Adversarially Robust Prompt Tuning through Selective Prompt Diversification and Injection

Matthias R. Simmons
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
matthias.work@ucf.edu

Kiran Melhatra
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
kmalhotra@colostate.edu

Chetan R. Mishra
Department of Computer Science, University of Houston, Houston, TX, USA.
chetanmishra75@uh.edu

Abstract

The paradigm of prompt tuning has emerged as a highly parameter-efficient means of adapting large pre-trained language models to downstream tasks, yet its vulnerability to adversarial perturbations threatens deployment in safety-critical and high-stakes environments. This paper presents a system-level framework for adversarially robust prompt tuning grounded in two interlocking mechanisms: selective prompt diversification and controlled prompt injection. The framework moves beyond static soft prompt representations by stochastically generating a candidate pool of diverse prompts and then selecting a subset according to robustness-aware criteria before integrating them into the model's forward pass. We analyze the resulting architecture through the lenses of structural trade-offs, inference-time overhead, deployment infrastructure, governance, fairness, and long-term sustainability, arguing that robust prompt tuning is as much a systems design problem as it is a machine learning optimization challenge. The discussion delineates how diversification strategies can be calibrated to balance adversarial resilience against computational cost, how injection controllers can be implemented as lightweight gating modules suitable for edge and cloud environments, and how the interplay between prompt selection and input semantics raises novel fairness and interpretability concerns. Without introducing mathematical formalisms, the paper provides a conceptual blueprint for next-generation robust language interfaces and situates the proposal within broader socio-technical discourse on trustworthy artificial intelligence.

Keywords

Adversarial robustness, prompt tuning, selective diversification, prompt injection, language models, system design, deployment infrastructure, sustainability.

1. Introduction

The rapid maturation of large pre-trained language models has catalyzed a shift from full-parameter fine-tuning to parameter-efficient adaptation strategies, among which soft prompt tuning occupies a prominent position [1, 2]. By learning a small set of continuous embeddings prepended to an input sequence and keeping the entire pre-trained model frozen, prompt tuning drastically reduces the number of trainable parameters while preserving competitive

performance across a wide range of natural language understanding and generation tasks. Despite its architectural elegance and resource efficiency, soft prompt tuning inherits and in some cases amplifies the brittleness of neural language models when confronted with adversarially crafted inputs. Attackers can exploit small perturbations to the input text or directly manipulate the learned prompt representations, causing catastrophic degradation in model behavior and undermining trust in deployed systems [3, 4]. The growing prevalence of adversarial attacks on natural language processing pipelines, coupled with the expanding use of prompt-based interfaces in consumer-facing products, necessitates a systematic re-examination of how robustness can be engineered into the prompt tuning lifecycle.

Existing defense mechanisms predominantly focus on model-level adversarial training or certified robustness guarantees, often treating the prompt as a fixed component that is tuned once and then frozen [5]. Yet the very modularity of prompt tuning opens a distinct design space: instead of hardening a single prompt vector, one can introduce controlled diversity into the prompt representation and inject the resulting representations selectively during inference. This idea repositions robustness from a purely statistical learning problem to a broader systems engineering challenge that involves prompt generation policies, selection controllers, and runtime orchestration. In the present work, we articulate an integrated framework for adversarially robust prompt tuning through selective prompt diversification and injection. We conceptualize a pipeline in which multiple candidate prompts are generated via stochastic diversification strategies, evaluated through a robustness-aware selector, and adaptively blended into the model's forward computation. The framework is presented not as a narrow algorithmic contribution but as a system-level architecture that foregrounds the structural trade-offs, deployment practicality, and governance implications that arise when robustness is treated as a first-class design objective.

The remainder of this paper develops this perspective in depth. We begin by surveying related work on parameter-efficient tuning, adversarial threats, and prompt-based defenses, establishing the context in which selective diversification and injection constitute a meaningful advance. We then detail the proposed system architecture, unpack the adversarial threat landscape, and analyze the design principles of selective diversification and injection. Subsequent sections examine deployment infrastructure, governance and fairness, and sustainability considerations, before concluding with an outlook on the future of robust, trustworthy prompt-tuned systems.

2. Background and Related Work

Parameter-efficient transfer learning has redefined how practitioners interact with large language models. Prompt tuning [1] introduces a small set of trainable prefix tokens whose embeddings are optimized while the core model remains static, achieving strong results with a fraction of the parameters required for full fine-tuning. Prefix tuning [2] extends this concept by inserting learned vectors into every transformer layer, offering additional representational capacity. These methods have given rise to a rich ecosystem of prompt design techniques that treat the prompt as a learnable, continuous object rather than a discrete template. However, the resilience of such continuous prompts in the face of adversarial manipulation has only recently attracted systematic scrutiny.

Adversarial attacks on natural language processing were first demonstrated in reading comprehension and text classification, exposing the susceptibility of models to semantically equivalent but lexically perturbed inputs [3]. A comprehensive survey [4] cataloged attack surfaces ranging from character-level substitutions to syntactic paraphrasing and backdoor

triggers, underscoring that linguistic models lack the input-space smoothness often assumed in vision. Adversarial training, where models are trained on perturbed examples, has become a standard defense [5], yet applying it directly to prompt-tuned systems raises complications because the prompt parameters themselves become an additional attack vector. Recent work has revealed that prompts can be adversarially manipulated to induce targeted misclassification, and that attackers can learn universal adversarial triggers that transfer across prompts [6]. BadPrompt, for instance, demonstrated that continuous prompts are susceptible to gradient-based attacks that alter the embedding space significantly beyond the training distribution.

In parallel, the search for robust and efficient prompt generation has spurred methods such as AutoPrompt [7], which automatically discovers discrete trigger words that elicit desired behaviors, and few-shot in-context learning [8], which leverages large models to perform tasks without any parameter updates. Although these approaches sidestep explicit prompt tuning, they do not eliminate adversarial vulnerabilities and often introduce different failure modes related to prompt sensitivity and selection. Parameter-efficient adapters and low-rank adaptation (LoRA) [9] have also been explored, showing that a small number of inserted parameters can achieve robustness when combined with weight ensembling, but the interaction between adapter-based defenses and continuous prompts remains underexplored.

More recently, the community has proposed adversarial training regimes specifically for continuous prompts, such as adversarial prompt reparameterization through WARP [10], domain-adaptive prompt tuning that improves generalization across distribution shifts [11], and adversarial training methods originally developed for semi-supervised text classification that have been adapted to the prompt tuning setting [12]. Robustness-oriented sample reweighting schemes that prioritize hard or adversarial examples during prompt optimization have also shown promise [13]. Despite these advances, a critical gap persists: most defenses produce a single robust prompt, ignoring the possibility that a carefully curated set of prompts, combined in a principled manner, could provide greater adversarial resilience than any single representation.

The evaluation of prompt robustness has been facilitated by benchmarks such as PromptBench [15], which systematically measures model behavior under diverse adversarial perturbations. These benchmarks reveal that even state-of-the-art prompt tuning methods degrade sharply under attack, motivating the need for defensive architectures that can dynamically adapt their prompt representations. A complementary line of research investigates how prompts can be selectively inserted or combined to improve clean-task performance [16], yet without explicit consideration of adversarial threat models. Our framework bridges these two currents by infusing selective prompt insertion with diversity-driven robustness strategies, yielding a comprehensive systems approach to adversarially robust prompt tuning.

3. System Architecture and Prompt Engineering Paradigm

The proposed architecture is structured around a closed-loop pipeline composed of three core modules: a diversification engine, a robustness-aware selector, and an injection controller. The diversification engine receives a base prompt initialized through standard prompt tuning or a pre-existing template and generates a pool of candidate prompts by applying stochastic transformations in the continuous embedding space. These transformations include introducing controlled isotropic Gaussian noise, linearly interpolating between plausible prompt embeddings drawn from different training runs, and applying semantic

reparameterizations that preserve the approximate meaning while shifting the embedding geometry. The goal is not to achieve maximal entropy but to cover a representative set of prompt variations that might be encountered under adversarial stress, thereby creating an implicit ensemble of soft prompts.

The robustness-aware selector then examines each candidate prompt from the pool and assigns a score based on how consistently the prompt yields correct predictions across a set of adversarial views generated on the fly from the current input. This selection process can be realized as a lightweight neural scorer trained jointly with the diversification engine, or as a non-parametric voting mechanism that privileges prompts with low variance in output distribution under input perturbation. Crucially, the selector does not optimize for clean accuracy alone; instead, it balances predictive performance with a stability metric that penalizes prompts that are hypersensitive to small modifications of the surrounding input. The selected subset, which may range from one to a handful of prompts depending on the computational budget and latency requirements, is then passed to the injection controller.

The injection controller implements the final integration into the model's forward pass. Rather than simply concatenating a single prompt with the input, it can blend the embeddings of multiple selected prompts through a learned attention gate, or conditionally activate different prompts for different segments of the input based on semantic cues. This dynamic mixing endows the system with the ability to respond differently to adversarial perturbations that might target a single prompt vector: if one prompt component is compromised, the injection controller can down-weight it in favor of others that retain their effectiveness. From a systems perspective, this architecture introduces a novel prompt engineering paradigm in which prompts become not static artifacts but active, runtime-managed resources that are created, assessed, and assembled by specialized infrastructural components.

4. Adversarial Threat Landscape and Robustness Challenges

The adversarial threat landscape for prompt-tuned models is broad and multi-layered, encompassing attacks on the input text, on the prompt parameters, and on the interaction between the two. Word-level attacks, such as synonym substitution and character-level perturbations, can disrupt the model's attention patterns and cause the continuous prompt to lose its alignment with the underlying linguistic content. More insidious are prompt-level attacks, where an adversary with white-box or even partial access to the prompt embeddings can apply gradient-based perturbations that are invisible in the discrete textual input yet devastating to model output. Universal adversarial triggers, which are fixed token sequences that induce failure regardless of the sample, pose a particular challenge because they can be appended to any prompt and hijack the entire classification or generation pipeline [6, 24]. Additionally, data poisoning during the prompt tuning phase can embed backdoors that remain dormant until specific input patterns appear, circumventing conventional adversarial training defenses.

Robustness challenges in prompt tuning stem from the very properties that make it attractive: the extreme parameter compression concentrates representational capacity into a small embedding space, which can amplify overfitting to spurious correlations present in the training data. A single prompt vector lacks the redundancy necessary to absorb adversarial perturbations; once it is pushed outside the manifold of benign embeddings, the model's behavior becomes unpredictable. Moreover, the transferability of adversarial examples implies that a vulnerability discovered on one prompt-tuned model can potentially be exploited on others sharing the same backbone, creating systemic risk across a deployed fleet

of services. The interplay between input noise and prompt representation also complicates certified robustness approaches, which typically assume a fixed model and bounded input perturbations, whereas prompt-tuned systems require simultaneous consideration of two interacting parameter spaces.

From a socio-technical standpoint, robustness failures in prompt-tuned models can have far-reaching consequences when these models serve as interfaces for medical decision support, legal document analysis, or financial risk assessment. In such domains, an adversary could manipulate a carefully crafted prompt to produce biased or factually incorrect outputs while evading superficial inspection. Therefore, robustness must be engineered as a multi-component property involving not only the learning algorithm but also the architectural separation of duties among diversification, selection, and injection, each of which can be independently monitored, patched, and fortified in response to emerging threats.

5. Selective Prompt Diversification: Design Principles and Structural Trade-offs

Selective prompt diversification is the process of generating a controlled variety of soft prompt embeddings and then identifying those that maximize adversarial resilience while preserving task performance. The design principles underlying this module are redundancy, representational coverage, and controllability. Redundancy ensures that no single prompt embedding constitutes a single point of failure; if one prompt is corrupted, others can compensate. Representational coverage requires that the set of candidate prompts spans a sufficiently wide region of the embedding manifold so that adversarial perturbations, which tend to exploit narrow directions of vulnerability, are unlikely to simultaneously compromise all candidates. Controllability imposes that the diversification intensity can be tuned according to available resources and threat posture, allowing system operators to dial up diversity during high-risk periods and reduce it when latency constraints are tight.

Implementing diversification involves a structural trade-off between robustness gains and computational overhead. Generating many candidate prompts and evaluating them against adversarial views increases both memory consumption and inference latency. However, because prompt embeddings are small relative to the entire model, the cost of maintaining a pool of, for instance, ten to twenty candidate vectors remains moderate. The primary expense lies in the forward passes required by the selection process. This trade-off can be mitigated by employing a cascade architecture where a fast, lightweight scorer filters the majority of candidates, and only a small number undergo full evaluation. Further efficiency can be gained by caching diversified prompts for frequently encountered input types and by sharing the diversification mechanism across multiple downstream tasks that operate on the same base model.

An important structural consideration is the relationship between diversification and the underlying pre-training objective. Language models pre-trained with contrastive or self-supervised objectives often exhibit smoothness in certain embedding directions; diversification strategies that align with the natural geometry of the pre-trained space are likely to yield candidates that are both diverse and semantically coherent. Conversely, naive diversification that ignores this geometry may produce prompts that the model interprets as out-of-distribution noise, undermining clean accuracy while providing only marginal robustness. The selector must therefore be tuned to recognize this tension and prefer candidates that not only remain stable under attack but also reside in regions well-supported by the pre-trained representations. This dual requirement redefines the robustness-accuracy trade-off as a multi-objective system design problem requiring explicit operational constraints.

6. Prompt Injection Mechanisms: Controlled Integration for Robust Inference

Controlled prompt injection refers to the process by which the selected diversified prompts are integrated into the model's forward computation in a manner that adapts to both the input characteristics and the prevailing adversarial conditions. Unlike static prompt tuning, where the same set of continuous embeddings is prepended to all inputs, our framework introduces a dynamic injection scheme. The injection controller can implement a soft attention mechanism that computes an input-dependent mixture of prompted representations, or a hard gating network that activates precisely one prompt per forward pass. The choice between soft and hard injection has significant implications for robustness and latency: soft blending provides a smoother decision boundary and can insulate against adversarial inputs that seek to flip the gating decision, while hard gating reduces compute and may be more interpretable.

A key insight is that the injection controller can be designed to leverage the selective prompt insertion principles explored in the SPT framework [16], where a model learns to strategically insert prompts only when they improve predictive accuracy on clean data. We extend this idea by coupling insertion decisions with adversarial awareness, so that prompts are not merely inserted or withheld based on expected performance gain, but also modulated according to the detection of anomalous input patterns that may signal an ongoing attack. The injection controller can incorporate a lightweight anomaly detector, trained separately on benign and adversarial examples, that adjusts the gating logits or attention weights to suppress prompts that have been flagged as potentially compromised. In this way, the system realizes a closed-loop defense: diversification generates redundancy, selection prunes the candidate set toward stability, and injection orchestrates the final, context-sensitive integration.

The dynamic nature of prompt injection also opens avenues for runtime adaptation. In environments where the threat model evolves over time, the injection controller can be periodically updated through online learning without retraining the entire prompt pool. This decoupling of the diversification, selection, and injection modules allows for modular maintenance and targeted security patches, aligning with modern DevOps practices for machine learning systems. Furthermore, the injection controller can be implemented as a lightweight neural component with negligible memory overhead, making it suitable for deployment on resource-constrained edge devices where full model adaptation is infeasible. The ability to swap injection strategies without modifying the core language model or the prompt tuning data further strengthens the system's long-term maintainability.

7. Infrastructure and Deployment Considerations

Deploying adversarially robust prompt tuning at scale demands infrastructure capable of orchestrating the additional computational workload introduced by diversification and dynamic injection. Inference serving pipelines must be extended with a prompt management layer that maintains the candidate pool, executes selection logic, and interfaces with the injection controller. Because prompt embeddings are small and the selection process can be parallelized across candidates, the overhead can be absorbed through batch processing and optimized GPU memory allocation. Caching frequently used diversified prompts reduces the need to recompute selection scores for each request, while a distributed key-value store can synchronize prompt state across model replicas in a load-balanced serving system. Edge deployments, where latency budgets are stricter, can adopt a streamlined configuration using a single selected prompt and a simplified anomaly detector, thereby trading some robustness margin for real-time responsiveness.

The infrastructure must also accommodate the increased memory footprint of storing multiple prompt variants. Although soft prompts typically comprise only a few thousand floating-point numbers, maintaining separate pools for hundreds of tasks or tenants in a multi-tenant platform can accumulate. Compression techniques such as quantization-aware prompt tuning and parameter sharing across tasks can mitigate this overhead. The injection controller itself can be distilled into an extremely compact form, ensuring that the additional parameter count remains a tiny fraction of the overall model size [17]. Monitoring dashboards must track robustness metrics, such as the percentage of adversarial inputs successfully deflected and the distribution of prompt usage, to provide visibility into the system's operational health.

Energy consumption and carbon footprint are salient deployment-level concerns. The extra forward passes incurred by candidate evaluation increase the total FLOP count per inference request, which translates into higher energy usage if not carefully managed [18]. This cost must be weighed against the societal risk of deploying fragile models that could cause harm. A tiered deployment model can offer configurable robustness levels, allowing applications in low-risk settings to use minimal diversification and aggressive caching, while high-stakes domains activate the full pipeline with periodic adversarial audits. By exposing these configuration options, platform providers enable their clients to make informed trade-offs between sustainability, cost, and safety, embodying the principle that robustness is not an absolute property but a resource-aware system objective.

8. Governance, Fairness, and Policy Implications

The introduction of diversified and selectively injected prompts raises important governance questions surrounding transparency, fairness, and accountability. A model whose behavior depends on an opaque selection and blending of multiple internal prompts may be more difficult for external auditors to scrutinize than a model using a single, static prompt. To address this, system designers should expose interpretable summaries of which prompts were activated for a given input and provide user-facing explanations grounded in the injection controller's attention patterns [25]. Regulatory frameworks that mandate explainability for high-risk artificial intelligence systems will need to evolve to accommodate architectures that shift internal representations dynamically, and the approach we describe can be designed with logging and traceability mechanisms from the outset.

Fairness considerations arise because prompt diversification and selection may interact with biases encoded in the pre-trained model and the training data. If the diversification process systematically suppresses prompts that are effective for minority demographic subgroups, the system could exhibit disparate robustness: some populations might enjoy reliable model outputs while others suffer degraded performance under the same distribution of adversarial attacks. Mitigating this risk requires fairness-aware prompt selection criteria that evaluate candidate prompts not only on aggregate stability metrics but also on performance parity across protected groups [14]. The injection controller can be regularized to prevent attention from focusing solely on prompts that benefit majority classes, ensuring that robustness gains are equitably distributed.

At the policy level, adversarially robust prompt tuning aligns with emerging standards for trustworthy artificial intelligence, which emphasize security and resilience as foundational pillars [19]. Certification procedures for language-based systems may in the future require evidence of adversarial robustness testing, and modular architectures like the one described here facilitate such audits by isolating the defense components. Moreover, the ability to selectively fortify prompts without altering the base model can simplify compliance with

model governance policies that restrict modification of pre-trained weights for liability or intellectual property reasons. This architectural separation of concerns supports a regulatory model in which the responsibility for prompt-level robustness can be distributed among model providers, application developers, and deployment operators.

9. Sustainability and Long-term Maintenance

Sustaining adversarial robustness over the operational lifetime of a prompt-tuned system requires continual adaptation as new attack techniques emerge and data distributions drift. The modularity of the proposed framework supports lifelong learning paradigms where the diversification engine, selector, and injection controller can be updated incrementally without retraining the entire pipeline [22]. For example, when a novel adversarial strategy is identified, the system can generate additional diversified prompts specifically designed to counter it and fine-tune the selector to recognize the associated perturbation patterns, leaving the injection controller's core logic largely intact. This targeted patching reduces the computational cost of maintaining robustness over time and aligns with sustainable software engineering practices.

The environmental sustainability of repeatedly evaluating and updating diversified prompt pools can be managed through resource-aware scheduling. Off-peak hours can be used to refresh the candidate prompt bank and retrain the selector on newly collected adversarial data, while online inference continues with the existing robust configuration. The carbon impact of these periodic retraining cycles can be offset by leveraging energy-efficient hardware accelerators and carbon-aware data center orchestration. Furthermore, the compact nature of soft prompts means that model updates can be distributed as lightweight patches rather than full model redistributions, reducing network bandwidth and storage costs. This property is especially beneficial for edge devices and mobile applications where over-the-air update payloads must be minimized.

From a socio-technical sustainability perspective, the long-term viability of adversarially robust prompt tuning depends on the establishment of a collaborative ecosystem in which threat intelligence is shared among stakeholders without compromising proprietary information. Industry consortia could maintain shared repositories of adversarial prompts and robustness benchmarks, analogous to cybersecurity information sharing platforms, enabling collective defense. Our framework's architecture, with its clear separation between base model, prompt management, and injection control, facilitates such sharing because threat-adaptive modules can be developed and distributed independently of the underlying models. This modularity encourages a specialization of labor in which security researchers, platform engineers, and domain experts each contribute to different components, promoting a resilient and evolving defense community.

10. Conclusion

Adversarially robust prompt tuning represents a critical frontier in the design of trustworthy language technologies. In this paper, we have articulated a system-level framework that reimagines prompt tuning not as the optimization of a single static vector but as a managed process of diversification, selection, and controlled injection. By analyzing the structural trade-offs between adversarial resilience and computational cost, the deployment infrastructure required for dynamic prompt orchestration, and the governance and fairness implications of multi-prompt architectures, we have positioned robust prompt tuning within a broader interdisciplinary discourse. Selective prompt diversification provides the redundancy necessary to absorb adversarial perturbations, while a context-aware injection controller

adaptively synthesizes the most stable prompt representations during inference. The resulting system is inherently modular, facilitating incremental updates, regulatory compliance, and equitable performance across user populations. As language models continue to permeate high-stakes societal applications, the architectural principles outlined here offer a pathway toward systems that are not only accurate and efficient but also resilient in the face of deliberate manipulation, thereby contributing to a safer and more trustworthy artificial intelligence ecosystem.

References

1. Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 3045–3059). Association for Computational Linguistics.

2. Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (pp. 4582–4597). Association for Computational Linguistics.

3. Jia, R., & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2021–2031). Association for Computational Linguistics.

4. Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Computing Surveys, 53(3), Article 57.

5. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations.

6. Xu, L., Chen, Y., Ghosh, S., Natarajan, P., & Chang, S.-F. (2022). BadPrompt: A black-box adversarial framework for prompt-tuning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 7165–7179). Association for Computational Linguistics.

7. Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (pp. 4222–4235). Association for Computational Linguistics.

8. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (Vol. 33, pp. 1877–1901).

9. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

10. Hambardzumyan, K., Khachatrian, H., & May, J. (2021). WARP: Word-level adversarial re-programming. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (pp. 4921–4933). Association for Computational Linguistics.

11. Gu, Y., Han, X., Liu, Z., & Huang, M. (2022). PPT: Pre-trained prompt tuning for few-shot learning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (pp. 8410–8423). Association for Computational Linguistics.

12. Miyato, T., Dai, A. M., & Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. In International Conference on Learning Representations.

13. Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In International Conference on Machine Learning (pp. 4334–4343). PMLR.

14. Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and mitigating unintended bias in text classification. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (pp. 67–73). ACM.

15. Li, J., Cheng, S., Li, J., Takanobu, R., Huang, M., & Feng, Y. (2023). PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. arXiv preprint arXiv:2306.04528.

16. Zhu, W., & Tan, M. (2023, December). SPT: Learning to selectively insert prompts for better prompt tuning. In Proceedings of the 2023 conference on empirical methods in natural language processing (pp. 11862-11878).

17. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In NIPS Deep Learning and Representation Learning Workshop.

18. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645–3650). Association for Computational Linguistics.

19. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R. B., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

20. Jia, R., Raghunathan, A., Göksel, K., & Liang, P. (2019). Certified robustness to adversarial word substitutions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (pp. 4120–4133). Association for Computational Linguistics.

21. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318). ACM.

22. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54–71.

23. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning (pp. 2790–2799). PMLR.

24. He, K., Zhu, J., Lin, Z., & Liang, P. (2023). Defending against universal adversarial triggers. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 4998–5017). Association for Computational Linguistics.

25. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). ACM.