Privacy-Preserving LLM Inference in Practice: A Comparative Survey of Techniques, Trade-Offs, and Deployability

Davide Andreoletti[1], Alessandro Rudi[2*], Emanuele Carpanzano[2†], Francesco Lelli[1], Tiziano Leidi[1]

[1]Information Systems and Networking Institute
University of Applied Sciences and Arts of Southern Switzerland, Switzerland
davide.andreoletti@supsi.ch, francesco.lelli@supsi.ch, tiziano.leidi@supsi.ch

[2]Prem AI

Abstract

Large Language Models (LLMs) are increasingly deployed as cloud services, raising practical concerns about the confidentiality of user prompts and generated completions. In this paper, we survey privacy-preserving inference solutions for Transformer-based LLMs with the explicit goal of supporting operational choices in real-world deployments. We adopt a strong operational notion of privacy: only the client can read the prompt and the corresponding completion, end to end. The review is organised around the main families of Privacy-Enhancing Technologies (PETs). For each family, we examine representative systems and how they address key bottlenecks in confidential LLM inference, such as non-linear layers and autoregressive decoding. We then compare these approaches in terms of trust assumptions, scalability, and deployment maturity. This comparison characterises the current practical landscape of privacy-preserving LLM inference and motivates a trust-minimising deployment trajectory: from TEE-based solutions that enable large-scale confidential inference today; through crypto-augmented designs that reduce reliance on hardware trust at higher computational cost; toward Fully Homomorphic Encryption as a principled long-term endpoint for non-interactive confidentiality.

1 Introduction

Large Language Models (LLMs), such as ChatGPT [63] and LLaMA [55], are increasingly adopted for a wide range of tasks, including question answering and document summarization. In most real-world deployments, these models are hosted in the cloud and offered as Software-as-a-Service (SaaS). Cloud-based deployment provides clear operational advantages, including elastic scaling, reduced operational overhead, and simplified maintenance [2]. At the same time, this setting implies that user prompts and model outputs are processed within infrastructure controlled by the service provider, making them potentially visible to the cloud operator and associated software stack.

This deployment model raises significant privacy concerns. Users may submit sensitive inputs—such as business-critical, financial, or healthcare documents—and may be unwilling to disclose either their prompts or the generated completions to third parties. A common workaround is to deploy LLMs on the client's own premises. However, this shift introduces two distinct trade-offs. First, on-premises deployment forfeits many of the practical benefits of cloud infrastructures. Second, it exposes model parameters to the client, which model owners typically regard as sensitive, for example because model weights may leak information about proprietary training data or constitute valuable intellectual property [21].

As a result, LLM deployment in practice gives rise to a two-sided privacy problem. On the one hand, *content privacy* requires that no third party be able to observe, infer, or persist user prompts or the corresponding model completions. On the other hand, *model privacy* requires that providers do not reveal model weights, intermediate activations, or information that could disclose proprietary or legally protected training data. While related, these objectives are not equivalent and are not always addressed simultaneously by existing techniques. In this work, we explicitly prioritise content privacy, and in particular end-to-end confidentiality of *prompt+completion*, as our primary operational requirement. This choice is motivated by the observation that protecting the user's prompt alone is insufficient if the corresponding completion can still be inspected by the provider or leaked via logging and telemetry. Model-privacy guarantees are instead treated as an important but technique-dependent dimension, which is not uniformly provided across approaches.

Reconciling end-to-end content confidentiality with the operational advantages of cloud-based deployment constitutes a central research challenge. This challenge has motivated a growing body of work on privacy-preserving LLM inference, i.e., techniques that enable LLMs to run in the cloud without exposing sensitive

*alessandro@premai.io, office Prem s.r.l., Via Giuseppe Verdi 6, 70017 Putignano (BA), Italy

†emanuele@premai.io, office Prem s.a., Crocicchio Cortogna 6, 6900 Lugano, Switzerland

information belonging to either the client or the model owner. These approaches rely on a diverse set of Privacy-Enhancing Technologies (PETs), each offering different security guarantees and performance trade-offs.

In this paper, we review existing privacy-preserving LLM solutions with a focus on the inference phase. We organise the review around four main PET families: Secure Multi-Party Computation (MPC), which enables joint evaluation over secret-shared inputs; Fully Homomorphic Encryption (FHE), which allows arbitrary computation directly on encrypted data; Trusted Execution Environments (TEEs), which provide hardware-isolated enclaves for confidential execution; and non-cryptographic approaches based on reducing the information contained in the queries, such as prompt obfuscation. These PETs differ substantially in their trust assumptions, computational overhead, and privacy guarantees. Consequently, no single approach is universally suitable, and many recent systems adopt hybrid designs—such as combining MPC with HE or complementing TEEs with lightweight cryptography—to balance confidentiality with practical latency and bandwidth constraints. Moreover, while TEEs and modern MPC frameworks increasingly support full-scale LLMs, some PETs, notably FHE, remain practical primarily only for some model components; we therefore also include works that secure specific sub-blocks when they are explicitly aimed at LLM inference. These trade-offs are summarised in Table 1, which provides a high-level comparison of the main PET families with respect to trust assumptions, interaction requirements, dominant bottlenecks, and deployability for LLM inference.

System and threat model. To reason concretely about the considered privacy problem, we assume a standard client–cloud inference setting with optional retrieval-augmented generation (RAG) [52, 28]. The client owns the prompt (and any retrieved documents) and is assumed to be the only party able to read the completion in plaintext. The service operator and the underlying cloud stack—including the operating system, hypervisor, administrators, and observability pipeline—are treated as potentially adversarial: they may inspect memory, storage, network traffic, logs, and telemetry, and in stronger threat models may deviate from prescribed behaviour [7]. The protected assets include the prompt, the completion, and any user-provided or retrieved context; depending on the PET, model weights may additionally be protected [38]. Our primary security goal is end-to-end confidentiality of *prompt+completion*.

The remainder of the paper is structured as follows. Section 2 provides the necessary background on i) Transformer-based LLM inference, focusing on architectural features such as non-linear layers and autoregressive decoding, and ii) on the main PET families considered in this survey. Section 3 then discusses MPC-based privacy-preserving inference, including systems for BERT, GPT, and LLaMA and recent advances in non-linear protocols, quantisation, and malicious security. Section 4 covers FHE-based approaches and encryption-friendly architectures. Section 5 surveys hybrid HE/MPC designs targeting low-interaction or non-interactive protocols. Section 6 reviews TEE-based confidential inference on CPUs and GPUs. Section 7 describes solutions based on information reduction. ection 8 synthesises cross-cutting trade-offs and open challenges. Section 9 then presents our vision of a trust-minimising deployment trajectory for confidential LLM inference, ranging from TEE-based serving to crypto-augmented designs and, ultimately, fully homomorphic encryption. Finally, Section 10 concludes the paper.

2 Background

This section introduces the minimum technical background needed to interpret the practical bottlenecks discussed throughout the survey. We first summarise the high-level mechanics of Transformer-based LLM inference, with emphasis on where computation, interaction, and leakage surfaces arise. We then provide a compact operational overview of the main PET families considered in the paper, especially focusing on computation models and trust assumptions.

2.1 Transformer-based LLM inference at a glance

Transformer blocks and where the cost comes from. Most modern LLMs are stacks of repeated *Transformer blocks* [58, 17, 6]. At a high level, each block alternates (i) large linear transformations and (ii) a small set of non-linear operations. Linear layers dominate floating-point compute in plaintext inference: they include the projections that produce queries, keys and values (Q/K/V), the output projection of attention, and the feed-forward network (FFN) layers [58]. By contrast, non-linear components such as `Softmax` (in attention), activation functions such as `GeLU`/`SiLU` (in the FFN), and normalisation layers (LayerNorm/RMSNorm) contribute less to plaintext FLOPs but often dominate overhead in privacy-preserving settings because they require comparisons, divisions, exponentials, or data-dependent control flow [23, 28].

Self-attention and the KV-cache. In self-attention, the model computes attention weights between the current token representation and previous token representations [58]. Concretely, for a new token at time step t, the model forms Q_t from the current hidden state and compares it to the set of keys $\{K_1, \ldots, K_t\}$, producing attention scores, applies `Softmax`, and then mixes values $\{V_1, \ldots, V_t\}$. In practice, inference systems store the past keys and values in a *key/value cache* (KV-cache) so that each new step reuses past K, V rather than recomputing them [15]. This cache is a major systems object: it grows with sequence length, is bandwidth-intensive to move across memory hierarchies, and, from a privacy perspective, contains sensitive intermediate representations that must remain protected if end-to-end confidentiality is required [46].

PET	Trust	Interaction	Main bottleneck	Deployability
FHE	Cryptographic (single-server)	Low	Computation complexity	Low
MPC	Cryptographic (multi-party)	High	Bandwidth	Medium
Hybrid	Mixed	Medium	Protocol complexity	Medium
TEE	Hardware + attestation	Low	Trusted Computer Base + side channels	High
Information reduction	Partial information exposure	None	Utility/privacy trade-off	High

Table 1: High-level comparison of privacy-enhancing technologies (PETs) for LLM inference. *Trust* indicates the underlying trust assumption or protection mechanism (e.g., cryptographic, hardware-based, or information-limiting); *Interaction* refers the amount of online communication between client and server required during inference; *Main bottleneck* summarises the dominant source of overhead or limitation; *Deployability* reflects the current maturity and practicality of deploying the approach in real-world systems.

Autoregressive generation as an iterative protocol. Decoder-only LLMs generate text *autoregressively*: they predict one token at a time [47, 48, 6]. Each generation step performs a full forward pass to produce a vector of logits over the vocabulary, then applies a *decoding rule* to select the next token, appends it to the context, and repeats. This structure is central to practical privacy-preserving inference because any per-step overhead is multiplied by the number of generated tokens [52]. It is also central to interaction patterns: if a privacy mechanism requires online rounds for one forward pass, it will generally require those rounds *per token* during generation [28].

From logits to tokens: decoding and sampling. Given logits $z \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$, inference applies a decoding strategy to choose the next token. Common strategies include: *greedy decoding* (take $\arg\max_i z_i$), *beam search* (maintain multiple hypotheses), and stochastic decoding methods such as *temperature sampling* and *top-k*/*top-p* (nucleus) sampling [20, 25, 30]. While these steps are often treated as lightweight in plaintext, they become important in privacy-preserving settings because they involve arg max, comparisons, sorting/selection, and random sampling [38]. In MPC, these are typically interactive and communication-heavy; in FHE, they are difficult to express efficiently without additional interaction or approximation; in TEEs, they are straightforward but contribute to the output-side leakage surface if streaming is used [27].

Prompting and context. Operationally, a "prompt" is the full input context presented to the model at inference time: system instructions, user messages, tool outputs, and optionally RAG [35, 32]. From the viewpoint of this paper, *content privacy* applies to all of this context and to the generated completion. Importantly, the prompt is not just raw text: it is tokenised into a sequence of integers (token IDs) and then mapped to dense vectors by an embedding table [50, 31]. Both token IDs and embeddings can leak sensitive information if exposed, and many PET designs must explicitly handle the embedding layer efficiently (e.g., large vocabulary lookups under encryption or secret sharing) [38, 52].

Why these details matter for PETs. The above mechanics explain recurring bottlenecks in privacy-preserving inference [23]: (1) large linear layers are compute-heavy but structurally regular (good for SIMD and hardware acceleration); (2) non-linearities and decoding steps are structurally small but cryptographically expensive; (3) autoregressive generation multiplies both costs across many steps; (4) the KV-cache and streaming outputs create additional systems-level confidentiality and leakage considerations.

2.2 Operational overview of PET families

We briefly summarise the PET families considered in this survey in terms of their computation model, trust assumption, and the implications for LLM inference [52, 38].

Secure Multi-Party Computation (MPC). MPC protocols allow two or more parties to jointly compute $y = f_\theta(x)$ without revealing x (client input) or θ (model parameters), typically by representing values as secret shares and performing arithmetic on those shares [62, 51]. The security guarantee is cryptographic and composable under explicit adversary models (often semi-honest, increasingly malicious) [7]. The main practical consequence is *interaction*: many operations require multiple online rounds and significant communication [3]. For LLMs, MPC is often bottlenecked by (i) bandwidth for large linear layers and (ii) interactive protocols for non-linearities and decoding primitives (comparisons, arg max, top-k) [38, 28].

Fully Homomorphic Encryption (FHE). FHE enables a server to evaluate functions directly on encrypted data: the client encrypts x, the server computes on ciphertexts, and the client decrypts the result [22]. The appeal is low online interaction and strong end-to-end confidentiality under a single-server model [5]. The practical constraints are computational: ciphertext operations are expensive, non-linearities must be approximated (often with low-degree polynomials), and depth is limited by ciphertext noise growth, sometimes requiring costly bootstrapping [23]. For LLMs, the combination of large matrix multiplications and repeated autoregressive steps makes fully end-to-end FHE

inference challenging at scale, motivating architecture co-design and acceleration [4].

Hybrid HE/MPC designs. Hybrid systems combine paradigms to exploit complementary strengths [28]. A common pattern uses HE/FHE to reduce communication for large linear layers (amortising cost via packing/SIMD), while using MPC/FSS-style protocols for non-linearities, comparisons, and decoding logic [49]. The core engineering challenge is efficient and correct conversion between encrypted representations and secret shares, and managing approximation/quantisation so that accuracy remains close to plaintext [52].

Trusted Execution Environments (TEEs). TEEs protect computation through hardware-isolated enclaves with attestation [13]. The model and prompt are processed in plaintext *inside* the enclave, and confidentiality relies on hardware isolation plus correct system integration (attestation, key provisioning, encrypted channels, and disciplined logging/telemetry) [27]. The practical advantage is near-plaintext performance for full-scale LLMs, including GPU inference, with low interaction [56]. The main drawbacks are the hardware trust chain and residual leakage surfaces (side channels, metadata such as timing/length, and output streaming) [42].

Information-reduction approaches. Information-reduction techniques operate at the client side by transforming the prompt to reduce sensitive content before sending it to an untrusted API endpoint [19]. These methods require no changes to the hosting infrastructure and have excellent deployability, but provide weaker and task-dependent guarantees [1]. They primarily protect the prompt (not the completion end-to-end) and must trade utility for reduced disclosure via perturbation (e.g., DP mechanisms) or semantic obfuscation with local restoration [8].

Connecting PETs to the paper's privacy goal. Throughout this survey we adopt an operational notion of privacy: only the client can read the prompt and the corresponding completion end-to-end [52]. Under this goal, the main differentiators among PET families are: (i) where trust is placed (cryptography vs. hardware vs. partial disclosure), (ii) whether inference requires online interaction per generated token, and (iii) how non-linearities, decoding, and KV-cache handling affect both performance and leakage surfaces in realistic LLM deployments.

3 MPC-Based Solutions

Secure Multi-Party Computation (MPC) enables multiple parties to jointly evaluate a function over their private inputs without revealing them to one another. In the client–cloud inference setting, the client holds a private prompt x, the server holds private model parameters θ, and both parties collaboratively compute $y = f_\theta(x)$ through a cryptographic protocol that preserves the confidentiality of both x and θ. This makes MPC a natural candidate for addressing the two-sided privacy problem in outsourced LLM inference.

Applying MPC to Transformer-based LLMs, however, remains challenging due to a small number of structural cost factors that dominate performance. First, the large matrix multiplications in linear layers largely determine bandwidth consumption and runtime. Second, non-linear components such as Softmax, GeLU, and LayerNorm require interactive protocols that are expensive under secret sharing. Third, the iterative nature of autoregressive decoding amplifies all costs across many generation steps. These factors are intrinsic to Transformer architectures and jointly define the design space for MPC-based LLM inference.

Existing systems explore different points in this design space by mitigating one or more of these structural bottlenecks through protocol design, model restructuring, or system-level optimisations. In the remainder of this section, we organise prior work along a set of recurring analytical lenses: handling of non-linearities, communication and system-level optimisation, reduced-precision computation, support for autoregressive decoding, and adversarial security.

Non-linearities and model restructuring. Non-linear operations constitute one of the primary performance bottlenecks in MPC-based Transformer inference. Early systems addressed this challenge by replacing costly non-linear components with MPC-friendly surrogates or by restructuring the model to avoid expensive operations. MPCFORMER follows this approach by introducing approximations for non-linear layers and subsequently distilling a student model to recover accuracy [33]. Puma generalises this strategy to much larger models, including LLaMA-7B, using carefully designed polynomial approximations for Softmax and GeLU together with secure Embedding and LayerNorm protocols [18]. It achieves near-plaintext accuracy without fine-tuning and performs LLaMA-7B inference in roughly five minutes per token, representing the first MPC demonstration at that scale. SecFormer further improves non-linear handling by providing efficient protocols for `argmax`, GeLU, and LayerNorm, yielding higher accuracy than MPCFORMER and faster inference than Puma [37]. Subsequent refinements include Nimbus, which tailors low-degree polynomial approximations to Transformer activation distributions [34], and SHAFT, which introduces constant-round Softmax and Fourier-based GeLU approximations to substantially reduce communication [29].

Communication and system-level optimisation. Beyond individual operators, communication overhead remains the dominant bottleneck for MPC in practical wide-area deployments. Several systems therefore focus on reducing bandwidth and interaction through packing strategies, dimension reduction, or selective outsourcing of computation. PrivTF introduces optimised protocols for embeddings and other core components; its PrivEmbedding mechanism ensures that communication does not scale with vocabulary size, reducing

a 128-dimensional embedding query to approximately 3.1 MB [45]. PermLLM pushes bandwidth reduction further by outsourcing costly non-linear computations to the client within a randomly permuted domain. While this setting is close in spirit to multi-party inference, it does not provide standard cryptographic guarantees, as confidentiality relies on the hardness of recovering sensitive values from the applied permutation and the resulting permuted computation domain [66].

Quantisation, distillation, and reduced-precision computation. Complementary to protocol- and system-level optimisations, several systems reduce MPC overhead by lowering numerical precision or shifting parts of the workload outside secure arithmetic. Ditto integrates quantisation directly into the MPC pipeline through quantisation-aware distillation and new primitives for secure type conversion [60]. Its static dyadic quantisation avoids MPC-expensive dynamic operations such as clipping and per-batch maxima, substantially reducing bitwidth, arithmetic cost, and communication. As a result, Ditto achieves significant speedups over MPCFORMER and Puma on BERT and GPT-2 while incurring minimal accuracy loss.

Autoregressive LLMs and GPT-style decoding. Autoregressive generation poses a distinct challenge for MPC because each output token requires a full forward pass, multiplying all underlying costs. This has motivated specialised protocols for GPT-style architectures. SIGMA leverages function secret sharing (FSS) to implement complex non-linearities such as Softmax, GeLU, and SiLU, and accelerates them on GPUs [24]. It improves latency by 12–19× compared to prior GPU-based approaches and performs secure inference of LLaMA2-13B in 38 s and GPT-2 in 1.5 s. CipherGPT focuses on two-party GPT inference and introduces a secure matrix multiplication protocol achieving a 6.2× speedup and 4.1× bandwidth savings over prior work [26]. It also provides new protocols for GeLU and top-k sampling, addressing core steps in secure autoregressive decoding.

Malicious security. The systems discussed so far primarily target semi-honest adversaries, i.e., adversaries that are presumed to follow the protocol specification correctly but may attempt to infer additional information from the messages they observe. A complementary line of work strengthens the adversarial model to account for malicious parties that may arbitrarily deviate from the protocol. Mosformer is the first maliciously secure three-party computation framework tailored to large Transformer models such as BERT and GPT [9]. Using verifiable distributed point functions (VDPFs), Mosformer constructs constant-round comparison and lookup protocols for complex non-linearities. Combined with optimised modulus conversion, it achieves up to 5.3× speedups and 4.3× communication savings relative to prior maliciously secure protocols, while matching or surpassing the online performance of semi-honest frameworks such as BOLT and BumbleBee.

Summary. Viewed through the analytical lenses outlined above, MPC-based LLM inference now spans BERT-scale models with near-plaintext accuracy (e.g., MPCFORMER, SecFormer, PrivTF), LLaMA- and ChatGLM-scale models with multi-second to multi-minute per-token latency (e.g., Puma, BumbleBee, PermLLM, SIGMA, CipherGPT), specialised protocols for non-linear components and embeddings (e.g., Nimbus, SHAFT, PrivTF), and maliciously secure Transformer-block protocols (Mosformer). Despite steady progress, communication and interaction remain the dominant bottlenecks, particularly for autoregressive decoding, motivating ongoing research into approximations, quantisation, and architectural restructuring that reduce the secure computational footprint.

4 FHE-Based Solutions

Fully Homomorphic Encryption (FHE) enables arbitrary computation over encrypted data and thus supports conceptually non-interactive inference: the client encrypts its input once, the server performs homomorphic evaluation over ciphertexts, and the client locally decrypts the result. This execution model provides strong end-to-end confidentiality and a clean client–server separation, making FHE an appealing primitive for privacy-preserving LLM inference.

Applying FHE to Transformer-based models, however, introduces a set of structural constraints that fundamentally shape the design space. First, homomorphic linear algebra incurs significant ciphertext expansion, making large matrix multiplications computationally expensive. Second, non-linear components such as Softmax, GeLU, and LayerNorm must be approximated by low-degree polynomials, introducing accuracy–efficiency trade-offs. Third, homomorphic encryption schemes rely on the deliberate injection of *noise* into ciphertexts to ensure security; while this noise is essential to hide plaintext values, it grows with each homomorphic operation and eventually renders the ciphertext undecryptable. As a result, the depth of computable circuits is bounded, and deep computation graphs such as Transformers often require costly *bootstrapping* operations, which refresh ciphertexts by homomorphically reducing accumulated noise at substantial computational expense. Together, these constraints define the practical envelope within which FHE-based Transformer inference can operate.

Existing work can be understood as exploring this envelope along a small number of largely orthogonal optimisation axes. In the remainder of this section, we organise prior systems according to three such axes: protocol-level advances for homomorphic linear and non-linear evaluation, architecture-level co-design of Transformer models adapted to FHE constraints, and system-level acceleration techniques that mitigate the computational cost of homomorphic execution. We conclude by discussing the resulting practical limitations of FHE for large-scale and autoregressive LLMs.

Protocol-level advances for homomorphic linear and non-linear layers. A central line of work improves the efficiency of homomorphic evaluation through protocol-engineering techniques that target both linear algebra and non-linear components. THOR exemplifies this approach by optimising homomorphic matrix multiplication using giant-step techniques and by intro-

ducing advanced approximation schemes and adaptive iterative evaluation for Softmax, LayerNorm, GeLU, and Tanh [39]. These techniques reduce noise growth and enable encrypted inference for $\text{BERT}_{\text{BASE}}$ with sequence lengths typical of standard benchmarks, while maintaining accuracy comparable to plaintext execution.

More recent work demonstrates that protocol-level advances can scale to full end-to-end Transformer inference under a fully non-interactive execution model. NEXUS follows this direction by relying exclusively on FHE to evaluate both linear and non-linear components of Transformer models, eliminating the round-by-round interaction characteristic of MPC-based inference [65]. Techniques such as ciphertext compression, slot folding, and FHE-compatible `argmax` enable full-model evaluation within a request–response paradigm, while GPU acceleration mitigates the substantial computational cost of homomorphic execution. Together, these results illustrate how protocol-level improvements alone can support non-interactive, end-to-end secure inference for Transformer-based architectures.

Architecture-level co-design for FHE. Rather than directly evaluating standard Transformer architectures, a complementary line of work redesigns the model itself to better align with FHE's algebraic constraints. Powerformer argues that unmodified Transformers are inefficient under homomorphic execution and proposes an architecture explicitly co-designed for FHE [44]. In Powerformer, Softmax and LayerNorm are replaced with power and linear functions amenable to polynomial evaluation, while GeLU and Tanh are approximated via a pseudo-sign composite function. Combined with a matrix multiplication method optimised for homomorphic arithmetic, this co-designed architecture substantially reduces circuit depth and homomorphic workload, while preserving accuracy on a distilled $\text{BERT}_{\text{BASE}}$ model.

System-level acceleration and GPU offloading. A third, largely orthogonal axis focuses on mitigating the computational cost of homomorphic evaluation through system-level optimisations. EncryptedLLM demonstrates privacy-preserving inference for GPT-2 variants using the CKKS scheme [10] with GPU-accelerated bootstrapping [16]. By offloading computationally intensive homomorphic operations to GPUs and carefully selecting polynomial approximations for Softmax, GeLU, and LayerNorm, EncryptedLLM achieves accuracy close to plaintext performance across a range of NLP benchmarks. Techniques such as batched evaluation, slot re-use, and ciphertext packing are critical in amortising the cost of homomorphic execution and enabling inference over longer input sequences.

Practical envelope and limitations of FHE for LLMs. Despite steady progress along all three optimisation axes, FHE-based inference remains substantially more computationally demanding than MPC for large-scale models. Even with protocol-level improvements, FHE-friendly architectures, and GPU acceleration, current systems operate most effectively on medium-sized models or reduced sequence lengths. Bootstrapping remains a dominant cost driver, and autoregressive decoding multiplies the expense of each homomorphic forward pass, severely limiting throughput for generative LLMs. Ongoing work therefore focuses on further reducing polynomial degree and circuit depth, improving packing and slot re-use, and exploring hybrid designs that combine FHE with MPC or TEEs to mitigate its high computational overhead while preserving non-interactive confidentiality.

5 Hybrid HE/MPC Solutions

Pure MPC and pure FHE each face structural limitations when applied to Transformer-based LLMs. MPC incurs high interaction and communication overhead, particularly for large linear layers and autoregressive decoding, while FHE suffers from high computational cost, ciphertext expansion, bootstrapping overhead, and sensitivity to circuit depth. Hybrid HE/MPC designs arise from the observation that these limitations are largely complementary: MPC excels at non-linear control flow, comparisons, and conditional logic, whereas FHE can execute large linear transformations in a non-interactive, Single-Input-Multiple-Output (SIMD)-friendly manner. By combining the two paradigms, hybrid systems aim to exploit their respective strengths while mitigating their individual weaknesses.

Rather than forming a single design pattern, hybrid HE/MPC systems span a design space defined by multiple orthogonal axes. Key dimensions include where homomorphic computation is placed within the model (e.g., block-level or operator-level), how interaction is structured, and how efficiently data can be converted between encrypted and secret-shared representations.

Block-level hybridisation: homomorphic linear layers within MPC. A conservative point in the hybrid design space incorporates homomorphic encryption selectively inside an otherwise MPC-based pipeline. In this block-level approach, FHE is used to accelerate bandwidth-heavy linear layers, while non-linear and control-heavy components remain in MPC. BOLT exemplifies this strategy by combining FHE with additive secret sharing [43]. Compact packing and word elimination substantially reduce communication for linear operations, while MPC handles Softmax, GeLU, and LayerNorm. BumbleBee follows a similar pattern, relying on interactive MPC for non-linear components and applying FHE to reduce communication in linear transformations [36]. These systems demonstrate that selective hybridisation can yield tangible efficiency gains without fundamentally altering the MPC execution model.

Operator-level hybridisation and HE–MPC conversion. At the most fine-grained end of the design space, hybridisation is performed at the level of individual operators. BLB adopts this strategy by dynamically selecting between HE and MPC based on the cost profile of each component [61]. Adjacent linear operators are fused and evaluated using the CKKS scheme [10] to mitigate ciphertext expansion, while non-linearities and control flow remain in MPC. Central to this approach is an explicit protocol for conversion between ciphertexts and MPC shares, enabling seamless alternation between encrypted domains even as ciphertext precision changes.

Experiments on $\text{BERT}_{\text{BASE}}$, $\text{BERT}_{\text{LARGE}}$, and $\text{GPT-2}_{\text{BASE}}$ demonstrate that such fine-grained operator-level reasoning can significantly reduce communication, at the cost of increased protocol complexity.

Summary and open challenges. Hybrid HE/MPC systems should be viewed not as a single class of protocols, but as points in a multi-dimensional design space that trades protocol simplicity for flexibility. By delegating bandwidth-heavy linear algebra to FHE and retaining MPC or FSS for non-linear control flow, these systems achieve practical efficiency improvements that neither paradigm can deliver alone. At the same time, hybridisation introduces new challenges, including increased protocol-engineering complexity, the cost and correctness of HE–MPC conversion, and the need to ensure that approximations and quantisation strategies preserve model accuracy as systems scale to larger LLMs. Addressing these challenges remains central to making hybrid designs robust, composable, and deployable at scale.

6 TEE-based Solutions

Trusted Execution Environments (TEEs) provide an orthogonal approach to privacy-preserving inference compared to cryptographic techniques such as MPC and FHE [41]. Rather than protecting computation through cryptography, TEEs rely on hardware-isolated execution environments in which code and data are shielded from the rest of the system, including privileged software such as operating systems and hypervisors [14]. Large Language Models execute in plaintext inside the enclave, while confidentiality and integrity are enforced by hardware isolation and remote attestation [11]. As a result, the primary challenges for TEE-based inference are not algorithmic but system-level: understanding performance characteristics on CPU and GPU platforms, managing encrypted memory movement at scale, and ensuring that the surrounding deployment faithfully realises end-to-end confidentiality guarantees [12].

Deployment pattern for confidential inference. In practice, achieving end-to-end confidentiality of *prompt+completion* with TEEs requires a concrete deployment workflow that ties hardware guarantees to the threat model introduced in Section 1 [27]. A standard pattern consists of the following steps. (1) *Remote attestation*: the client verifies the identity and measurement of the enclave workload, ensuring that the expected model, runtime, and configuration are executing inside a genuine TEE. (2) *Key provisioning*: only after successful attestation does the client release session keys or decryption material to the enclave. (3) *Secure channel establishment*: prompts are transmitted to the enclave over an encrypted channel, decrypted and processed in plaintext exclusively inside the enclave, and the resulting completions are encrypted before leaving it. (4) *Output handling*: system policies explicitly forbid plaintext logging, tracing, or telemetry outside the enclave boundary. Optionally, model weights may be loaded into the enclave at startup and sealed or cached across restarts, at the cost of increased cold-start latency [57]. This workflow makes explicit how attestation, key management, and secure communication compose to deliver the end-to-end confidentiality guarantees attributed to TEE-based inference.

CPU and GPU TEEs for large LLMs. Recent evaluations show that this deployment model is compatible with large-scale LLM inference at near-production performance [11]. A systematic study across Intel SGX, Intel TDX, and NVIDIA H100 Confidential Compute demonstrates that TEEs can support full-scale LLMs with overheads that remain within practical bounds. On CPUs, LLaMA-2 models of varying sizes run inside TDX and SGX with throughput and latency closely tracking non-confidential execution, aided by hardware extensions such as AMX [11]. On GPUs, the H100's confidential-compute mode introduces a modest performance penalty that becomes less pronounced for larger batch sizes and longer sequences, as also described in Ref. [67]. The evaluation further covers Retrieval-Augmented Generation pipelines executed entirely inside TEEs, with overheads remaining similarly contained. These results position TEEs as the only currently viable mechanism for end-to-end confidential inference of large LLMs across both CPU and GPU hardware at near-production performance.

Mitigating confidential-compute overheads in GPU memory management. While enclave execution itself is efficient, encrypted data movement between GPU memory and host memory introduces a significant bottleneck when model size exceeds on-device capacity [12]. Large LLMs such as OPT-30B and OPT-66B surpass the 80 GB memory of the H100, leading to frequent swapping of parameters and KV-cache states. In naive confidential-compute configurations, this results in severe throughput degradation. PipeLLM addresses this challenge by introducing speculative pipelined encryption: it predicts upcoming memory accesses, pre-encrypts the corresponding pages, and overlaps encryption with GPU computation [53]. A lightweight pipeline-relinquishing mechanism handles mispredictions. This design sustains throughput close to non-confidential serving systems such as vLLM, PEFT, and FlexGen, substantially reducing the penalty imposed by encrypted memory paging for models ranging from 13B to 175B parameters.

Leakage surface, trust assumptions, and mitigations. Unlike cryptographic approaches, the security of TEE-based inference depends on a hardware trust chain and careful system engineering [40]. Although TEEs protect enclave memory and execution state, they do not automatically prevent all forms of information leakage. In the context of LLM inference, text completion privacy can be compromised by system-level mechanisms such as token-by-token streaming, logging, tracing, prompt caching, and performance metrics [27]. Similarly, coarse-grained metadata—including prompt and completion length, timing, token rate, and I/O patterns—may leak unless explicitly accepted in the threat model or mitigated [59]. The trusted computing base further includes the CPU or GPU hardware, firmware, microcode, and the attestation infrastructure itself, implying that patch

management and timely response to hardware vulnerabilities are integral to the security posture.

Practical deployments therefore pair TEEs with additional mitigations, including hardened enclave runtimes, configuration choices to reduce side-channel exposure (e.g., disabling simultaneous multithreading where appropriate), batching and rate limiting to smooth timing leakage, traffic shaping for streaming outputs, and rigorous auditing of logging and telemetry paths to enforce encryption at enclave boundaries [42].

Summary and trade-offs. TEE-based approaches avoid the heavy communication and cryptographic computation intrinsic to MPC and FHE, yielding overheads that remain small even for multi-tens-of-billions-parameter LLMs [11]. Their limitations arise from the hardware trust model and the need to manage side channels, metadata leakage, and output handling at the system level. Despite these constraints, TEEs currently offer the most practical performance–confidentiality trade-off for large-scale cloud deployment. Moreover, their ability to integrate naturally with lightweight cryptographic and information-reduction techniques makes them an especially flexible and deployable foundation for confidential LLM inference today.

7 Information-Reduction-Based Solutions

The approaches discussed so far place trust either in cryptographic guarantees (MPC and FHE) or in hardware-enforced isolation (TEEs). In contrast, information-reduction strategies rely on a different principle: privacy is achieved by reducing or distorting the information contained in the prompt itself before it is disclosed, rather than by preventing observation altogether. This class of methods targets closed-box LLM deployments accessed through an API, where neither the model nor the hosting infrastructure can be modified.

Information-reduction-based techniques operate entirely on the client side. The client transforms the prompt prior to submission so as to limit the exposure of sensitive information, while preserving the ability of the model to produce a useful response. The scope of protection is therefore explicitly restricted to the prompt content; completions are returned in the clear, and model parameters, inference internals, and querying patterns remain outside the protection boundary. The central challenge in this setting is to balance the amount of information removed or distorted against the resulting loss in output utility, particularly for long or information-dense prompts.

Formal information distortion via differential privacy. Differential privacy (DP) provides a principled framework for limiting the information revealed about individual components of the prompt through controlled randomisation. InferDPT exemplifies this approach for proprietary, closed-box LLMs such as GPT-4 [54]. Its pipeline consists of a DP-based perturbation module followed by an extraction module that reconstructs coherent text from perturbed generations in a distillation-like manner. The core mechanism, RANTEXT, defines adjacency and noise addition over token spans using random adjacency lists, enabling fine-grained control over the privacy–utility trade-off. Experimental results across multiple datasets show that InferDPT preserves output quality close to non-private GPT-4 while providing strong protection against embedding inversion attacks at moderate privacy budgets, outperforming earlier DP-based mechanisms such as SANTEXT+ and CUSTEXT+.

Semantic obfuscation with local restoration. A complementary line of work removes sensitive information explicitly rather than perturbing it statistically. PrivacyRestore adopts a semantic obfuscation strategy based on selective redaction and subsequent local restoration [64]. The server pre-trains restoration vectors associated with categories of sensitive entities, such as names, dates, or identifiers, and distributes them to clients. During inference, the client detects and removes sensitive spans from the prompt, aggregates the corresponding restoration vectors into a meta vector, perturbs this vector under a d_χ-privacy mechanism, and submits the redacted prompt together with the noisy meta vector to the server. After receiving the model's response, the client locally re-inserts the removed information. By decoupling privacy cost from prompt length, this design avoids the cumulative privacy-budget growth typical of token-level DP mechanisms. Evaluations in medical and legal domains show that PrivacyRestore maintains strong task utility while effectively concealing sensitive content.

Summary and applicability. Information-reduction-based techniques require no changes to the model or hosting infrastructure, incur negligible computational overhead, and offer flexible privacy–utility trade-offs that are well suited to API-based deployments. Their guarantees, however, are inherently weaker and more task-dependent than those provided by cryptographic or hardware-based approaches. Differential privacy introduces controlled distortion, with stronger privacy implying greater utility loss, while obfuscation depends on accurate identification of sensitive spans and may still allow leakage through contextual cues. Moreover, these methods do not protect model parameters, inference internals, or adaptive multi-query interactions. In practice, they are best suited to scenarios where deployability and low friction are paramount and where prompt privacy—rather than full end-to-end confidentiality—is the primary concern.

8 Discussion: Trade-Offs and Design Considerations

The surveyed approaches span cryptographic protocols, hardware-isolated execution, and lightweight client-side mechanisms. Rather than forming a linear hierarchy, these techniques occupy distinct regions of a shared design space defined by a small number of common axes: the source of trust (cryptography, hardware, or information reduction), performance overhead and la-

Criterion	MPC	FHE	Hybrid	TEE	Info-Reduction
End-to-end confidentiality	✓	✓	✓	✓†	×
Cryptographic trust only	✓	✓	✓	×	×
Near-plaintext latency	×	×	×	✓	✓
Supports large LLMs (B+)	×	×	×	✓	✓
Online interaction	High	None	Medium	None	None
Deployability (no infra changes)	×	×	×	△	✓

Table 2: Qualitative comparison of PET families for LLM inference. † Confidentiality relies on hardware trust assumptions. △ Deployability depends on the availability of trusted hardware and enclave-enabled cloud infrastructure.

tency, deployability in existing cloud infrastructures, and the strength and scope of the resulting privacy guarantees. Understanding how each family positions itself along these axes is essential for making informed design choices.

Cryptographic trust vs. interaction overhead. MPC-based approaches place trust in cryptographic guarantees and offer strong, composable security under well-defined adversarial models. They naturally support complex control flow and non-linear layers with limited architectural modification. However, interaction and communication remain inherent bottlenecks, particularly for large-scale autoregressive decoding. Recent systems such as BumbleBee, PermLLM, SIGMA, CipherGPT, and SHAFT [36, 66, 24, 26, 29] have substantially reduced these costs, yet still operate orders of magnitude slower than plaintext inference for large LLMs.

Non-interactive confidentiality vs. computational feasibility. FHE-based solutions explore the opposite end of the design space by eliminating interaction entirely. Their appeal lies in non-interactive encrypted inference and a clean client–server separation. In practice, even with GPU-accelerated bootstrapping and encryption-friendly architectures—such as EncryptedLLM [16], Powerformer [44], THOR [39], and end-to-end systems such as NEXUS [65]—the computational overhead of homomorphic matrix multiplication and polynomial approximation limits feasibility to small-sized models and relatively short sequences.

Balancing bandwidth reduction and protocol complexity. Hybrid HE/MPC protocols occupy intermediate points in this space. Systems such as BLB [61] combine homomorphic evaluation of bandwidth-heavy linear layers with MPC-based handling of non-linearities and control flow, reducing communication and interaction while preserving accuracy. These gains come at the cost of increased protocol-engineering complexity and the need for efficient HE–MPC conversion mechanisms.

Near-plaintext performance under hardware trust assumptions. Hardware-based isolation represents a fundamentally different design point. TEE systems built on Intel SGX, Intel TDX, or NVIDIA H100 Confidential Compute [11] achieve near-plaintext performance for full LLM and RAG pipelines by shifting trust from cryptography to hardware-enforced isolation. Techniques such as PipeLLM [53] further mitigate encrypted memory-movement overheads for models that exceed on-device capacity. At present, TEEs are the only approach that supports multi-billion-parameter models with production-level latency, albeit under a hardware trust model with residual side-channel and metadata-leakage considerations.

Maximum deployability with reduced privacy guarantees. Finally, information-reduction-based techniques—such as InferDPT [54] and PrivacyRestore [64]—target closed-box API deployments. By operating entirely on the client side, they achieve maximal deployability and negligible overhead, at the cost of weaker and task-dependent guarantees that are limited to prompt content (i.e., text completion is not protected).

A practical decision framework. Taken together, these trade-offs suggest a pragmatic framework for selecting an appropriate PET in real-world deployments. Table 2 summarises the main considerations across PET families.

Open challenges and research directions. Across all families, several open challenges remain. Cryptographic methods must scale to production-grade autoregressive decoding while preserving accuracy under quantisation and polynomial approximation. Achieving malicious security at full-model scale remains costly, and the composability of heterogeneous PETs is not yet well understood. For TEEs, improving side-channel resilience, managing metadata leakage, and reducing reliance on the hardware and firmware stack remain active areas of research. Finally, the lack of standardised benchmarks hampers meaningful comparison across solutions under comparable threat models and workloads.

9 Outlook and Deployment Roadmap

Within the threat model considered in this work—where the cloud provider and underlying infrastructure are treated as potentially adversarial—our current deployments prioritise privacy-preserving LLM inference as

a trust-minimising process rather than a single-point solution. We explicitly acknowledge that different techniques occupy different points along a continuum of trust, performance, and deployability, and that practical systems must evolve as underlying cryptographic and hardware capabilities mature.

Current approach: TEE-based inference as a pragmatic foundation. Our current deployments prioritise TEE-based inference as a pragmatic foundation for privacy-preserving LLM serving. TEEs are, today, the only approach that simultaneously supports large-scale models, near-plaintext latency, and production-grade deployability. We explicitly acknowledge the associated hardware trust chain and residual leakage surfaces, and mitigate them through careful system design, operational controls, and conservative deployment practices. Importantly, this choice is not viewed as an endpoint, but rather as a practical baseline that enables immediate protection of prompt and completion content at scale.

Medium-term approach: crypto-augmented inference with reduced hardware trust. Our near-term direction focuses on crypto-augmented designs that preserve much of the performance and deployability of TEEs while progressively shifting parts of the trust boundary toward cryptographic guarantees. Concretely, we are exploring formulations in which LLM inference is expressed over enriched algebraic domains, allowing inputs to be masked with noise removable only by the client, while ensuring that model transformations preserve this masking so that outputs remain confidential and client-decodable. These approaches aim to reduce reliance on trusted hardware without incurring the computational overhead of fully homomorphic encryption, and can be viewed as intermediate points along a trust-minimisation trajectory.

Long-term approach: fully cryptographic confidentiality via FHE. In the longer term, fully homomorphic encryption represents the natural asymptotic endpoint of this trajectory: inference directly over ciphertexts, without enclaves or trusted hardware, and with confidentiality guaranteed purely by cryptography. While current performance and latency constraints preclude its widespread deployment for large autoregressive models, progress in protocol design, architecture co-design, and hardware acceleration continues to steadily expand the practical envelope of FHE-based inference. Taken together, this staged roadmap—from TEEs as a pragmatic foundation today, through crypto-augmented intermediate designs, toward fully cryptographic confidentiality in the long term—captures the direction we envision for the evolution of privacy-preserving LLM inference.

10 Conclusion

In this paper, we surveyed the landscape of privacy-preserving inference for Transformer-based LLMs, spanning cryptographic protocols (MPC and FHE), hybrid HE/MPC designs, hardware-isolated execution via TEE, and client-side information-reduction techniques. We examined how these approaches address fundamental challenges such as non-linear layers and autoregressive decoding, and highlighted the architectural and systems-level trade-offs that shape their practical viability. A central conclusion of this survey is that privacy-preserving LLM inference does not admit a single dominant solution. Instead, existing approaches represent different ways of placing trust: in cryptographic hardness, in hardware-enforced isolation, or in deliberate information reduction. Each occupies a well-defined region of the design space, with distinct implications for performance, deployability, and the strength of privacy guarantees. Cryptographic methods continue to advance toward larger models and stronger threat models, yet remain constrained by interaction in MPC and computational cost in FHE. Hybrid designs demonstrate that combining paradigms can yield meaningful efficiency gains, while TEEs currently offer the only practical path to end-to-end confidential inference for very large models at production latency. Lightweight client-side techniques play a complementary role in closed-box settings where prompt privacy is the primary concern. Taken together, these findings suggest that realistic deployments increasingly rely on the careful selection and composition of multiple privacy-enhancing technologies, guided by application-specific requirements and trust assumptions. More broadly, the field is moving along a trust-minimising trajectory, in which reliance on providers and infrastructure is progressively reduced as cryptographic techniques mature. This evolution—incremental, multi-layered, and grounded in practical constraints—defines both the current state and the future direction of privacy-preserving LLM inference.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. Elasticity in cloud computing: state of the art and research challenges. *IEEE Transactions on services computing*, 11(2):430–447, 2017.

[3] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 805–817, 2016.

[4] Fabian Boemer, Yixing Lao, Rosario Cammarota, and Casimir Wierzynski. ngraph-he: a graph compiler for deep learning on homomorphically encrypted data. In *Proceedings of the 16th ACM inter-*

national conference on computing frontiers, pages 3–13, 2019.

[5] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Annual cryptology conference*, pages 505–524. Springer, 2011.

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

[7] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[9] Ke Cheng, Yuheng Xia, Anxiao Song, Jiaxuan Fu, Wenjie Qu, Yulong Shen, and Jiaheng Zhang. Mosformer: Maliciously secure three-party inference framework for large transformers. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 4124–4138, 2025.

[10] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International conference on the theory and application of cryptology and information security*, pages 409–437. Springer, 2017.

[11] Marcin Chrapek, Marcin Copik, Etienne Mettaz, and Torsten Hoefler. Confidential llm inference: Performance and cost across cpu and gpu tees. *arXiv preprint arXiv:2509.18886*, 2025.

[12] Marcin Chrapek, Marcin Copik, Etienne Mettaz, and Torsten Hoefler. Confidential llm inference: Performance and cost across cpu and gpu tees. *arXiv preprint arXiv:2509.18886*, 2025.

[13] Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.

[14] Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.

[15] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

[16] Leo de Castro, Daniel Escudero, Adya Agrawal, Antigoni Polychroniadou, and Manuela Veloso. Encryptedllm: Privacy-preserving large language model inference via gpu-accelerated fully homomorphic encryption. In *Forty-second International Conference on Machine Learning*.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[18] Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Chen. Puma: Secure inference of llama-7b in five minutes. *arXiv preprint arXiv:2307.12533*, 2023.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[20] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

[21] Hao Fang, Yixiang Qiu, Hongyao Yu, Wenbo Yu, Jiawei Kong, Baoli Chong, Bin Chen, Xuan Wang, Shu-Tao Xia, and Ke Xu. Privacy leakage on dnns: A survey of model inversion attacks and defenses. *arXiv preprint arXiv:2402.04013*, 2024.

[22] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

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

[24] Kanav Gupta, Neha Jawalkar, Ananta Mukherjee, Nishanth Chandran, Divya Gupta, Ashish Panwar, and Rahul Sharma. Sigma: Secure gpt inference with function secret sharing. *Cryptology ePrint Archive*, 2023.

[25] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[26] Xiaoyang Hou, Jian Liu, Jingyu Li, Yuhan Li, Wenjie Lu, Cheng Hong, and Kui Ren. Ciphergpt: Secure two-party gpt inference. *Cryptology ePrint Archive*, 2023.

[27] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. Chiron: Privacy-preserving machine learning as a service. *arXiv preprint arXiv:1803.05961*, 2018.

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

[29] Andes YL Kei and Sherman SM Chow. Shaft: Secure, handy, accurate, and fast transformer inference. In *Network and Distributed System Security Symposium, NDSS*, volume 2025, 2025.

[30] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International conference on machine learning*, pages 3499–3508. PMLR, 2019.

[31] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[33] Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P Xing, and Hao Zhang. Mpcformer: fast, performant and private transformer inference with mpc. *arXiv preprint arXiv:2211.01452*, 2022.

[34] Zhengyi Li, Kang Yang, Jin Tan, Wen-jie Lu, Haoqi Wu, Xiao Wang, Yu Yu, Derun Zhao, Yancheng Zheng, Minyi Guo, et al. Nimbus: Secure and efficient two-party inference for transformers. *Advances in Neural Information Processing Systems*, 37:21572–21600, 2024.

[35] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[36] Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and WenGuang Chen. Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*, 2023.

[37] Jinglong Luo, Yehong Zhang, Zhuo Zhang, Jiaqi Zhang, Xin Mu, Hui Wang, Yue Yu, and Zenglin Xu. Secformer: fast and accurate privacy-preserving inference for transformer models via smpc. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13333–13348, 2024.

[38] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[39] Jungho Moon, Dongwoo Yoo, Xiaoqian Jiang, and Miran Kim. Thor: Secure transformer inference with homomorphic encryption. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 3765–3779, 2025.

[40] Antonio Muñoz, Ruben Ríos, Rodrigo Román, and Javier López. A survey on the (in) security of trusted execution environments. *Computers & Security*, 129:103180, 2023.

[41] Krishna Giri Narra, Zhifeng Lin, Yongqin Wang, Keshav Balasubramaniam, and Murali Annavaram. Privacy-preserving inference in machine learning services using trusted execution environments. *arXiv preprint arXiv:1912.03485*, 2019.

[42] Oleksii Oleksenko, Bohdan Trach, Robert Krahn, Mark Silberstein, and Christof Fetzer. Varys: Protecting {SGX} enclaves from practical {Side-Channel} attacks. In *2018 Usenix Annual Technical Conference (USENIX ATC 18)*, pages 227–240, 2018.

[43] Qi Pang, Jinhao Zhu, Helen Möllering, Wenting Zheng, and Thomas Schneider. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4753–4771. IEEE, 2024.

[44] Dongjin Park, Eunsang Lee, and Joon-Woo Lee. Powerformer: Efficient and high-accuracy privacy-preserving language model with homomorphic encryption. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11090–11111, 2025.

[45] Ziwei Peng, Kaiping Xue, Bin Zhu, Yaxuan Huang, Jian Li, Meng Hao, and Tianwei Zhang. Privtf: Efficient and privacy-preserving transformer inference with optimized protocols for secure nlp. *IEEE Transactions on Network Science and Engineering*, 2025.

[46] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[49] Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow2: Practical 2-party secure inference. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 325–342, 2020.

[50] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725, 2016.

[51] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[52] Wenting Zheng Srinivasan, PMRL Akshayaram, and Popa Raluca Ada. Delphi: A cryptographic inference service for neural networks. In *Proc. 29th USENIX secur. symp*, volume 3, 2019.

[53] Yifan Tan, Cheng Tan, Zeyu Mi, and Haibo Chen. Pipellm: Fast and confidential large language model services with speculative pipelined encryption. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pages 843–857, 2025.

[54] Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for closed-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 22(5):4625–4640, 2025.

[55] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[56] Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

[57] Jean-Baptiste Truong, William Gallagher, Tian Guo, and Robert J Walls. Memory-efficient deep learning inference in trusted execution environments. In *2021 IEEE International Conference on Cloud Engineering (IC2E)*, pages 161–167. IEEE, 2021.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[59] Guanlong Wu, Zheng Zhang, Yao Zhang, Weili Wang, Jianyu Niu, Ye Wu, and Yinqian Zhang. I know what you asked: Prompt leakage via kv-cache sharing in multi-tenant llm serving. In *Proceedings of the 2025 Network and Distributed System Security (NDSS) Symposium. San Diego, CA, USA*, 2025.

[60] Haoqi Wu, Wenjing Fang, Yancheng Zheng, Junming Ma, Jin Tan, Yinggui Wang, and Lei Wang. Ditto: Quantization-aware secure inference of transformers upon mpc. *arXiv preprint arXiv:2405.05525*, 2024.

[61] Tianshi Xu, Wen-jie Lu, Jiangrui Yu, Yi Chen, Chenqi Lin, Runsheng Wang, and Meng Li. Breaking the layer barrier: Remodeling private transformer inference with hybrid {CKKS} and {MPC}. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2653–2672, 2025.

[62] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[63] Gokul Yenduri, Muthukumaran Ramalingam, G Chemmalar Selvi, Y Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G Deepti Raj, Rutvij H Jhaveri, B Prabadevi, Weizheng Wang, et al. Gpt (generative pre-trained transformer)—a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE access*, 12:54608–54649, 2024.

[64] Ziqian Zeng, Jianwei Wang, Junyao Yang, Zhengdong Lu, Haoran Li, Huiping Zhuang, and Cen Chen. Privacyrestore: Privacy-preserving inference in large language models via privacy removal and restoration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10821–10855, 2025.

[65] Jiawen Zhang, Xinpeng Yang, Lipeng He, Kejia Chen, Wen-jie Lu, Yinghao Wang, Xiaoyang Hou, Jian Liu, Kui Ren, and Xiaohu Yang. Secure transformer inference made non-interactive. *Cryptology ePrint Archive*, 2024.

[66] Fei Zheng, Chaochao Chen, Zhongxuan Han, and Xiaolin Zheng. Permllm: Private inference of large language models within 3 seconds under wan. *arXiv preprint arXiv:2405.18744*, 2024.

[67] Jianwei Zhu, Hang Yin, and Shunfan Zhou. Confidential computing on nvidia h100 gpu: A performance benchmark study. *arXiv e-prints*, pages arXiv–2409, 2024.