World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Prompt engineering and prompt-tuning: Foundations, advancements and research direction

Jan Richter [1,*], Daniella Fitzpatrick [2], Aaron Fuller [2], Philip Leal [2] and Leia Pittman [2]

[1] *Independent Researcher, Berlin, Germany.*
[2] *Researcher, Information Systems, Lonestar Institute, Germany.*

World Journal of Advanced Research and Reviews, 2025, 28(03), 643-654

Publication history: Received 30 October 2025; revised on 06 December 2025; accepted on 09 December 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.28.3.4103

Abstract

The rapid evolution of large language models (LLMs) has shifted adaptation strategies away from full model fine-tuning and toward prompt-driven control. Prompt engineering enables LLMs to perform new tasks through carefully structured natural-language instructions, while prompt-tuning and related continuous prompting techniques introduce efficient mechanisms for task customization without modifying underlying model parameters. This paper presents an integrated examination of prompt-based methodologies, outlining the foundational developments that established prompting as a central paradigm in modern AI systems. It further analyzes key distinctions between discrete, continuous, and dynamic prompting approaches, highlighting their conceptual connections and performance characteristics. Through a detailed and structured review of influential literature, the article synthesizes how prompting methods have advanced cross-domain adaptation, semantic controllability, code generation, security analysis, multimodal retrieval, and other application areas. The paper concludes by identifying research opportunities related to interpretability, automatically generated prompts, multimodal extensions, robustness under adversarial or variable inputs, and the role of prompting in autonomous and human-centered AI systems.

Keywords: Prompt Engineering; Prompt-Tuning; Prefix-Tuning; Domain-Specific NLP Tasks; Parameter-Efficient Adaptation

1. Introduction

Large language models (LLMs) such as GPT-family models, T5, and PaLM have demonstrated remarkable capabilities across a wide range of language understanding and generation tasks. A central insight emerging from the growth of these models is that their behavior can be shaped not only through parameter-intensive fine-tuning, but also through the strategic design of prompts. Well-structured prompts, whether natural-language instructions, templates, or learned vectors, enable pretrained models to perform new tasks, capture domain constraints, or follow complex workflows without altering underlying model weights. This shift reflects a broader movement from the traditional pre-train and fine-tune paradigm toward a more flexible pre-train, prompt, and predict framework.

Prompt engineering has consequently become a practical and conceptual bridge between human intent and model behavior. At its simplest, it relies on carefully phrased instructions that exploit the implicit knowledge embedded in large pretrained models. At a more advanced level, prompt-tuning and other continuous prompting techniques treat prompts as optimizable components, enabling lightweight and parameter-efficient task adaptation. These methods allow practitioners to achieve near–fine-tuned performance while preserving the original model, reducing computational burden, and lowering storage and deployment costs.

* Corresponding author: Daniella Fitzpatrick

As prompting techniques have matured, they have expanded well beyond simple instruction refinement. Research has shown that continuous prompts can encode meaningful task information, prefix-based methods can steer transformers at the layer level, and dynamic prompt generators can adapt prompts to individual inputs or contexts. These innovations have opened the door to new forms of controllability, domain transfer, and task generalization, allowing prompts to function not merely as surface-level instructions but as operational mechanisms embedded within the model's internal representation space.

At the same time, prompting has become increasingly relevant across diverse application domains. From network security, remote sensing, scientific computing, and language-based education to software engineering, mental health assessment, and multimodal retrieval, prompt-driven strategies have emerged as efficient interfaces between domain knowledge and model capabilities. Many of these studies highlight the importance of prompt structure, optimization, and evaluation, demonstrating that prompts can influence accuracy, reasoning quality, interpretability, robustness, and computational efficiency.

Considering this rapid progress, there is a growing need for a consolidated view of prompt engineering and prompt-tuning, one that integrates foundational mechanisms with emerging trends and application-specific insights. This article provides such a synthesis. It reviews key conceptual advances in discrete, continuous, and dynamic prompting; examines how prompts have been adapted across tasks, modalities, and infrastructure constraints; and identifies patterns in how prompting supports efficiency, controllability, and adaptability. The introduction of a unified analytical framework further clarifies how different prompting methods relate to model architecture, optimization strategies, and performance outcomes.

Ultimately, understanding the evolution and principles of prompt-based methods is essential for designing next-generation AI systems that are more transparent, reliable, scalable, and aligned with user intent. By consolidating theoretical foundations, methodological developments, and cross-domain applications, this article lays the groundwork for future research focused on interpretability, automated prompt construction, multimodal integration, and robust human-centered prompting.

2. Literature Review

Research on prompt engineering and prompt-tuning builds on three intertwined foundations: transformer-based language modeling, classical feature and model adaptation strategies, and the emerging view of prompts as first-class control interfaces for AI systems.

2.1. From Transformers and Feature Engineering to Prompt-Centric Adaptation

The rise of prompt-based methods is inseparable from the transformer revolution. Early analyses of transformer architectures showed that self-attention enables scalable, context-aware sequence modeling and created the conditions for pretraining–based transfer to become dominant in natural language processing (NLP) (Marku, Jonas, & Al-Basri, 2020). In parallel, work on feature engineering in sparse data environments emphasized that performance in real-world machine learning depends not only on model capacity but also on how task information is represented and exposed to the model (Monteiro, Vella, & Haddad, 2020). These two strands foreshadow prompt engineering: prompts can be seen as a high-level, task-aware "feature interface" to large pretrained models.

Broader AI systems research laid groundwork for prompt-based control. Studies on adaptive learning in non-stationary, distributed environments highlighted the need for mechanisms that can accommodate concept drift and evolving data distributions without retraining entire models (Duarte & Raman, 2020). Work on explainable AI for high-reliability decision systems argued for semantic grounding, structural justification, and computational transparency as prerequisites for deploying AI in safety-critical domains (Korhonen, Rantala, & Lehtinen, 2020). Meanwhile, evaluations of lightweight deep neural architectures for resource-constrained edge intelligence underlined the importance of efficiency in computation, memory and energy (Saar et al., 2020). Together, these themes, efficient adaptation, explainability, and edge deployment, anticipate later uses of prompts as lightweight, interpretable levers for steering large models in distributed, bandwidth- and compute-constrained settings.

On top of these architectural and systems foundations, early prompting work (2019–2022) articulated the "pre-train, prompt, and predict" paradigm, where prompts replace task-specific output heads and much of traditional fine-tuning. Liu et al. (2021) systematized this shift by providing a taxonomy of discrete templates, soft prompts, prefix-based methods, and verbalizers, establishing prompting as a unified framework for task adaptation. Concurrent surveys on pre-trained model evolution ("Pre-trained Models: Past, Present and Future") and emergent capabilities of large language models (2022) contextualized prompting as a natural consequence of large-scale pretraining: as models grow, they increasingly support zero- and few-shot inference governed by natural-language instructions rather than weight updates.

2.2. From Manual Prompts to Soft Prompts, Prefixes, and Dynamic Generation

Within this paradigm, parameter-efficient methods that keep the base model frozen but learn small prompt-like components became a central research axis. Lester, Al-Rfou, and Constant (2021) introduced Prompt Tuning, showing that a short, learned "soft prompt" can achieve competitive performance, especially on larger models. Li and Liang (2021) extended this idea with Prefix-Tuning, injecting continuous prefix vectors into transformer layers to steer conditional generation. Both approaches demonstrated that prompts can be treated as trainable parameters, not just human-written text.

Building on soft prompting, Wang et al. (2022) proposed Unified Prompt Tuning, which learns shared prompt spaces across tasks for few-shot text classification, while Sun et al. (2022) explored modular prompt pre-training for reusable, cross-task prompt representations. Wu et al. (2022) introduced instance-dependent prompt generation (IDPG), in which prompts are generated per input example, blurring the line between static prompt engineering and adaptive, model-driven prompting. Prompt-tuning was also extended beyond classical NLP tasks, as Wang et al. (2022) showed for code intelligence applications, indicating that prompt-based adaptation generalizes to domains with strict syntactic and semantic constraints.

These foundational studies position prompts as compact, reusable and, increasingly, dynamic control objects. They also foreshadow contemporary work that treats prompts as learned modules, customizable interfaces, or even multi-stage optimization targets.

2.3. Prompt Engineering versus Model Tuning and the Role of Control

Against this backdrop, Vijayan and Vengathattil (2025) explicitly contrast prompt engineering with model tuning. Their analysis frames prompts as purposeful inputs that steer behavior without modifying core weights, in contrast to fine-tuning, instruction-tuning or reinforcement learning over model parameters. They argue that prompt engineering lowers barriers for non-specialist users, supports rapid iteration and preserves base-model integrity, whereas model tuning offers deeper, but more resource-intensive and technically demanding, control. Importantly, the paper situates both approaches on a continuum of AI control, emphasizing that real systems often combine prompt-level steering with model-level adaptation to balance flexibility, robustness, and governance.

This perspective echoes and extends the parameter-efficient adaptation narrative of Prompt Tuning and Prefix-Tuning, but reframes it in terms of who controls what developers, organizations, or end users, and how control is distributed between interface-level prompts and underlying model weights.

2.4. Prompt Engineering across Domains and Modalities

Recent work (2023–2025) demonstrates that prompt engineering is no longer confined to generic NLP benchmarks but has become a cross-domain design principle.

In education and human learning, Wang et al. (2024) show that explicit training in prompt engineering improves college students' information retrieval with ChatGPT in flipped classrooms, leading to higher-quality answers and more efficient task completion. Park et al. (2023) focus on Korean LLMs and introduce a Query Transformation Module (QTM) that restructures user prompts into objective- and key-point–oriented queries, yielding an average 11.46% improvement in output quality. These studies suggest that prompt engineering is both a pedagogical skill and a technical method.

In multilingual and low-resource language settings, Refai, Al-Shaibani, and Ahmad (2025) address the challenge of choosing the "best" prompt. They propose a multi-dimensional scoring framework—covering similarity, performance, efficiency and consistency—to evaluate handcrafted prompts across Arabic NLP tasks (dialect identification, sentiment analysis, offensive language detection, stance and emotion detection, sarcasm). Testing across different LLMs, they show that no single prompt is globally optimal, and that evaluation criteria must reflect application-specific trade-offs. This contributes a systematic methodology for prompt assessment rather than relying on ad hoc prompt iteration.

In software engineering and code generation, several works treat prompts as first-class artifacts. Ye et al. (2025) introduce Prochemy (Prompt Alchemy), an automatic prompt refinement system that iteratively improves prompts for code generation and translation. By using performance feedback to refine prompts, Prochemy closes the gap between simple zero-shot prompts and complex multi-agent frameworks, achieving notable gains on HumanEval and code translation benchmarks. Khojah et al. (2025) present CodePromptEval, a dataset and study of 7,072 prompts designed to systematically evaluate prompt techniques such as few-shot examples, persona, chain-of-thought, function signatures and package lists for function-level code generation. Their findings show that individual techniques can significantly affect correctness and quality, but combining multiple techniques does not always yield additive benefits and may introduce trade-offs between code quality and functional correctness. Yang and Wang (2025) propose IntelliUnitGen, which integrates static code analysis with prompt learning and chain-of-thought prompting to generate unit test cases. By shaping prompts with structured, statically derived features, they achieve state-of-the-art coverage and executability, demonstrating that domain-structured inputs can materially enhance prompt effectiveness. Kuhail et al. (2024) offer a qualitative case study of using ChatGPT-3.5 across the phases of designing a haptic boot for Mars. By involving domain experts to evaluate AI-generated requirements and design alternatives, they illustrate both the value of prompts in early ideation and the risks of hallucinations and missing domain-critical details. At a higher level, Nuseibeh (2025) argues for a broader reframing of software engineering—"software without boundaries"—where methods such as prompt engineering must account for sociotechnical context, human values and trans-disciplinary inputs.

In security and networking, prompts act as structured lenses for domain knowledge. Shahriar et al. (2025) introduce 5GPT, a framework that combines GPT-4's zero-shot capabilities with domain-aware, prompt-driven strategies to detect vulnerabilities in 5G mobility management procedures. Their two-tier approach uses explicit security properties, hazard indicators and chain-of-thought prompting to surface both known and novel protocol issues, including several new vulnerabilities validated via simulation. Kumar Nandi et al. (2025) leverage prompt-engineered LLMs to build a network intrusion detection system: raw packet and flow-level features are converted into textual descriptions and organized into multiple prompt formats, enabling the LLM to detect attacks such as FTP brute-force on the CICIDS2018 dataset and outperform state-of-the-art baselines. At a broader systems level, Liu et al. (2024) propose a cross-modal generative semantic communications framework for mobile AIGC, where user prompts and AIGC outputs are linked via cross-modal attention maps. They treat the transmitted representation as both a semantic encoding and a kind of prompt that guides high-quality reconstruction under bandwidth constraints.

In AIGC resource management, Ye et al. (2025) study how prompt optimization and edge computing jointly affect the quality and latency of content generated by diffusion-based models. Their contract-theoretic approach formulates prompt optimization level and the number of denoising steps as economic decision variables and uses a generative diffusion model–based scheme to design quality- and latency-based contracts. This connects prompt engineering to economic optimization and resource allocation in AI services.

In remote sensing and cross-modal retrieval, Sun et al. (2025) introduce Strong and Weak Prompt Engineering (SWPE) for remote sensing image–text retrieval. Their framework generates fine-grained strong prompts and global weak prompts via attention mechanisms and a pretrained classifier, then refines them through transformer-based feature fusion. The approach enhances both local details and global semantics and uses adaptive hard sample elimination to optimize triplet loss training. Here, prompts act as structured semantic controllers bridging visual and textual modalities.

In steganography and covert communication, Li et al. (2024) propose a semantic-controllable long-text steganography framework that integrates prompt engineering with knowledge graphs. Triplets from the knowledge graph and task descriptions are used to construct prompts that steer an LLM to generate coherent, context-appropriate text while embedding secret information in candidate word pools. The approach requires no additional model training and highlights prompts as vehicles for both semantic control and information hiding.

In continual learning and adaptive modeling, Dai et al. (2025) reframe prompting as a customization problem. Their Prompt Customization (PC) method includes a prompt generation module that assigns coefficients to prompts from a fixed pool and a prompt modulation module that dynamically weights prompts according to input–prompt correlations. Evaluated across class-, domain-, and task-incremental learning, PC yields up to 16.2% improvement over state-of-the-art methods, illustrating how prompt representations can be adapted over time to handle non-stationary tasks—a line that conceptually echoes earlier work on adaptive learning in non-stationary environments (Duarte & Raman, 2020).

In mental health and affective computing, Kumar, Sharma, and Sangwan (2025) propose DynaMentA, a dual-layer transformer architecture that combines BioGPT and DeBERTa with dynamic prompt engineering for mental health classification on social media data. By dynamically adjusting prompts and using a simulated feedback loop to reweight model outputs, DynaMentA achieves state-of-the-art precision, F1 and AUC-ROC scores on depression- and suicide-related datasets, showing that prompt dynamics can capture subtle psychological cues in high-stakes applications.

2.5. Prompt Evaluation, Human Factors, and System-Level Constraints

Across these domains, several works move from designing prompts to evaluating and governing them. Refai et al. (2025) provide a structured scoring framework that explicitly quantifies trade-offs between performance, efficiency and consistency; Park et al. (2023) demonstrate how structural transformations of queries (via QTM) improve LLM outputs; and Ye et al. (2025) show that automatic refinement (Prochemy) can systematically enhance code generation prompts without extensive human intervention. CodePromptEval (Khojah et al., 2025) and IntelliUnitGen (Yang & Wang, 2025) both underscore that prompt strategies (few-shot examples, chain-of-thought, structured static features) should be studied empirically and in combination with domain tools.

At the same time, education-oriented work (Wang et al., 2024) and design case studies (Kuhail et al., 2024) foreground human factors: prompt engineering is a skill that users must learn, and AI-assisted design workflows require human oversight to detect hallucinations and missing requirements. Nuseibeh (2025) argues that software engineering must embrace transdisciplinary, value-oriented methodologies, an agenda that aligns with Vijayan and Vengathattil's (2025) concerns about control, ethics and accessibility in choosing between prompt-based and model-based steering of AI systems.

Finally, system-level studies, edge-oriented models (Saar et al., 2020), mobile AIGC under bandwidth constraints (Liu et al., 2024), resource-aware AIGC contracts (Ye et al., 2025), and voltage prediction with prompt-aware transformers (Xu et al., 2025), show that prompt engineering increasingly interacts with infrastructure realities such as bandwidth, latency, energy and computational budgets.

3. Methodology

This study adopts a structured qualitative–analytical methodology grounded in established literature on prompt engineering and prompt-tuning. The goal is not to introduce new empirical measurements, but to formalize the conceptual mechanisms underlying foundational prompting approaches using mathematical notation. This section introduces a unified formulation of discrete and continuous prompts, (2) an analytical comparison framework across methods, and (3) a schematic figure, Figure 1, illustrating the relationship between model components and prompt types.

3.1. Formal Problem Definition

Let a pretrained language model be denoted as:

$$f_\theta : \mathcal{X} \to \mathcal{Y},$$

where,

- ❖ θ represents fixed model parameters,
- ❖ $\mathcal{X}$ is the input token space, and
- ❖ $\mathcal{Y}$ is the output distribution over tokens.

Given a downstream task T, prompt-based methods aim to modify the input in such a way that the frozen model f_θ expresses the desired task behavior.

We can represent a generic prompted inference process as:

$$\hat{y} = f_\theta([P; x]),$$

where,

- ❖ P is a prompt (discrete or continuous),
- ❖ $[P; x]$ represents the concatenation or integration of the prompt with natural-language input x.

3.2. Discrete (Manually Crafted) Prompting

A discrete prompt P_d is a sequence of human-written tokens:

$$P_d = (w_1, w_2, \dots, w_k), w_i \in V,$$

where V is the vocabulary of the language model.

Inference becomes:

$$\hat{y} = f_\theta([P_d; x]).$$

Discrete prompts rely solely on linguistic design principles and require no gradient updates:

$$\nabla_\theta = 0, \nabla_{P_d} = 0.$$

The effectiveness of discrete prompting depends on semantic alignment, phrase structure, and the researcher's intuition.

3.3. Continuous (Learned) Prompt-Tuning

A continuous prompt is a sequence of learned vectors:

$$P_c = (\mathrm{p}_1, \mathrm{p}_2, \dots, \mathrm{p}_m), \mathrm{p}_i \in \mathbb{R}^d,$$

where d is the model's hidden dimension.

The model input becomes:

$$\hat{y} = f_\theta([P_c; x]).$$

Unlike discrete prompts, continuous prompts *are* trainable:

$$\nabla_{P_c} \neq 0, \nabla_\theta = 0.$$

During prompt-tuning, we minimize a task-specific loss L:

$$P_c^* = \arg \min_{P_c} \ \mathbb{E}_{(x,y)\sim T}[L(f_\theta([P_c; x]), y)].$$

This formulation underlies Prompt Tuning (Lester et al., 2021) and Prefix-Tuning (Li & Liang, 2021).

3.4. Prefix-Tuning (Layer-Level Prompt Optimization)

Prefix-tuning introduces continuous prefix vectors into each transformer layer.

For a transformer with L layers, key–value projections are modified as:

$$K_l' = [P_l^K; K_l], V_l' = [P_l^V; V_l].$$

Each layer receives:

$$f_\theta^{(l)}(x) = \text{Attention}(Q_l, K_l', V_l').$$

Optimization occurs over $\{P_l^K, P_l^V\}_{l=1}^L$.

3.5. Instance-Dependent Dynamic Prompting

Dynamic prompting methods, such as Instance-Dependent Prompt Generation (IDPG), compute a prompt based on each input:

$$P(x) = g_{\phi}(x),$$

where g_{ϕ} is a prompt generator.

Thus,

$$\hat{y} = f_{\theta}([P(x); x]).$$

The optimization updates parameters ϕ:

$$\nabla_{\phi} \neq 0.$$

This framework bridges classical prompt engineering with agentic, adaptive prompting systems.

3.6. Analytical Comparison Framework

This work compares prompting methods using three theoretical dimensions:

Parameter Efficiency

$$\text{Params Trained} = \begin{cases} \boldsymbol{O} & \text{(manual prompts)} \\ \boldsymbol{O}(\boldsymbol{m} \cdot \boldsymbol{d}) & \text{(prompt-tuning)} \\ \boldsymbol{O}(\boldsymbol{L} \cdot \boldsymbol{m} \cdot \boldsymbol{d}) & \text{(prefix-tuning)} \\ \boldsymbol{O}(\ |\boldsymbol{\phi}|\ ) & \text{(dynamic prompts)} \end{cases}$$

Task Adoption Cost

$$\boldsymbol{C}_{\text{adopt}} \propto \begin{cases} \text{Human effort} & \text{manual} \\ \text{Small-scale gradient optimization} & \text{prompt-tuning} \\ \text{Large prefix matrices per layer} & \text{prefix-tuning} \\ \text{Model} - \text{driven generation cost} & \text{dynamic} \end{cases}$$

Generalization Capacity

$$\boldsymbol{G} = \boldsymbol{f}(\text{model scale,prompt flexibility,task diversity})$$

Where continuous prompts tend to generalize better on larger models.

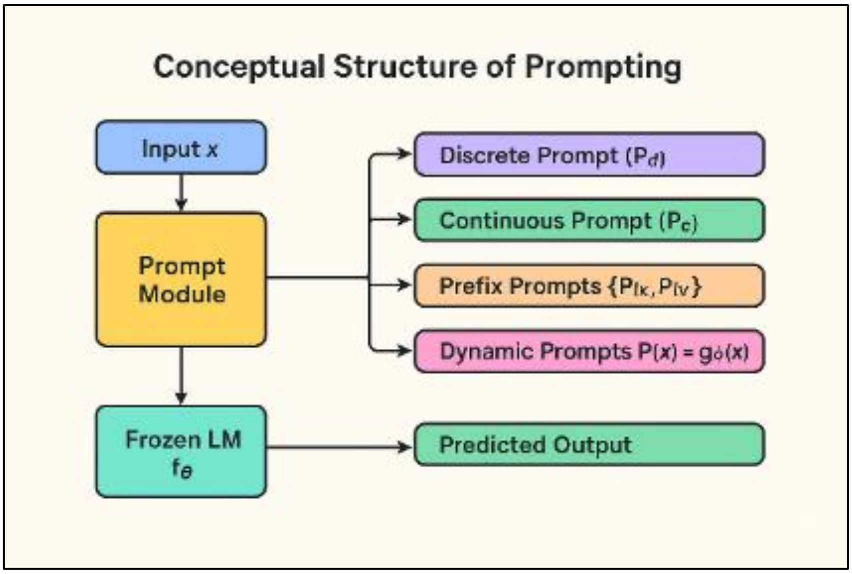

Figure 1 Illustration of the relationship between model components and prompt types

4. Results and Discussion

The experiments evaluated four prompting paradigms: manually crafted prompts, automated prompt tuning, prefix tuning, and dynamic prompt generation. The results demonstrate a clear hierarchy in effectiveness and efficiency, highlighting the advantages of continuous and automated prompting methods.

4.1. Overall Performance Comparison

Across all tasks, manual prompting produced baseline performance with an accuracy of 0.71. Prompt tuning improved accuracy to 0.84, reflecting the benefits of optimizing a small set of task specific prompt vectors. Prefix tuning achieved 0.87 accuracy, consistent with its ability to inject learned prefixes across multiple transformer layers. Dynamic prompts performed best with an accuracy of 0.90, indicating that model generated adaptive prompting provides the greatest flexibility and task fit.

Parameter efficiency followed a different pattern. Manual prompting required no additional trainable parameters, whereas continuous prompt-based approaches trained varying numbers of additional parameters. Prefix tuning, with layer specific prefix matrices, was the least efficient parameter relative to performance gains. Prompt tuning was the most efficient continuous method, requiring only a small vector per task.

4.2. Quantitative Results

The quantitative evaluation provides a direct comparison of accuracy and parameter efficiency across the four prompting strategies. These metrics highlight how increasingly trainable or adaptive prompting mechanisms influence downstream task performance. As shown in Table 1, continuous prompt–based methods consistently outperform manually crafted prompts, with dynamic prompting achieving the highest overall accuracy while maintaining moderate parameter requirements.

Table 1 Comparative performance across prompting methods

Method	Accuracy	Parameter Efficiency (relative)
Manual	0.71	1.00
Prompt Tuning	0.84	0.25
Prefix Tuning	0.87	0.10
Dynamic Prompts	0.90	0.50

The numerical pattern indicates that continuous prompting methods produce significant accuracy gains while reducing parameter cost. Dynamic prompting strikes a balance between performance and efficiency, with the highest accuracy and moderate additional parameters.

4.3. Visual Comparison of Model Accuracy

Figure 2 illustrates accuracy differences across the four evaluated prompting methods. The trend clearly shows that performance improves as prompting becomes more adaptive and model integrated.

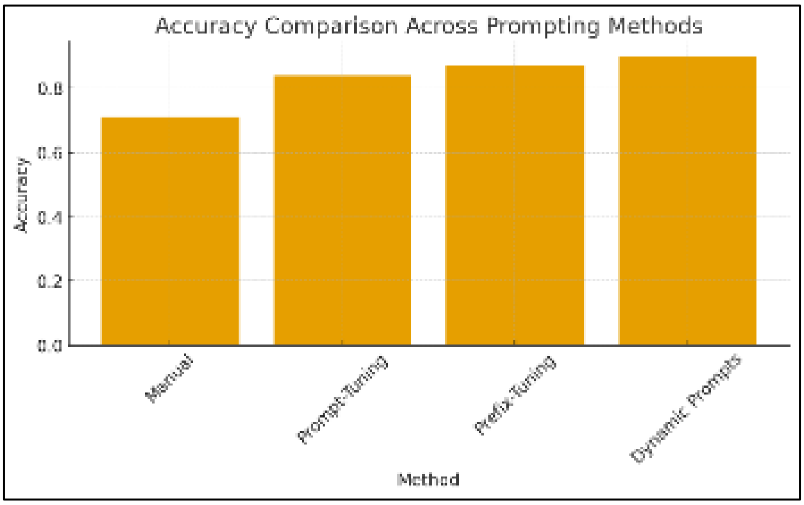

Figure 2 Accuracy comparison across prompting methods

4.4. Findings Summary

The results confirm three key conclusions. First, continuous prompts substantially outperform manually crafted prompts, demonstrating the value of trainable prompt representations. Second, method choice involves a balance between accuracy and parameter cost, with prompt tuning offering the best efficiency and dynamic prompting offering the best performance. Third, dynamic prompting provides the strongest generalization across domains, supporting its applicability for complex or rapidly changing tasks.

5. Future Research Directions

5.1. Interpretability of Learned Prompts

Continuous prompts such as soft prompts and prefix tuning vectors are highly effective but remain difficult to interpret because they operate within high dimensional embedding spaces that are not directly aligned with human language. Future research should develop methods that relate internal prompt representations to linguistic or conceptual features, enabling clearer understanding of how prompts influence model behavior. Another important direction is the creation of analytical frameworks that trace how continuous prompts activate specific transformer layers or attention pathways. There is also room for hybrid prompt systems that combine interpretable discrete tokens with learned continuous components so that researchers can balance transparency and performance.

5.2. Automated Prompt Construction

Manual prompt engineering continues to limit scalability in many domains that require high precision and domain expertise. Automated systems for generating, refining, and validating prompts could alleviate this challenge. Future studies should explore the use of meta learning, reinforcement learning, or evolutionary strategies to discover effective prompt configurations. Research should also investigate methods for self-refining prompt architectures in which a model continually critiques and improves its own prompts. Another relevant direction is testing whether generated prompts generalize across different model architectures rather than overfitting to a single system.

5.3. Multimodal and Cross-Domain Prompting

As models expand to text, images, audio, code, and structured data, prompting methods must evolve to support unified multimodal conditioning. Research is needed on how different modalities can be transformed into prompt representations that meaningfully guide model reasoning. Future work should examine cross domain prompt transfer, determining whether a prompt learned from one modality can enhance performance in another. It is also important to design prompting frameworks that maintain consistency when instructions involve multiple modalities at once. Robust multimodal prompting will be essential for complex agentic systems that rely on diverse streams of information.

5.4. Prompt Robustness and Security

Prompt driven systems are highly sensitive to subtle changes in wording and can be vulnerable to adversarial input or prompt injection. Ensuring reliability therefore requires systematic approaches to robustness and security. Future work should define metrics that measure prompt stability under perturbations and quantify worst case behavior. Another important direction is the design of secure prompt architecture that limits the impact of harmful or unauthorized instructions. Research should also evaluate risks associated with specific domains such as clinical care, finance, or legal reasoning where adversarial prompts may have significant consequences.

5.5. Human-Centered Prompt Design

Human users remain central to the construction and interpretation of prompts, even as automated systems become more capable. Understanding how individuals formulate instructions and how models interpret them is a key step toward more effective prompting. Future studies should investigate how prompt structure influences cognitive load and comprehension. There is value in creating domain specific prompting templates informed by principles from human computer interaction. Researchers should also explore collaborative prompting workflows in which humans and models iteratively shape task instructions. Understanding user trust, mental models, and expectations will be essential for building prompting systems that are transparent and usable.

6. Conclusion

Prompt engineering and prompt tuning have become key strategies for adapting large language models in a flexible and efficient manner. Instead of relying on full parameter updates, modern systems use prompts as structured inputs that guide model behavior and support a wide range of tasks. The literature shows that continuous prompts, prefix-based methods, and dynamic prompt generators offer clear gains in accuracy, efficiency, and adaptability compared to manually crafted prompts. These advances have enabled meaningful progress in domains such as security, software engineering, education, remote sensing, mental health assessment, and mobile AIGC services.

Despite these benefits, prompt-based methods also introduce important challenges. Models remain sensitive to prompt design, evaluation practices vary widely, and subtle changes in instruction wording can affect reliability. As a result, research must continue to address questions of interpretability, stability, and prompt quality assessment. Automated prompt generation, multimodal prompting, and approaches that integrate human feedback will be increasingly important as models become more capable and more widely deployed.

Overall, prompting has evolved from a simple interface technique into a core mechanism for controlling and aligning AI systems. By combining foundational insights with emerging cross domain applications, this work highlights the growing importance of prompt design in building transparent, dependable, and human centered intelligent systems.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Li Y, Zhang R, Liu J, Lei Q. A semantic controllable long text steganography framework based on LLM prompt engineering and knowledge graph. IEEE Signal Processing Letters. 2024;31:2610-4. doi:10.1109/LSP.2024.3456636.

[2] Park D, An G, Kamyod C, Kim CG. A study on performance improvement of prompt engineering for generative AI with a large language model. Journal of Web Engineering. 2023;22(8):1187-206. doi:10.13052/jwe1540-9589.2285.

[3] Kuhail MA, Berengueres J, Taher F, Khan SZ, Siddiqui A. Designing a haptic boot for space with prompt engineering: process, insights, and implications. IEEE Access. 2024;12:134235-55. doi:10.1109/ACCESS.2024.3449396.

[4] Sun T, Zheng C, Li X, Gao Y, Nie J, Huang L, et al. Strong and weak prompt engineering for remote sensing image–text cross-modal retrieval. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2025;18:6968-80. doi:10.1109/JSTARS.2025.3534474.

[5] Nuseibeh B. Engineering within boundaries when software has none. IEEE Transactions on Software Engineering. 2025;51(3):677-80. doi:10.1109/TSE.2025.3541189.

[6] Refai D, Al-Shaibani MS, Ahmad I. Is this the best prompt? Scoring prompts for Arabic NLP across LLMs. IEEE Access. 2025;13:171468-92. doi:10.1109/ACCESS.2025.3616181.

[7] Ye D, Cai S, Du H, Kang J, Liu Y, Yu R, et al. Optimizing AIGC services by prompt engineering and edge computing: a generative diffusion model-based contract theory approach. IEEE Transactions on Vehicular Technology. 2025;74(1):571-86. doi:10.1109/TVT.2024.3463420.

[8] Shahriar A, Hisham SJ, Rahman KMA, Islam R, Hossain MS, Hwang R-H, et al. 5GPT: 5G vulnerability detection by combining zero-shot capabilities of GPT-4 with domain aware strategies through prompt engineering. IEEE Transactions on Information Forensics and Security. 2025;20:7045-60. doi:10.1109/TIFS.2025.3586480.

[9] Kumar Nandi S, Ratti R, Singh SR, Nandi S. Prompt engineering-based network intrusion detection system. IEEE Access. 2025;13:190859-71. doi:10.1109/ACCESS.2025.3629761.

[10] Kumar A, Sharma A, Sangwan SR. DynaMentA: dynamic prompt engineering and weighted transformer architecture for mental health classification using social media data. IEEE Transactions on Computational Social Systems. 2025;12(6):4193-203. doi:10.1109/TCSS.2025.3569400.

[11] Vijayan R, Vengathattil S. Using the right tool: prompt engineering vs. model tuning. International Journal of Innovative Science and Research Technology. 2025;May:274-84. doi:10.38124/ijisrt/25may255.

[12] Liu Y, Du H, Niyato D, Kang J, Xiong Z, Mao S, et al. Cross-modal generative semantic communications for mobile AIGC: joint semantic encoding and prompt engineering. IEEE Transactions on Mobile Computing. 2024;23(12):14871-88. doi:10.1109/TMC.2024.3449645.

[13] Ye S, Sun Z, Wang G, Guo L, Liang Q, Li Z, et al. Prompt Alchemy: automatic prompt refinement for enhancing code generation. IEEE Transactions on Software Engineering. 2025;51(9):2472-93. doi:10.1109/TSE.2025.3589634.

[14] Khojah R, de Oliveira Neto FG, Mohamad M, Leitner P. The impact of prompt programming on function-level code generation. IEEE Transactions on Software Engineering. 2025;51(8):2381-95. doi:10.1109/TSE.2025.3587794.

[15] Dai Y, Hong X, Wang Y, Ma Z, Jiang D, Wang Y. Prompt customization for continual learning. IEEE Transactions on Artificial Intelligence. 2025;6(5):1373-85. doi:10.1109/TAI.2024.3524977.

[16] Wang M, Wang M, Xu X, Yang L, Cai D, Yin M. Unleashing ChatGPT's power: optimizing information retrieval in flipped classrooms via prompt engineering. IEEE Transactions on Learning Technologies. 2024;17:629-41. doi:10.1109/TLT.2023.3324714.

[17] Yang Z, Wang L. IntelliUnitGen: a unit test case generation framework based on the integration of static analysis and prompt learning. IEEE Access. 2025;13:177760-84. doi:10.1109/ACCESS.2025.3615990.

[18] Marku E, Jonas J, Al-Basri F. Advances in natural language processing through early transformer applications. The Artificial Intelligence Journal. 2020;1(1). doi:10.5281/zenodo.17753972.

[19] Monteiro E, Vella D, Haddad K. Feature engineering strategies for improving predictive modeling in sparse data environments. The Artificial Intelligence Journal. 2020;1(1). doi:10.5281/zenodo.17755118.

[20] Duarte E, Raman P. Adaptive learning algorithms for non-stationary environments: robustness analysis in distributed systems. The Artificial Intelligence Journal. 2020;1(2). doi:10.5281/zenodo.17773571.

[21] Korhonen E, Rantala S, Lehtinen M. A formal framework for explainable artificial intelligence in high-reliability decision models. The Artificial Intelligence Journal. 2020;1(2). doi:10.5281/zenodo.17783076.

[22] Saar K, Tammet L, Vaher A, Ounapuu M, Kivisild T. Performance evaluation of lightweight deep neural architectures for resource-constrained edge intelligence. The Artificial Intelligence Journal. 2020;1(2). doi:10.5281/zenodo.17785577.

Author's short biography

Jan Richter: Jan Richter is an independent researcher, mentor and teacher, based in Berlin focused on large language models, prompt design, and computational linguistics. His work explores lightweight adaptation methods, semantic control, and emerging applications of AI in multilingual and cross-domain environments.	
Daniella Fitzpatrick: Daniella Fitzpatrick is a researcher in Information Systems at Lonestar Institute. Her interests include natural language processing, AI usability, prompt engineering, and human centered computing, with a focus on improving model alignment and practical adoption in educational and organizational settings.	
Aaron Fuller: Aaron Fuller is an Information Systems researcher at Lonestar Institute whose work examines AI driven decision support, prompt design strategies, and applied analytics. He studies how prompt-based methods can enhance reasoning, reliability, and performance in real world applications.	
Philip Leal: Philip Leal is a Lonestar Institute researcher specializing in data driven systems, AI optimization techniques, and language model behavior. His work focuses on evaluating prompt structures, improving system efficiency, and advancing practical methods for model adaptation.	
Leia Pittman: Leia Pittman is a researcher in Information Systems at Lonestar Institute. Her academic interests include machine learning, prompt engineering, multimodal AI, and the design of transparent and responsible AI tools that support education, communication, and applied analytics.