REVIEW ARTICLE

Reinforcement Learning for Prompt Optimization in Language Models: A Comprehensive Survey of Methods, Representations, and Evaluation Challenges

Zhangqi Liu[1,*]

[1] Brown University, Providence, RI 02912, United States

Abstract

The growing prominence of prompt engineering as a means of controlling large language models has given rise to a diverse set of methods, ranging from handcrafted templates to embedding-level tuning. Yet, as prompts increasingly serve not merely as input scaffolds but as adaptive interfaces between users and models, the question of how to systematically optimize them remains unresolved. Reinforcement learning, with its capacity for sequential decision-making and reward-driven adaptation, has been proposed as a possible framework for discovering effective prompting strategies. This survey explores the emerging intersection of RL and prompt engineering, organizing existing research along three interdependent axes: the representation of prompts (symbolic, soft, and hybrid), the design of RL-based optimization mechanisms, and the challenges of evaluating and generalizing learned prompt policies. Rather than presenting a single unified framework, the discussion reflects the fragmented, often experimental nature of current approaches, many of which remain constrained by unstable reward signals, limited generalizability, and a lack of reproducible evaluation standards. By analyzing methodological innovations and points of friction alike, this work aims to foster a more critical and reflective understanding of what it means to "learn to prompt" in complex, real-world language modeling contexts.

Keywords: prompt engineering, reinforcement learning, language models, prompt optimization, reward design, prompt representation.

1 Introduction

As large language models continue to permeate a wide range of natural language processing tasks, from open-domain question answering to domain-specific summarization, the role of prompt engineering has grown from a peripheral concern into a central design problem. Unlike traditional supervised learning pipelines, where fine-tuned parameters dictate model behavior, prompting relies on manipulating the input context to elicit desired outputs from largely frozen architectures [1]. This seemingly superficial layer of control—composed of instructions, examples, or

Submitted: 01 August 2025
Accepted: 23 August 2025
Published: 14 September 2025

Vol. 2, **No.** 4, 2025.
10.62762/TETAI.2025.790504

***Corresponding author:**
✉ Zhangqi Liu
15322274141@163.com

Citation
Liu, Z. (2025). Reinforcement Learning for Prompt Optimization in Language Models: A Comprehensive Survey of Methods, Representations, and Evaluation Challenges. *ICCK Transactions on Emerging Topics in Artificial Intelligence*, 2(4), 173–181.

stylistic cues—has proven surprisingly influential in shaping model performance. Yet, it remains deeply underformalized.

Despite a proliferation of prompting techniques (see Figure 1 for a taxonomy), ranging from handcrafted templates and instruction-based formats to soft prompt tuning embedded in the model's input space, much of current practice is ad hoc and task-specific. Manually composed prompts, though interpretable and straightforward to implement, often require extensive domain knowledge and iterative human experimentation [2]. On the other hand, embedding-based prompt tuning methods, while automatable, suffer from a lack of transparency and are difficult to generalize across tasks or models. The field thus finds itself suspended between the need for structured, explainable prompt systems and the practical limitations of current design methodologies.

In recent work, reinforcement learning has emerged as a potentially unifying paradigm for prompt optimization. By treating prompt generation as a sequential decision process—with actions modifying prompt components, and rewards derived from downstream model behavior—RL introduces a feedback-driven mechanism to explore and refine prompt strategies [3]. To some extent, this reframes the prompt not merely as an input artifact but as a dynamic, learnable policy interface. However, such integration is far from resolved; RL-based prompting remains fragmented, methodologically diverse, and empirically unstable, partly due to reward sparsity, model stochasticity, and a lack of standardized evaluation frameworks.

This survey aims to provide a structured examination of the emerging intersection between reinforcement learning and prompt engineering. The discussion is organized around three interrelated dimensions: the representation of prompts and their associated action spaces [4], the design and adaptation of reinforcement learning algorithms for prompt optimization, and the practical challenges associated with evaluating learned prompt strategies across models and tasks [5]. Before developing this taxonomy, Section 2 contextualizes recent developments in prompt engineering and RL-based NLP. Finally, Section 5 synthesizes the findings and reflects on the open questions that continue to shape this nascent but increasingly consequential field.

2 Related Work

The intersection of reinforcement learning and prompt engineering arises from two relatively independent research trajectories that have only recently begun to converge. On one hand, prompting has evolved from a heuristic technique to a subject of technical inquiry, with growing efforts toward systematic design and automation [6]. On the other, reinforcement learning has long served as a control mechanism for adaptive behaviors in natural language tasks. Recent attempts to combine these approaches reveal a space of possibilities—but also of tensions—between symbolic control, continuous optimization, and the unpredictable behavior of large language models [7]. Understanding this intersection requires briefly revisiting each trajectory before examining their integration.

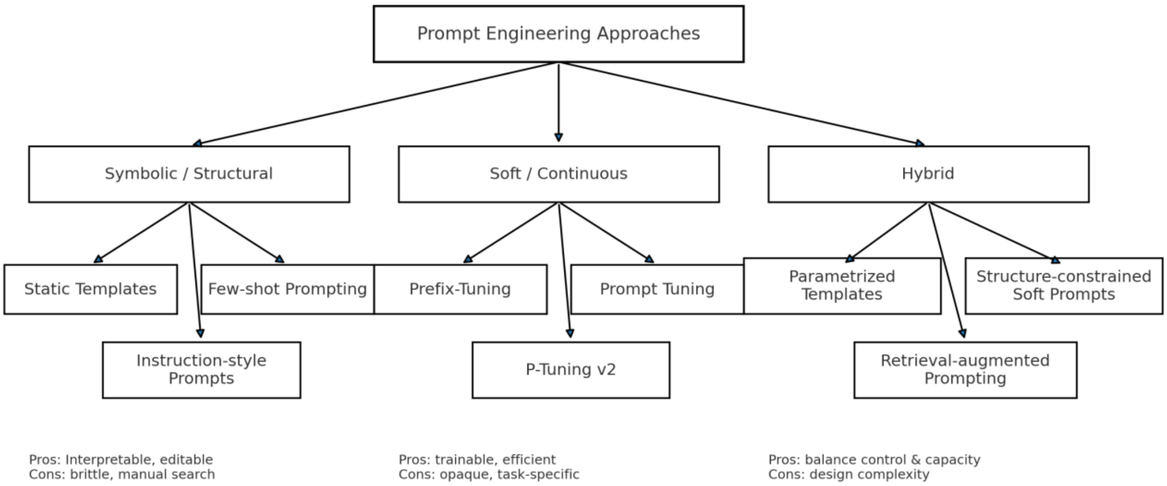
Prompt Engineering Approaches
Symbolic / Structural
Soft / Continuous
Hybrid
Static Templates
Few-shot Prompting
Prefix-Tuning
Prompt Tuning
Parametrized Templates
Structure-constrained Soft Prompts
Instruction-style Prompts
P-Tuning v2
Retrieval-augmented Prompting
Pros: Interpretable, editable
Cons: brittle, manual search
Pros: trainable, efficient
Cons: opaque, task-specific
Pros: balance control & capacity
Cons: design complexity

Figure 1. Taxonomy of prompt engineering approaches.

2.1 Prompt Engineering Techniques

Prompting began as a manual craft, with static templates tailored to elicit specific behaviors from language models. Few-shot prompting extended this by embedding task demonstrations in the prompt, improving performance in some tasks but proving highly sensitive to input order and style. In pursuit of generality and efficiency, researchers introduced soft prompting, in which trainable embeddings replace discrete tokens. While such methods integrate seamlessly into model architectures, they sacrifice interpretability and offer limited structural control [8].

Subsequent methods like AutoPrompt and Prefix-Tuning sought to automate prompt creation, using gradient signals or tunable vectors. Although they improved scalability, these approaches often lacked adaptability across tasks and models [9]. More recent instruction-based prompting techniques, such as those used in InstructGPT, attempt to align model behavior with user goals via natural-language directives [10]. Yet even these remain brittle under distributional shifts, and rely on careful phrasing and implicit human priors. Across this spectrum, one recurring limitation persists: most prompt optimization methods are static, context-agnostic, and poorly suited to dynamic, feedback-driven environments [11].

2.2 Reinforcement Learning in NLP and LLM Alignment

Reinforcement learning has found wide application in NLP scenarios requiring sequential decisions, from dialogue management to summarization. Its most prominent role in recent years has been in reinforcement learning from human feedback (RLHF), which fine-tunes LLMs based on learned reward models aligned with human preferences [12]. However, RLHF primarily optimizes model parameters, treating the prompt as fixed context [13].

Prompt optimization reframes this dynamic by considering the prompt itself as a learnable object. This leads to a natural analogy: if prompt formulation is a policy, then RL becomes an appropriate tool to train it [14]. The reward, in this case, is not simply correctness, but task success, fluency, or alignment with user goals [15]. Despite this conceptual fit, applying RL at the prompt level poses unique challenges—particularly in defining reliable rewards and managing unstable policy updates in high-dimensional, discrete action spaces.

2.3 RL-Based Prompt Optimization Methods

Several early-stage efforts have attempted to bring reinforcement learning into prompt engineering. PromptAgent casts prompt selection as a multi-step decision process [16], learning to choose from predefined templates. AdaPrompt applies PPO to update prompt structures dynamically, with some success in cross-task adaptation. RLPrompt, by contrast, emphasizes symbolic prompt editing—reordering terms, adjusting tone—and optimizes these decisions using policy gradients [17].

Despite variation in formulation, these methods share limitations. Exploration is often inefficient due to the combinatorial nature of prompt structures [18]. Reward signals are noisy, delayed, or brittle, especially when derived from unstable LLM outputs. Generalization remains limited; strategies learned on one task frequently fail to transfer. In aggregate, these works form a fragmented but growing subfield—ambitious in scope, but still searching for theoretical clarity and empirical robustness [19]. A comparative summary of these methods is provided in Table 1.

3 Methodology

Efforts to optimize prompts using reinforcement learning span a broad space of design choices, many of which remain underexplored or only partially formalized [20]. This section aims to synthesize the field's methodological diversity by categorizing current approaches along four dimensions: the representation of prompts, the construction of action spaces and policies, the formulation of reward signals, and the choice of reinforcement learning algorithms. While these dimensions are conceptually separable, they are in practice tightly coupled—representational choices shape the available action space; reward design constrains policy learning; and algorithmic limitations feed back into how prompts are structured and evaluated.

3.1 Prompt Representations

The representation of a prompt plays a central role in determining the granularity and controllability of the optimization process. Broadly, existing work can be divided into three classes.

Symbolic prompts refer to discrete, human-readable structures composed of lexical items, syntactic frames, or stylistic markers. Symbolic formats offer interpretability and facilitate human-in-the-loop

Table 1. Summary of RL-based prompt optimization methods.

Method (Citation)	RL Algorithm	Prompt Representation	Action Space	Reward Signal	Primary Tasks / Datasets	Notes
PromptAgent [9]	Policy gradient / selection (reported as RL template selection)	Symbolic templates	Choose among predefined templates; context-aware switching	Task success (accuracy / EM); LM-based proxies (as reported)	Classification, QA (reported toy-to-medium scale)	Early RL framing of prompt selection; limited structural edits; depends on template pool
AdaPrompt [10]	PPO (policy gradient)	Symbolic structure with learnable parameters	Token/segment edits; structure toggles	Task metrics + optional LM confidence (dense)	Classification (SST-2, MNLI), QA	Adaptive updates per task; sensitivity to reward shaping and decoding settings
Universal Prompt Optimization (UPO-RL) [13]	Policy gradient (reported)	Soft / hybrid (task-conditioned)	Edit / compose prompt embeddings and slots	Mixed: task scores, LM signals	Classification & generation (various)	Aims for cross-task generality; transfer gains modest; tuning cost remains
Prefix-Tuning [5] (baseline)	— (non-RL baseline)	Soft (continuous prefix)	N/A (learned prefix vectors)	Supervised loss / generation objective	Generation tasks (e.g., summarization)	Parameter-efficient but opaque; strong baseline for RL methods
AutoPrompt [4] (baseline)	— (non-RL baseline)	Symbolic (trigger tokens)	Token-level insertions via gradient heuristics	Proxy gradients (MLM/LM objectives)	Classification (cloze-style)	Interpretable triggers; brittle and task-specific; useful comparison point

adjustment [21]. However, the space of meaningful symbolic variations is vast and sparsely populated with effective configurations, making automated search both computationally intensive and sample-inefficient.

In contrast, soft prompts are represented as learned continuous embeddings injected into the model's input layer. These vectors are typically initialized randomly or derived from existing tokens, and optimized via gradient descent or reinforcement feedback. Soft prompts, as seen in methods like P-tuning or Prompt Tuning, excel in parameter efficiency and can be trained quickly. Yet their lack of linguistic transparency makes debugging difficult, and generalization to new tasks or models is not guaranteed [22].

Some recent systems explore hybrid representations, which attempt to combine the structure and interpretability of symbolic prompts with the learning flexibility of soft embeddings. These may involve composing symbolic templates whose components are parameterized by learned vectors, or applying embedding-level tuning to prompts that follow a fixed grammar. Such representations could, in principle, support more robust and generalizable optimization—but their effectiveness remains speculative and empirically inconsistent.

3.2 Action Space and Policy Design

The design of the action space defines what the RL agent can do to a prompt. At one end of the spectrum are token-level operations, including the replacement, insertion, or deletion of individual tokens. These fine-grained actions allow for detailed edits, and are well-suited to symbolic prompts [23]. However, their combinatorial nature poses significant challenges for credit assignment and policy convergence, especially in sparse-reward settings.

At a higher level, structure-level actions manipulate larger prompt components, such as switching between predefined templates, toggling stylistic attributes, or altering sentence order. These actions offer greater abstraction and can reduce the size of the decision space. Yet they often rely on rigid schema definitions, which limit expressiveness, and may require extensive predefinition by human experts.

Policy design further complicates this space. Some systems adopt deterministic policies optimized via policy gradients, while others explore stochastic or hierarchical approaches that select actions in stages. The choice of granularity influences not only learning dynamics but also interpretability—coarse policies may generalize better, but at the cost of transparency and editability. One example of a fine-grained action space and policy architecture is illustrated in Figure 2. To date, there exists no consensus on the optimal level of abstraction, and trade-offs are often task-dependent.

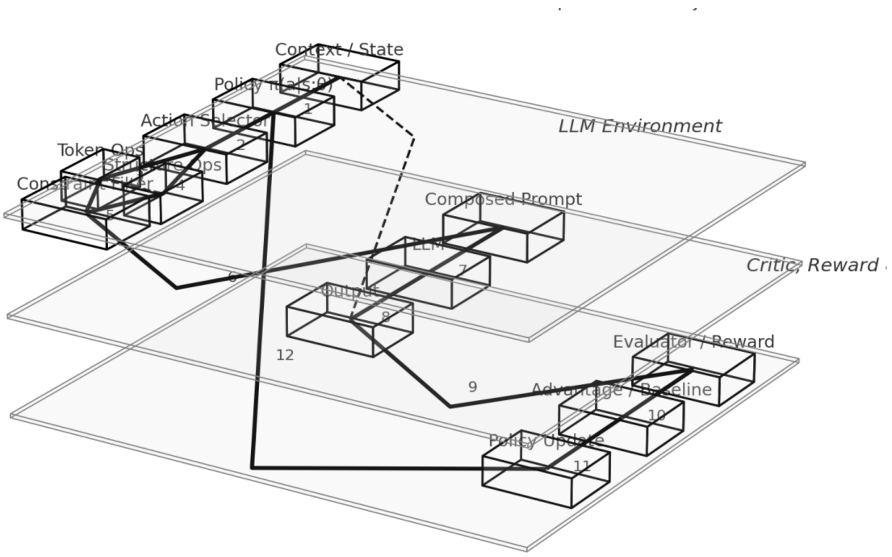

Figure 2. RLPrompt action space and policy architecture.

3.3 Reward Formulation

The reward function is arguably the most consequential and least standardized component in RL-based prompt optimization. One common approach uses task-based metrics such as accuracy, BLEU or ROUGE scores, or exact match. While these metrics align with downstream evaluation, they tend to be sparse and may not provide sufficient gradient signal for prompt refinement, particularly early in training.

Alternatively, some methods utilize language model confidence signals, such as token log-probabilities, perplexity scores, or the entropy of generated distributions [24]. These signals are dense and easier to compute, but may correlate weakly with actual task success or user satisfaction. Moreover, LLMs often assign high probability to generic or evasive responses, which can confound optimization.

A third category employs human-aligned reward models, either via preference learning or proxy scorers like GPTScore. These can reflect nuanced quality judgments—fluency, helpfulness, coherence—but introduce additional complexity in training and evaluation. Furthermore, reward models are themselves imperfect and may encode hidden biases from training data.

Many current studies combine multiple reward sources, either through weighted sums or staged objectives. However, tuning such multi-objective formulations remains more art than science, and the interaction effects between different reward types are poorly understood. This opens space for future exploration, including adaptive reward shaping, task-conditioned weighting, and curriculum-style training regimes.

3.4 RL Algorithms for Prompt Learning

The choice of reinforcement learning algorithm shapes the learning dynamics of prompt optimization, often constraining what representations and reward functions are tractable. The majority of existing work employs policy gradient methods, particularly REINFORCE and PPO, due to their compatibility with high-dimensional, non-differentiable action spaces. While these methods support flexible exploration, they are sensitive to reward variance and require careful baseline estimation to reduce gradient noise.

Value-based methods like Q-learning are less commonly used, in part because of their difficulty in handling large or continuous prompt spaces. However, some hybrid approaches attempt to combine policy and value learning through actor-critic architectures, though empirical results remain mixed.

Several practical challenges arise across algorithmic choices. Reward sparsity limits learning signal in early episodes, especially when the initial prompts perform poorly. Policy instability—amplified by noisy rewards and sensitive model outputs—can lead to oscillatory or brittle behavior [25]. Moreover, most systems do not support prompt transfer across tasks, making training expensive and use-case specific.

Emerging directions include meta-reinforcement learning, which aims to train agents capable of few-shot prompt adaptation, and multi-objective optimization, which balances task success with auxiliary criteria such as robustness, interpretability, or user satisfaction. However, these remain largely theoretical or preliminary in implementation.

4 Experiments

This section surveys the current experimental practices in reinforcement learning–based prompt optimization, focusing not only on task types and model platforms but also on how evaluation metrics and generalization protocols affect the credibility and interpretability of findings [26]. While RLPrompt systems have shown promising results in improving task-specific performance, substantial variance remains in how experiments are designed, evaluated, and reported. These inconsistencies pose challenges to meaningful comparison and generalization, and to some extent, they also reflect the immaturity of the field.

4.1 Task Settings and Models

The empirical evaluation of RL-based prompt learning methods has predominantly centered around classification and generation tasks, often with relatively narrow scope. Classification settings such as sentiment analysis, natural language inference (MNLI), and topic classification are commonly used due to their well-defined output space and ease of accuracy-based evaluation. Meanwhile, text generation tasks such as summarization, open-domain question answering, and even dialogue generation (e.g., MultiWOZ, PersonaChat) have emerged as testbeds for more complex prompt control.

However, these tasks differ not only in structure but in how reward signals are defined and how model sensitivity manifests, which complicates direct performance comparisons across studies. Some prompt optimization methods perform well in classification but fail to show stable improvements in free-form generation tasks, where output evaluation is inherently noisier and often model-dependent [27].

The choice of underlying language model also introduces significant variation. Most existing studies rely on commercial or open-access models like GPT-3, GPT-3.5, T5, or LLaMA-2, though the specific version used is not always disclosed or consistent. This lack of transparency can obscure how much of a method's success stems from the prompt policy itself versus the inherent robustness of the LLM being used. Moreover, model versioning affects prompt behavior in subtle and sometimes unpredictable ways, meaning a prompt optimized for GPT-3 may perform poorly when transferred to GPT-3.5 or GPT-4. These platform dependencies raise concerns regarding both reproducibility and the long-term relevance of learned prompt strategies.

4.2 Evaluation Metrics

The evaluation of RLPrompt methods hinges on how reward and success are operationalized, yet there is currently no unified metric framework. For classification tasks, accuracy remains the dominant measure. For generation tasks, researchers typically report BLEU, ROUGE, or METEOR, though the correlation between these metrics and perceived output quality—especially in instruction-following settings—is often weak. In response to this, some have introduced language model–based scorers, such as log-probabilities, perplexity, or GPTScore, to serve as dense reward signals or post-hoc evaluation tools.

Nevertheless, each metric brings its own biases and limitations. Token-level metrics like BLEU may reward surface-level similarity while ignoring semantic relevance. Perplexity may favor fluent but vacuous outputs. Human evaluations are arguably more reliable but are costly, inconsistent, and difficult to scale. Moreover, when used as reward functions, these metrics may skew the optimization process, leading to degenerate prompt strategies that exploit quirks in scoring algorithms rather than truly improving task performance.

Another issue that remains underdiscussed is output variance. LLMs are inherently stochastic, and small changes in prompt structure or model temperature can yield drastically different outputs. Yet few studies report confidence intervals or run multiple seeds, making it difficult to assess whether observed improvements are statistically meaningful or simply artifacts of model volatility.

4.3 Generalization and Reproducibility

Beyond raw performance, a key concern for any optimization method is whether it generalizes—across tasks, across domains, and across models. Existing RLPrompt systems, however, often show limited transferability. A prompt policy trained on sentiment classification rarely improves inference or summarization tasks, and structural variations in task instructions can significantly degrade performance. These sensitivities suggest that current prompt learning agents are overfitting to narrow behavioral regimes rather than learning broadly applicable prompting strategies. This limited generalization is quantitatively evidenced in Figure 3, which shows significant performance degradation across tasks.

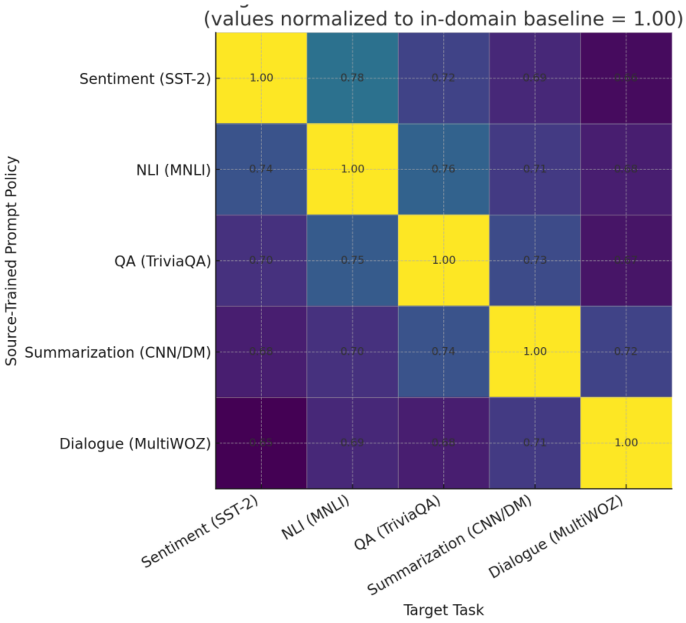

Figure 3. Cross-task generalization performance.

Reproducibility poses another fundamental challenge. Many papers rely on proprietary APIs or closed-source models, where internal updates to the model weights or system instructions can silently invalidate earlier results. Even in open models, non-determinism in decoding, lack of seed control, and insufficient hyperparameter reporting hinder replication. In some cases, access to specific model checkpoints or tokenization versions is no longer possible, limiting the longevity of empirical findings [28].

Moreover, the field currently lacks standardized evaluation benchmarks or testing suites tailored to RL-based prompt learning. Without shared datasets, evaluation scripts, and reproducibility checklists, it is difficult to isolate algorithmic improvements from dataset selection or prompt engineering heuristics.

Efforts to introduce prompt generalization tests, such as cross-task transfer or robustness under paraphrasing, remain limited in scope and largely anecdotal. This limitation constrains the ability to draw principled conclusions about what kinds of prompts—or what forms of reinforcement learning—yield reliably effective strategies.

5 Conclusion

The emergence of reinforcement learning as a mechanism for optimizing prompts challenges the long-standing assumption that prompts are static, fixed inputs. Instead, as the surveyed literature suggests, prompts may be more appropriately understood as learnable control policies—entities that can be adapted, shaped, and strategically deployed in response to model behavior and task demands. This shift in framing does not merely offer a new methodological toolset [29]; it reflects a more profound reconceptualization of the interface between humans and language models, where control is not imposed exogenously but learned endogenously through feedback and exploration.

Within this new paradigm, reinforcement learning provides a natural pathway for exploring the structure space of prompt design. From token-level edits to structural reconfigurations, from hand-crafted templates to embedding-based representations, RL allows systems to discover prompt policies that may outperform handcrafted solutions, particularly in non-obvious or high-dimensional contexts [30]. Yet, the apparent promise of these systems must be weighed against the methodological and theoretical difficulties they continue to face.

Among these, the design of reward functions remains particularly unsettled. Whether based on task-specific metrics, model-derived scores, or human feedback, reward signals tend to be noisy, brittle, and often misaligned with long-term learning goals. Additionally, many of the prompt optimization strategies surveyed demonstrate limited generalizability—they work well within narrowly defined tasks but struggle to transfer across domains, models, or even slight variations in instruction phrasing. Likewise, prompt representations remain an open question: symbolic prompts offer interpretability but are difficult to optimize, while soft prompts are trainable but opaque and hard to control.

Considering these limitations, there is a growing need for a more systematic, reproducible, and collaborative research ecosystem around RL-based prompt engineering [31]. This includes the development of standardized benchmark tasks, shared evaluation protocols, and open-source toolkits that support transparent experimentation and ablation. Without such shared infrastructure, meaningful progress risks becoming fragmented and difficult to validate.

More broadly, this survey has aimed to map the conceptual and methodological landscape of RLPrompt systems without prematurely resolving it. Many of the assumptions underlying current work—about what makes a good prompt, how learning should occur, and what success looks like—remain open to redefinition. Engaging with these uncertainties is not a weakness of the field, but perhaps its most valuable opportunity for theoretical growth.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Zheng, H., Shen, L., Tang, A., Luo, Y., Hu, H., Du, B., ... & Tao, D. (2025). *Learning from models beyond fine-tuning. Nature Machine Intelligence, 7*(1), 6-17. [Crossref]

[2] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys, 55*(9), 1-35. [Crossref]

[3] Banerjee, C., & Nazir, M. S. Zero-Shot Llms in Human-in-The-Loop Rl: Replacing Human Feedback for Reward Shaping. *Available at SSRN 5218722*. [Crossref]

[4] Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., ... & Zhao, L. (2023). Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*. [Crossref]

[5] Xin, X., Pimentel, T., Karatzoglou, A., Ren, P., Christakopoulou, K., & Ren, Z. (2022, July). Rethinking reinforcement learning for recommendation: A prompt perspective. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1347-1357). [Crossref]

[6] Lester, B., Al-Rfou, R., & Constant, N. (2021, November). 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). [Crossref]

[7] Lu, D., Wu, S., & Huang, X. (2025, March). Research on personalized medical intervention strategy generation system based on group relative policy optimization and time-series data fusion. In *Proceedings of the 2025 International Conference on Health Big Data* (pp. 86-91). [Crossref]

[8] Wu, S., Huang, X., & Lu, D. (2025, March). Psychological health knowledge-enhanced LLM-based social network crisis intervention text transfer recognition method. In *Proceedings of the 2025 International Conference on Health Big Data* (pp. 156-161). [Crossref]

[9] Shih, K., Deng, Z., Chen, X., Zhang, Y., & Zhang, L. (2025, May). DST-GFN: A Dual-Stage Transformer Network with Gated Fusion for Pairwise User Preference Prediction in Dialogue Systems. In *2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 715-719). IEEE. [Crossref]

[10] Xing, Y., & Liu, P. (2023, June). Prompt and instruction-based tuning for response generation in conversational question answering. In *International conference on applications of natural language to information systems* (pp. 156-169). Cham: Springer Nature Switzerland. [Crossref]

[11] Meynhardt, C., Meybohm, P., Kranke, P., & Hölzing, C. R. (2025). Advanced Prompt Engineering in Emergency Medicine and Anesthesia: Enhancing Simulation-Based e-Learning. *Electronics, 14*(5), 1028. [Crossref]

[12] Feng, H., Dai, Y., & Gao, Y. (2025). Personalized Risks and Regulatory Strategies of Large Language Models in Digital Advertising. *arXiv preprint arXiv:2505.04665*. [Crossref]

[13] Huang, T., Yi, J., Yu, P., & Xu, X. (2025, March). Unmasking digital falsehoods: A comparative analysis of LLM-based misinformation detection strategies. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)* (pp. 2470-2476). IEEE. [Crossref]

[14] Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., & Gan, C. (2022, June). Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning* (pp. 24631-24645). PMLR.

[15] Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020, November). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4222-4235). [Crossref]

[16] Do, X. L., Dinh, D., Nguyen, N. H., Kawaguchi,

K., Chen, N., Joty, S., & Kan, M. Y. (2025, July). What Makes a Good Natural Language Prompt?. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 5835-5873). [Crossref]

[17] Yi, Q., He, Y., Wang, J., Song, X., Qian, S., Yuan, X., ... & Shi, T. (2025). Score: Story coherence and retrieval enhancement for ai narratives. *arXiv preprint arXiv:2503.23512*.

[18] Saletta, M., & Ferretti, C. (2024, July). Exploring the prompt space of large language models through evolutionary sampling. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 1345-1353). [Crossref]

[19] Liu, S., Fang, Y., Cheng, H., Pan, Y., Liu, Y., & Gao, C. (2023, November). Large Language Models guided Generative Prompt for Dialogue Generation. In *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 10-17). IEEE. [Crossref]

[20] Tao, Y., Wang, Z., Zhang, H., Wang, L., & Gu, J. (2025, July). Nevlp: Noise-robust framework for efficient vision-language pre-training. In *International Conference on Intelligent Computing* (pp. 74-85). Singapore: Springer Nature Singapore. [Crossref]

[21] Nadizar, G., Rovito, L., De Lorenzo, A., Medvet, E., & Virgolin, M. (2024). An analysis of the ingredients for learning interpretable symbolic regression models with human-in-the-loop and genetic programming. *ACM Transactions on Evolutionary Learning and Optimization, 4*(1), 1-30. [Crossref]

[22] Jain, A. M., & Jindal, M. (2025, March). Systematic survey of various prompt optimization methods and their classifications. In *2025 11th International Conference on Computing and Artificial Intelligence (ICCAI)* (pp. 524-536). IEEE. [Crossref]

[23] Huang, T., Xu, Z., Yu, P., Yi, J., & Xu, X. (2025). A hybrid transformer model for fake news detection: Leveraging Bayesian optimization and bidirectional recurrent unit. *arXiv preprint arXiv:2502.09097*.

[24] Shorinwa, O., Mei, Z., Lidard, J., Ren, A. Z., & Majumdar, A. (2025). A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*. [Crossref]

[25] He, Y., Wang, J., Wang, Y., Li, K., Zhong, Y., Song, X., ... & Chen, J. (2025). Enhancing intent understanding for ambiguous prompt: A human-machine co-adaption strategy. *arXiv preprint arXiv:2501.15167*.

[26] Milani, S., Topin, N., Veloso, M., & Fang, F. (2024). Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys, 56*(7), 1-36. [Crossref]

[27] Huang, T., Cui, Z., Du, C., & Chiang, C. E. (2025). CL-ISR: A Contrastive Learning and Implicit Stance Reasoning Framework for Misleading Text Detection on Social Media. *arXiv preprint arXiv:2506.05107*.

[28] Ciniselli, M., Cooper, N., Pascarella, L., Mastropaolo, A., Aghajani, E., Poshyvanyk, D., ... & Bavota, G. (2021). An empirical study on the usage of transformer models for code completion. *IEEE Transactions on Software Engineering, 48*(12), 4818-4837. [Crossref]

[29] Sheilsspeigh, P., Larkspur, M., Carver, S., & Longmore, S. (2024). Dynamic context shaping: A new approach to adaptive representation learning in large language models. [Crossref]

[30] Samuel, J., Khanna, T., Esguerra, J., Sundar, S., Pelaez, A., & Bhuyan, S. S. (2025). The Rise of Artificial Intelligence Phobia! Unveiling News-Driven Spread of AI Fear Sentiment using ML, NLP and LLMs. *IEEE Access*. [Crossref]

[31] Deng, Z., Ma, W., Han, Q. L., Zhou, W., Zhu, X., Wen, S., & Xiang, Y. (2025). Exploring DeepSeek: A Survey on Advances, Applications, Challenges and Future Directions. *IEEE/CAA Journal of Automatica Sinica, 12*(5), 872-893. [Crossref]