CoTrust: Privacy-Preserving Collaboration Between Large and Small Language Models in Trusted Execution Environments

Zhenya Ma[1], **Tingyi Wang**[1], **Yongheng Deng**[1*], **Ziqing Qiao**[1], **Yinggui Wang**[2], **Tao Wei**[2], **Lei Wang**[2], **Ju Ren**[1]
[1]Department of Computer Science and Technology, Tsinghua University
[2] Ant Group, China
mzy23@mails.tsinghua.edu.cn dengyh1013@gmail.com

Abstract

Services powered by large language models (LLMs) provide powerful text generation capabilities, but accessing sensitive user inputs raises serious privacy concerns. Trusted Execution Environments (TEEs) provide a secure computation environment, enabling sensitive inputs to be safely processed. However, directly deploying high-capacity LLMs in TEEs is often prohibitively expensive due to computation and memory constraints. To reconcile privacy, efficiency, and generation quality, we propose CoTrust, a privacy-preserving collaborative inference framework that combines LLMs with small language models (SLMs) inside TEE. CoTrust uses multiple de-identified views to let the LLM produce a consensus scaffold capturing answer reasoning without exposing private information, which the SLM then grounds in the full input to generate the final response. Experiments on multiple question answering and summarization benchmarks show that CoTrust approaches the performance of unconstrained LLMs, outperforms existing privacy-preserving baselines, and maintains strong privacy protection, while remaining efficient in a TDX-based TEE implementation.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Zhang et al., 2022) have rapidly become core components of modern AI systems, powering personal assistants, productivity tools, and domain-specific applications in areas like healthcare, finance, and software engineering (Li et al., 2025; Yang et al., 2024). In real-world deployments, LLM-powered services process user inputs that frequently contain privacy-sensitive information, such as personally identifiable information (PII), proprietary business data, or domain-specific records. The processing of such inputs during inference inevitably introduces significant privacy risks. Therefore, ensuring the privacy of user inputs is essential for the safe and trustworthy deployment of LLM-based services.

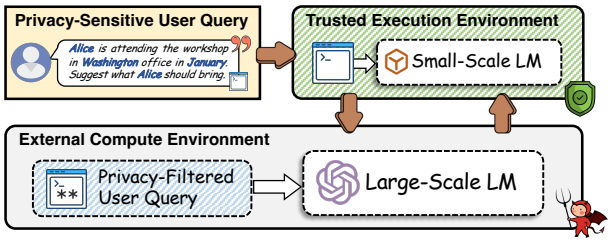

Figure 1: A collaborative inference framework: An SLM processes user inputs inside TEE to preserve privacy, while an LLM outside the trust boundary operates on a privacy-filtered view to harness its powerful capabilities.

Trusted execution environments (TEEs) (Alves, 2004; Intel Corporation, 2021; McKeen et al., 2013) provide a hardware-based isolation mechanism that guarantees the confidentiality and integrity of data and computations. Unlike traditional cryptographic techniques or access control policies, TEEs allow sensitive information to be processed directly in plaintext within a secure environment, offering strong privacy protection. Despite these advantages, TEEs are inherently constrained in memory and computational throughput, which limits the size and complexity of models that can be executed entirely within the TEE. Therefore, deploying modern LLMs with billions of parameters inside a TEE to provide powerful services remains impractical. These limitations pose a fundamental challenge: *how can one leverage the powerful capabilities of modern LLMs while still fully preserving the privacy of sensitive user inputs?*

Addressing this challenge is non-trivial. Existing approaches often protect sensitive information by masking or obfuscating portions of the input (Chen et al., 2023; Shen et al., 2024; Qiang et al., 2023) before sending private inputs to the LLM. Although

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 21423–21440

such methods reduce privacy exposure, they inevitably sacrifice some valuable information from user inputs, which compromises the generation performance of LLMs. To resolve this tension, we propose a novel collaborative framework as illustrated in Figure 1: a small-scale language model (SLM) within the TEE processes the full input to preserve sensitive information, while a larger LLM outside the trust boundary operates on a privacy-filtered version to exploit its powerful reasoning and generation capabilities. In this way, our framework preserves the full-fidelity content of user inputs while simultaneously leveraging the powerful capabilities of larger LLMs. In this paradigm, the key challenge is *how to orchestrate the SLM inside the TEE, which has full access to user inputs, with the large LLM outside the TEE, which possesses powerful capabilities but operates on a privacy-filtered version, to achieve efficient, high-quality, privacy-preserving collaborative inference.*

To address this challenge, we introduce **CoTrust**, a privacy-preserving collaborative inference framework that decouples access to sensitive information from high-capacity reasoning and generation. The key idea of CoTrust is to let the external LLM produce a de-identified scaffold that captures high-level reasoning and answer structure, while delegating all privacy-sensitive instantiation to a TEE-resident SLM with full access to the original input. To obtain such a scaffold without exposing private information, CoTrust provides the LLM with multiple de-identified views of the same private input, encouraging the LLM to generate complementary candidate references under privacy constraints. These references are then canonicalized and induced into a single consensus scaffold that distills the LLM's generative capability and retains strong priors over the answer's structure and reasoning across diverse de-identified views. Finally, this scaffold is brought back to the TEE and used to guide the SLM to regenerate the final answer by following the scaffold's reasoning cues while grounding them in the full-fidelity private input.

Extensive experiments demonstrate that CoTrust delivers high generation quality under strict privacy constraints, closely approaching direct LLM inference while keeping all sensitive information confined within the TEE. CoTrust is resource-efficient in a TDX-based TEE implementation; it significantly reduces the overhead of running a full LLM within the TEE, while incurring only modest latency and memory overhead compared to the SLM-only baseline. Comprehensive privacy evaluations demonstrate that CoTrust preserves strong resistance to privacy-content extraction, offering privacy protection comparable to single-masking approaches while enabling cross-model collaboration.

2 Background and Related Work

2.1 Trusted Execution Environments

Trusted Execution Environments (TEEs) provide hardware-enforced isolation that protects sensitive code, data, and computation from external adversaries, including compromised operating systems and privileged software. This isolation establishes a trusted execution boundary in which memory contents and control flow are inaccessible to untrusted software. Beyond isolated execution, modern TEEs secure the full data lifecycle. Sensitive inputs can be delivered through secure communication channels that ensure confidentiality and integrity during transmission, and once inside the TEE, memory encryption and access control prevent host-side inspection or tampering (Gu et al., 2025). TEEs also support remote attestation, enabling external parties to verify the integrity of the trusted execution before provisioning secrets or private inputs.

Widely deployed TEE implementations include Intel SGX (McKeen et al., 2013), ARM TrustZone (Alves, 2004), AMD SEV/SEV-SNP (AMD, 2016), and Intel TDX (Intel Corporation, 2021; Cheng et al., 2024). In particular, TDX provides transparent memory encryption, strong isolation from the host and VMM, and hardware-backed attestation (Zhan et al., 2025), extending confidential computing to full virtual machines known as Trusted Domains. Despite these strong guarantees, TEEs impose strict constraints on memory and computation, making it impractical to execute full-scale LLMs inside the trust boundary. Prior works (Zhang et al., 2024d; Shen et al., 2022; Hashemi et al., 2021) primarily use TEEs to protect lightweight DNN inference or sensitive data, leaving open the challenge of securely leveraging high-capacity LLMs without exposing private inputs—an issue our work directly addresses.

2.2 Privacy Protection for User Inputs to LLMs

A common paradigm for protecting input privacy to LLMs is content sanitization, where sensitive spans are first detected—using predefined patterns, rule-based systems, or lightweight classifiers—and

then obfuscated through masking, substitution, or anonymization. Representative techniques (Chen et al., 2023; Shen et al., 2024; Qiang et al., 2023) include lexical token masking, pseudonymization, learned anonymization–deanonymization pipelines, and perturbation mechanisms inspired by local differential privacy (Yue et al., 2021). These methods are simple and practical, and have been widely adopted in real systems. However, such methods may provide limited protection: they often preserve enough contextual and structural cues for adversaries to infer sensitive information, and may miss domain-specific semantic nuances. Moreover, researchers go beyond simple substitution and explore structural transformations that incorporate perturbation, collaborative candidate selection, and hierarchical inference (e.g., embedding the query among decoys (Yao et al., 2024), providing multiple candidate tokens (Zhang et al., 2024b), or separating public and private prompt components via device-cloud inference (Zhang et al., 2024a)) to balance privacy and usability better. Despite their practicality, these methods may generalize poorly across tasks, over-sanitize prompts, and discard critical semantic cues, significantly degrading LLM response quality.

2.3 Collaborative Inference between LLMs and SLMs

Collaborative inference between LLMs and SLMs (Yao et al., 2022; Lv et al., 2022; Gu et al., 2024) has emerged as a promising approach to combine the high reasoning capacity of LLMs with the efficiency and personalization of SLMs (Labrak et al., 2024; Xiang et al., 2023). By jointly leveraging LLMs and SLMs, these methods can accelerate inference, adapt to local contexts without sacrificing generation quality. Most existing approaches (Lin et al., 2025; Zhang et al., 2024c; Yu et al., 2024) focus on device–cloud scenarios, where LLMs reside in the cloud and SLMs operate at the mobile/edge device to provide low-latency, task-specific processing. While effective for performance optimization, these designs typically focus on optimizing the generation quality or efficiency, which limits their applicability in privacy-sensitive contexts. In contrast, CoTrust considers a TEE scenario, where all sensitive input processing occurs on a TEE-resident SLM, and the LLM only accesses de-identified views and provides generation guidance for the SLM. This setup enables the SLM to ground its generation in the full-fidelity private input, while still benefiting from the LLM's guidance. By doing so, CoTrust achieves privacy-preserving cross-model collaboration, combining the advantages of LLM reasoning and SLM efficiency without exposing sensitive data. To the best of our knowledge, this is the first work to enable privacy-preserving LLM–SLM collaboration in TEE settings, providing both high-quality generation and strong privacy guarantees.

3 Threat Model and Goal

We consider a deployment setting where user input may contain sensitive or personally identifiable information (PII). The SLM, PII detection and de-identification logic within the TEE are assumed to be fully trusted. Only de-identified inputs produced from TEE are sent to the external environment via a secure communication channel. The external environment hosting the LLM is treated as honest-but-curious: it executes inference faithfully, but may attempt to inspect or extract sensitive information from any artifacts observable outside the TEE. Under this threat model, our goal is to enable collaborative inference between an SLM deployed in the trusted environment and an LLM deployed in an untrusted environment, such that private user inputs can be processed efficiently with privacy protection while still achieving high generation quality.

4 Methodology

4.1 Overview

CoTrust orchestrates collaborative inference between an SLM within a TEE and an LLM deployed in an external untrusted environment, so that private user inputs can be fully utilized while still benefiting from the LLM's strong generative capabilities. Inside the TEE, CoTrust transforms each private input into multiple complementary de-identified views, each masking sensitive entities in a different yet valid manner while preserving different subsets of non-sensitive semantics. These de-identified views are sent to the external LLM, which independently generates candidate references. Since the views mask entities differently, the candidate references contain view-specific placeholders and are not directly aligned. CoTrust brings them back to the TEE, reconciles them into canonicalized references, and returns these to the external LLM, which induces them into a consensus de-identified scaffold. This scaffold distills the LLM's strong generative capabilities across diverse views into a

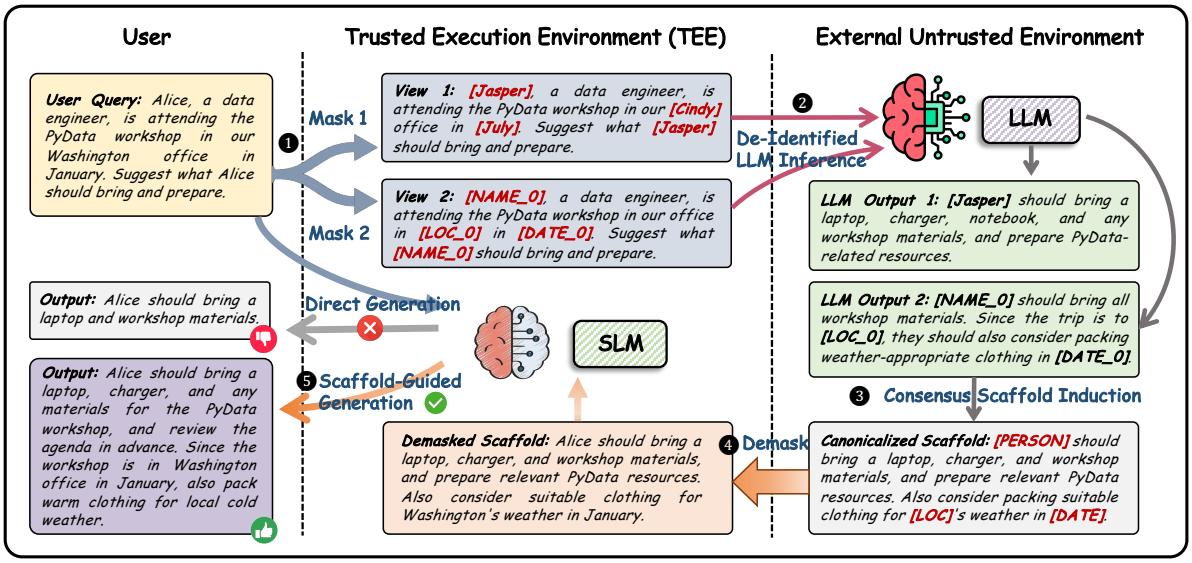

Figure 2: Overview of CoTrust. The SLM within TEE processes full private input. The external LLM contributes reasoning and generation on de-identified views; outputs are induced into a consensus and privacy-preserving scaffold to guide final answer generation.

coherent, privacy-preserving reference. Finally, the scaffold is instantiated with the private entities and combined with the raw user input to guide the SLM in the TEE to produce the final answer. In this way, CoTrust allows the LLM to contribute rich generative capability, while all privacy-sensitive content is processed strictly within the TEE. A schematic overview of CoTrust is shown in Figure 2.

4.2 Multi-View De-Identified LLM Inference

When a user submits a private input, it often contains sensitive information such as names, organizations, or locations. To leverage the external LLM outside the TEE, a straightforward approach might be to simply mask these entities using an off-the-shelf masking strategy before sending the input to the LLM. However, a single fixed masking policy is often brittle: it may discard different subsets of task-relevant non-sensitive semantics (e.g., entity roles, temporal cues, and discourse relations), and can also behave inconsistently on ambiguous mentions (e.g., masking *Washington* as a person name in one context but as a location in another), leading to unstable and lower-quality LLM outputs.

To overcome this challenge, CoTrust first performs *Multi-View De-Identified Input Construction*. Inside the TEE, for each private user input x, it applies multiple masking or transformation functions $\{T^{(1)}, \ldots, T^{(K)}\}$ to construct several complementary de-identified views of the input:

$$v^{(k)} = T^{(k)}(x), \quad k = 1, \ldots, K. \tag{1}$$

Each $T^{(k)}$ masks sensitive entities in a different yet valid manner, while preserving distinct subsets of non-sensitive semantics. For example, one view may replace name *Alice* with a generic type token (*[NAME_0]*), while another may use consistent pseudonyms (*Jasper*) to better preserve conversational roles. During this process, the TEE records a de-identification mapping $\mathcal{M}$ that links each PII span e in x to its masked form $\mathcal{M}_k(e)$ in each view:

$$\mathcal{M}_k(e) = v^{(k)}(e), \quad k = 1, \ldots, K. \tag{2}$$

These multiple views are safely transmitted to the external LLM for *De-Identified LLM Inference*, which treats each view $v^{(k)}$ independently and produces a candidate reference for each:

$$r^{(k)} = \mathrm{LLM}(v^{(k)}), \quad k = 1, \ldots, K. \tag{3}$$

From the LLM's perspective, it is akin to reading several drafts of the same narrative, each preserving different non-sensitive cues while never exposing the underlying private content. This naturally produces a diverse set of candidate references, capturing different lines of reasoning or highlighting complementary information about the input.

4.3 Consensus Scaffold Induction

The *Multi-View De-Identified LLM Inference* stage produces a set of candidate references that *collectively contain rich but fragmented knowledge* about the user's private input. However, these references are *not comparable or combinable*. Because each de-identified view obfuscates entities differently, the resulting outputs refer to the same underlying

entities using incompatible placeholders. For instance, the same person *Alice* may appear as *Jasper* in one reference and as *NAME_0* in another. A natural strategy is to bring these references back into the TEE and let the SLM directly induce multiple references after demasking to produce the final answer. However, we empirically find that this is ineffective due to the limited capacity of the SLM. In practice, the SLM struggles to effectively reconcile inconsistencies across multiple references and to synthesize a high-quality response.

Therefore, CoTrust turns to the external LLM for reference induction, as it is substantially more capable of synthesizing multiple candidate references into coherent and high-quality guidance. However, directly inducing the references $r^{(k)}$ outside the TEE is not straightforward. Because the references are generated from different de-identified views, they use view-specific placeholders and encode incompatible representations of the same underlying entities. Without resolving these inconsistencies through *Cross-view Entity Canonicalization*, the external LLM cannot reliably identify entities across references, which undermines effective induction.

Crucially, *Cross-view Entity Canonicalization* requires access to the de-identification mapping, which is privacy-sensitive and must remain inside the TEE. CoTrust therefore separates canonicalization from induction: it first uses the TEE to reconcile view-specific placeholders into a shared canonical masking space, producing canonicalized references. These references are then sent to the external LLM for *Canonical-view Scaffold Induction*, which can effectively aggregate them and induce a single consensus *de-identified scaffold* S: a canonical reference that distills the LLM's strong generative capability across diverse de-identified views and turns the LLM's output diversity into a single coherent and informative reference template.

4.4 Scaffold-Guided Private SLM Inference

In the final stage, CoTrust returns to the TEE to produce the final answer. By this point, the external LLM has distilled its high-capacity reasoning and generation into a de-identified scaffold S, but instantiating private identities remains a critical step that must be performed within the TEE. Accordingly, CoTrust brings S back into the TEE and deterministically instantiates it using the canonical de-identification mapping maintained within the TEE. This yields a privacy-aware scaffold S^{priv}, in which all placeholders are resolved to their original entities without exposing any privacy beyond the TEE boundary. The TEE then pairs S^{priv} with the original full-fidelity private input x and feeds them jointly to the TEE-resident SLM:

$$\hat{y} = \text{SLM}_{\text{TEE}}(x, S^{\text{priv}}). \tag{4}$$

Crucially, the SLM is not asked to perform high-level reasoning on its own. Instead, it follows the scaffold distilled by the external LLM, using it as a strong prior over the answer's organization and reasoning, while leveraging direct access to the full-fidelity private input to inject the correct user-specific details. In this way, CoTrust confines all privacy-sensitive instantiation and refinement to the TEE, while allowing the external LLM's generative capability to shape the final response indirectly. At no point are concrete identities or sensitive attributes revealed outside the trust boundary.

5 Experiments

Dataset. We evaluate CoTrust on four datasets that naturally contain privacy-sensitive information. Specifically, we use *MedQA* (Jin et al., 2020) and the *Stanford Question Answering Dataset (SQuAD)* (Rajpurkar et al., 2018) for question answering (QA), and *SAMSum* (Gliwa et al., 2019) and *DialogSum* (Chen et al., 2021) for dialogue summarization. These datasets involve personal names, locations, and other identifiable entities, making them suitable for evaluating privacy-preserving inference under realistic conditions.

Models and Masking Methods. We report results under three mixed-scale SLM/LLM pairings: *Qwen3-4B* (SLM) with *Qwen3-8B* (LLM) (Yang et al., 2025), *Ministral-3B* (SLM) with *Ministral-8B* (LLM) (Mistral AI, 2025), and *Llama-3.2-3B* (SLM) with *Llama-3.1-8B* (LLM) (Grattafiori et al., 2024). We set K=2 and apply two widely used off-the-shelf masking tools within the TEE: Spacy (Honnibal et al., 2020) and Presidio (Microsoft, 2018), to identify privacy-sensitive spans and apply de-identification transformations.

Evaluation Metric. We evaluate CoTrust from three perspectives: generation quality, efficiency, and privacy protection. For generation evaluation on QA tasks, we report *Accuracy*, *Recall*, and *F1* for *MedQA*. For *SQuAD*, we report *Exact Match (EM)* and *F1* following prior work (Rajpurkar et al., 2018). For summarization tasks, we use *Qwen3-Max* (Qwen Team, 2025) as an LLM-as-a-judge to

Method	MedQA			SQuAD		DialogSum	SAMSum
	Accuracy	Recall	F1	EM	F1	QM-Score	QM-Score
Qwen3-4B/8B							
SLM-Only	58.29	58.07	58.37	45.93	56.71	5.802	5.894
Single-Masked-LLM	58.44	58.58	58.52	49.64	59.64	5.904	5.549
Multi-Masked-LLM	60.69	61.17	61.05	47.46	56.48	6.106	5.747
LLM-Guided-SLM	60.42	59.84	59.93	42.07	51.87	5.974	6.003
LLM-w/o-Mask	**62.84**	**62.82**	**62.93**	51.81	62.84	6.008	**6.098**
CoTrust (ours)	61.51	61.27	61.40	**54.68**	**63.75**	**6.144**	6.077
Ministral-3B/8B							
SLM-Only	54.42	54.46	54.15	60.08	66.74	5.932	6.049
Single-Masked-LLM	59.23	59.04	58.77	55.13	61.83	5.872	5.623
Multi-Masked-LLM	59.94	59.75	59.47	63.93	71.87	5.864	6.031
LLM-Guided-SLM	59.07	59.01	58.50	56.46	62.70	6.126	5.997
LLM-w/o-Mask	**61.74**	**61.53**	**61.56**	62.71	69.53	6.126	**6.330**
CoTrust (ours)	60.02	59.82	59.49	**68.22**	**74.48**	**6.168**	6.110
Llama3.2-3B/3.1-8B							
SLM-Only	53.02	53.07	53.03	42.62	47.85	5.246	5.601
Single-Masked-LLM	56.64	55.63	56.22	36.60	43.50	5.301	5.477
Multi-Masked-LLM	57.19	56.12	56.70	43.92	49.60	5.486	5.438
LLM-Guided-SLM	57.42	56.77	57.22	42.48	**51.55**	5.794	5.660
LLM-w/o-Mask	**59.78**	**59.17**	**59.62**	**45.87**	50.72	5.542	**5.880**
CoTrust (ours)	57.89	57.20	57.66	45.63	50.99	**5.826**	5.698

Table 1: Generation quality comparison across multiple QA and summarization tasks. CoTrust consistently outperforms privacy-preserving baselines and achieves performance close to, or even surpassing, the unmasked LLM upper bound. We **bold** the best result and underline the second.

measure summary quality, denoted as *QM-Score* (on a 1–7 scale). For efficiency evaluation, we measure *Peak TEE memory Footprint* and *End-to-End Inference Latency*. For privacy protection evaluation, we use *Qwen3-Max* as both the privacy-content extractor and evaluator of privacy protection to quantify leakage risks using *Extract Success Rate (ESR)* and *Privacy Protection Score (PPS)*.

Baselines. We consider four privacy-preserving baselines that strictly satisfy the privacy constraints, where no raw private information is ever exposed to the LLM outside the TEE. (1) *SLM-Only*: the private user input is fed into the SLM inside the TEE, and the SLM directly generates the final output. (2) *Single-Masked-LLM*: the private user input is first masked by a specific masking algorithm, and the privacy-filtered input is sent to the LLM for generation. The LLM output is demasked inside the TEE and directly returned as the final answer. (3) *Multi-Masked-LLM*: the private user input is first masked by multiple masking algorithms, and the derived multiple de-identified views of the input are processed independently by the LLM. The LLM then induces these outputs into a single de-identified response, which is demasked and returned as the final answer. (4) *LLM-Guided-SLM*: the private user input is first masked by a specific masking algorithm, and the privacy-filtered input is sent to the LLM for generation. The resulting LLM output is treated as a hint to guide the SLM in generating the final response within the TEE. In contrast, we also include: (5) *LLM-w/o-Mask*: the private user input is directly sent to the LLM for generation without any masking or protection. While this setting is expected to achieve the best generation quality in theory, it fundamentally violates the privacy constraints. We therefore include it only as an oracle **upper bound** for performance comparison.

We provide more details of datasets in Appendix B, models in Appendix C, system configurations, experimental setups as well as evaluation metrics in Appendix D, and prompts in Appendix E.

5.1 Generation Quality Evaluation

We evaluate the generation quality of CoTrust on both QA and summarization tasks under three SLM/LLM pairings, comparing against five baselines. As shown in Table 1, across all datasets, CoTrust consistently outperforms existing privacy-preserving baselines and achieves performance closest to the oracle upper bound *LLM-w/o-Mask*.

Compared with the privacy-preserving baselines, CoTrust delivers the best results in most settings. It substantially outperforms *SLM-Only*, demonstrat-

Method	Latency (s)	Memory (GB)	F1	EM
LLM-in-TEE	32.49	21.92	62.84	51.81
SLM-Only	15.65	10.85	56.71	45.93
CoTrust (ours)	21.47	10.97	63.75	54.68

Table 2: Efficiency comparison on *SQuAD* using the *Qwen3-4B/8B* pairing. We report end-to-end inference latency (seconds per input) and peak TEE memory footprint (GB), as well as their F1 and EM scores.

ing that de-identified collaboration with an external LLM effectively compensates for the limited capacity of small models. Compared to masked LLM baselines (both single- and multi-masked), CoTrust yields higher task performance, indicating that masking alone—regardless of granularity—fails to preserve sufficient task-relevant information. In addition, CoTrust consistently surpasses *LLM-guided SLM*, showing that the scaffold induced from multi-view guidance is more effective than single-view LLM guidance. Overall, the results demonstrate the effectiveness of CoTrust in enabling collaborative inference between large and small language models under privacy constraints.

Relative to the oracle *LLM-w/o-Mask*, CoTrust incurs only a small performance gap on most benchmarks, despite never exposing raw private inputs outside the TEE. In some cases—particularly on *SQuAD* and *DialogSum*—CoTrust even surpasses *LLM-w/o-Mask*. We attribute this to the multi-view de-identification and scaffold induction mechanism: exposing the LLM to multiple complementary privacy-filtered views can reduce spurious correlations and encourage robust generation, whereas directly feeding raw inputs may amplify noise or overfit to sensitive but task-irrelevant details.

Overall, these results show that CoTrust achieves near-oracle generation quality while strictly preserving privacy, outperforming existing privacy-preserving inference approaches across both QA and summarization tasks.

5.2 Efficiency Evaluation

We evaluate the efficiency of CoTrust on *SQuAD* using the *Qwen3-4B/8B* pairing. We report two practical efficiency metrics: *end-to-end inference latency* (seconds per input) and *peak TEE memory footprint* (GB), both of which are measured inside a TDX-based TEE implementation. We compare CoTrust with two representative privacy-preserving alternatives: (i) hosting the LLM inside the TEE for inference, and (ii) an *SLM-Only* inference fully within the TEE.

As shown in Table 2, directly hosting the LLM within the TEE is highly resource-intensive, with 32.49s latency and 21.92GB memory footprint. By contrast, CoTrust achieves a substantial reduction in computation and resource usage, with a latency of 21.47s and a peak memory footprint of 10.97GB, corresponding to a 33.92% reduction in latency and a 49.96% reduction in memory usage. Compared with the *SLM-Only* setting, CoTrust introduces a modest latency overhead (21.47s vs. 15.65s) while maintaining nearly identical memory usage (10.97 GB vs. 10.85 GB). Importantly, this slight increase in latency is offset by a marked improvement in generation quality, as demonstrated in the F1 and EM scores (63.75 vs. 56.71, 54.68 vs. 45.93). These results indicate that CoTrust offers a practical efficiency–quality trade-off: it avoids the prohibitive costs of running a full LLM in the secure environment while preserving the strong generation quality enabled by collaborative inference.

5.3 Privacy Protection Evaluation

CoTrust relies on multiple de-identified views to guide the external LLM, while keeping all sensitive information fully inside the TEE. All computations inside the TEE are assumed to fully protect sensitive information. To ensure that this multi-view strategy does not compromise privacy, we evaluate the potential leakage from the masked texts generated by CoTrust, comparing them against single-mask baselines (Spacy and Presidio). Following prior work (Ma et al., 2025), we employ *Qwen3-Max* both as a privacy-content extractor and as an evaluator of privacy protection, reporting three widely used metrics: *Privacy Protection Score* (PPS; higher is better), *Extraction Success Rate* (ESR; lower is better), and *Privacy Enhancement* EΔ (higher is better), computed as the difference between the privacy content ratio and ESR.

The results shown in Table 3 demonstrate that, across all datasets, CoTrust consistently achieves high PPS and low ESR, staying close to the single-masking baselines. On QA tasks (MedQA and SQuAD), CoTrust maintains a PPS above 0.82 with an ESR below 2%. A similar trend is observed on summarization benchmarks (DialogSum and SAMSum), where CoTrust consistently yields PPS values around 0.90 and ESR values around or below 1.35%. Compared to single-mask baselines, CoTrust incurs only a minor absolute reduction in PPS, ranging from approximately 0.003 to 0.045 in most cases, while the increase in ESR remains limited, typically within 0.03% to 0.4% across dif-

Method	MedQA			SQuAD			DialogSum			SAMSum		
	PPS↑	ESR(%)↓	EΔ (%)↑	PPS↑	ESR(%)↓	EΔ (%)↑	PPS↑	ESR(%)↓	EΔ (%)↑	PPS↑	ESR(%)↓	EΔ (%)↑
Spacy	0.8251	1.47	7.07	0.9059	1.65	18.12	0.9111	0.56	6.04	0.9236	1.09	16.56
Presidio	0.8678	1.73	6.81	0.8715	1.83	17.94	0.9035	0.73	5.87	0.9169	1.06	16.59
CoTrust	0.8222	1.76	6.78	0.8617	1.93	17.84	0.8979	0.95	5.65	0.9117	1.35	16.3

Table 3: Privacy protection performance of CoTrust compared with single-mask baselines (Spacy and Presidio).

	SQuAD-EM	SQuAD-F1
CoTrust	**54.68**	**63.75**
Single-Mask LLM	42.07	51.87
Single-Mask LLM + TTS	50.43	58.91
w.o. Canonicalization	49.84	58.76
Single-Candidate Scaffold	50.87	59.67
Induction with SLM	48.68	58.19
w.o. Scaffold	45.93	56.71
Multi-Candidate Scaffold	53.62	61.67

Table 4: Ablation results on SQuAD with Qwen3-4B/8B. Removing or modifying any stage of CoTrust (multi-view LLM inference, scaffold induction, or scaffold-guided SLM inference) results in noticeable drops in EM and F1, confirming that all stages are essential for high-quality, privacy-preserving inference.

ferent tasks. The resulting privacy enhancement metric EΔ decreases by less than 0.4% in most cases, indicating that the overall privacy leakage risk remains largely unchanged. Overall, these results suggest that CoTrust does not significantly compromise privacy preservation comparable to single-masking methods, while enabling higher generation quality.

5.4 Ablation Study

Table 4 reports ablation results on SQuAD with Qwen3-4B/8B, designed to isolate the contribution of each stage in our framework. The full CoTrust achieves the best performance (54.68% EM / 63.75% F1), while all ablated variants exhibit clear degradation, indicating that the three stages are complementary and jointly necessary for high-quality privacy-preserving inference.

Effect of Multi-View De-Identified LLM Inference. We first replace multiple masked views with a single masked input to the LLM (*Single-Mask LLM*). This leads to the most severe degradation (–12.61% EM / –11.88% F1), suggesting that a single masked view is brittle and fails to capture sufficient task-relevant semantics under privacy constraints. We further test whether test-time scaling (TTS) can compensate for the lack of multi-view diversity. Although TTS (*Single-Mask LLM + TTS*) improves over the single-mask setting, it still underperforms the full CoTrust by a large margin (–4.25% EM / –4.84% F1). These results indicate that scaling alone cannot replace the structured diversity provided by multiple de-identified views.

Effect of Consensus Scaffold Induction. Next, we study the effect of how the scaffold is induced from LLM outputs. Directly inducing a scaffold from unaligned candidates (*w.o. Canonicalization*) significantly degrades performance (49.84% EM / 58.76% F1), indicating that canonicalization is critical for stabilizing downstream reasoning. Replacing consensus induction with selecting a single candidate as the scaffold (*Single-Candidate Scaffold*) yields only marginal improvement (50.87% / 59.67%), suggesting that relying on any individual LLM output is unreliable. Finally, moving the induction process into the SLM (*Induction with SLM*) further degrades performance (48.68% / 58.19%), highlighting the necessity of LLM-level reasoning for robust scaffold construction.

Effect of Scaffold-Guided Private SLM Inference. Removing the scaffold entirely (*w.o. Scaffold*) results in a substantial performance drop (–8.75% EM / –7.04% F1), demonstrating that the scaffold provides essential structure and hints for the SLM under limited capacity. Simply feeding multiple LLM candidates to the SLM without consensus induction (*Multi-Candidate Scaffold*) partially recovers performance but still falls short of the full system (–1.06% EM / –2.08% F1). This confirms that the benefit of the scaffold stems not from raw candidate diversity, but from a distilled and aligned representation.

6 Conclusion

We present CoTrust, a framework enabling LLM–SLM collaboration under strict privacy constraints. By combining multi-view de-identified LLM scaffolds with TEE-based SLM inference, CoTrust delivers high-quality generation across QA and summarization tasks, outperforms existing privacy-preserving baselines, and approaches unconstrained LLM performance. Its TDX-based

implementation demonstrates practical efficiency, while privacy evaluations confirm strong protection of sensitive information.

7 Acknowledgement

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62402267, 62432004, Ant Group, and a grant from the Guoqiang Institute, Tsinghua University.

Limitations

First, CoTrust relies on PII detection to identify privacy-sensitive spans before constructing de-identified views. Although we employ multiple complementary masking strategies (*multi-view de-identification*) to reduce brittleness compared with a single-masking method and our privacy protection experiments empirically demonstrate effective protection (low ESR and high PPS), the overall privacy guarantee can still be impacted by detection errors (e.g., missed or misclassified spans) or domain-specific PII patterns that are not well covered by the detector. Improving the robustness of PII detection and masking under diverse domains remains an important direction. Second, our threat model assumes the TEE as the trusted computing base (TCB) and does not consider side-channel attacks that may compromise TEEs. While some side-channel attacks may threaten TEE confidentiality (Yuan et al., 2025, 2024; Qiu et al., 2019), mitigating such vectors is orthogonal to our focus on privacy-preserving collaborative inference and is out of scope for this paper.

Ethical Considerations

Our work aims to enable high-quality, privacy-preserving LLM inference by ensuring that privacy-sensitive input content remains inside the TEE while still benefiting from the strong generation capability of LLMs outside the trust boundary. In our privacy protection evaluation, CoTrust achieves a low *Extract Success Rate (ESR)* and a high *Privacy Protection Score (PPS)*, suggesting that sensitive information is difficult to recover from the content exposed outside the TEE under the threat model considered in this paper. Besides, we conduct our evaluation on public datasets and open-source models, and do not collect, store, or release any real user inputs or proprietary private data, thereby demonstrating effectiveness without introducing additional disclosure risk.

References

Alibaba Group. 2025. Alibaba group: Technology and innovation.

T. Alves. 2004. ARM Security Technology Building a Secure System using TrustZone Technology. White paper, ARM Limited. Available from https://developer.arm.com/documentation/PRD29-GENC-009492/latest/.

AMD. 2016. AMD Secure Encrypted Virtualization (SEV).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *arXiv preprint arXiv:2309.03057*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Pau-Chen Cheng, Wojciech Ozga, Enriquillo Valdez, Salman Ahmed, Zhongshu Gu, Hani Jamjoom, Hubertus Franke, and James Bottomley. 2024. Intel tdx demystified: A top-down approach. *ACM Computing Surveys*, 56(9):1–33.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Zhongshu Gu, Enriquillo Valdez, Salman Ahmed, Julian James Stephen, Michael Le, Hani Jamjoom, Shixuan Zhao, and Zhiqiang Lin. 2025. Nvidia gpu confidential computing demystified. *arXiv preprint arXiv:2507.02770*.

Hanieh Hashemi, Yongqin Wang, and Murali Annavaram. 2021. Darknight: An accelerated framework for privacy and integrity preserving deep learning using trusted hardware. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 212–224.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Hugging Face. 2019. knkarthick/samsum dataset card.

Intel Corporation. 2021. Intel® Trust Domain Extensions (Intel® TDX) Architecture Specification. Technical report, Intel Limited.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Kaggle. 2025. Dialog summarization (dialogsum).

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Meiziniu Li, Dongze Li, Jianmeng Liu, Jialun Cao, Yongqiang Tian, and Shing-Chi Cheung. 2025. Enhancing differential testing with llms for testing deep learning libraries. *ACM Transactions on Software Engineering and Methodology*.

Zheng Lin, Guanqiao Qu, Qiyuan Chen, Xianhao Chen, Zhe Chen, and Kaibin Huang. 2025. Pushing large language models to the 6g edge: Vision, challenges, and opportunities. *IEEE Communications Magazine*, 63(9):52–59.

Chengfei Lv, Chaoyue Niu, Renjie Gu, Xiaotang Jiang, Zhaode Wang, Bin Liu, Ziqi Wu, Qiulin Yao, Congyu Huang, Panos Huang, Tao Huang, Hui Shu, Jinde Song, Bin Zou, Peng Lan, Guohuan Xu, Fei Wu, Shaojie Tang, Fan Wu, and Guihai Chen. 2022. Walle: An end-to-end, general-purpose, and large-scale production system for device-cloud collaborative machine learning. *Preprint*, arXiv:2205.14833.

Hongru Ma, Wenpeng Lu, Yanjie Liang, Tianyi Wang, Qi Zhang, Yingjie Zhu, and Jiasheng Si. 2025. Alsa: Context-sensitive prompt privacy preservation in large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2042–2053.

Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. 2013. Innovative instructions and software model for isolated execution. *Hasp@ isca*, 10(1).

Meta Platforms, Inc. 2025. Meta ai: Building the future of artificial intelligence.

Microsoft. 2018. Context aware, pluggable and customizable pii de-identification service for text and images. GitHub repository.

Mistral AI. 2025. Mistral ai — open and efficient ai models.

Jipeng Qiang, Kang Liu, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Parals: Lexical substitution via pretrained paraphraser. *arXiv preprint arXiv:2305.08146*.

Pengfei Qiu, Dongsheng Wang, Yongqiang Lyu, and Gang Qu. 2019. Voltjockey: Breaking sgx by software-controlled voltage-induced hardware faults. In *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pages 1–6. IEEE.

Qwen Team. 2025. Qwen: Official open source large language models by alibaba.

Pranav Rajpurkar. 2025. The stanford question answering dataset (squad) explorer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018,*, pages 784–789.

Tianxiang Shen, Ji Qi, Jianyu Jiang, Xian Wang, Siyuan Wen, Xusheng Chen, Shixiong Zhao, Sen Wang, Li Chen, Xiapu Luo, Fengwei Zhang, and Heming Cui. 2022. {SOTER}: Guarding black-box inference for general neural networks at the edge. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pages 723–738.

Zhili Shen, Zihang Xi, Ying He, Wei Tong, Jingyu Hua, and Sheng Zhong. 2024. The fire thief is also the keeper: Balancing usability and privacy in prompts. *arXiv preprint arXiv:2406.14318*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vals AI, Inc. 2025. Medqa benchmarks.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. 2023. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhen Yang, Fang Liu, Zhongxing Yu, Jacky Wai Keung, Jia Li, Shuo Liu, Yifan Hong, Xiaoxue Ma, Zhi Jin, and Ge Li. 2024. Exploring and unleashing the power of large language models in automated code translation. *Proceedings of the ACM on Software Engineering*, 1(FSE):1585–1608.

Jiangchao Yao, Shengyu Zhang, Yang Yao, Feng Wang, Jianxin Ma, Jianwei Zhang, Yunfei Chu, Luo Ji, Kunyang Jia, Tao Shen, Anpeng Wu, Fengda Zhang, Ziqi Tan, Kun Kuang, Chao Wu, Fei Wu, Jingren Zhou, and Hongxia Yang. 2022. Edge-cloud polarization and collaboration: A comprehensive survey for ai. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6866–6886.

Yixiang Yao, Fei Wang, Srivatsan Ravi, and Muhao Chen. 2024. Privacy-preserving language model inference with instance obfuscation. *arXiv preprint arXiv:2402.08227*.

Zhongkai Yu, Shengwen Liang, Tianyun Ma, Yunke Cai, Ziyuan Nan, Di Huang, Xinkai Song, Yifan Hao, Jie Zhang, Tian Zhi, Yongwei Zhao, Zidong Du, Xing Hu, Qi Guo, and Tianshi Chen. 2024. Cambricon-llm: A chiplet-based hybrid architecture for on-device inference of 70b llm. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1474–1488. IEEE.

Yuanyuan Yuan, Zhibo Liu, Sen Deng, Yanzuo Chen, Shuai Wang, Yinqian Zhang, and Zhendong Su. 2024. Hypertheft: Thieving model weights from tee-shielded neural networks via ciphertext side channels. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 4346–4360.

Yuanyuan Yuan, Zhibo Liu, Sen Deng, Yanzuo Chen, Shuai Wang, Yinqian Zhang, and Zhendong Su. 2025. Ciphersteal: Stealing input data from tee-shielded neural networks with ciphertext side channels. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 4136–4154. IEEE.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3853–3866. Association for Computational Linguistics.

Jiangou Zhan, Wenhui Zhang, Zheng Zhang, Huanran Xue, Yao Zhang, and Ye Wu. 2025. Portcullis: A scalable and verifiable privacy gateway for third-party llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1022–1030.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024a. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4295–4312.

Mengke Zhang, Tianxing He, Tianle Wang, Lu Mi, Niloofar Mireshghallah, Binyi Chen, Hao Wang, and Yulia Tsvetkov. 2024b. Latticegen: Hiding generated text in a lattice for privacy-aware large language model generation on cloud. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2674–2690.

Shuhang Zhang, Qingyu Liu, Ke Chen, Boya Di, Hongliang Zhang, Wenhan Yang, Dusit Niyato, Zhu Han, and H Vincent Poor. 2024c. Large models for aerial edges: An edge-cloud model evolution and communication paradigm. *IEEE Journal on Selected Areas in Communications*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Ziqi Zhang, Chen Gong, Yifeng Cai, Yuanyuan Yuan, Bingyan Liu, Ding Li, Yao Guo, and Xiangqun Chen. 2024d. No privacy left outside: On the (in-) security of tee-shielded dnn partition for on-device ml. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3327–3345. IEEE.

A Algorithm of CoTrust

The overall workflow of CoTrust is summarized in Algorithm 1.

B Dataset Details

We evaluate CoTrust on four widely used NLP benchmarks covering question answering and summarization tasks. In addition to their task diversity, these datasets naturally contain privacy-relevant information (e.g., person names, organizations, locations, demographics, and health-related narratives), making them suitable for evaluating privacy-preserving LLM inference. Furthermore, we ensure all datasets and models are utilized following their original licenses and intended purposes. Table 5 reports summary statistics for all datasets.

MedQA. *MedQA* (Jin et al., 2020) is a multiple-choice QA benchmark derived from professional medical exam questions (e.g., USMLE (Jin et al., 2020)). Each instance typically presents a clinical vignette and asks the model to choose the best answer among four options, capturing clinically grounded reasoning. Such vignettes often include patient-centered details (e.g., age, symptoms, medications, and clinical history), which can constitute sensitive or quasi-identifying information in real deployments. We use the standard test split in our experiment (Vals AI, Inc., 2025).

SQuAD. The *Stanford Question Answering Dataset (SQuAD)* (Rajpurkar et al., 2018) is a reading comprehension benchmark built from Wikipedia passages, where answers are spans from the provided context. Because Wikipedia articles frequently contain named entities (e.g., persons, places, organizations) and other identifying descriptions, masking these contexts while preserving answer fidelity can be non-trivial in practice. In our experiments, we use the standard SQuAD v2.0 dev split (Rajpurkar, 2025), which contains total 11,873 questions from 1,207 contexts; it combines the 5,928 answerable questions in SQuAD1.1 with over 5,945 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones.

DialogSum. *DialogSum* is a large-scale benchmark for dialogue summarization research. It contains conversations drawn from many everyday contexts, reflecting how people communicate across a broad range of real-world situations. The dialogues cover diverse topics such as school and work, medication and shopping, leisure activities, and travel. They also represent varied interaction settings, including exchanges between friends and coworkers as well as customer–service conversations (Kaggle, 2025). Dialogues often involve privacy-sensitive information (e.g., family, travel, work, and healthcare-related discussions) and may include explicit personal references (names, relationships, ages, and other biographical cues). We use the test split for evaluation.

SAMSum. *SAMSum Corpus* (Gliwa et al., 2019) is an English dialogue summarization dataset consisting of messenger-style conversations paired with human-written abstractive summaries. It contains roughly 16k chat dialogues (Hugging Face, 2019) created by linguists to mimic everyday messaging behavior, featuring diverse registers (informal to formal) and realistic chat phenomena such as slang, emoticons, and typos. Dialogues include explicit speaker names, and each reference summary is written in the third person as a concise description of the main events in the conversation. These characteristics make SAMSum well-suited for evaluating privacy-preserving summarization, as dialogue-style inputs commonly involve identifiable entities and personal-context cues that must be protected while maintaining faithful generation. We use the standard test split in our experiments.

C Model Details

Qwen3-4B/8B: A pair of dense Transformer-based language models released by Alibaba Cloud (Alibaba Group, 2025) under the Qwen3 series (Yang et al., 2025). Both models support a native 32K context window, and some variants further enable longer-context inference via RoPE-scaling techniques (e.g., YaRN). The Qwen3 family also exposes a unified interface for thinking / non-thinking inference modes to balance multi-step reasoning quality and response latency. In our evaluation, the Qwen3-4B model serves as the TEE-resident SLM for efficient private inference, while the 8B model offers stronger instruction following and reasoning capacity for generating higher-quality references in collaborative inference.

Ministral-3B/8B. We use *Ministral-3B* and *Ministral-8B*, a lightweight model family released by Mistral AI (Mistral AI, 2025; Jiang et al., 2023) that targets strong performance below 10B parame-

Algorithm 1: Workflow of CoTrust

Input: Private input x; PII detector Detect; PII Transformations $\{T^{(k)}\}_{k=1}^{K}$; $\mathrm{LLM}_{\mathrm{GPU}}$; $\mathrm{SLM}_{\mathrm{TEE}}$
Output: Final answer $\hat{y}$

// Step 1: Multi-View De-Identified LLM Inference
$m \leftarrow \mathrm{Detect}(x)$ /* TEE: Detect PII spans/types */
for $k \leftarrow 1$ **to** K **do**
  $(v^{(k)}, \mathcal{M}_k) \leftarrow T^{(k)}(x, m)$ // TEE: Multi-View De-Identified Input Construction
$\mathcal{M} \leftarrow \{\mathcal{M}_k\}_{k=1}^{K}$ // TEE: Form initial de-identification mappings
$\{r^{(k)}\}_{k=1}^{K} \leftarrow \mathrm{LLM}_{\mathrm{GPU}}(\{v^{(k)}\}_{k=1}^{K})$ // GPU: Batched De-Identified LLM Inference
// Step 2: Consensus Scaffold Induction
for $k \leftarrow 1$ **to** K **do**
  $\tilde{r}^{(k)} \leftarrow \mathrm{Canonicalize}(r^{(k)}, m, \mathcal{M})$ // TEE: Cross-view entity canonicalization
  Update $\mathcal{M}$ // TEE: Update de-identification mappings
$S \leftarrow \mathrm{Induce}_{\mathrm{LLM}}(\mathrm{LLM}_{\mathrm{GPU}}, \{\tilde{r}^{(k)}\}_{k=1}^{K})$ // GPU: Canonical-view scaffold induction
// Step 3: Scaffold-Guided Private SLM Inference
$S^{\mathrm{priv}} \leftarrow \mathrm{Demask}(S, \mathcal{M})$ // TEE: Recover Entities in S
$\hat{y} \leftarrow \mathrm{SLM}_{\mathrm{TEE}}(x, S^{\mathrm{priv}})$ // TEE: SLM Inference for the final answer
return $\hat{y}$

Dataset	Task	Description	Samples
MedQA	QA	Medical multiple choice	1273
SQuAD	QA	Extractive QA	11873
SAMSum	Summ.	Conversation	819
DialogSum	Summ.	Dialogue	500

Table 5: Statistics of datasets used in our evaluation.

ters with efficient inference. The Ministral models support up to a 128K-token context window, although the effective usable length may vary across deployments depending on system and kernel support. The 8B variant is reported to adopt an attention design optimized for long-context efficiency (e.g., sliding-window or interleaved patterns), improving speed and memory usage on long inputs. In our setting, Ministral-3B serves as the TEE-resident SLM, while Ministral-8B is deployed as the GPU-hosted LLM to provide stronger generation quality in collaborative inference.

Llama-3.2-3B/Llama-3.1-8B: We use *Llama-3.2-3B* as the SLM and *Llama-3.1-8B* as the LLM, both released by Meta (Meta Platforms, Inc., 2025) under the Llama family (Grattafiori et al., 2024). The models support long-context inference (up to a 128K-token context window in common configurations) and provide strong instruction-following and multilingual capabilities. Notably, this pairing spans two *different series* within the same model family; we include it to examine whether CoTrust can enable effective collaborative inference even when the SLM and LLM come from different Llama releases rather than a strictly matched series. The Llama-3.2-3B model is suitable for resource-constrained environments, while the larger Llama-3.1-8B model provides stronger generation and reasoning ability for producing higher-quality reference outputs.

Qwen3-Max. *Qwen3-Max* is an API-served model released by the Qwen team as the flagship model in the Qwen3 series (Qwen Team, 2025). It is designed to provide strong instruction following, reasoning, multilingual understanding, and agent-oriented capabilities at scale. In our evaluation, we use Qwen3-Max as a unified *LLM-as-a-judge* to compute QM-Score for both summarization and QA outputs, complementing token-overlap metrics by providing a semantic quality assessment. In addition, we use Qwen3-Max in the privacy protection experiments as both an *adversarial extractor* and a *privacy protection evaluator*. Specifically, given only artifacts observable outside the TEE boundary (e.g., de-identified inputs and intermediate outputs), it attempts to reconstruct sensitive entities to measure the *Extraction Success Rate (ESR)*, and it further produces a model-based *Privacy Protection Score (PPS)*.

D Evaluation Details

Experiment Platform. We conduct experiments on a host with an Intel Xeon Silver 4514Y CPU (32 cores) and 128 GB DDR5 memory. The system is equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB) connected through a PCIe 4.0 x16 interface. The host runs Intel's customized Linux kernel (Linux 6.8.0-1022-intel) optimized for TDX optimization, while the TDX guest (TD) runs

Linux 6.8.0-71-generic and is launched through *qemu*. The TD is configured with 8 vCPUs and 64 GB private memory.

Generation Hyperparameters. For response generation with both LLMs and SLMs, we generate outputs using greedy decoding (sampling disabled; `do_sample=False, num_beams=1`) with the other generation hyperparameters following each model's default `generation_config`.

Accuracy, Recall and F1 metric for MedQA. MedQA is a four-way multiple-choice benchmark with label space $\{Y\} = \{A, B, C, D\}$. We report Accuracy, macro-Recall, and macro-F1. Accuracy is computed as the fraction of questions for which the predicted option matches the gold answer. For Recall and F1, we first compute per-class recall and F1 for each option $c \in \{Y\}$ by treating c as the positive class and the remaining options as negative. We then average over the four classes to obtain macro-Recall and macro-F1, respectively, so that each answer option contributes equally regardless of class frequency.

EM and F1 metrics on SQuAD. We follow the standard SQuAD evaluation protocol (Rajpurkar et al., 2018) and report Exact Match (EM) and token-level F1. Let $\mathrm{norm}(\cdot)$ denote the official normalization that lowercases text, removes punctuation and articles, and collapses whitespace. For a prediction $\hat{a}$ and a set of reference answers $\mathcal{A}$, EM is defined as

$$\mathrm{EM}(\hat{a}, \mathcal{A}) = \max_{a \in \mathcal{A}} \mathbb{I}[\mathrm{norm}(\hat{a}) = \mathrm{norm}(a)] \,.$$

For token-level F1, let $T(\cdot)$ map a normalized answer string to a bag of tokens, and let $|\cdot|$ denote token counts with multiplicity. For a reference a, define the token overlap $o(\hat{a}, a) = |T(\hat{a}) \cap T(a)|$, then

$$P = \frac{o(\hat{a}, a)}{|T(\hat{a})|}, \quad R = \frac{o(\hat{a}, a)}{|T(a)|}, \quad F1(\hat{a}, a) = \frac{2PR}{P + R},$$

and the per-question score is $\max_{a \in \mathcal{A}} F1(\hat{a}, a)$. For SQuAD v2.0, unanswerable questions are evaluated by including the empty string as a reference answer when applicable.

E Prompt Details

Prompts designed for *Multi-View De-Identified LLM Inference*, *Consensus Scaffold Induction* and *Scaffold-Guided Private SLM Inference* are outlined in Figure 3, 5 and 4, respectively. Prompts used for Privacy-Content Extraction and Privacy Protection Evaluation with Qwen3-MAX are depicted in Figure 6 and 7.

Prompt

You are a precise extraction model for reading comprehension.
Your goal is to extract the answer from the Context as a **concise text span**.

Strict Instructions:
1. Extract the **exact text fragment** from the Context. Do NOT write full sentences.
2. Do NOT add explanations or extra words.
3. If the answer is not present, return "unanswerable".
4. Output MUST be a valid JSON object.

Example:
Context: "The Eiffel Tower is located in Paris, France."
Question: "Where is the Eiffel Tower?"
JSON Output: {{ "answer": "Paris, France" }}

Task:

Context:
{context}

Question:
{question}

JSON Output:

Figure 3: Prompts for De-Identified LLM Inference.

Prompt

You are a model participating in a SQuAD (Stanford Question Answering Dataset) style reading comprehension task.
Your task is to provide the most accurate and concise answer to a question based on a provided text passage. You have received expert feedback to help improve your answer.

Expert Tips to Follow:
{tips}

Context:
{context}

Question:
{question}

Your Task:
Provide a new, improved answer. The answer MUST be a concise fragment extracted directly from the context. If the answer cannot be found, your answer should be the single word: "unanswerable".

New, Improved Answer:

Figure 4: Prompts for Scaffold-Guided Private SLM Inference.

Prompt

You are a senior editor and fact-checker for the SQuAD task.

Your goal is to synthesize the best possible answer from two candidate answers provided by different AI assistants.

Context:
- The context and answers may contain abstract placeholders (e.g., '[PERSON_1]').
- Answer A and Answer B were generated by the same powerful model but based on slightly different masked versions of the context.

— Start of Content —

Context:
{context}

Question:
{question}

Answer A:
{answer_A}

Answer B (Placeholders Unified):
{answer_B}

— End of Content —

Your Task:
Review both Answer A and Answer B. Your goal is to produce a new, final answer that is the most accurate and concise, combining the best elements of both. If one is already perfect, use it. If both are flawed, derive the correct answer from the context. If the question is unanswerable, respond with "unanswerable".

Synthesized Answer:

Figure 5: Prompts for Canonical-view Scaffold Induction.

Prompt

You are an expert in text reconstruction for SQuAD-style QA data.
Below are TWO masked versions of the SAME concatenated string:
1. The CONTEXT passage appears first.
2. Then ALL QUESTIONS for that context appear afterwards, in order.
Sensitive information has been replaced with placeholder tags like [DATE_1], [GPE_2], [PERSON_3], etc.

Your task:
Reconstruct the original concatenated string as accurately as possible by replacing these placeholders with plausible, contextually appropriate words or phrases. You MUST not delete any placeholders directly.

Important:
1. Keep sentence structure EXACTLY the same.
2. REPLACE each placeholder with a plausible value, based on context from either version! Never delete the placeholder.
3. Do NOT ask for any unmasked original text. You only have the two masked versions below.
4. Do NOT invent new facts not supported by the text. Only restore what is likely missing from context.
5. Use BOTH versions jointly: if one version reveals information that the other masks, use the revealed information.
6. Do NOT add/remove/reorder questions. Do NOT merge or split sentences. Only replace placeholders with words/phrases.

Return ONLY the reconstructed concatenated string, with no extra explanation and no JSON.

Here are the two masked versions:
SpaCy Masked:
{spacy_masked_concat}

Presidio Masked:
{presidio_masked_concat}

Reconstructed concatenated string:

Figure 6: Prompts for Data Reconstruction with *Qwen3-Max* on SQuAD.

Prompt

You are an expert in privacy protection evaluation for protecting SQuAD-style QA data.
You will see:
1. The ORIGINAL concatenated string (context first, then all questions)
2. SpaCy masked concatenated string
3. Presidio masked concatenated string

Task:
Give a Privacy Protection Score in [0, 1] (float).

Interpretation:
1.00 = Excellent privacy protection: identifying details are effectively removed; re-identification risk is very low.
0.00 = No privacy protection: the masked text still reveals sensitive details almost completely.

Consider:
1. Identifiers in BOTH the context and the appended questions (names, precise dates, locations, IDs, rare specifics).
2. Whether masked details can still be trivially inferred from the remaining text (lower score).
3. Consistency and coverage of masking across the whole concatenated string.

Original:
{original_concat}

SpaCy Masked:
{spacy_masked_concat}

Presidio Masked:
{presidio_masked_concat}

Return your answer as a JSON with the privacy protection score between 0.0 and 1.0. Return without extra explanation. Example: {{"score": 0.85}}

Your response:

Figure 7: Prompts for Privacy Protection Evaluation with *Qwen3-Max* on SQuAD.