mathematics

MDPI

Article

Balancing Privacy and Robustness in Prompt Learning for Large Language Models

Chiyu Shi [1], Junyu Su [2], Chiawei Chu [1], Baoping Wang [3,*] and Duanyang Feng [1]

[1] Faculty of Data Science City, University of Macau, Macau 999078, China
[2] Faculty of Art and Communication, Kunming University of Science and Technology, Kunming 650032, China
[3] School of Management, Guangdong University of Science and Technology, Dongguan 523070, China
* Correspondence: wangbaoping@gdust.edu.cn

Abstract: This paper tackles the critical issue of privacy in Natural Language Processing (NLP) systems that process sensitive data by introducing a novel framework combining differential privacy and adversarial training. The proposed solution ensures formal privacy guarantees by minimizing the influence of individual data points on the model's behavior, effectively preventing information leakage. Simultaneously, adversarial training is applied to strengthen model robustness against privacy attacks by exposing it to adversarial examples during training. The framework is rigorously evaluated across various NLP tasks, demonstrating its capability to balance privacy preservation with high utility effectively. These results mark a significant advancement in developing secure and reliable NLP systems, particularly for applications requiring stringent data confidentiality, such as healthcare and finance.

Keywords: privacy protextion; large language model; prompt learning

MSC: 68P27

check for updates

Citation: Shi, C.; Su, J.; Chu, C.; Wang, B.; Feng, D. Balancing Privacy and Robustness in Prompt Learning for Large Language Models. *Mathematics* **2024**, *12*, 3359. https://doi.org/10.3390/math12213359

Academic Editor: Florin Leon

Received: 12 September 2024
Revised: 2 October 2024
Accepted: 24 October 2024
Published: 26 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, Natural Language Processing (NLP) has undergone transformative changes, primarily driven by the advent of large-scale pre-trained language models such as GPT-3, BERT, and T5. These models have significantly enhanced the ability of machines to understand and generate human language, resulting in breakthroughs in a wide array of NLP tasks, including text generation, machine translation, sentiment analysis, and question answering. The underlying strength of these models lies in their capacity to learn from vast amounts of textual data, enabling them to generalize effectively to various tasks with minimal additional training. This capability has led to their widespread adoption in diverse fields, ranging from customer service chatbots and content creation to automated translation and personal assistants.

However, the deployment of these powerful models in real-world applications, especially in domains that involve handling sensitive data—such as healthcare, finance, and personal communication—has raised significant concerns about data privacy and security. These concerns are exacerbated in scenarios where large language models are fine-tuned or adapted using datasets that contain sensitive information. The fine-tuning process, which aims to optimize model performance for specific tasks, can inadvertently lead to the leakage of private information through model outputs or gradients. This risk is particularly acute in the context of prompt learning—a technique that has recently gained traction due to its efficiency in adapting pre-trained language models to new tasks.

Prompt learning involves crafting specific prompts that steer the model towards generating desired outputs, thereby reducing the need for extensive task-specific data and simplifying the adaptation process. Despite its advantages, prompt learning introduces unique privacy challenges. Since prompts can elicit responses based on underlying

patterns in the training data, there is a risk that sensitive information may be exposed. Moreover, the interaction between prompts and the model's internal representations can reveal insights into the training data, making it possible for adversaries to extract sensitive information or infer private details.

Addressing these privacy concerns is crucial for ensuring the safe deployment of NLP systems in sensitive applications. As a response to these challenges, privacy-preserving machine learning has emerged as a vital area of research. Among the various techniques developed to enhance data privacy, differential privacy (DP) has garnered considerable attention due to its rigorous theoretical foundations and practical applicability. Differential privacy provides formal privacy guarantees by ensuring that the inclusion or exclusion of any single data point in a dataset has a minimal impact on the model's output. This is achieved by introducing controlled noise into the learning process, which obscures the contribution of individual data points, thus preventing information leakage and making it difficult to infer specific details about the data.

In addition to privacy concerns, the robustness of NLP models against adversarial attacks is a critical issue. Adversarial attacks involve intentionally manipulating model inputs to cause incorrect or misleading outputs, thereby compromising the reliability and security of the models. Adversarial training is a well-established technique to counter these threats, wherein models are trained on adversarially perturbed examples [1]. Through exposing models to these adversarial examples during training, adversarial training enhances the model's ability to resist manipulation and maintain accurate predictions even in the presence of malicious inputs. While adversarial training has traditionally been employed to improve model robustness, it also offers potential benefits for privacy preservation. This is because the techniques used to generate adversarial examples can also be applied to identify and mitigate vulnerabilities that may lead to privacy breaches.

Given the dual challenges of privacy and robustness in NLP systems, this paper proposes a novel framework that integrates differential privacy and adversarial training into the prompt learning paradigm. The goal is to create a privacy-preserving and robust environment for large language models, enabling them to handle sensitive data securely while maintaining high utility and robustness. The proposed framework addresses the need for robust privacy guarantees by incorporating differential privacy into the gradient-based learning process. This approach ensures that the impact of individual data points on the model's behavior is minimized, thereby safeguarding sensitive information and providing formal privacy guarantees. Concurrently, adversarial training is employed to enhance the model's robustness against privacy attacks. By systematically exposing the model to adversarial examples designed to exploit potential vulnerabilities, the framework ensures that the model can withstand attacks aimed at extracting sensitive information or compromising model outputs.

The integration of differential privacy and adversarial training into prompt learning represents a significant advancement in the development of secure NLP systems. This approach not only enhances the privacy guarantees of prompt-based models but also improves their resilience to adversarial threats. The dual protection offered by this framework is particularly relevant in applications where privacy and security are of utmost importance. For example, in healthcare, where patient data must be handled with strict confidentiality under regulations such as HIPAA in the United States and GDPR in the European Union, the proposed model enhances compliance by integrating differential privacy into the NLP learning process. This approach ensures that individual data points are not directly exposed or reconstructed, thereby reducing the risk of privacy breaches. The added layer of adversarial training further protects sensitive information against potential attacks, helping healthcare applications not only meet legal requirements but also improve the robustness of data security under these regulatory frameworks. Similarly, in the financial sector, where the integrity of sensitive transactions and customer data is critical, the framework can safeguard against privacy breaches and ensure the reliability of NLP applications.

The contribution of this paper are summarized as follows:

- We introduce a novel framework that combines differential privacy with adversarial training within the context of prompt learning. This integration ensures that NLP models can handle sensitive data securely while maintaining robustness against adversarial attacks.
- By incorporating differential privacy into the gradient-based learning process, the proposed framework offers strong privacy guarantees. This approach minimizes the impact of individual data points on the model's behavior, thereby preventing information leakage and safeguarding sensitive data.
- Our framework employs adversarial training to expose models to potential vulnerabilities systematically. This exposure enhances the model's ability to withstand adversarial attacks that could otherwise extract sensitive information or manipulate model outputs.

The remainder of this paper is structured as follows: Section 2 reviews related work, covering the evolution of prompt learning in NLP, key methodologies, and associated challenges, along with advancements in differential privacy and adversarial training. Section 3 describes the proposed methodology, detailing the framework's structure and the integration of differential privacy and adversarial training within prompt learning. Section 4 presents experimental results, evaluating the framework's performance across various NLP tasks, focusing on its privacy-preserving capabilities and robustness to adversarial attacks. Finally, Section 5 concludes the paper by summarizing the key findings, discussing their implications for the future of privacy-preserving NLP systems, and suggesting directions for further research.

2. Related Work

2.1. Prompt Learning in NLP and Large Language Models

Prompt learning has rapidly become one of the most influential techniques in Natural Language Processing, especially with the advent of large pre-trained language models [2,3]. By leveraging the extensive pre-training of models like GPT, BERT, T5, and others, prompt learning enables the adaptation of these models to a wide variety of downstream tasks with minimal additional training. This section provides a detailed exploration of the evolution of prompt learning, key methodologies, significant applications, ongoing challenges, and emerging research directions [4–6]. A comparison of related works in given in Table 1.

The concept of prompt learning in NLP can be traced back to early methods in transfer learning and few-shot learning [7,8]. These methods aimed to adapt pre-trained models to specific tasks with limited task-specific data, but the traditional approaches often required fine-tuning the entire model, which was both computationally expensive and prone to overfitting [9]. Early approaches to transfer learning involved task-specific fine-tuning, where the model was retrained on a small labeled dataset relevant to the new task [10,11]. This approach was effective but computationally costly, especially for large models. Additionally, the lack of sufficient task-specific data often led to overfitting, reducing the generalization ability of the model [12,13].

The introduction of prompt learning marked a significant shift in this paradigm. Instead of fine-tuning the entire model, prompt learning involves crafting input prompts that frame the task as a language modeling problem, effectively reusing the knowledge already embedded in the pre-trained model [14,15]. This approach leverages the model's existing capabilities to perform a new task, thereby reducing the need for extensive task-specific training and mitigating the risk of overfitting [16]. The initial success of prompt learning in few-shot learning scenarios demonstrated its potential to significantly reduce the data and computational requirements for adapting models to new tasks [17]. This early work laid the foundation for a series of innovations that have expanded the applicability and effectiveness of prompt learning across a wide range of NLP tasks [17,18].

Table 1. Comparison of related works.

Method	Advantages	Disadvantages	Primary Findings
Differential Privacy in NLP	Provides formal privacy guarantees by limiting individual data impact.	Degrades model utility and performance due to noise.	Balances privacy and utility effectively across NLP tasks.
Adversarial Training in NLP	Enhances model robustness against attacks.	Reduces accuracy on clean data; increases complexity.	Maintains model performance even under privacy constraints.
Prompt Learning for LLMs	Adapts pre-trained models efficiently with minimal training.	Risks exposing sensitive data via prompt responses.	Reduces data needs while preventing overfitting.
Privacy Attacks in ML	Identifies model vulnerabilities leading to privacy breaches.	Risks data confidentiality and inference of private details.	Highlights the need for privacy-preserving techniques.

2.2. Automated Prompt Engineering

Automated prompt engineering addresses the limitations of manual prompt design by using algorithms to generate and optimize prompts [19–21]. This approach leverages machine learning techniques to explore a large space of potential prompts and identify those that yield the best performance on a given task [22]. One common method in automated prompt engineering is prompt search, where a search algorithm iteratively generates and evaluates prompts based on a performance metric [23]. This search can be conducted using techniques such as grid search, random search, or more sophisticated methods like evolutionary algorithms [24]. The goal is to identify prompts that maximize task performance while minimizing the need for human intervention [25].

Automated prompt engineering has the advantage of scalability, as it can generate and evaluate a large number of prompts in a systematic manner [18]. It also reduces the reliance on human expertise, making it accessible for a broader range of tasks. However, it introduces new challenges, such as the need for efficient search algorithms and the risk of overfitting to specific prompts during the meta-learning process [26].

2.3. Privacy Attacks in Machine Learning

Privacy attacks in machine learning have become a critical area of research, particularly as machine learning models are increasingly deployed in sensitive applications such as healthcare, finance, and personalized services [27]. These attacks exploit vulnerabilities in models to extract private information about the training data, potentially leading to significant breaches of confidentiality. This section reviews the major types of privacy attacks, key methodologies, and ongoing challenges in mitigating these threats [28]. One of the earliest and most studied forms of privacy attacks is the membership inference attack. In this type of attack, an adversary aims to determine whether a specific data point was part of the training dataset used to train a machine learning model [29]. Membership inference attacks exploit the overfitting tendencies of models, where the model behaves differently on training data compared to unseen data. For example, Shokri et al. demonstrated that adversaries could use shadow models to mimic the behavior of the target model and predict membership status with high accuracy [30].

Gradient-based attacks are another critical area of concern. These attacks involve adversaries with access to the model's gradients, typically in federated learning or collaborative training scenarios [31]. By analyzing gradients, adversaries can infer sensitive information about the training data, such as individual data records or even entire input features. For example, Zhu et al. demonstrated that an attacker could reconstruct high-quality images from gradients in a collaborative learning environment [32].

Recent advancements in privacy-preserving techniques, such as differential privacy [33,34], aim to mitigate these risks by adding noise to the data or model outputs, thus obscuring

the contribution of any individual data point. However, the trade-off between privacy and model utility remains a significant challenge, as overly aggressive noise addition can degrade the model's performance [35]. Research continues to explore ways to balance privacy and utility, with approaches such as private aggregation of teacher ensembles (PATE) [36] and local differential privacy [37] gaining attention. In addition, adversarial attacks also intersect with privacy concerns, as adversarially perturbed inputs can reveal vulnerabilities in model training and data handling processes [38,39]. These attacks, while primarily aimed at degrading model performance, can also be used to exploit privacy weaknesses, such as identifying sensitive attributes of the input data.

3. Methodology

Our proposed framework for privacy-preserving prompt learning in NLP systems integrates differential privacy (DP) and adversarial training to create a robust and secure environment for handling sensitive data. This section provides an in-depth and rigorous overview of the framework's structure, detailing the mathematical formulations and interactions that ensure privacy while maintaining the effectiveness of NLP models.

3.1. Differential Privacy

We adopt differential privacy during the gradient update process. Consider a dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, where $x_i \in \mathcal{X}$ denotes an individual data point. Let $\theta \in \mathbb{R}^d$ represent the parameters of the model, and $\mathcal{L}(\theta; \mathcal{D})$ denote the empirical loss function, defined as:

$$\mathcal{L}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; x_i), \tag{1}$$

where $\ell(\theta; x_i)$ represents the loss incurred by the model on the individual data point x_i. The objective is to minimize $\mathcal{L}(\theta; \mathcal{D})$ through gradient descent. At each iteration t, the gradient of the loss function with respect to the model parameters is computed as:

$$g^{(t)} = \nabla \mathcal{L}(\theta^{(t)}; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \nabla \ell(\theta^{(t)}; x_i). \tag{2}$$

To ensure differential privacy, we perturb the gradient $g^{(t)}$ by adding Gaussian noise:

$$\tilde{g}^{(t)} = g^{(t)} + \mathcal{N}(0, \sigma^2 I_d), \tag{3}$$

where $\mathcal{N}(0, \sigma^2 I_d)$ is a multivariate Gaussian distribution with mean zero and covariance matrix $\sigma^2 I_d$, and I_d is the $d \times d$ identity matrix. The added noise ensures that the influence of any individual data point on the gradient is obfuscated, providing a guarantee of privacy. The model parameters are updated using the perturbed gradient:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \tilde{g}^{(t)}, \tag{4}$$

where $\eta > 0$ is the learning rate. This update rule ensures that the parameter updates are differentially private, meaning that the presence or absence of a single data point in $\mathcal{D}$ does not significantly alter the model's behavior.

To quantify the privacy guarantees offered by differential privacy, consider two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ by at most one data point. A randomized mechanism $\mathcal{A}(\cdot)$, which maps input datasets to a distribution over outputs, is said to be (ϵ, δ)-differentially private if, for any measurable subset S of the output space, the following condition holds:

$$\Pr[\mathcal{A}(\mathcal{D}) \in S] \leq \exp(\epsilon) \Pr[\mathcal{A}(\mathcal{D}') \in S] + \delta, \tag{5}$$

where $\epsilon > 0$ is the privacy budget, determining the strength of the privacy guarantee, and $\delta \geq 0$ is a small probability indicating the chance of privacy violation. A smaller ϵ

implies stronger privacy, and δ allows for a controlled relaxation of strict privacy. The sensitivity of the gradient, Δg, plays a crucial role in determining how much influence a single data point can exert on the model's parameters. It is defined as the maximum change in the gradient vector between two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$:

$$\Delta g = \max_{\mathcal{D},\mathcal{D}'} \|g^{(t)}(\mathcal{D}) - g^{(t)}(\mathcal{D}')\|_2, \tag{6}$$

where $g^{(t)}(\mathcal{D})$ is the gradient of the model's loss function at iteration t computed over dataset $\mathcal{D}$, and $\|\cdot\|_2$ denotes the ℓ_2-norm, measuring the magnitude of the vector difference. Bounding Δg is essential for maintaining differential privacy, as it controls how much a single data point can influence the learning process, thereby limiting privacy leakage.

To satisfy (ϵ, δ)-differential privacy, the variance of the noise added to the gradient must be calibrated to the sensitivity Δg:

$$\sigma \geq \frac{\Delta g}{\epsilon} \sqrt{2 \log\left(\frac{1.25}{\delta}\right)}. \tag{7}$$

This ensures that the gradient update mechanism satisfies the desired differential privacy guarantees.

3.2. Adversarial Training

Adversarial training is a technique used to improve the robustness of the model by training it on adversarial examples. These adversarial examples are generated by adding small, carefully crafted perturbations to the input data, which are designed to maximize the model's loss. The perturbation for an input data point x_i is computed as:

$$\tilde{x}_i = x_i + \epsilon \cdot \text{sign}(\nabla_{x_i} \ell(\theta; x_i)), \tag{8}$$

where $\epsilon > 0$ controls the magnitude of the perturbation, and $\text{sign}(\cdot)$ denotes the sign function, which is applied element-wise to the gradient of the loss with respect to the input x_i. The loss incurred by the model on these adversarial examples is given by:

$$\mathcal{L}_{\text{adv}}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; \tilde{x}_i). \tag{9}$$

To train a model that is robust to adversarial attacks, we minimize a combination of the original loss and the adversarial loss:

$$\mathcal{L}_{\text{total}}(\theta; \mathcal{D}) = (1 - \lambda)\mathcal{L}(\theta; \mathcal{D}) + \lambda \mathcal{L}_{\text{adv}}(\theta; \mathcal{D}), \tag{10}$$

where $\lambda \in [0, 1]$ is a hyperparameter that controls the trade-off between natural training and adversarial training. The parameter λ can be tuned depending on the desired level of robustness against adversarial attacks.

3.3. Gradient Descent with Differential Privacy and Adversarial Training

In this section, we explore the integration of differential privacy (DP) and adversarial training within the gradient descent optimization framework. The goal is to ensure that the learning process not only protects individual data privacy but also enhances the model's robustness against adversarial attacks. This approach is critical in scenarios where models are deployed in environments susceptible to both privacy breaches and adversarial manipulations.

We revisit gradient descent, commonly used for optimizing model parameters θ. Given a dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, the gradient of the loss function $\mathcal{L}(\theta; \mathcal{D})$ at iteration t is:

$$g^{(t)} = \frac{1}{N} \sum_{i=1}^{N} \nabla \ell(\theta^{(t)}; x_i), \tag{11}$$

where $\ell(\theta; x_i)$ is the loss for data point x_i. To incorporate differential privacy, the gradient is perturbed by adding Gaussian noise, and the model parameters are updated simultaneously as:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \left(g^{(t)} + \mathcal{N}(0, \sigma^2 I_d) \right), \tag{12}$$

where $\eta > 0$ is the learning rate, σ controls the noise scale, and I_d is the identity matrix. This combined step ensures privacy by limiting the impact of any single data point on the gradient. The update rule for the model parameters under differential privacy becomes:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \tilde{g}^{(t)}, \tag{13}$$

This update rule provides a formal privacy guarantee by ensuring that the output of the learning algorithm is statistically indistinguishable when any single data point is added or removed from the dataset, within the bounds defined by the differential privacy parameters ϵ and δ.

Next, we consider the incorporation of adversarial training into the gradient descent process. Adversarial training is a technique that enhances the model's robustness by training it on adversarial examples—inputs that have been intentionally perturbed to maximize the model's loss. The perturbation for a given input x_i is computed as follows:

$$\tilde{x}_i = x_i + \epsilon \cdot \text{sign}(\nabla_{x_i} \ell(\theta; x_i)), \tag{14}$$

Here, $\epsilon > 0$ is a small constant that controls the magnitude of the perturbation, and $\text{sign}(\cdot)$ is the element-wise sign function, which returns the sign of each component of the gradient of the loss with respect to the input x_i. This perturbation is designed to push the input x_i in the direction that maximizes the loss, thereby creating a worst-case scenario that the model must learn to handle. The loss function for the adversarial examples is defined as:

$$\mathcal{L}_{\text{adv}}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; \tilde{x}_i), \tag{15}$$

where $\tilde{x}_i$ represents the adversarially perturbed input. The gradient of this adversarial loss with respect to the model parameters is given by:

$$g_{\text{adv}}^{(t)} = \nabla \mathcal{L}_{\text{adv}}(\theta^{(t)}; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \nabla \ell(\theta^{(t)}; \tilde{x}_i). \tag{16}$$

To ensure that the adversarial gradient $g_{\text{adv}}^{(t)}$ also satisfies differential privacy, we perturb it with Gaussian noise in the same manner as the original gradient:

$$\tilde{g}_{\text{adv}}^{(t)} = g_{\text{adv}}^{(t)} + \mathcal{N}(0, \sigma^2 I_d), \tag{17}$$

This step is crucial because it guarantees that the privacy of the dataset is preserved even when training on adversarially perturbed inputs. The noise added to both the original and adversarial gradients ensures that the model's updates remain differentially private throughout the training process.

The total gradient used for updating the model parameters is a weighted combination of the differentially private gradients from both the original and adversarial losses:

$$\tilde{g}_{\text{total}}^{(t)} = (1-\lambda)\tilde{g}^{(t)} + \lambda\tilde{g}_{\text{adv}}^{(t)}, \tag{18}$$

where $\lambda \in [0,1]$ is a hyperparameter that controls the trade-off between focusing on the original loss and the adversarial loss. The parameter λ plays a critical role in balancing privacy, robustness, and utility. A larger λ increases the emphasis on adversarial training, which can enhance robustness but may require more noise to maintain privacy, potentially degrading model performance. Conversely, a smaller λ places more focus on the original loss, which might preserve utility but could leave the model more vulnerable to adversarial attacks. The model parameters are then updated using the combined differentially private gradient:

$$\theta^{(t+1)} = \theta^{(t)} - \eta\tilde{g}_{\text{total}}^{(t)}, \tag{19}$$

This update rule reflects the integration of differential privacy and adversarial training within a unified gradient descent framework. The noise added to both the original and adversarial gradients ensures that the model's updates adhere to differential privacy constraints while simultaneously improving the model's robustness against adversarial attacks.

3.4. Privacy–Utility Trade-Off

The integration of differential privacy (DP) and adversarial training within a gradient descent framework introduces a multi-faceted trade-off among privacy, utility, and robustness. In this section, we explore this trade-off in greater mathematical depth, examining how various parameters influence the learning dynamics and the resulting performance of the model.

3.4.1. Differential Privacy: Mathematical Impact on Utility

Differential privacy ensures that the behavior of the learning algorithm is stable with respect to changes in individual data points. This is achieved by adding noise to the gradient updates, a process that inherently introduces randomness into the learning procedure. Mathematically, the noise added is typically Gaussian, and the perturbed gradient at iteration t is given by:

$$\tilde{g}^{(t)} = g^{(t)} + \mathcal{N}(0, \sigma^2 I_d), \tag{20}$$

where $g^{(t)} = \nabla\mathcal{L}(\theta^{(t)}; \mathcal{D})$ is the true gradient of the loss function $\mathcal{L}(\theta; \mathcal{D})$ with respect to the model parameters θ, and $\mathcal{N}(0, \sigma^2 I_d)$ represents Gaussian noise with covariance matrix $\sigma^2 I_d$. The parameter σ controls the scale of the noise, and it is determined based on the desired level of privacy, characterized by the (ϵ, δ) parameters. Specifically, the noise scale σ is chosen to satisfy the differential privacy constraint:

$$\sigma \geq \frac{\Delta g}{\epsilon}\sqrt{2\log\left(\frac{1.25}{\delta}\right)}, \tag{21}$$

where Δg is the global sensitivity of the gradient, defined as:

$$\Delta g = \max_{\mathcal{D},\mathcal{D}'} \|g(\mathcal{D}) - g(\mathcal{D}')\|_2, \tag{22}$$

with $\mathcal{D}$ and $\mathcal{D}'$ being neighboring datasets that differ by a single data point. The introduction of this noise has a direct impact on the convergence of the gradient descent algorithm. The expected update to the model parameters at iteration t is now:

$$\mathbb{E}[\theta^{(t+1)}] = \theta^{(t)} - \eta\mathbb{E}[\tilde{g}^{(t)}] = \theta^{(t)} - \eta g^{(t)}, \tag{23}$$

where $\eta > 0$ is the learning rate. Although the expectation of the perturbed gradient $\tilde{g}^{(t)}$ equals the true gradient $g^{(t)}$, the variance introduced by the noise affects the magnitude and direction of the updates. This can be captured by analyzing the variance of the perturbed gradient:

$$\mathrm{Var}(\tilde{g}^{(t)}) = \mathrm{Var}(g^{(t)}) + \sigma^2 I_d. \tag{24}$$

The additional noise term $\sigma^2 I_d$ increases the overall variance of the gradient estimates, which can lead to slower convergence and may cause the model to converge to a suboptimal solution. The impact of this increased variance on the loss function's expected decrease per iteration can be analyzed using the Taylor expansion around $\theta^{(t)}$:

$$\mathbb{E}[\mathcal{L}(\theta^{(t+1)})] \approx \mathcal{L}(\theta^{(t)}) - \eta \|\nabla \mathcal{L}(\theta^{(t)})\|^2 + \frac{\eta^2}{2} \mathbb{E}[\|\tilde{g}^{(t)}\|^2], \tag{25}$$

where $\|\nabla \mathcal{L}(\theta^{(t)})\|$ denotes the norm of the gradient of the loss function at iteration t. The expectation $\mathbb{E}[\|\tilde{g}^{(t)}\|^2]$ can be decomposed as:

$$\mathbb{E}[\|\tilde{g}^{(t)}\|^2] = \|\nabla \mathcal{L}(\theta^{(t)})\|^2 + \mathrm{Tr}(\sigma^2 I_d), \tag{26}$$

where $\mathrm{Tr}(\sigma^2 I_d) = d\sigma^2$ is the trace of the covariance matrix of the added noise. Substituting this into the expected loss decrease, we obtain:

$$\mathbb{E}[\mathcal{L}(\theta^{(t+1)})] \approx \mathcal{L}(\theta^{(t)}) - \eta \|\nabla \mathcal{L}(\theta^{(t)})\|^2 + \frac{\eta^2}{2} (\|\nabla \mathcal{L}(\theta^{(t)})\|^2 + d\sigma^2). \tag{27}$$

3.4.2. Adversarial Training: Impact on Utility and Robustness

Adversarial training modifies the learning process by incorporating adversarial examples into the training data, thereby improving the model's robustness to adversarial attacks. The adversarial examples are generated by perturbing the input data points in the direction that maximizes the model's loss. Mathematically, this can be expressed as:

$$\tilde{x}_i = x_i + \epsilon \cdot \mathrm{sign}(\nabla_{x_i} \ell(\theta; x_i)), \tag{28}$$

where $\epsilon > 0$ is the perturbation magnitude, and $\mathrm{sign}(\cdot)$ is the element-wise sign function. The adversarial loss function $\mathcal{L}_{\mathrm{adv}}(\theta; \mathcal{D})$ is then given by:

$$\mathcal{L}_{\mathrm{adv}}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; \tilde{x}_i). \tag{29}$$

The gradient of the adversarial loss with respect to the model parameters is:

$$g_{\mathrm{adv}}^{(t)} = \nabla \mathcal{L}_{\mathrm{adv}}(\theta^{(t)}; \mathcal{D}), \tag{30}$$

Then, incorporating adversarial training into the learning process alters the optimization landscape, as the model must now minimize a loss function that accounts for worst-case perturbations of the input data. The adversarial training update rule is:

$$\theta^{(t+1)} = \theta^{(t)} - \eta g_{\mathrm{adv}}^{(t)}. \tag{31}$$

Adversarial training generally makes the optimization problem more challenging because the adversarial loss function $\mathcal{L}_{\mathrm{adv}}(\theta; \mathcal{D})$ is non-convex and often more complex than the original loss function $\mathcal{L}(\theta; \mathcal{D})$. As a result, the model may converge more slowly, and the risk of converging to a suboptimal solution increases. This trade-off between

robustness and utility is governed by the hyperparameter λ, which controls the relative weight of adversarial training in the overall loss function:

$$\mathcal{L}_{\text{total}}(\theta; \mathcal{D}) = (1 - \lambda)\mathcal{L}(\theta; \mathcal{D}) + \lambda \mathcal{L}_{\text{adv}}(\theta; \mathcal{D}). \tag{32}$$

The corresponding gradient of the total loss is:

$$\nabla \mathcal{L}_{\text{total}}(\theta; \mathcal{D}) = (1 - \lambda)\nabla \mathcal{L}(\theta; \mathcal{D}) + \lambda \nabla \mathcal{L}_{\text{adv}}(\theta; \mathcal{D}). \tag{33}$$

After applying differential privacy, the perturbed gradient becomes:

$$\tilde{g}_{\text{total}}^{(t)} = (1 - \lambda)\Big(g^{(t)} + \mathcal{N}(0, \sigma^2 I_d)\Big) + \lambda\Big(g_{\text{adv}}^{(t)} + \mathcal{N}(0, \sigma^2 I_d)\Big). \tag{34}$$

This perturbed gradient is used to update the model parameters:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \tilde{g}_{\text{total}}^{(t)}. \tag{35}$$

The introduction of both differential privacy and adversarial training modifies the learning dynamics in several ways. First, the noise added for differential privacy increases the variance of the gradient estimates, which can slow down convergence and lead to less accurate final models. Second, the adversarial training component increases the complexity of the loss landscape, potentially making it harder for the model to converge to a globally optimal solution. To analyze the combined impact on utility, we can examine the expected decrease in the total loss function per iteration:

$$\mathbb{E}[\mathcal{L}\text{total}(\theta^{(t+1)})] \approx \mathcal{L}\text{total}(\theta^{(t)}) - \eta |\nabla \mathcal{L}\text{total}(\theta^{(t)})|^2 + \frac{\eta^2}{2}\mathbb{E}[|\tilde{g}\text{total}^{(t)}|^2]. \tag{36}$$

Expanding the norm of the perturbed gradient, we have:

$$\mathbb{E}[|\tilde{g}\text{total}^{(t)}|^2] = |\nabla \mathcal{L}\text{total}(\theta^{(t)})|^2 + \text{Var}(\tilde{g}_{\text{total}}^{(t)}), \tag{37}$$

where the variance of the total perturbed gradient is given by:

$$\text{Var}(\tilde{g}_{\text{total}}^{(t)}) = (1 - \lambda)^2 \sigma^2 d + \lambda^2 \sigma^2 d = \sigma^2 d\Big((1 - \lambda)^2 + \lambda^2\Big). \tag{38}$$

Substituting this back into the expected loss decrease:

$$\begin{aligned} \mathbb{E}[\mathcal{L}\text{total}(\theta^{(t+1)})] \approx \mathcal{L}\text{total}(\theta^{(t)}) - \eta |\nabla \mathcal{L}\text{total}(\theta^{(t)})|^2 \\ + \frac{\eta^2}{2}\Big(|\nabla \mathcal{L}\text{total}(\theta^{(t)})|^2 + \sigma^2 d\Big((1 - \lambda)^2 + \lambda^2\Big)\Big). \end{aligned} \tag{39}$$

4. Experiment

4.1. Settings

The experimental setup is designed to rigorously evaluate the proposed privacy-preserving prompt learning framework using the BERT model. BERT, known for its deep contextual understanding, is fine-tuned on three NLP tasks: sentiment analysis, question answering, and topic classification, utilizing the IMDB Movie Reviews, SQuAD, and AG News datasets, respectively. Each task employs carefully crafted prompts to align with BERT's pre-training objectives and maximize task performance.

For sentiment analysis on the IMDB dataset, the prompt is structured to guide BERT in identifying whether a movie review is positive or negative. In the SQuAD dataset for question answering, the prompts are designed to direct BERT to extract the correct answer span from a passage. For the AG News topic classification task, the prompts help BERT classify news articles into one of four categories: World, Sports, Business, or Science.

To ensure privacy preservation, differential privacy is integrated into the training process. We experiment with privacy budgets (ϵ) of 1.0, 0.5, and 0.1, each corresponding to different levels of privacy guarantees. The noise scale (σ) added to the gradients is calculated based on the chosen privacy budget, ensuring that the impact of individual data points on model predictions is minimized. Gradients are clipped to a norm of 1.0 before noise addition to maintain bounded sensitivity, which is crucial for upholding the differential privacy guarantees.

Adversarial training is incorporated to enhance the model's robustness against adversarial attacks. Adversarial examples are generated using the Fast Gradient Sign Method (FGSM) with perturbation magnitudes (ϵ_{adv}) set to 0.01 and 0.05. These examples are introduced during training to ensure that BERT learns to resist manipulations aimed at compromising model predictions. The trade-off between standard training and adversarial training is controlled by the hyperparameter λ, with values of 0.1, 0.3, and 0.5 explored to understand their impact on robustness and model utility.

The fine-tuning of BERT is conducted using the Adam optimizer, configured with a learning rate of 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. A batch size of 16 is consistently used across all experiments, ensuring that the model has sufficient data per update while balancing memory usage on the GPU. The model is trained for up to 10 epochs, with early stopping applied if validation performance does not improve over three consecutive epochs, preventing overfitting. Dropout with a rate of 0.1 is employed to further mitigate the risk of overfitting during fine-tuning.

The implementation of the experiments is performed using the Hugging Face Transformers library, which provides robust tools for model fine-tuning and evaluation. The training is carried out on NVIDIA V100 GPUs, which are capable of handling the computational demands of fine-tuning large-scale models like BERT.

Evaluation metrics are carefully chosen to comprehensively assess the performance of the framework. For sentiment analysis and topic classification, accuracy is the primary metric, while the F1 score is used to provide additional insight, particularly for imbalanced datasets. In the SQuAD question answering task, Exact Match (EM) and F1 scores are used to measure the model's ability to correctly predict answer spans. The robustness of the model is evaluated by introducing adversarial examples during testing and measuring the performance drop compared to clean data. Privacy is quantified by the privacy budget ϵ, and the corresponding utility degradation is analyzed to assess the effectiveness of the privacy-preserving mechanisms.

The experiments were conducted on a high-performance computing platform with the following specifications: an NVIDIA Tesla V100 GPU (32 GB memory), 256 GB of RAM, and an Intel Xeon Gold 6248 CPU. The system ran on Ubuntu 20.04 LTS, with Python 3.8 as the main programming language. Key libraries used included TensorFlow 2.5 and PyTorch 1.9, which provided support for deep learning models and differential privacy frameworks. Adversarial attacks and training were implemented using the Adversarial Robustness Toolbox (ART) and Differential Privacy for PyTorch (Opacus).

4.2. Results and Analysis

Results from Tables 2–4 for sentiment analysis on the IMDB dataset demonstrate that the accuracy of the BERT model decreases as the privacy budget (ϵ) is reduced, reflecting the trade-off between privacy and model utility. Without privacy constraints, the model achieves a high accuracy of 94.5%, which gradually declines to 89.2% at $\epsilon = 0.1$. Introducing adversarial training, with increasing values of the hyperparameter λ, generally leads to a further reduction in accuracy on clean data. However, it significantly improves the model's robustness, as evidenced by the increase in accuracy on adversarially perturbed data, reaching up to 90.2% when $\lambda = 1.0$.

In the SQuAD question–answering task, the Exact Match (EM) and F1 scores follow a similar trend, where stronger privacy guarantees (ϵ) result in lower performance. The model's EM/F1 scores start at 81.2%/88.3% without privacy but decrease to 71.5%/80.2% at $\epsilon = 0.1$.

The incorporation of adversarial training helps to mitigate the performance drop on adversarial examples, with the most notable improvement seen when $\lambda = 1.0$, where the model achieves EM/F1 scores of 79.8%/87.0% on adversarially perturbed data, demonstrating enhanced robustness.

Overall, these results underscore the delicate balance between privacy, utility, and robustness in the BERT model's performance across different NLP tasks. As the privacy constraints become more stringent, there is a clear reduction in accuracy and EM/F1 scores. Nonetheless, the inclusion of adversarial training enhances the model's resistance to adversarial attacks, particularly as the strength of adversarial training (λ) increases. The framework's ability to maintain relatively high performance even under stringent privacy settings highlights its effectiveness in managing the trade-offs inherent in privacy-sensitive NLP applications.

Table 2. Impact of privacy budget on sentiment analysis accuracy (%) on IMDB dataset.

Privacy Budget (ϵ)	$\lambda = 0$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 1.0$
No Privacy	94.5	93.7	92.8	90.5
1.0	93.8	92.9	91.7	89.3
0.5	92.5	91.4	90.2	87.9
0.3	91.1	90.0	88.8	86.5
0.1	89.2	87.3	85.8	83.4

Table 3. Adversarial accuracy (%) on IMDB dataset for different values of λ.

λ	**0**	**0.1**	**0.3**	**0.5**	**0.7**	**1.0**
Adversarial Accuracy	72.3	81.3	85.1	87.6	89.0	90.2

Table 4. Question answering on SQuAD dataset (EM/F1 %).

Privacy Budget (ϵ)	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 1.0$
No Privacy	81.2/88.3	80.5/87.6	79.8/87.1	78.3/86.2	77.0/85.0	75.5/83.7
1.0	79.5/86.7	78.9/86.0	77.9/85.2	76.4/84.3	75.0/83.1	73.5/81.7
0.5	76.2/84.3	75.6/83.6	75.3/82.5	74.1/81.2	72.8/80.0	71.3/78.4
0.3	74.1/82.0	73.5/81.4	73.2/80.3	72.0/79.0	70.7/77.7	69.2/76.1
0.1	71.5/80.2	70.8/79.5	69.4/78.1	68.3/77.0	66.9/75.7	65.4/74.2
Adversarial EM/F1	63.4/72.1	70.3/78.7	74.8/82.5	76.9/84.2	78.5/85.8	79.8/87.0

Table 5 presents results for topic classification on the AG News dataset indicate that the BERT model's accuracy decreases as the privacy budget (ϵ) is reduced, reflecting the trade-off between privacy and model performance. Without privacy constraints, the model achieves an accuracy of 93.1%, which gradually declines to 85.8% at $\epsilon = 0.1$. The introduction of adversarial training with increasing values of the hyperparameter λ further reduces accuracy on clean data but significantly enhances robustness against adversarial attacks, with adversarial accuracy improving from 67.8% (no adversarial training) to 86.3% at $\lambda = 1.0$. This demonstrates the effectiveness of adversarial training in maintaining model robustness, even as privacy constraints are tightened.

Figure 1 presents the F1 score performance for three NLP tasks—Sentiment Analysis, Question Answering, and Topic Classification—under varying privacy budgets (ϵ) and adversarial training strengths (λ). Each subplot corresponds to a different task and demonstrates how increasing λ values generally lead to a decrease in F1 scores, indicating a trade-off between robustness to adversarial attacks and model accuracy. For Sentiment Analysis, the F1 score starts high but shows a noticeable decline as λ increases, especially under stricter privacy settings (lower ϵ).

Table 5. Topic classification on AG News dataset (Accuracy %).

Privacy Budget (ϵ)	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 1.0$
No Privacy	93.1	92.7	92.4	91.6	91.0	89.8
1.0	91.7	91.3	90.8	89.9	89.3	87.8
0.5	89.4	89.0	88.3	87.2	86.5	84.9
0.3	87.6	87.2	86.5	85.3	84.6	83.0
0.1	85.8	85.2	84.1	82.9	82.0	80.4
Adversarial Accuracy	67.8	76.5	80.9	83.4	85.0	86.3

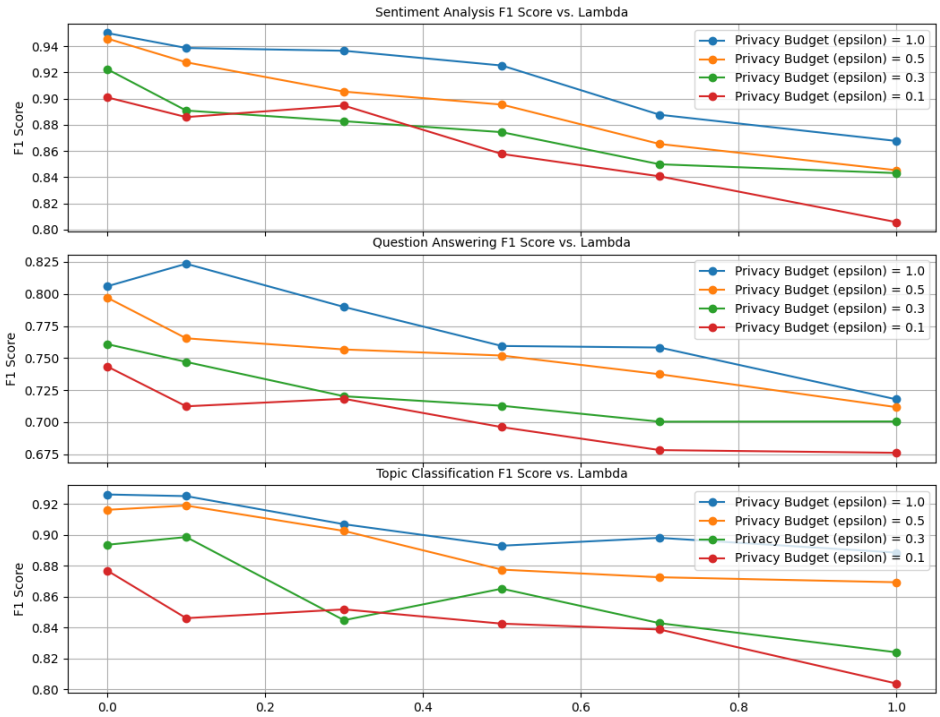


Figure 1. F1 score performance across different privacy budgets and adversarial training strengths for NLP tasks.

In the Question Answering task, a similar trend is observed, with F1 scores gradually decreasing as adversarial training strength grows, highlighting the challenge of balancing precision and recall while maintaining model robustness. The Topic Classification subplot also shows a consistent decline in F1 scores with higher λ values, although the impact of the privacy budget is slightly less pronounced compared to the other tasks. These results underline the critical trade-offs in machine learning model design, where increasing adversarial training to protect against attacks can degrade performance, particularly when combined with stringent privacy requirements. Thus, this analysis emphasizes the importance of carefully tuning both ϵ and λ to achieve an optimal balance between privacy, robustness, and utility in NLP applications.

Tables 6–8 present the accuracy performance of three NLP tasks—Sentiment Analysis, Question Answering, and Topic Classification—under varying privacy budgets (ϵ) and adversarial training strengths (λ). For Sentiment Analysis, the accuracy starts high at 96% for $\epsilon = 1.0$ and $\lambda = 0.0$, indicating minimal privacy constraints and no adversarial training. As λ increases, accuracy gradually declines, with more significant drops observed under stricter privacy settings (lower ϵ). This trend highlights the trade-off between maintaining high model accuracy and enhancing robustness to adversarial attacks, especially when privacy constraints are tighter.

Table 6. Sentiment Analysis accuracy across different privacy budgets and adversarial training strengths.

Privacy Budget ϵ	**λ = 0.0**	**λ = 0.1**	**λ = 0.3**	**λ = 0.5**	**λ = 0.7**	**λ = 1.0**
1.0	0.960	0.955	0.950	0.945	0.940	0.935
0.5	0.950	0.945	0.940	0.935	0.930	0.925
0.3	0.940	0.935	0.930	0.925	0.920	0.915
0.1	0.930	0.925	0.920	0.915	0.910	0.905

Table 7. Question Answering accuracy across different privacy budgets and adversarial training strengths.

Privacy Budget ϵ	**λ = 0.0**	**λ = 0.1**	**λ = 0.3**	**λ = 0.5**	**λ = 0.7**	**λ = 1.0**
1.0	0.870	0.865	0.860	0.855	0.850	0.845
0.5	0.860	0.855	0.850	0.845	0.840	0.835
0.3	0.850	0.845	0.840	0.835	0.830	0.825
0.1	0.840	0.835	0.830	0.825	0.820	0.815

Table 8. Topic Classification accuracy across different privacy budgets and adversarial training strengths.

Privacy Budget ϵ	**λ = 0.0**	**λ = 0.1**	**λ = 0.3**	**λ = 0.5**	**λ = 0.7**	**λ = 1.0**
1.0	0.940	0.935	0.930	0.925	0.920	0.915
0.5	0.930	0.925	0.920	0.915	0.910	0.905
0.3	0.920	0.915	0.910	0.905	0.900	0.895
0.1	0.910	0.905	0.900	0.895	0.890	0.885

A similar pattern is observed for the Question Answering and Topic Classification tasks. In the Question Answering task, the accuracy decreases from 87% to 81.5% as both adversarial training strength and privacy constraints increase. This reduction reflects the challenge of balancing precision with the need for privacy and robustness. For Topic Classification, the accuracy shows a consistent decline from 94% to 88.5% across different λ values, again emphasizing the compromise between high performance and security measures. These results collectively suggest that while adversarial training can protect models against attacks, it often reduces their overall performance, particularly under stringent privacy settings.

4.3. Discussion

The data suggest that the optimal balance between privacy and adversarial training depends on the application's tolerance for reduced utility under strict privacy constraints. Large systems should adopt a tiered approach, prioritizing adversarial training to enhance robustness in high-risk contexts while adjusting privacy parameters based on the acceptable trade-off between model performance and data protection.

5. Conclusions

This paper presents a novel framework that integrates differential privacy and adversarial training into the prompt learning process for NLP systems, enhancing both privacy and robustness. Empirical evaluations across multiple NLP tasks demonstrate that the framework effectively balances the trade-offs between privacy preservation and utility, while stricter privacy settings lead to reduced performance, this is mitigated by the model's improved resilience to adversarial attacks. These results are particularly significant for privacy-sensitive domains like finance and healthcare, where data confidentiality and robustness are critical. However, the framework faces limitations, such as the performance-utility trade-off under strong privacy constraints and the challenge of scalability to larger datasets or models. Future work will focus on optimizing these trade-offs, exploring scal-

able solutions, and extending the framework to broader applications, thereby advancing secure and trustworthy NLP systems.

Author Contributions: Conceptualization, C.S., J.S. and B.W. Methodology, C.S., J.S., B.W. and D.F.; Software, C.C. and B.W.; Validation, C.S., J.S., C.C. and B.W.; Formal analysis, C.S., J.S. and C.C.; Investigation, B.W.; Data curation, D.F.; Writing—review & editing, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
2. Xu, Q.; Wei, Y.; Yuan, S.; Wu, J.; Wang, L.; Wu, C. Learning emotional prompt features with multiple views for visual emotion analysis. *Inf. Fusion* **2024**, *108*, 102366. [CrossRef]
3. Shi, K.; Lu, J.; Fang, Z.; Zhang, G. Unsupervised Domain Adaptation Enhanced by Fuzzy Prompt Learning. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 4038–4048. [CrossRef]
4. Oesterling, A.; Ma, J.; Calmon, F.P.; Lakkaraju, H. Fair Machine Unlearning: Data Removal while Mitigating Disparities. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 2–4 May 2024; Dasgupta, S., Mandt, S., Li, Y., Eds.; Proceedings of Machine Learning Research; PMLR: London, UK, 2024; Volume 238, pp. 3736–3744.
5. Pan, Y.; Yuan, Y.; Yin, Y.; Shi, J.; Xu, Z.; Zhang, M.; Shang, L.; Jiang, X.; Liu, Q. Preparing Lessons for Progressive Training on Language Models. In Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, Vancouver, BC, Canada, 20–27 February 2024; Wooldridge, M.J., Dy, J.G., Natarajan, S., Eds.; AAAI Press: Washington, DC, USA, 2024; pp. 18860–18868. [CrossRef]
6. Lu, D.; Wang, Z.; Wang, T.; Guan, W.; Gao, H.; Zheng, F. Set-level Guidance Attack: Boosting Adversarial Transferability of Vision-Language Pre-training Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, 1–6 October 2023; pp. 102–111. [CrossRef]
7. Li, G.; Wu, W.; Sun, Y.; Shen, L.; Wu, B.; Tao, D. Visual Prompt Based Personalized Federated Learning. *Trans. Mach. Learn. Res.* **2024**, *2024*, 1–21.
8. Lei, Y.; Li, J.; Li, Z.; Cao, Y.; Shan, H. Prompt learning in computer vision: A survey. *Front. Inf. Technol. Electron. Eng.* **2024**, *25*, 42–63. [CrossRef]
9. Liello, L.D.; Garg, S.; Soldaini, L.; Moschitti, A. Pre-training Transformer Models with Sentence-Level Objectives for Answer Sentence Selection. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 11806–11816. [CrossRef]
10. Fang, Y.; Chen, Z.; Fan, X.; Zhang, N.; Chen, H. Knowledge-Informed Molecular Learning: A Survey on Paradigm Transfer. In Proceedings of the Knowledge Science, Engineering and Management—17th International Conference, KSEM 2024, Birmingham, UK, 16–18 August 2024; Cao, C., Chen, H., Zhao, L., Arshad, J., Asyhari, A.T., Wang, Y., Eds.; Proceedings, Part I; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2024; Volume 14884, pp. 86–98. [CrossRef]
11. Guo, W.; Zhuang, F.; Zhang, X.; Tong, Y.; Dong, J. A comprehensive survey of federated transfer learning: Challenges, methods and applications. *Front. Comput. Sci.* **2024**, *18*, 186356. [CrossRef]
12. Zhu, Z.; Lin, K.; Jain, A.K.; Zhou, J. Transfer Learning in Deep Reinforcement Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13344–13362. [CrossRef]
13. Jin, D.; Jin, Z.; Hu, Z.; Vechtomova, O.; Mihalcea, R. Deep Learning for Text Style Transfer: A Survey. *Comput. Linguist.* **2022**, *48*, 155–205. [CrossRef]
14. Bu, K.; Liu, Y.; Ju, X. Efficient utilization of pre-trained models: A review of sentiment analysis via prompt learning. *Knowl. Based Syst.* **2024**, *283*, 111148. [CrossRef]
15. Qin, Y.; Wang, X.; Su, Y.; Lin, Y.; Ding, N.; Yi, J.; Chen, W.; Liu, Z.; Li, J.; Hou, L.; et al. Exploring Universal Intrinsic Task Subspace for Few-Shot Learning via Prompt Tuning. *IEEE ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 3631–3643. [CrossRef]
16. Ma, C.; Liu, Y.; Deng, J.; Xie, L.; Dong, W.; Xu, C. Understanding and Mitigating Overfitting in Prompt Tuning for Vision-Language Models. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4616–4629. [CrossRef]
17. Gu, Z.; Fan, J.; Tang, N.; Cao, L.; Jia, B.; Madden, S.; Du, X. Few-shot Text-to-SQL Translation using Structure and Content Prompt Learning. *Proc. ACM Manag. Data* **2023**, *1*, 147. [CrossRef]

18. Yue, Z.; Zeng, H.; Lan, M.; Ji, H.; Wang, D. Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, ON, Canada, 9–14 July 2023; Rogers, A., Boyd-Graber, J.L., Okazaki, N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 7928–7943. [CrossRef]
19. Strobelt, H.; Webson, A.; Sanh, V.; Hoover, B.; Beyer, J.; Pfister, H.; Rush, A.M. Interactive and Visual Prompt Engineering for Ad-hoc Task Adaptation with Large Language Models. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 1146–1156. [CrossRef]
20. Shtedritski, A.; Rupprecht, C.; Vedaldi, A. What does CLIP know about a red circle? Visual prompt engineering for VLMs. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, 1–6 October 2023; pp. 11953–11963. [CrossRef]
21. Sorensen, T.; Robinson, J.; Rytting, C.M.; Shaw, A.G.; Rogers, K.J.; Delorey, A.P.; Khalil, M.; Fulda, N.; Wingate, D. An Information-theoretic Approach to Prompt Engineering Without Ground Truth Labels. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 819–862. [CrossRef]
22. Edemacu, K.; Wu, X. Privacy Preserving Prompt Engineering: A Survey. *arXiv* **2024**. [CrossRef]
23. Du, Y.; Yin, Z.; Xie, R.; Zhang, Q. Prompt template construction by Average Gradient Search with External Knowledge for aspect sentimental analysis. *Expert Syst. Appl.* **2024**, *238*, 122271. [CrossRef]
24. Korzynski, P.; Mazurek, G.; Krzypkowska, P.; Kurasinski, A. Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT. *Entrep. Bus. Econ. Rev.* **2023**, *11*, 25–37. [CrossRef]
25. Polak, M.P.; Morgan, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nat. Commun.* **2024**, *15*, 1569. [CrossRef]
26. Mao, K.; Dou, Z.; Mo, F.; Hou, J.; Chen, H.; Qian, H. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 1211–1225. [CrossRef]
27. Rigaki, M.; García, S. A Survey of Privacy Attacks in Machine Learning. *ACM Comput. Surv.* **2024**, *56*, 101. [CrossRef]
28. Elmazi, D.; Karras, D.A.; Alkholidi, A.; Çapari, K. Cybersecurity and Privacy Attacks Detection in IoT Networks with Improved Data Engineering and Machine Learning Methods. In Proceedings of the IEEE Ninth International Conference on Big Data Computing Service and Applications, BigDataService 2023, Athens, Greece, 17–20 July 2023; pp. 223–228. [CrossRef]
29. Hu, H.; Pang, J. Membership Inference Attacks against GANs by Leveraging Over-representation Regions. In Proceedings of the CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, 15–19 November 2021; Kim, Y., Kim, J., Vigna, G., Shi, E., Eds.; ACM: New York, NY, USA, 2021; pp. 2387–2389. [CrossRef]
30. Ye, J.; Maddi, A.; Murakonda, S.K.; Bindschaedler, V.; Shokri, R. Enhanced Membership Inference Attacks against Machine Learning Models. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, 7–11 November 2022; Yin, H., Stavrou, A., Cremers, C., Shi, E., Eds.; ACM: New York, NY, USA, 2022; pp. 3093–3106. [CrossRef]
31. Xue, D.; Yang, H.; Ge, M.; Li, J.; Xu, G.; Li, H. Fast Generation-Based Gradient Leakage Attacks against Highly Compressed Gradients. In Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications, New York, NY, USA, 17–20 May 2023; pp. 1–10. [CrossRef]
32. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 14747–14756.
33. Zhang, L.; Zhu, T.; Xiong, P.; Zhou, W.; Yu, P.S. A Robust Game-Theoretical Federated Learning Framework With Joint Differential Privacy. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3333–3346. [CrossRef]
34. Mao, Y.; Ye, Q.; Wang, Q.; Hu, H. Differential Privacy for Time Series: A Survey. *IEEE Data Eng. Bull.* **2024**, *47*, 67–92.
35. Huang, G.; Wu, Q.; Sun, P.; Ma, Q.; Chen, X. Collaboration in Federated Learning With Differential Privacy: A Stackelberg Game Analysis. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *35*, 455–469. [CrossRef]
36. Boenisch, F.; Dziedzic, A.; Schuster, R.; Shamsabadi, A.S.; Shumailov, I.; Papernot, N. Reconstructing Individual Data Points in Federated Learning Hardened with Differential Privacy and Secure Aggregation. In Proceedings of the 8th IEEE European Symposium on Security and Privacy, EuroS&P 2023, Delft, The Netherlands, 3–7 July 2023; pp. 241–257. [CrossRef]
37. Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.B.; Song, D.; Erlingsson, Ú.; et al. Extracting Training Data from Large Language Models. In Proceedings of the 30th USENIX Security Symposium, USENIX Security 2021, Online, 11–13 August 2021; Bailey, M.D., Greenstadt, R., Eds.; USENIX Association: Berkeley, CA, USA, 2021; pp. 2633–2650.

38. Lu, P.; Zhang, L.; Liu, M.; Sridhar, K.; Sokolsky, O.; Kong, F.; Lee, I. Recovery from Adversarial Attacks in Cyber-physical Systems: Shallow, Deep, and Exploratory Works. *ACM Comput. Surv.* **2024**, *56*, 211. [CrossRef]
39. Sánchez, P.M.S.; Celdrán, A.H.; Bovet, G.; Pérez, G.M. Adversarial attacks and defenses on ML- and hardware-based IoT device fingerprinting and identification. *Future Gener. Comput. Syst.* **2024**, *152*, 30–42. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.