 2023), its use in LLMs is nascent. Notably, the integration of private inference in prompt learning settings and prompt ensembles remains an under-explored area, presenting a frontier yet to be ventured into the field.

By pursuing private inference tailored for prompt ensemble learning, we aim to bridge the gap between utility, robustness, and privacy, thereby realizing the benefits of prompted LLMs without compromising user trust and data integrity.

2.2 Private Inference via Fully Homomorphic Encryption

The FHE scheme used in this paper is the full *residue number system* (RNS) variant of Cheon-Kim-Kim-Song (CKKS) (Cheon et al., 2017, 2019). RNS-CKKS is a *leveled* FHE, which can support computations up to a multiplicative depth L. Both the plaintexts and ciphertexts of RNS-CKKS are elements in a polynomial ring :

$$\mathcal{R}_Q = \mathbb{Z}_Q[X]/(X^N + 1)$$

where $Q = \Pi_{i=0}^{L} q_i$ with distinct primes q_i. Once a ciphertext's level becomes too low, a *bootstrapping* operation is required to refresh it to a higher level, enabling more computations. In a nutshell, bootstrapping homomorphically evaluates the decryption circuit and raises the modulus from q_0 to q_L by leveraging the isomorphism $\mathcal{R}_{q_0} \cong \mathcal{R}_{q_0} \times \mathcal{R}_{q_1} \times \cdots \times \mathcal{R}_{q_L}$ (Bossuat et al., 2021). Suppose the bootstrapping consumes K levels, then a fresh ciphertext can support $L - K$ levels of computations.

2.3 Prompt Ensembling for Robust LLMs

The brittleness of LLMs to slight input modifications often leads to varied/inaccurate and sometimes even malicious/harmful outputs, highlighting the essential need for enhanced robustness for LLMs (Talmor et al., 2020; Schick et al., 2020; Jiang et al., 2020). Robustness in this context refers to LLM's ability to provide consistent predictions regardless of slight changes to the inference data, aiming for more predictable and stable responses.

Building on the success of prompt learning, prompt ensemble learning (Lu et al., 2022; Allingham et al., 2023) demonstrates the potential to offer efficient, effective, and robust predictions. Prompt ensemble utilizes a series of prompts to allow for the aggregation of multiple responses for the same inference data, leading to more robust predictions.

Prompt ensembling, in which the masked language model $\mathcal{L}$ is directly tasked with "auto-completing" natural language prompts. For instance, for the inference data x_{in}, the template into which the inference data is inserted that x_{prompt} = "It was `MASK`" is concatenated (i.e., $x_i = x_{in} \oplus x_{\text{prompt}}$), The prompt typically includes one or more masked tokens `[MASK]` that the model $\mathcal{L}$ is expected to fill in, making it a structured query that directs the model's response.

The single output refers to the model's prediction for each prompt, drawing on the context of the prompt and input data present, like determining the sentiment of a movie review. When multiple prompts or input variations are used to obtain a range of model responses, the aggregated output synthesizes these individual outputs to derive a more robust or accurate prediction. This aggre-

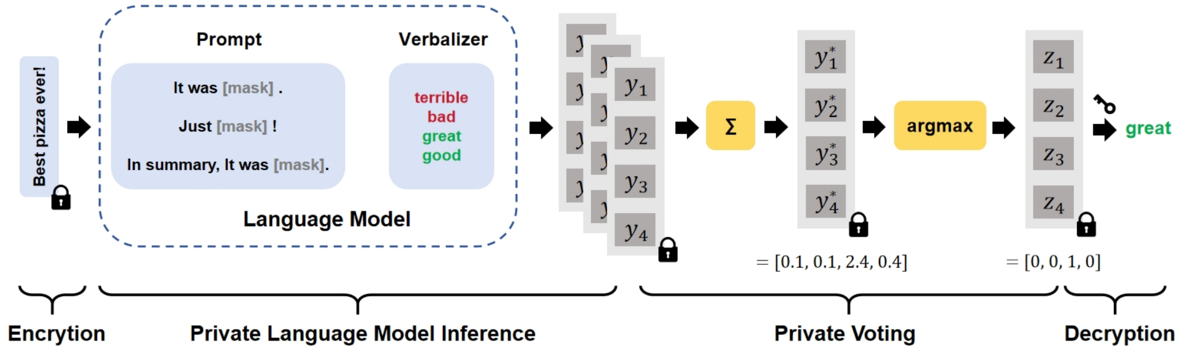

FIGURE 2 – An illustration of secPE, which enables homomorphically encrypted LLM inference with guarantees.

gation could involve combining the model's responses to enhance prediction reliability or accuracy, especially in tasks where nuanced understanding or multiple aspects of the input data are considered.

Suppose there are m prompt templates, the verifier takes a question and a candidate reasoning path as input and outputs the probability that the reasoning path leads to the correct answer (Li et al., 2023b).

$$y^* = \text{argmax}(\sum_{i=1}^{m} \mathcal{L}(x_{in} \oplus x_{\text{prompt}})),$$

where $f(\cdot)$ is the probability produced by the verifier.

3 Proposed Method : SecPE

We propose a new private inference framework tailor-made for the prompt ensembling. Private inference for prompt ensembling raises a critical, unaddressed issue : the challenge of integrating private contextual inference. Incorporating privacy-preserving mechanisms into prompt ensembles remains a significant and complex challenge, despite progress in leveraging prompt-based learning to improve model effectiveness in downstream tasks. Our work aims to break new ground by developing a comprehensive framework that not only improves model performance through optimized prompt selection but also prioritizes the integration of robust privacy safeguards.

3.1 SecPE Framework

We give an illustration of SecPE in Fig 2, the overall process is divided into the following four steps :

1. **Encryption.** User encrypts m inputs $x_i = x_{in} \oplus x_{\text{prompt}}, i \in [1, m]$ using FHE and sends them to the server, where m is the number of prompt templates.
2. **Private Language Model Inference.** Server uses the language model $\mathcal{L}$ classifying m inputs into one of n classes, n is the number of labels. the inputs are propagated through $\mathcal{L}$ utilizing the homomorphic operations of the FHE scheme (Chen et al., 2022; Hao et al., 2022) to obtain m encrypted logits $y_i, i \in [1, m]$.
3. **Private Voting.** Server aggregates the encrypted logits $y^* \leftarrow \sum_{i=1}^{m} y_i$ and then evaluates `Argmax` function in FHE. In particular, this step transforms the logit vector y^* into a one-hot vector z. Then the server sends z to the User.
4. **Decryption.** User decrypts z with its secret key, where the single non-zero entry represents the index of the predicted classification label.

In the workflow of SecPE described above, Steps 1 and 4 involve basic FHE encryption and decryption. Step 2 has been implemented in many recent works (Chen et al., 2022; Hao et al., 2022; Pang et al., 2024). These three steps are orthogonal to the efficiency designs of prompt ensembling. The key challenge lies in using FHE to evaluate `Argmax` in Step 3. As FHE does not allow evaluation of control flow (e.g., branching), and ciphertext comparison (e.g., checking inequality) is not directly supported by the homomorphic primitives of the RNS-CKKS FHE scheme, we therefore cannot canonically implement `Argmax`. Instead, we aim for an efficient approximation to circumvent this efficiency bottleneck raised by prompt ensembling.

3.2 Efficient Private Inference for Prompt Ensembling

As mentioned above, the design core of efficient private inference for prompt ensembling lies at the private aggregation operator, i.e., the `Argmax` operation. Therefore, our goal is to approximate the following function on an RNS-CKKS ciphertext logit vector :

$$[y_1, ..., y_n, 0^{N-n}] \to [z_1, ..., z_n, \#^{N-n}], \quad (1)$$

where $z_i = 1$ for the index i corresponding to the largest value among $[y_1, y_2, ..., y_n]$ (and 0 elsewhere).

The state-of-the-art protocol that can achieve this goal is Phoneix (Jovanovic et al., 2022), which requires $(m+1)$ times $Sign$ operations and $(m+1)$ times ciphertext rotations. Our method only requires $(\log n + 1)$ times $Sign$ operations and $(\log n + 1)$ times ciphertext rotations.

We innovatively proposed an `Argmax` evaluation method as :

$$z_i \leftarrow Sign(y_i - y_{max}) + 1. \quad (2)$$

To enable encrypted comparisons, we leverage the polynomial approximation of the sign function :

$$Sign(x) = \begin{cases} -1 & -1 \le x \le -2^{-\alpha} \\ 0 & x = 0 \\ 1 & 2^{-\alpha} \le x \le 1 \end{cases} \quad (3)$$

The approximation (Cheon et al., 2020) involves a composition of polynomials :

$$Sign(x) = f^{d_f}(g^{d_g}(x)), \quad (4)$$

where $f(),\ g()$ are two polynomials and $d_f,\ d_g$ are the number of repetitions for them. In our implementation, both $f()$ and $g()$ are 9-degree polynomials ; we set $\alpha = 12, d_f = 2, d_g = 2$, so the max error bound is less than 10^{-4}. To reduce the multiplicative depth, we evaluate the polynomials using the Baby-Step-Giant-Step algorithm (Han and Ki, 2020).

Before proceeding, we comment on the basic input requirement of $Sign(x)$, namely that its inputs are in $[-1, 1]$. Suppose the inputs $x_i \in [D_{min}, D_{max}]$, to ensure this requirement, for those inputs that need to be different from each other, we need to normalize $\hat{x}_i \in [0, 1]$:

$$\hat{x}_i = \frac{x_i - D_{min}}{D_{max} - D_{min}}, \quad (5)$$

meaning that for all $i \neq j$, $\hat{x}_i - \hat{x}_j \in [-1, 1]$, satisfying the requirement in Algo.1

Algorithm 1 `Argmax` on RNS-CKKS

Input: $[y_1, y_2, ..., y_n, 0^{N-n}]$
Output: $[z_1, z_2, ..., z_n, \#^{N-n}]$) as in Eq. 1

```
 1: function Argmax(y)
 2:     y ← y ⊕ RotR(y, n)
 3:     y_max ← QuickMax(y)
 4:     y ← y ⊖ y_max
 5:     z ← Sign(y)
 6:     z ← z ⊕ 1
 7:     return z
 8: end function
 9: function QuickMax(y)
10:     l ← log_2 n
11:     for i = 0 to log n − 1 do
12:         r ← RotL(y, 2^i)
13:         r ← Max(r, y)
14:         y ← r
15:     end for
16:     return y
17: end function
```

In order to get x_{max}, with the help of the $Sign$ function, we can calculate the maximum value of a and b by :

$$Max(a, b) = \frac{a+b}{2} + \frac{a-b}{2} \cdot Sign(a-b). \quad (6)$$

Then, the selection vector can be easily computed as described in Algorithm 1.

In Fig. 3, we illustrate how Alg. 1 processes a toy example. The algorithm first duplicates the logits (Line 2), then use $QuickMax$ to get the maximum value of $[y_1, y_2, ..., y_n]$. Unlike phoenix (Jovanovic et al., 2022), we do not rotate only one step at a time, but rotate $2^i,\ i \in [0, \log n - 1]$ steps each time, which greatly reduces our number of rotations and the number of $Sign$ operations.

4 Experiments

4.1 Experimental setup

Tasks and Datasets. In the experiments, we utilize 8 tasks from popular benchmarks to thoroughly evaluate the utility, robustness, and efficiency of `secPE`.

i) Benign NLP tasks We evaluate `secPE` on six tasks from the **GLUE** benchmark (Wang et al., 2018). In detail, the evaluated tasks are (1) SST-2 (Socher et al., 2013) ; (2) QQP ; (3) MNLI-matched ; (4) MNLI-mismatched (Williams et al., 2017), (5) RTE (Giampiccolo et al., 2007), and (6)

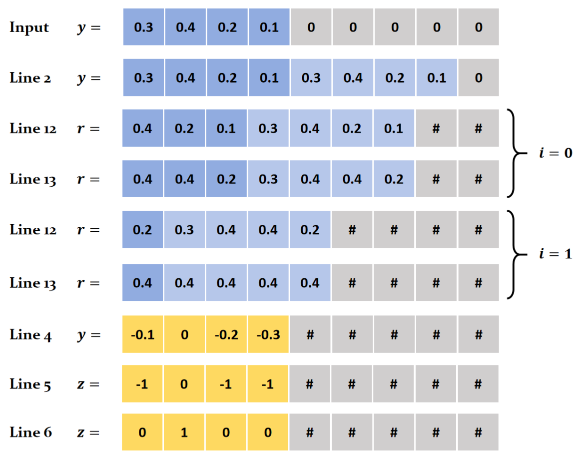

FIGURE 3 – Example run of Algorithm 1.

Task	Template	Verbalizer
SST-2	It was [MASK] . $< S_1 >$ $< S_1 >$. All in all, it was [MASK]. Just [MASK] ! $< S_1 >$ In summary, the movie was [MASK].	bad / good bad / good bad / good bad / good
QQP	$< S_1 >$ [MASK], $< S_2 >$ $< S_1 >$ [MASK], I want to know $< S_2 >$ $< S_1 >$ [MASK], but $< S_2 >$ $< S_1 >$ [MASK], please, $< S_2 >$	No / Yes No / Yes No / Yes No / Yes
MNLI	$< S_1 >$? [MASK], $< S_2 >$ $< S_1 >$? [MASK], $< S_2 >$" $< S_1 >$? [MASK], $< S_2 >$	Wrong/Right/Maybe No/Yes/Maybe Wrong/Right/Maybe
RTE	"$< S_2 >$? [MASK], $< S_1 >$" "$< S_2 >$? [MASK], $< S_1 >$ "$< S_1 >$? [MASK]. $< S_2 >$	No/Yes No/Yes No/Yes
QNLI	$< S_1 >$? [MASK], $< S_2 >$ $< S_1 >$? [MASK], $< S_2 >$ "$< S_1 >$? [MASK], $< S_2 >$"	No/Yes Wrong/Right Wrong/Right No/Yes

TABLE 1 – Manual template and verbalizer pairs. $< S_1 >$ and $< S_2 >$ are the input sentences.

QNLI—range (Rajpurkar et al., 2016), which range from sentiment analysis to question answering, diversifying in different inference data formats from sentences to pairs of sentences.

ii) Adversarial NLP tasks We evaluate the robustness of secPE on six adversarial tasks in the Adversarial-GLUE (**AdvGLUE**) benchmark (Wang et al., 2021), which are adversarial counterparts to the above benign GLUE tasks. The AdvGLUE benchmark is enriched with task-specific adversarial examples generated by 14 different textual attack methods, coming different adversarial perturbation strategies including word-level, sentence-level, and human-generated. Recognizing the potential problem of invalid adversarial constructs identified by Wang et al. (Wang et al., 2021), where up to 90% of automatically generated examples may be flawed, we also incorporate human validation. This step allows for a more accurate and robust evaluation of secPE by ensuring that the adversarial examples in our benchmark are legitimate and that the perturbations maintain the integrity of the original task.

iii) Arithmetic reasoning tasks We evaluate the self-consistency of SecPEon two arithmetic reasoning benchmarks : GSM8K (Cobbe et al., 2021) and MultiArith (Roy and Roth, 2016). GSM8K contains grade-school-level mathematical word problems requiring models to perform complex arithmetic reasoning and multi-step calculations. MultiArith contains multiple arithmetic operations within a single problem, testing a model's ability to comprehend and execute a sequence of calculations, reflecting the complexity of mathematical reasoning needed for higher accuracy in various problem-solving contexts.

Models for Evaluation. In the context of the SecPEframework, we specifically implement private inference both on ALBERT-XXLarge-v2 (Lan et al., 2019) and GPT-3 code-davinci001 engine (Chen et al., 2021). This allows techniques like LM-BFF and PET to process ciphertext inputs, thereby facilitating privacy-preserving inference.

i) ALBERT-XXLarge-v2 For tasks within the GLUE and AdvGLUE benchmarks, we use the ALBERT-XXLarge-v2 model to generate different contextual representations. This combined text is fed into the model to obtain the language model results. This method allows us to assess the relationship between questions and their corresponding answers, taking advantage of the model's pre-trained capabilities.

ii) GPT-3 code-davinci-001 For reasoning tasks such as MultiArith and GSM8K, we used the GPT-3 model, specifically the code-davinci-001 variant. This model was chosen for its advanced ability to handle complex language patterns and to generate coherent, contextually relevant text completions.

Baseline Methods. We compare SecPE with three different baseline methods : Classic fine-tuning (Devlin et al., 2019), LM-BFF (Gao et al., 2021), and PET (Schick and Schütze, 2021). Below, we briefly describe the FSL methods and explain our rationale for considering them in our study.

— **LM-BFF (Gao et al., 2021) :** It involves concatenating the input example, which is modified to follow the prompting template with a [MASK] in place of the verbalizer, with semantically similar examples. During inference, LM-BFF

Method	Prompt	Setting	SST-2	QQP	MNLI-m	MNLI-mm	RTE	QNLI
LM-BFF (Plaintext)	Single	Cln	94.0	80.1	76.7	78.3	78.1	81.4
		Adv	54.1	46.2	47.1	40.1	58.8	61.5
LM-BFF (Ciphertext)	Single	Cln	93.7	79.2	76.0	77.6	77.5	81.0
		Adv	53.8	46.1	46.4	39.5	58.2	61.1
PET (Plaintext)	Ensemble	Cln	93.4	73.7	74.6	75.7	74.2	84.6
		Adv	61.7	59.3	55.6	44.8	54.0	67.9
SecPE	Ensemble	Cln	93.0	73.1	73.2	74.7	72.2	81.1
		Adv	61.3	59.3	55.4	43.9	53.2	66.8

TABLE 2 – Performance comparison on GLUE (Cln) and Adversarial GLUE (Adv) benchmarks. We report the average and standard deviation in the accuracy values of 5 different runs.

ensembles the predictions made by concatenating the input example with all demonstrations from the few-shot training set. (i.e., demonstrations) from the few-shot training set. For each test example, we ensemble the predictions over different possible sets of demonstrations. we perform random sampling and subsequent training of LM-BFF for 5 times and 1000 training steps, for each task.

— **PET (Schick and Schütze, 2021) :** It is a simple prompt-based few-shot fine-tuning approach where the training examples are converted into templates, and the `[MASK]` tokens are used to predict the verbalizer, which indicates the output label. To understand the role of using multiple prompts in robustness, we use PET to fine-tune models with different template-verbalizer pairs and ensemble their predictions during inference. The pairs used for different tasks are listed in Table 1. We train the model on four different sets of manual template-verbalizer pairs for 250 training steps.

Private Inference Implementation. We develop encryption functions with C++ and integrate the SEAL library for RNS-CKKS homomorphic encryption. To improve performance on Intel CPUs, we include HEXL acceleration. Our configuration adheres to homomorphic encryption standards, setting the polynomial degree to $N = 2^{16}$ and the ciphertext modulus to 1763 bits for 128-bit security. We set a multiplicative depth of $L = 35$ and a bootstrapping depth of $K = 14$, resulting in an effective multiplicative depth of 21.

4.2 Evaluation Results on GLUE and Adversarial GLUE Tasks

In Table 2, we present evaluation results on GLUE and AdvGLUE tasks, reporting metrics F1 score for QQP and accuracy for the other five tasks). BERT is used as the large pre-trained language model. For baselines LM-BFF and PET, we implement the same private ALBERT-xxlarge-v2 for fair comparison.

According to Table 2, we have the following experiment results :

— Compared with prompt ensembles without privacy preservation, SecPE exhibits almost no accuracy loss on GELU and AdvGELU benchmarks. This suggests that SecPE is capable of maintaining both high utility and robustness while providing privacy protection.

— Compared with the private inference of a single prompt template, SecPE has demonstrated better adversarial robustness than LM-BFF(Ciphertext).

4.3 Comparison on Arithmetic Reasoning Tasks

Self Consistency (Wang et al., 2022a) uses different prompt templates to generate a diverse set of reasoning paths, each reasoning path might lead to a different final answer, so we determine the optimal answer by marginalizing out the sampled reasoning paths using a voting verifier (aggregate-then-argmax) (Li et al., 2023b) to find the most consistent answer in the final answer set.

We implemented Self Consistency's privacy inference under the SecPE framework. The baseline we compare to is chain-of-thought prompting with greedy decoding (Wei et al., 2022). Compared with Self Consistency's inference results under plaintext, the accuracy of ciphertext inference is similar to it and much higher than the baseline. Figure 4 and 5 shows the performance on GSM8K and MultiArith with the different number of reasoning paths.

4.4 Efficiency Comparison

Figure 6 illustrates the efficiency comparison of SecPE with Phoenix (Jovanovic et al., 2022) under

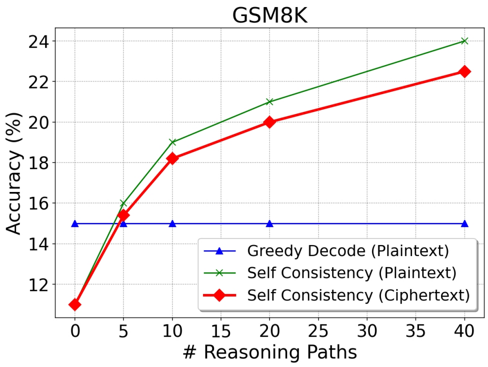

FIGURE 4 – Performance on GSM8K with the different number of reasoning paths.

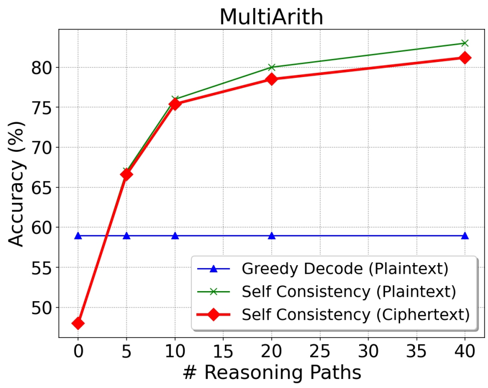

FIGURE 5 – Performance on MultiArith with the different number of reasoning paths.

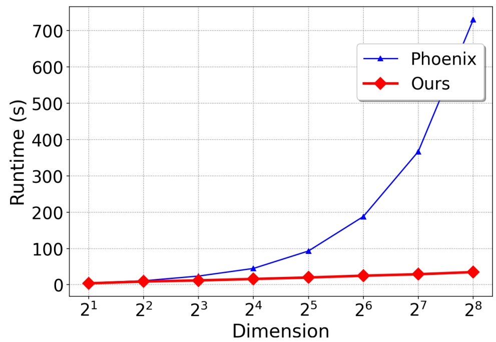

FIGURE 6 – Performance of `Argmax` on RNS-CKKS for different dimensions of input.

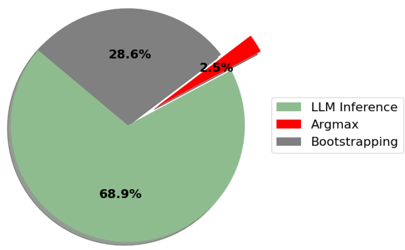

FIGURE 7 – Runtime breakdown.

different input dimensions. In particular, we focus on the essential `Argmax` operation, which incurs one of the major overheads of prompt ensemble under private inference. For an input length of n, Phoenix (Jovanovic et al., 2022) adopts a sequential comparison approach to obtain the sign bit, resulting in $(n+1)$ $Sign$ operations and $(n+1)$ ciphertext rotations. In contrast, `SecPE`'s Algorithm 1 only requires $(\log n + 1)$ $Sign$ operations and $(\log n + 1)$ ciphertext rotations. This significantly reduces the execution time, which is depicted in Figure 6. For the input length of 256, `SecPE` achieves $20.8\times$ speedup for `Argmax`.

Figure 7 shows the time distribution of different building blocks in `SecPE`. Due to the numerous non-linear operations (GELU, Softmax, LayerNorm) involved in LLM private inference, which require multiple bootstrapping, they contribute significantly to the overall overhead. We show that the `Argmax` computation accounts for only 2.5% of the total time. Therefore, `SecPE` incurs an additional cost of only 2.5% compared to private inference with LLM without prompt ensembling.

It indicates that while Prompt Ensembling necessitates multiple inference runs, this overhead is justified. Despite the additional computational cost, as visualized by the substantial slice of the pie chart allocated to LLM inference, the benefits of Prompt Ensembling cannot be overstated. The improved robustness and accuracy provided by multiple inferences, where different prompts are evaluated to derive a final answer, results in more reliable and accurate model performance. This benefit often outweighs the cost of increased inference time, making Prompt Ensembling a valuable technique in scenarios where high-quality predictions are paramount.

5 Conclusions

We propose `SecPE`, the first attempt to our best knowledge to jointly enable privacy-preserving and adversarial robustness for LLM inference. `SecPE` synergizes the strengths of private inference and prompt ensembling, previously explored in isolation, and overcomes inefficiency challenges incurred by a naive combination of existing techniques. Our extensive experiments have demonstrated that `SecPE` not only preserves high clean accuracy but also significantly bolsters robustness, all with a minimal efficiency overhead when compared to existing private inference methods. Therefore, `SecPE` manifests a satisfactory "accuracy-robustness-efficiency" tradeoff.

Limitations

Although our work can ensure the privacy and robustness of LLM, the privacy inference efficiency of LLM is low due to the efficiency of homomorphic encryption. Even for MPC-based privacy inference, communication traffic will bring a lot of overhead. In addition, Prompt Ensemble requires multiple inferences, which also improves our latency. In addition, SecPE only provides empirical robustness and does not extend the certified robustness.

Ethics Concern

Our effort to integrate privacy and robustness into LLM inference is a first step, and we're aware of the ethical weight it carries. While we strive to respect user privacy and enhance security, we recognize the complexity of these issues and welcome further insight and guidance from the community.

References

James Urquhart Allingham, Jie Ren, Michael W Dusenberry, Xiuye Gu, Yin Cui, Dustin Tran, Jeremiah Zhe Liu, and Balaji Lakshminarayanan. 2023. A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models. In *International Conference on Machine Learning*, pages 547–568. PMLR.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit : A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attentional mixtures of soft prompt tuning for parameter-efficient multi-task knowledge sharing. *arXiv preprint arXiv :2205.11961*, 3.

Hrishav Bakul Barua. 2021. Data science and machine learning in the clouds : A perspective for the future. *arXiv preprint arXiv :2109.01661*.

Jean-Philippe Bossuat, Christian Mouchet, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. 2021. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 587–617. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33 :1877–1901.

Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha. 2019. Low latency privacy preserving inference. In *International Conference on Machine Learning*, pages 812–821. PMLR.

Cecily Mauran. 2023. Samsung bans chatgpt, ai chatbots after data leak blunder. https://mashable.com/article/samsung-chatgpt-leak-leads-to-employee-ban. Accessed : 2023-05-02.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv :2107.03374*.

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. THE-X: Privacy-preserving transformer inference with homomorphic encryption. In *Findings of the Association for Computational Linguistics : ACL 2022*, pages 3510–3520, Dublin, Ireland. Association for Computational Linguistics.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2019. A full rns variant of approximate homomorphic encryption. In *Selected Areas in Cryptography–SAC 2018 : 25th International Conference, Calgary, AB, Canada, August 15–17, 2018, Revised Selected Papers 25*, pages 347–368. Springer.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017 : 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pages 409–437. Springer.

Jung Hee Cheon, Dongwoo Kim, and Duhyeong Kim. 2020. Efficient homomorphic comparison methods with optimal complexity. In *Advances in Cryptology–ASIACRYPT 2020 : 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 221–256. Springer.

Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. 2018. Faster cryptonets : Leveraging sparsity for real-world encrypted inference. *arXiv preprint arXiv :1811.09953*.

J Christoph, L Griebel, I Leb, I Engel, F Köpcke, D Toddenroth, H-U Prokosch, J Laufer, K Marquardt, and M Sedlmayr. 2015. Secure secondary use of clinical data with cloud-based nlp services. *Methods of information in medicine*, 54(03) :276–282.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv :2110.14168*.

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

In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. 2019. Diversity with cooperation : Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3723–3731.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Craig Gentry. 2009. *A fully homomorphic encryption scheme*. Stanford university.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets : Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR.

Kyoohyung Han and Dohyeong Ki. 2020. Better bootstrapping for approximate homomorphic encryption. In *Cryptographers' Track at the RSA Conference*, pages 364–390. Springer.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. 2022. Iron : Private inference on transformers. *Advances in Neural Information Processing Systems*, 35 :15718–15731.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know ? *Transactions of the Association for Computational Linguistics*, 8 :423–438.

Nikola Jovanovic, Marc Fischer, Samuel Steffen, and Martin Vechev. 2022. Private and reliable neural network inference. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1663–1677.

Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. 2018. {GAZELLE} : A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert : A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv :1909.11942*.

Eunsang Lee, Joon-Woo Lee, Junghyun Lee, Young-Sik Kim, Yongjune Kim, Jong-Seon No, and Woosuk Choi. 2022. Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In *International Conference on Machine Learning*, pages 12403–12422. PMLR.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv :2104.08691*.

Lei Li, Yongfeng Zhang, and Li Chen. 2023a. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1348–1357.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Yizhe Li, Yu-Lin Tsai, Chia-Mu Yu, Pin-Yu Chen, and Xuebin Ren. 2023c. Exploring the benefits of visual prompting in differential privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5158–5167.

Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. 2017. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 619–631.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. 2020. An ensemble of epoch-wise empirical bayes for few-shot learning. In *Computer Vision–ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 404–421. Springer.

Qian Lou and Lei Jiang. 2019. She : A fast and accurate deep neural network for encrypted data. *Advances in neural information processing systems*, 32.

Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. 2022. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215.

Oliver Masters, Hamish Hunt, Enrico Steffinlongo, Jack Crawford, Flavio Bergamaschi, Maria E Dela Rosa, Caio C Quini, Camila T Alves, Feranda de Souza, and Deise G Ferreira. 2019. Towards a homomorphic machine learning big data pipeline for the financial services sector. *Cryptology ePrint Archive*.

Cecily Mauran. 2023. Whoops, samsung workers accidentally leaked trade secrets via chatgpt. *Mashable [online]. Dostupné z : https ://mashable.com/article/samsungchatgpt-leak-details*.

Natasha Lomas. 2023. Italy orders chatgpt blocked citing data protection concerns. https://techcrunch.com/2023/03/31/chatgpt-blocked-italy/. Accessed : 2023-05-28.

Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. 2024. Bolt : Privacy-preserving, accurate and efficient inference for transformers. *IEEE Symposium on Security and Privacy (SP)*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1) :5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad : 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv :1606.05250*.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv :1608.01413*.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. *arXiv preprint arXiv :2010.13641*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv :2001.07676*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2007. Privacy protection in personalized search. In *ACM SIGIR Forum*, volume 41, pages 4–17. ACM New York, NY, USA.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Wenting Zheng Srinivasan, PMRL Akshayaram, and Popa Raluca Ada. 2019. Delphi : A cryptographic inference service for neural networks. In *Proc. 29th USENIX Secur. Symp*, pages 2505–2522.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8 :743–758.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot : Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv :2110.07904*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue : A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv :1804.07461*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue : A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv :2111.02840*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

Yongqin Wang, G Edward Suh, Wenjie Xiong, Benjamin Lefaudeux, Brian Knott, Murali Annavaram, and Hsien-Hsin S Lee. 2022b. Characterization of mpc-based private inference for transformer-based models. In *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 187–197. IEEE.

Yuxin Wang, Yuhan Chen, Zeyu Li, Zhenheng Tang, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. 2024. Towards efficient and reliable llm serving : A real-world workload study. *arXiv preprint arXiv :2401.17644*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35 :24824–24837.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv :1704.05426*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv :1904.02232*.

Junjielong Xu, Ziang Cui, Yuan Zhao, Xu Zhang, Shilin He, Pinjia He, Liqun Li, Yu Kang, Qingwei Lin, Yingnong Dang, et al. 2024. Unilog : Automatic logging via llm and in-context learning. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, pages 1–12.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv :1902.01718*.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019b. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv :1903.10972*.

Wenxuan Zeng, Meng Li, Wenjie Xiong, Tong Tong, Wen-jie Lu, Jin Tan, Runsheng Wang, and Ru Huang. 2023. Mpcvit : Searching for accurate and efficient mpc-friendly vision transformer with heterogeneous attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5052–5063.