A Systematic Survey of Automatic Prompt Optimization Techniques

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, Haozhu Wang, Han Ding, Yuzhe Lu, Zhichao Xu, Yun Zhou, Balasubramaniam Srinivasan, Qiaojing Yan, Yueyan Chen, Haibo Ding, Panpan Xu, and Lin Lee Cheong
Amazon Web Services
{raxkiran,zhoukang,shguan,soumish,xuaqi,donshen, wshui,sangminw,sullamij, yawenwan, haozhuw, handing, yuzhelu, xzhichao, yunzzhou, srbalasu, qiaojiny, yyanc, hbding, xupanpan, lcheong}@amazon.com

Abstract

Since the advent of large language models (LLMs), prompt engineering has been a crucial step for eliciting desired responses for various Natural Language Processing (NLP) tasks. However, prompt engineering remains an impediment for end users due to rapid advances in models, tasks, and associated best practices. To mitigate this, Automatic Prompt Optimization (APO) techniques have recently emerged that use various automated techniques to help improve the performance of LLMs on various tasks. In this paper, we present a comprehensive survey summarizing the current progress and remaining challenges in this field. We provide a formal definition of APO, a 5-part unifying framework, and then proceed to rigorously categorize all relevant works based on their salient features therein. We hope to spur further research guided by our framework.

1 Introduction

Since McCann et al. (2018) cast multi-task NLP as Question Answering, using prompts as inputs has become the standard way to elicit desired responses from Large Language Models (LLMs). Furthermore, LLMs' few-shot learning (Brown et al., 2020), instruction-following (Ouyang et al., 2022), and zero-shot reasoning capabilities (Kojima et al., 2023) have led to a widespread proliferation of prompting tricks for various tasks and model variants. However, LLMs still exhibit unpredictable sensitivity to various factors (explanation of the task (Li et al., 2023b),ordering (Liu et al., 2024a), stylistic formatting (Sclar et al.), etc.) causing a performance gap between two prompts that are semantically similar, thereby adding impediments for adoption by end users. Against this backdrop, Black-Box Automatic Prompt Optimization (APO) techniques have emerged that improve task performance via automated prompt improvements. The possess various attractive features - (1) they do not require parameter access on LLMs performing the task, (2) they systematically search through the prompt solution space, and (3) they retain human interpretability of prompt improvements. In this survey paper, we aim to highlight the advances in the field. Our core contribution is a 5-part APO taxonomy combined with a comprehensive fine-grained categorization of various design choices therein (see Fig. 1, Tables 2, 3, 4 in Appendix). We hope our framework will be informational for new and seasoned researchers alike, enabling further research on open questions.

2 Automatic Prompt Optimization Formulation

We formalize the process of automatic prompt optimization (APO) as follows. Given a task model M_{task}, initial prompt $\rho \in V$, the goal of an APO-system M_{APO} is to obtain the best performing prompt-template ρ^{opt} under a metric $f \in F$ and eval-set D_{val}

$$\rho^{opt} := \arg\max_{\rho \in V} E_{x \sim D_{val}}[f(M_{task}(\rho \oplus x))] \quad (1)$$

This objective function is not tractable for discrete prompt optimization as token-sequence search spaces are combinatorial. Instead, APO techniques follow the general anatomy as described in Algorithm 1 to obtain approximate solutions.

3 Initialize Seed Prompts

3.1 Manual Instructions

Several approaches use a seed of manually created instructions that offer interpretable and strong baselines as the basis for further improvement,*inter alia.*, ProteGi (Pryzant et al., 2023), GPS (Xu et al., 2022), SPRIG (Zhang et al., 2024b). While obtaining quality examples can be costly, APE (Zhou et al., 2022) [1] showed that a few hundred samples are sufficient for further optimization.

[1] Note: APE stands for Automatic Prompt Engineer method introduced by (Zhou et al., 2022), not to be confused with APO

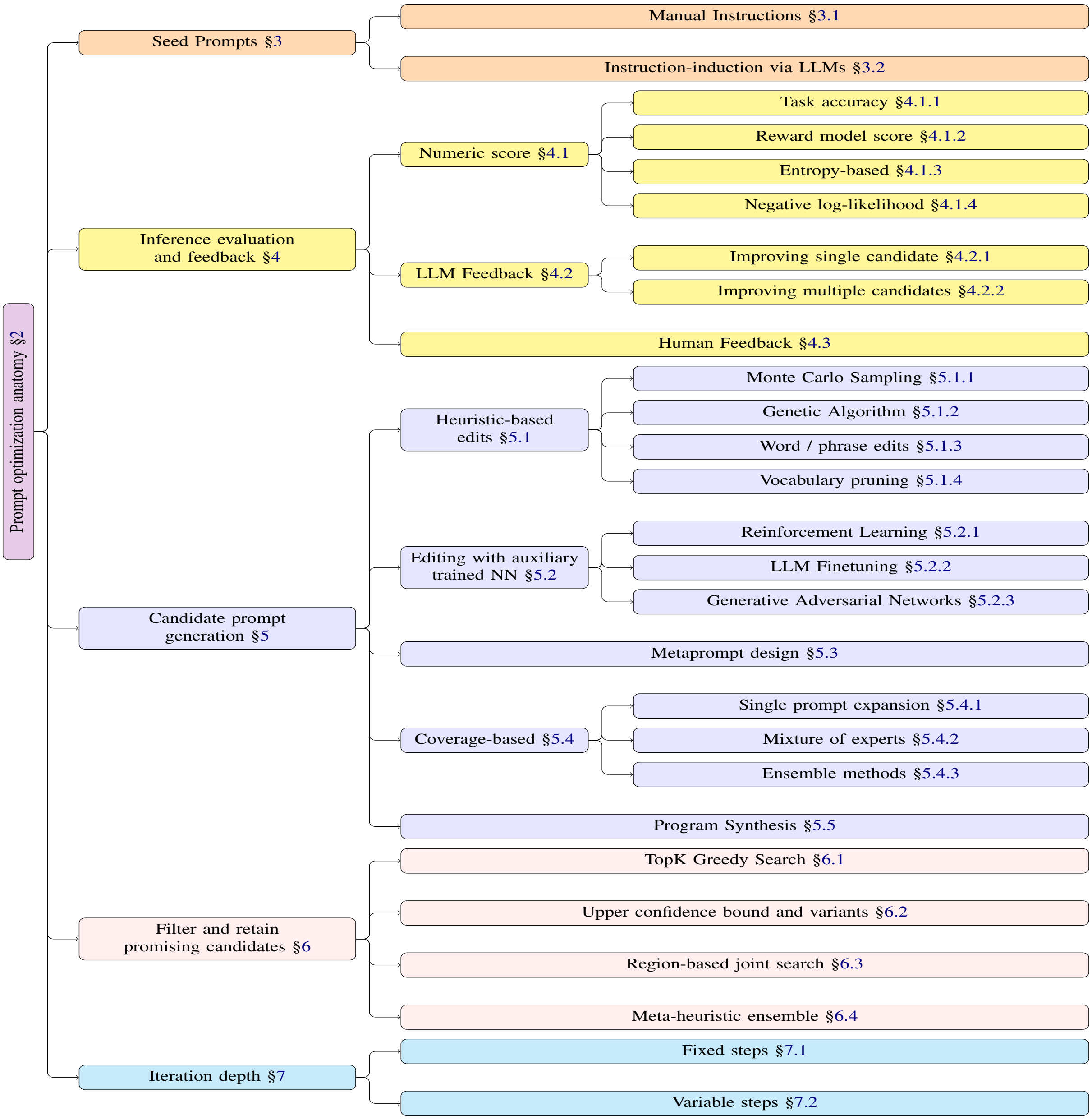
Prompt optimization anatomy §2
Seed Prompts §3
Manual Instructions §3.1
Instruction-induction via LLMs §3.2
Inference evaluation and feedback §4
Numeric score §4.1
Task accuracy §4.1.1
Reward model score §4.1.2
Entropy-based §4.1.3
Negative log-likelihood §4.1.4
LLM Feedback §4.2
Improving single candidate §4.2.1
Improving multiple candidates §4.2.2
Human Feedback §4.3
Candidate prompt generation §5
Heuristic-based edits §5.1
Monte Carlo Sampling §5.1.1
Genetic Algorithm §5.1.2
Word / phrase edits §5.1.3
Vocabulary pruning §5.1.4
Editing with auxiliary trained NN §5.2
Reinforcement Learning §5.2.1
LLM Finetuning §5.2.2
Generative Adversarial Networks §5.2.3
Metaprompt design §5.3
Coverage-based §5.4
Single prompt expansion §5.4.1
Mixture of experts §5.4.2
Ensemble methods §5.4.3
Program Synthesis §5.5
Filter and retain promising candidates §6
TopK Greedy Search §6.1
Upper confidence bound and variants §6.2
Region-based joint search §6.3
Meta-heuristic ensemble §6.4
Iteration depth §7
Fixed steps §7.1
Variable steps §7.2

Figure 1: Taxonomy of Automatic Prompt Optimization

Algorithm 1 Prompt optimization framework

```
$P_0 := \{\rho_1, \rho_2, \dots, \rho_k\}$    ▷ §3. Seed prompts
$D_{val} := \{(x_1, y_1)\}_{i=1}^{n}$    ▷ Validation set
$f_1, \dots, f_m \in F$    ▷ §4. Inference evaluation
for $t = 1, 2, \dots, N$ do    ▷ §7. Iteration depth
    ▷ §5. Generate prompt candidates
    $G_t := M_{APO}(P, D_{val}, F)$
    ▷ §6. Filter and retain candidates
    $P_t := Select(G_t, D_{val}, F)$
    ▷ §7. Optionally check for early convergence
    if $f_{convergence} \leq \epsilon$ then
        exit
return $\arg\max_{\rho \in P_N} E_{x \sim D_{val}} [f(M_{task}(\rho \oplus x))]$
```

3.2 Instruction Induction via LLMs

Honovich et al. (2023) were the first to propose inducing LLMs to infer human-readable prompts based on a few demonstrations E (see Appendix 14.1 for prompt). APE (Zhou et al., 2022) and DAPO (Yang et al., 2024c) use the induced seed instructions for further optimization, while MOP (Wang et al., 2025) and GPO (Li et al., 2023c) use APE to induce cluster-specific prompts. Apart from demonstrations, SCULPT (Kumar et al., 2024) induced instructions from task-READMEs, while UniPrompt (Juneja et al., 2024) used LLMs to fill-

which broadly refers to the entire area of Automatic Prompt Optimization in structured templates.

4 Inference Evaluation and Feedback

The evaluation step helps identify promising prompt candidates in each iteration. Some methods also use LLM feedback on prompt-response pairs to help generate more prompt candidates.

4.1 Numeric Score Feedback

4.1.1 Accuracy

Using task-specific accuracy metrics is the most straightforward and widespread way of eliciting feedback, i.a., (Zhou et al., 2022, 2023; Zhang et al., 2024b; Khattab et al., 2022). Classification and MCQ-based QA tasks use exact accuracy, while code-related tasks measure execution accuracy. Text generation tasks (summarization, translation, creative writing) employ flexible metrics like BLEU-N, Rouge-N, Rouge-N-F1, or embedding-based measures such as BERTScore (Zhang* et al., 2020) (Honovich et al., 2023; Dong et al., 2024b).

4.1.2 Reward-model Scores

Given the limitations of rigid accuracy metrics, some approaches proposed using learned reward models to provide more nuanced evaluations of prompts-response pairs (Deng et al., 2022; Sun et al., 2024a; Kong et al., 2024). OIRL (Sun et al., 2024a) trained an XGBoost-based reward model that takes query-prompt embedding pairs as input and predicts whether the prompt will elicit correct answers from the language model and use it to select appropriate prompts for specific queries using a best-of-N strategy. DRPO (Amini et al., 2024) follows an LLM-based reward modeling approach using both predefined and dynamic reward criteria. It first optimizes in-context learning examples E, and using that it optimizes the specific task prompt.

4.1.3 Entropy-based Scores

Entropy-based scores evaluate the entire output distribution induced by candidates, as opposed to a single inference instance. They are gradient-free but require access to the entire output probability distribution, something not usually possible with black-box LLMs. CLAPS (Zhou et al., 2023) leverages the negative incremental cross-entropy of $\pi_{(x_i \oplus v \in V)}$ v/s $\pi_{(x_i)}$ to identify promising words $v \in V$ to add to the prompt. The topK words are then used as candidate tokens from which to construct candidate prompts. GRIPS (Prasad et al., 2023) simply added an entropy term to the task-weighted accuracy $-\sum \pi_\rho(y)\, ln(\pi_\rho(y)) + \frac{1}{|T|}\sum \mathbf{1}(y = \hat{y})$ to prioritize output diversity in potential prompt candidates.

4.1.4 Negative Log-likelihood of Output

Some approaches like APE, GPS (Xu et al., 2022), PACE (Dong et al., 2024b) consider the negative log-likelihood (NLL) of token sequences under the target LLM, i.e., $-\log(\pi_\rho(y))$. This however requires the log-probabilities to be accessible during the decoding of each token, limiting its applicability. The NLL for ground truth one-hot token-sequence is equivalent to the cross-entropy.

4.2 LLM Feedback

A popular paradigm to augment or fully replace numeric scores is to use textual feedback generated by $LLM_{Evaluator}$ (Wang et al., 2024a; Long et al., 2024; Sinha et al., 2024). It is versatile because it can evaluate both the response as well as the prompt input. It can directly aid the prompt rewriting process while being flexible to individual tasks as it only needs natural language instructions for general-purpose LLMs as opposed to task-specific handcrafting of metrics. A potential downside is the inference cost incurred due to an additional LLM call. All the LLM feedback approaches provide multiple feedback data and broadly fall into two categories - improving a single prompt candidate versus improving multiple prompt candidates (discussed below, examples in Appendix 14.3).

4.2.1 Improving Single Candidate

SCULPT (Kumar et al., 2024) introduces a systematic method for tuning long, unstructured prompts by employing a **hierarchical tree structure** and two-step feedback loops - preliminary assessment and error assessment - to evaluate and correct prompts before and after execution. The feedback updates the hierarchical prompt tree which is then back-synthesized into a new prompt candidate. PACE (Dong et al., 2024b) applies an **actor-critic** editing framework to the prompt refinement process itself, allowing for more dynamic and adaptive adjustments. Overcoming the limitations of optimizing a single metric, CRISPO (He et al., 2025) adopts a **multi-aspect critique-suggestion** meta-prompt to highlight flaws in the generated response across multiple dimensions such as style, precision, and content alignment. Thereafter it leverages detailed, aspect-specific feedback and iteratively updates the prompts. Autohint (Sun et al., 2023)

Paper	Seed instructions	Iteration depth	Inference evaluation	Candidate generation	Search+filter strategy
ProTeGi (Pryzant et al., 2023)	Manually created	Fixed	LLM feedback + Task accuracy	LLM rewriter	UCB for trees
APE (Zhou et al., 2022)	Instruction induction	Fixed	Task accuracy	N/A	UCB
CRISPO (He et al., 2025)	Manually created	Fixed	LLM feedback + Task accuracy	LLM rewriter	TopK selection
MOP (Wang et al., 2025)	Instruction induction	Fixed	Task accuracy	Mixture of experts	Region-based joint search
DSPY (Khattab et al., 2024)	Manually created + Instruction induction	Variable	LLM feedback + Task accuracy	Program Synthesis	TopK selection
OPRO (Yang et al., 2024a)	Manually created	Variable	LLM feedback + Task accuracy	Metaprompt design	TopK selection
GATE (Joko et al., 2024)	Manually created	Variable	Human feedback	LLM rewriter	N/A

Table 1: Comparison of some APO techniques under our framework (Tables 2,3,4 show full comparison)

summarizes feedback for multiple incorrect inferences via **hints** to instill improvements into a single prompt candidate.

4.2.2 Improving Multiple Candidates

ProTeGi (Pryzant et al., 2023) and TextGrad (Yuksekgonul et al., 2024) leverage **textual “gradients”** to guide the discrete prompt optimization procedure, very similar to the gradient-descent style of continuous prompt optimization approaches. Different from continuous gradient-descent, ProTeGi sampled multiple “gradients” i.e. directions of improvement, and each such “gradient” is used to generate several prompt candidates for evaluation in the next iteration. PromptAgent (Wang et al., 2024a) similarly used an error collection approach to emulate expert-written prompts that consisted of clear sections like “Task description”, “Domain Knowledge”, “Solution Guidance”, “Exception Handling”, “Output Formatting”. PREFER (Zhang et al., 2024a) utilizes a feedback-reflect-refine cycle to aggregate feedback into multiple prompts in an **ensemble** to improve the model’s ability to generalize across various tasks. Survival of the Safest (SOS) (Sinha et al., 2024) added **safety-score** into a multi-objective prompt optimization framework that used an interleaved strategy to balance performance and security in LLMs simultaneously. To avoid accidentally damaging well-functioning prompts, StraGo (Wu et al., 2024) summarized strategic guidance based on both correct and incorrect predictions as feedback.

4.3 Human-feedback

A few works also incorporate human feedback, either during compile-time or inference-time in the prompt construction / optimization process. Joko et al. (2024) proposed “Generative Active Task Elicitation” to better capture human preferences. It prompts a language model to interactively ask questions and infer human preferences conditioned on the history of free-form interaction. Cheng et al. (2024) trained a smaller LLM to optimize input prompts based on user preference feedback, achieving up to 22% increase in win rates for ChatGPT and 10% for GPT-4. PROMST (Chen et al., 2024) tackles the challenges of multi-step tasks by incorporating human-designed feedback rules and a learned heuristic model. APOHF (Lin et al., 2024) focuses on optimizing prompts using only human preference feedback rather than numeric scores, employing a dueling bandits-inspired strategy to efficiently select prompt pairs for preference feedback, proving effective for tasks like text-to-image generation and response optimization.

5 Candidate Prompt Generation

In this step, one or more candidate prompts are generated that are most likely to result in an improvement in a metric of interest $f \in F$. The approaches reviewed below range from simple rule-based edits (sec. 5.1) to sophisticated agentic systems that combine with LLM-based evaluations (sec. 4.2) and various filtering strategies (sec. 6).

5.1 Heuristic-based Edits

Several works proposed heuristic-based mechanisms to make edits to intermediate prompt candidates to generate newer candidates. They range from edits at the word / phrase / sentence-level (either simple rule-based or LLM-generated), or metric-driven incremental search. While these strategies may not result in the most optimal solution, they help in making the discrete prompt optimization problem computationally tractable.

5.1.1 Monte Carlo Sampling

ProTeGi (Pryzant et al., 2023) uses Monte carlo sampling to explore combinatorial discrete solution spaces in an incremental fashion - it samples multiple textual gradients to use to generate prospective candidates, and spawns paraphrases as monte-carlo successors for evaluation. PromptAgent (Wang et al., 2024a) uses a tree-variant called Monte Carlo Tree Search (MCTS) which consists of 4 steps — Selection, Expansion, Simulation, and Backpropagation (also explained in Sec. 6).

5.1.2 Genetic Algorithm

A significant line of work applies the well-studied genetic algorithms to make discrete edits to texts. The common recipe for several genetic algorithms is 1/ Mutate and 2/ Cross-over components from promising candidates. **Token mutations:** SPRIG (Zhang et al., 2024b) and CLAPS perform token-level mutations. SPRIG uses a starting corpus of 300 components grouped into categories like COT, roles, styles, emotions, scenarios, and good properties. It performs add/rephrase/swap/delete, highlighting complementary strengths of optimizing system prompts alongside task-prompts (via methods like ProTeGi) to enhance accuracy across multiple diverse domains, languages, and tasks without needing repeated task-specific optimizations.

LLM-based mutation: LMEA (Liu et al., 2023), SOS (Sinha et al., 2024), and StraGo (Wu et al., 2024) uses mutation prompts with LLMs to overcome the traditional complexity of designing tailored operators for cross-over / mutation. PromptBreeder (Fernando et al., 2023) advocates self-referential improvement of all prompts in the prompt optimization system - Direct Mutation of task prompts, Hypermutation of mutation prompts themselves, Lamarckian Mutation where prompts are reverse-engineered from successful examples (similar to Instruction Induction Honovich et al. (2023), and finally Crossover and Shuffling to improve diversity of the prompt pool. EvoPrompt (Guo et al., 2024) use Differential Evolution - where differences between existing prompts is incorporated to form new prompt candidates to overcome the problem of local optima. AELP (Hsieh et al., 2024) also uses mutation operators to perform sentence-level edits in an iterative fashion. They include sentence-level histories of reward $\{(s_{t-1}, s_t, r_t)\}$ in the mutation prompt in order to avoid local optima and accidentally returning to sub-optimal versions. GPS (Xu et al., 2022) used Back-translation, Sentence Continuation, and Cloze transformations to perform prompt mutation. PromptWizard (Agarwal et al., 2024) proposed a pipeline combining several steps including iterative improvement, few shot example synthesis and selection, utilizing LLM's reasoning capability to improve and validate the prompt, and finally an expert persona to ensure consistency of the style of generated prompts.

5.1.3 Word / Phrase Level Edits

Several word-edit approaches first identify "influential" tokens in the prompts. COPLE (Zhan et al., 2024) argued that LLMs exhibit lexical sensitivity, showing that merely replacing a few words with their synonyms can yield significant improvements. First, "influential" tokens are identified where expected loss on dev-set $E_{D_{val}}[L(y, \hat{y})]$ drops the most after removing that token versus the original prompt, and then influential tokens are replaced using predictions from a Masked-Language Models. This token-replacement approach is also attractive as a standalone post-processing step for long prompts that are already optimized using other LLM-based approaches. GRIPS (Prasad et al., 2023) argues that phrase level edition is an effective and interpretable method to optimize prompts, leveraging 4 basic edit operations -add, delete, paraphrase, and swap

5.1.4 Vocabulary Pruning

Some works prune the vocabulary space V to V_{pruned} for decoding the next token for the optimized prompt ρ^*. CLAPS (Zhou et al., 2023) argued that general search spaces are highly redundant and use K-means clustering to find word-clusters and retain top-2000 words closest to cluster centroids. BDPL (Diao et al., 2022) used pairwise mutual information (PMI) to retain top co-occuring ngrams for decoding. PIN (Choi et al., 2024) instead added regularization in the form of Tsallis-entropy (ideal for heavy-tailed distributions like natural language) for the RL training of a prompt generation network, to reduce the probability mass for unlikely tokens and improve interpetability.

5.2 Editing via Auxiliary Trained NN

Some approaches leverage a trained auxiliary neural network to edit the initial prompt for obtaining desired improvements. We include approaches where the finetuned network is different and smaller than the task network.

5.2.1 Reinforcement-learning

Multi-objective Optimization techniques (Jafari et al., 2024) demonstrate superiority over simple reward averaging, particularly through volume-based methods that effectively balance competing objectives. Dynamic prompt modification strategies, introduced through **prompt rewriting** (Kong et al., 2024), directional stimulus prompting (Li et al., 2023d) and **test-time editing** (Zhang et al., 2022) solve the important goal of moving beyond static prompt generation. Prompt-OIRL (Sun et al., 2024a) also tackled test-time optimization objective by learning an **offline reward model** and subsequently using a best-of-N strategy to recommend the optimal prompt in a query-dependent fashion. BDPL (Diao et al., 2022) optimized discrete prompts using variance-reduced policy gradient algorithm to estimate gradients, allowing user devices to fine-tune tasks with limited API calls.

5.2.2 Finetuning LLMs

BPO (Cheng et al., 2024) trains a smaller 7B model to align itself to task-performance on individual LLMs using reward-free alignment. FIPO (Lu et al., 2025) trains a local model (7B - 13B) to perform prompt optimizations to preserve privacy and adapt to target models better leveraging both data diversification and strategic fine-tuning such as SFT, preference optimization, and iterative preference learning.

5.2.3 Generative Adversarial Networks

Long et al. (2024) framed the prompt optimization process in the GAN setting. The LLM generator takes question and the generation prompt to produce output. The (input, output) pairs are evaluated by an LLM powered discriminator, whose goal is to identify generated pairs from ground truth pairs. Both generator and the discriminator are jointly optimized using adversarial loss, by utilizing a prompt modifier LLM to rewrite their prompts.

5.3 Metaprompt Design

PE2 (Ye et al., 2024) argued that previous works under-explored meta-prompt search space. OPRO (Yang et al., 2024a) proposes a meta-prompt design (see Appendix 14.2) which includes the optimization problem description in natural language and previously generated solutions (multiple solutions per stage for diversity) and scores alongisde the meta-instruction for prompt refinement. DAPO (Yang et al., 2024c) utilizes a well-designed meta-instruction to guide the LLM in generating high-quality and structured initial prompts (contain task-specific info, e.g. task type and description, output format and constraints, reasoning process, professional tips) by observing given input-output exemplars. Then, DAPO iteratively optimizes the prompts at the sentence level, leveraging previous tuning experience to expand prompt candidates.

5.4 Coverage-based

Some approaches seek to "cover" the entire problem space - either within a single prompt, or using multiple prompts working individually or in an ensemble during inference.

5.4.1 Single Prompt-expansion

AMPO (Yang et al., 2024d) uses LLM feedback to enumerate all the failure cases based on the evaluation-set D_{val} and then enlists each of them in the meta-instruction in an if-then-else format using 3 modules - 1/ Pattern Recognition, 2/ Branch Adjustment, and 3/ Branch Pruning to decide whether to enhance existing branches, or to grow new branches. Similarly, UNIPROMPT focused on explicitly ensuring that various semantic facets of a task get represented in the final prompt. It designs a human-like (manual) prompt engineering approach (UniPrompt) with two stages: a) task facets initialization using background knowledge, and b) refinement using examples.

5.4.2 Mixture of Experts

Wang et al. (2025) introduced the Mixture-of-Expert-Prompts where each expert is a task-prompt to be used for specialized inference. MOP first clusters all demonstrations using K-means clustering. Then, the Region-based Joint Search (RBJS) (sec.6.3) algorithm generates the appropriate instruction for each exemplar-cluster via instruction induction (sec.3.2) based on a mix of in-cluster and out-of-cluster demonstrations to cover “blind-spots”. During inference, a single expert prompt is invoked whose cluster centroid μ_c is closest to the instance-embedding $\arg\min_C ||\phi(x_i) - \mu_c||_2$.

5.4.3 Ensemble Methods

PromptBoosting (Hou et al., 2023), Boosted-Prompting (Pitis et al., 2023), PREFER (Zhang et al., 2024a), etc. are ensemble methods that invoke multiple prompts during inference and combine them to generate the final output $\hat{y} = y_0 + \Sigma_m \beta_i y_i$. GPO (Li et al., 2023c) also uses labeled source data to generate an ensemble of prompts, which are applied to unlabeled target data to generate output through majority voting.

5.5 Program Synthesis

Program-synthesis based approaches transform LLM pipelines into structured, modular components that can be systematically optimized and composed. These optimization techniques iteratively refine instructions and demonstrations for each module to improve the entire pipeline's performance, DSP (Khattab et al., 2022) introduces a three-stage framework for retrieval-augmented inference: Demonstrate (generates task-specific demonstrations), Search (retrieves relevant information), and Predict (combines retrieved info with demonstrations). DSPY (Khattab et al., 2024) transforms LLM pipelines into text transformation graphs - introducing parameterized models, learning through demonstrations, and a compiler that optimizes pipelines. DLN (Sordoni et al., 2023) similarly considers chained LLM calls as stacked deep language networks performing variational inference, where the learnable parameters for each layer are task-decomposed prompt templates. MIPRO (Opsahl-Ong et al., 2024) automates the optimization of multi-stage language model programs by improving instructions and demonstrations for each module. SAMMO (Schnabel and Neville, 2024) proposed symbolic prompt programming, representing prompts as directed-acyclic-graphs (DAG). A set of user-defined node mutation rules guide the mutation-search to find the optimal DAG, which is then converted back to a prompt.

6 Filter and Retain Promising Prompts

In this step, promising prompt candidates are filtered for further optimization.

6.1 TopK Greedy Search

The simplest mechanism to iteratively search through prompt candidate sets is a greedy topK search where in each iteration of the optimization, the top-K best-performing candidates on mini-batch of data instances D_{val} are retained for further iterations (e.g. - ProTeGi, AELP. This differs from beam-search which judges partial solutions' based on the reward for the entire trajectory of prompt edits $r(\{\rho_1^1, \rho_2^1, \ldots, \rho_t^1\})$.

6.2 Upper Confidence Bound and Variants

Relying on a single static evaluation dataset can lead to biases in the selection procedure and finally suboptimal solutions. ProTeGi, SPRIG, *inter alia*, cast the candidate prompt selection problem as that of bandit search - identifying the most suitable arm (prompt candidate) operating on a fixed computation budget. They use the Upper Confidence Bounds (UCB, Algorithm 2) which balances exploration with exploitation. In each iteration of prompt optimization, they sample a different evaluation dataset $D_{sample} \in D_{val}$, and maintain a moving estimate of the optimality of each arm (i.e. prompt). In each iteration, the playout filters top-B prompt candidates with the greatest score for further exploration. PromptAgent uses a variation of UCB called UCB for Trees (UCT) which are used in the setting of contextual bandits (i.e. the action-space and the reward function is state-dependent). AELP (Hsieh et al., 2024) used a modification called Linear UCB (Li et al., 2010) which uses a closed form linear estimate based on the reward trajectories of previously sampled edits as well as prompt embedding $\phi(s)$ to select the next best-arm.

6.3 Region-based Joint Search

MOP (Wang et al., 2025) proposes a Mixture-of-Expert-Prompts performing prompt optimization for each expert individual. Once C exemplar-clusters are identified, the RBJS search first samples examples $D_{exemplars} \in D_C \cup D \setminus D_C$, and then uses APE to induct and optimize each expert instruction.

6.4 Metaheuristic Ensemble

PLUM (Pan et al., 2024) library offered a metaheuristic ensemble of different search algorithms like Hill climbing, Simulated Annealing, Genetic Algorithms, Tabu Search, and Harmony Search.

7 Iteration Depth

7.1 Fixed Steps

Most approaches choose to carry out the prompt optimization for a fixed number of steps N.

7.2 Variable number of steps

GRIPS (Prasad et al., 2023) concludes search when successive iterations with negative gains breach a patience parameter, whereas PromptAgent concluded APO when $r_t \leq \epsilon_{min} \vee r_t \geq \epsilon_{max}$.

8 Theoretical Perspectives

8.1 Upper Bound of Improvement from APO

AlignPro (Trivedi et al., 2025) establishes an upper bound on the gains realizable from discrete prompt optimization under a given prompt optimizer and also a suboptimality-gap w.r.t. RLHF-optimal policy π^*, while a lower bound is left unexplored.

8.2 Other Related Perspectives

Bhargava et al. (2024) proposed a control theoretic framework to establish bounds on the set of reachable LLM-outputs for self-attention in terms of the singular values of its weight matrices. Liu et al. (2024c) showed the existence of a strong transformer that can approximate any sequence-to-sequence Lipschitz function. They also showed the existence of "difficult" datasets that depth-limited transformers could not commit to memory.

9 Challenges and Future Directions

9.1 Task-agnostic APO

All the surveyed APO methods assume that the task type T is known beforehand; additionally offline APO methods also require an evaluation set D_{val}, something not explicitly available in production settings. Barring a few tasks covered by Joko et al. (2024); Sun et al. (2024a); Zhang et al. (2022); Choi et al. (2024), inference-time optimization of multiple unknown tasks is underexplored. More robust evaluations are needed for task-agnostic APO systems combining seen and unseen tasks.

9.2 Unclear Mechanisms

Melamed et al. (2024) showed that prompts have so-called 'evil twins' that are uninterpretable yet recover some of the performance of gold-standard prompts. Lu et al. (2024) showed that rare gibberish strings can serve as competitive delimiters τ in prompts. Yang et al. (2024b) showed that self-reflection by LLMs can suffer from incorrect error identification, prior biases, semantic invalidity, leading to failure in yielding improved prompts. More studies are needed to better uncover the mechanisms of prompt optimization.

9.3 APO for System Prompts / Agents

Although SPRIG explored optimizing system prompts in chat-style settings, scalability remains a challenge - optimizing system prompts required a predefined corpus and close to 60 hours whereas Protegi only needed $\tilde{1}0$ minutes per task. Similarly, optimizing prompts for several components in an agentic system in a concurrent fashion poses an exciting direction for future research.

9.4 Multimodal APO

Recently, textual prompt optimization has expanded to multimodal domains: text-to-image (Liu et al., 2024b; Mañas et al., 2024; Liu et al., 2024d), text-to-video (Ji et al., 2024), text-to-audio (Huang et al., 2023), and text-image alignment models like CLIP (Du et al., 2024; Mirza et al., 2024). Beyond textual prompts, Huang et al. (2023) explore optimizing multimodal inputs, such as images, to elicit better responses from large multimodal models. However, the interplay between modalities in prompt optimization remains underexplored. Future research could develop APO frameworks to jointly optimize multimodal prompts (eg - remove background noise from audio, add visual markers to videos, etc.) to fully leverage their synergies.

10 Conclusion

In this paper, we provide a comprehensive fine-grained review of existing APO techniques and identified key areas for future growth. It is our aim to spur future research spawning from our survey.

11 Limitations

While we attempted to cover all qualifying papers, it is possible that we may have unintentionally missed out on some relevant papers. We also mention some of the papers that were excluded in this survey with specific reasons in section 12.2. Also, we realize that fitting varied research works into a single unifying framework might risk broad categorizations for some papers, or skipping some characteristics for others (e.g. Tempera (Zhang et al., 2022) consists of both RL-based and word/phrase-level editing techniques, applied to both instructions and exemplars). In such cases, we categorize a paper based on its most salient features. Another challenge is that when presenting a survey paper under 8 pages, we had to make tradeoffs and only retain content in the main body that was deemed most necessary. This resulted in having to relegate a core contribution (Tables 2,3,4) which contained a rigorous comparison of all the surveyed papers into the appendix. We have attempted our best to strike the right balance between specificity and brevity to present a novel framework. We also provide copious references to interested researchers for further reading.

References

Eshaan Agarwal, Joykirat Singh, Vivek Dani, Raghav Magazine, Tanuja Ganu, and Akshay Nambi. 2024. Promptwizard: Task-aware prompt optimization framework.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9954–9972, Bangkok, Thailand. Association for Computational Linguistics.

R. Anantha, Svitlana Vakulenko, Zhucheng Tu, S. Longpre, Stephen G. Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. In *North American Chapter of the Association for Computational Linguistics*.

Jacob Andreas, Johannes Bufe, David Burkett, Charles C. Chen, Joshua Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Leo Wright Hall, Kristin Delia Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, C. H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Ann Short, Div Slomin, B Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, A. A. Vorobev, Izabela Witoszko, Jason Wolfe, A. G. Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. In *International Conference on Computational Linguistics*.

Aman Bhargava, Cameron Witkowski, Shi-Zhuo Looi, and Matt Thomson. 2024. What's the magic word? a control theory of llm prompting.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Daniel Matthew Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *International Workshop on Semantic Evaluation*.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*.

Yongchao Chen, Jacob Arkin, Yilun Hao, Yang Zhang, Nicholas Roy, and Chuchu Fan. 2024. PRompt optimization in multi-step tasks (PROMST): Integrating human feedback and heuristic-based sampling. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3859–3920, Miami, Florida, USA. Association for Computational Linguistics.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3201–3219, Bangkok, Thailand. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Minje Choi, Jiaxin Pei, Sagar Kumar, Chang Shu, and David Jurgens. 2023. Do llms understand social knowledge? evaluating the sociability of large language models with socket benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11370–11403.

Yunseon Choi, Sangmin Bae, Seonghyun Ban, Minchan Jeong, Chuheng Zhang, Lei Song, Li Zhao, Jiang Bian, and Kee-Eung Kim. 2024. Hard prompts made interpretable: Sparse entropy regularization for prompt tuning with rl.

Christopher Cieri, Mark Liberman, Sunghye Cho, Stephanie Strassel, James Fiumara, and Jonathan Wright. 2022. Reflections on 30 years of language resource development and sharing. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 543–550, Marseille, France. European Language Resources Association.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. *ArXiv*, abs/2004.04494.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *International Joint Conference on Natural Language Processing*.

Robert C. Detrano, András Jánosi, Walter Steinbrunn, Matthias Emil Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern Guppy, Stella Lee, and Victor Froelicher. 1989. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64 5:304–10.

Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. 2022. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *International Joint Conference on Natural Language Processing*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024a. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Yihong Dong, Kangcheng Luo, Xue Jiang, Zhi Jin, and Ge Li. 2024b. PACE: Improving prompt with actor-critic editing for large language model. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7304–7323, Bangkok, Thailand. Association for Computational Linguistics.

Yingjun Du, Wenfang Sun, and Cees GM Snoek. 2024. Ipo: Interpretable prompt optimization for vision-language models. *arXiv preprint arXiv:2410.15397*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *North American Chapter of the Association for Computational Linguistics*.

Stefan Daniel Dumitrescu, Petru Rebeja, Beáta Lőrincz, Mihaela Găman, Mihai Daniel Ilie, Andrei Pruteanu, Adriana Stan, Luciana Morogan, Traian Rebedea, and Sebastian Ruder. 2021. Liro: Benchmark and leaderboard for romanian language tasks. In *NeurIPS Datasets and Benchmarks*.

Ibrahim Abu Farha and Walid Magdy. 2020a. From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset. In *OSACT*.

Ibrahim Abu Farha and Walid Magdy. 2020b. From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset. In *OSACT*.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *ArXiv*, abs/2309.16797.

Rory A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7:179–188.

Noa Garcia, Chentao Ye, Zihua Liu, Qingtao Hu, Mayu Otani, Chenhui Chu, Yuta Nakashima, and Teruko Mitamura. 2020. A dataset and baselines for visual question answering on art. In *European Conference on Computer Vision*, pages 92–108.

Miguel Garc'ia-Orteg'on, Gregor N. C. Simm, Austin Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. 2021. Dockstring: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62:3486 – 3502.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Annual Meeting of the Association for Computational Linguistics*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Chulaka Gunasekara, Jonathan K. Kummerfeld, Lazaros Polymenakos, and Walter S. Lasecki. 2019. Dstc7 task 1: Noetic end-to-end response selection. *Proceedings of the First Workshop on NLP for Conversational AI*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Han He, Qianchu Liu, Lei Xu, Chaitanya Shivade, Yi Zhang, Sundararajan Srinivasan, and Katrin Kirchhoff. 2025. Crispo: Multi-aspect critique-suggestion-guided automatic prompt optimization for text generation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2022. Instruction induction: From few examples to natural language task descriptions. *ArXiv*, abs/2205.10782.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1935–1952, Toronto, Canada. Association for Computational Linguistics.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Conference on Empirical Methods in Natural Language Processing*.

Bairu Hou, Joe O'Connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Promptboosting: black-box text classification with ten forward passes. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Cho-Jui Hsieh, Si Si, Felix Yu, and Inderjit Dhillon. 2024. Automatic engineering of long prompts. In *Findings of the Association for Computational Linguistics: ACL 2024*, page 10672—10685, Bangkok, Thailand. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. 2023. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR.

Yasaman Jafari, Dheeraj Mekala, Rose Yu, and Taylor Berg-Kirkpatrick. 2024. Morl-prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization.

Yatai Ji, Jiacheng Zhang, Jie Wu, Shilong Zhang, Shoufa Chen, Chongjian GE, Peize Sun, Weifeng Chen, Wenqi Shao, Xuefeng Xiao, et al. 2024. Prompt-a-video: Prompt your video diffusion model via preference-aligned llm. *arXiv preprint arXiv:2412.15156*.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Kumar Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. In *Findings*.

Can Jin, Hongwu Peng, Shiyu Zhao, Zhenting Wang, Wujiang Xu, Ligong Han, Jiahui Zhao, Kai Zhong, Sanguthevar Rajasekaran, and Dimitris N. Metaxas. 2024. Apeer: Automatic prompt engineering enhances large language model reranking.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *ArXiv*, abs/2009.13081.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P De Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing personal laps: Llm-augmented dialogue construction for personalized multi-session conversational search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 796–806.

Gurusha Juneja, Nagarajan Natarajan, Hua Li, Jian Jiao, and Amit Sharma. 2024. Task facet learning: A structured approach to prompt optimization. *arXiv preprint arXiv:2406.10504*.

David Jurgens, Srijan Kumar, Raine Hoover, Daniel A. McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, D. Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *International Workshop on Semantic Evaluation*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Weize Kong, Spurthi Amba Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Prewrite: Prompt rewriting with reinforcement learning.

Shanu Kumar, Akhila Yesantarao Venkata, Shubhanshu Khandelwal, Bishal Santra, Parag Agrawal, and Manish Gupta. 2024. Sculpt: Systematic tuning of long prompts.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *ArXiv*, abs/1906.00300.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and L. Morgenstern. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Bei Li, Rui Wang, Junliang Guo, Kaitao Song, Xu Tan, Hany Hassan, Arul Menezes, Tong Xiao, Jiang Bian, and JingBo Zhu. 2023a. Deliberate then generate: Enhanced prompting framework for text generation.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023b. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 661–670, New York, NY, USA. Association for Computing Machinery.

Moxin Li, Wenjie Wang, Fuli Feng, Yixin Cao, Jizhi Zhang, and Tat-Seng Chua. 2023c. Robust prompt optimization for large language models against distribution shifts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1539–1554, Singapore. Association for Computational Linguistics.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023d. Guiding large language models via directional stimulus prompting. *arXiv preprint arXiv:2302.11520*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Prompt optimization with human feedback.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Annual Meeting of the Association for Computational Linguistics*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew Soon Ong. 2023. Large language models as evolutionary optimizers. *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Shihong Liu, Samuel Yu, Zhiqiu Lin, Deepak Pathak, and Deva Ramanan. 2024b. Language models as black-box optimizers for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12687–12697.

Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. 2024c. Automatic and universal prompt injection attacks against large language models.

Yilun Liu, Minggui He, Feiyu Yao, Yuhe Ji, Shimin Tao, Jingzhou Du, Duan Li, Jian Gao, Li Zhang, Hao Yang, et al. 2024d. What do you want? user-centric prompt generation for text-to-image synthesis via multi-turn guidance. *arXiv preprint arXiv:2408.12910*.

Xuan Do Long, Yiran Zhao, Hannah Brown, Yuxi Xie, James Xu Zhao, Nancy F. Chen, Kenji Kawaguchi, Michael Shieh, and Junxian He. 2024. Prompt optimization via adversarial in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7308–7327, Bangkok, Thailand. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL Conference*.

Junru Lu, Siyu An, Min Zhang, Yulan He, Di Yin, and Xing Sun. 2025. FIPO: Free-form instruction-oriented prompt optimization with preference dataset and modular fine-tuning schema. In *Proceedings of the 31st International Conference on Computational Linguistics*, page 11029—11047, Abu Dhabi, UAE. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Annual Meeting of the Association for Computational Linguistics*.

Yao Lu, Jiayi Wang, Raphael Tang, Sebastian Riedel, and Pontus Stenetorp. 2024. Strings from the library of babel: Random sampling as a strong baseline for prompt optimisation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, page 2221—2231, Mexico City, Mexico. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *ArXiv*, abs/1808.09602.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. 2024. Improving text-to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Rimon Melamed, Lucas H. McCabe, Tanay Wakhare, Yejin Kim, H. Howie Huang, and Enric Boix-Adsera. 2024. Prompts have evil twins.

M Jehanzeb Mirza, Mengjie Zhao, Zhuoyuan Mao, Sivan Doveh, Wei Lin, Paul Gavrikov, Michael Dorkenwald, Shiqi Yang, Saurav Jha, Hiromi Wakaki, et al. 2024. Glov: Guided large language models as implicit optimizers for vision language models. *arXiv preprint arXiv:2410.06154*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Conference on Computational Natural Language Learning*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Ehsan Nezhadarya, Yang Liu, and Bingbing Liu. 2019. Boxnet: A deep learning method for 2d bounding box estimation from bird's-eye view point cloud. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1557–1564. IEEE.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *ArXiv*, abs/1910.14599.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *ArXiv*, abs/1706.09254.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, page 9340—9366, Miami, Florida, USA. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Rui Pan, Shuo Xing, Shizhe Diao, Wenhe Sun, Xiang Liu, KaShun Shum, Jipeng Zhang, Renjie Pi, and Tong Zhang. 2024. Plum: Prompt learning using metaheuristics. In *Findings of the Association for Computational Linguistics: ACL 2024*, page 2177—2197, Bangkok, Thailand. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *ArXiv*, cs.CL/0409058.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Annual Meeting of the Association for Computational Linguistics*.

Arkil Patel, S. Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *North American Chapter of the Association for Computational Linguistics*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Silviu Pitis, Michael R Zhang, Andrew Wang, and Jimmy Ba. 2023. Boosted prompt ensembles for large language models. *arXiv preprint arXiv:2304.05970*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. Xcopa: A multilingual dataset for causal commonsense reasoning. *arXiv preprint arXiv:2005.00333*.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. Grips: Gradient-free, edit-based instruction search for prompting large language models.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 7957—7968, Singapore. Association for Computational Linguistics.

Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? *ArXiv*, abs/1804.06323.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *ArXiv*, abs/2311.12022.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI spring symposium series*.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *ArXiv*, abs/1608.01413.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Tobias Schnabel and Jennifer Neville. 2024. Symbolic prompt program search: A structure-aware approach to efficient compile-time prompt optimization.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Jingyuan Selena She, Christopher Potts, Sam Bowman, and Atticus Geiger. 2023. Scone: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Annual Meeting of the Association for Computational Linguistics*.

Zeru Shi, Zhenting Wang, Yongye Su, Weidi Luo, Fan Yang, and Yongfeng Zhang. 2024. Robustness-aware automatic prompt optimization.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Ankita Sinha, Wendi Cui, Kamalika Das, and Jiaxin Zhang. 2024. Survival of the safest: Towards secure prompt optimization through interleaved multi-objective evolution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1016–1027, Miami, Florida, US. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Gizem Sogancioglu, Hakime Öztürk, and Arzucan Özgür. 2017. Biosses: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33:i49 – i58.

Alessandro Sordoni, Eric Yuan, Marc-Alexandre Côté, Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. 2023. Joint prompt optimization of stacked llms using variational inference. In *Advances in Neural Information Processing Systems*, volume 36, pages 58128–58151. Curran Associates, Inc.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Hao Sun, Alihan Hüyük, and Mihaela van der Schaar. 2024a. Query-dependent prompt evaluation and optimization with offline inverse RL. In *The Twelfth International Conference on Learning Representations*.

Hong Sun, Xue Li, Yinchuan Xu, Youkow Homma, Qi Cao, Min Wu, Jian Jiao, and Denis Charles. 2023. Autohint: Automatic prompt optimization with hint generation. *arXiv preprint arXiv:2307.07415*.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yudong Liu, Zhixu Du, Yiran Chen, and Holger R. Roth. 2024b. Fedbpt: efficient federated black-box prompt tuning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937.

Prashant Trivedi, Souradip Chakraborty, Avinash Reddy, Vaneet Aggarwal, Amrit Singh Bedi, and George K. Atia. 2025. Align-pro: A principled approach to prompt optimization for llm alignment.

Nirali Vaghani and Mansi Thummar. 2023. Flipkart product reviews with sentiment dataset.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, and Sercan O. Arik. 2024. Teach better or show smarter? on instructions and exemplars in automatic prompt optimization.

Ruochen Wang, Sohyun An, Minhao Cheng, Tianyi Zhou, Sung Ju Hwang, and Cho-Jui Hsieh. 2025. One prompt is not enough: automated construction of a mixture-of-expert prompts. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022a. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Annual Meeting of the Association for Computational Linguistics*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. 2024a. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Annual Meeting of the Association for Computational Linguistics*.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Zixu, Zhu, Xiang-Bo Mao, Sitaram Asur, Na, and Cheng. 2024b. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39:165–210.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics*.

Yurong Wu, Yan Gao, Bin Benjamin Zhu, Zineng Zhou, Xiaodi Sun, Sheng Yang, Jian-Guang Lou, Zhiming Ding, and Linjun Yang. 2024. StraGo: Harnessing strategic guidance for prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10043–10061, Miami, Florida, USA. Association for Computational Linguistics.

Jasper Xian, Saron Samuel, Faraz Khoubsirat, Ronak Pradeep, Md Arafat Sultan, Radu Florian, Salim Roukos, Avirup Sil, Christopher Potts, and Omar Khattab. 2024. Prompts as auto-optimized training hyperparameters: Training best-in-class ir models from scratch with 10 gold labels. *arXiv preprint arXiv:2406.11706*.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. 2022. Gps: Genetic prompt search for efficient few-shot learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8162–8171.

Wei Xu, Alan Ritter, William B. Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *International Conference on Computational Linguistics*.

Weijia Xu, Andrzej Banburski-Fahey, and Nebojsa Jojic. 2024. Reprompting: automated chain-of-thought prompt inference through gibbs sampling. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024a. Large language models as optimizers.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024b. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Muchen Yang, Moxin Li, Yongle Li, Zijun Chen, Chongming Gao, Junqi Zhang, Yangyang Li, and Fuli Feng. 2024c. Dual-phase accelerated prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12163–12173, Miami, Florida, USA. Association for Computational Linguistics.

Sheng Yang, Yurong Wu, Yan Gao, Zineng Zhou, Bin Benjamin Zhu, Xiaodi Sun, Jian-Guang Lou, Zhiming Ding, Anbang Hu, Yuan Fang, et al. 2024d. Ampo: Automatic multi-branched prompt optimization. *arXiv preprint arXiv:2410.08696*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2024. Prompt engineering a prompt engineer.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic "differentiation" via text.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI, Vol. 2*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Pengwei Zhan, Zhen Xu, Qian Tan, Jie Song, and Ru Xie. 2024. Unveiling the lexical sensitivity of llms: Combinatorial optimization for prompt enhancement. In *Conference on Empirical Methods in Natural Language Processing*.

Chenrui Zhang, Lin Liu, Chuyuan Wang, Xiao Sun, Hongyu Wang, Jinpeng Wang, and Mingchen Cai. 2024a. Prefer: prompt ensemble learning via feedback-reflect-refine. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Lechen Zhang, Tolga Ergen, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024b. Sprig: Improving large language model performance by system prompt optimization. *ArXiv*, abs/2410.14826.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2022. Tempera: Test-time prompting via reinforcement learning.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. 2023. Survival of the most influential prompts: Efficient black-box prompt search via clustering and pruning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13064–13077, Singapore. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers.

12 Appendix

12.1 Notation

We now define the notation of key terms and expressions used throughout the paper.

1. T = Task type, I= Task instruction, $E = (xi, yi)_{i=1}^{e}$ Few shot demonstrations in the prompt, τ= Template delimiters, z = CoT recipe for a task-instance, $z_i \in I_i$
2. M_{task} target model, M_{APO} APO system
3. $\rho = concat([s_1, s_2, \ldots, s_m]) = concat(I, \tau, E)$ Prompt composed of m sentences, which comprise of Instruction, template delimiters and few-shot demonstrations.
4. $D = \{(x_i, y_i)\}_{i=1}^{m}$ collection of m input-output pairs. D_{val} is the validation set used to validate prompt performance, D_{train} is the training set used to finetune the language model(Reprompting).
5. $\{f_1, f_2, \ldots\} \in F$ metric function upon which to evaluate task-prompt performance
6. $r : S \times A \rightarrow R$= reward model score, where S is the state-space and A is the action-space
7. $|V|$ = length of vocabulary
8. $\phi : S \in V_* \rightarrow R_d$ embedding function which takes in a sentence generated as a finite sequence of tokens belonging to a vocabulary V, and generating a floating point array representation of dimension d
9. $\rho_* = argmax_{\rho \in V_*} E_{D_{val}}[f_i(\rho)]$ The best performing prompt based on the metric score on validation set
10. k = number of candidates for top-K search, B = Beam width for beam search, N = number of iterations for search
11. C = number of experts in a Mixture of Experts approach (MOP), μ_C= cluster centroid of cluster C (MOP).
12. LLM_{target}= target model which will be used for inference, $LLM_{rewriter}$= rewriter model which will be used for rewriter, $LLM_{evaluator}$= evaluator model which provides the LLM feedback to prompts / responses or both
13. λ with subscripts to denote different latency types: λ_t = Total training cost/latency, including all offline costs for data collection, preprocessing, and model fine-tuning, λ_i = per-example inference latency, λ_m = MLM inference latency per-example

12.2 Excluded works

FedBPT (Sun et al., 2024b) used federated learning to update soft prompts and not discrete tokens. **Deliberate-then-generate** (Li et al., 2023a) randomly sampled arbitrary noisy inference and prompted the task LLM to deliberate on the wrong inference, while **Reflexion** (Shinn et al., 2023) agents maintain an episodic buffer of past deliberations. Neither method optimizes the input prompt. **AutoPrompt** (Shin et al., 2020) required gradient access to the task LLM and therefore doesn't remain blackbox.

12.3 UCB based selection algorithm

Algorithm 2 $Select(\cdot)$ with UCB Bandits

```
Require: $n$ prompts $\rho_1, ..., \rho_n$, dataset $\mathcal{D}_{val}$, $T$ time steps, metric function $m$
Initialize: $N_t(\rho_i) \leftarrow 0$ for all $i = 1, \ldots, n$
Initialize: $Q_t(\rho_i) \leftarrow 0$ for all $i = 1, \ldots, n$
for $t = 1, \ldots, T$ do
    Sample uniformly $\mathcal{D}_{sample} \subset \mathcal{D}_{val}$
    $\rho_i \leftarrow \arg\max_\rho \left\{ \frac{Q_t(\rho)}{N_t(\rho_i)} + c\sqrt{\frac{\log t}{N_t(\rho)}} \right\}$
    Observe reward $r_{i,t} = m(\rho_i, \mathcal{D}_{sample})$
    $N_t(\rho_i) \leftarrow N_t(\rho_i) + |\mathcal{D}_{sample}|$
    $Q_t(\rho_i) \leftarrow Q_t(\rho_i) + r_{i,t}$
return $SelectTop_b(Q_T/N_T)$
```

13 Comparison of different approaches + Tasks

13.1 Comparison

Below we offer a comprehensive comparison of all the surveyed methods against our framework, covering the following aspects

1. **Seed instructions**
2. **Inference evaluation**
3. **Candidate generation**
4. **Search+filter strategy**
5. **Iteration depth**
6. **Optimization time complexity**
7. **Prompt generation model**
8. **Target models**

SNo.	Method	Seed instructions	Inference evaluation	Candidate generation	Search+filter strategy	Iteration depth	Optimization time complexity	Prompt generation model	Target models
1	GPS (Xu et al., 2022)	Manually created	Task accuracy	Genetic Algorithm: Back translation, Cloze, Sentence continuation	Metaheuristic ensemble	Fixed	$O(T*N*k*\lambda_i)$		T0
2	GRIPS (Prasad et al., 2023)	Manually created	Entropy-based score+ Task accuracy	Phrase level add/remove/swap/paraph	TopK selection	Fixed	$O(k*N*\|D_{val}\|*B)$	PEGASUS paraphrase model	InstructGPT
3	Instruction induction (Honovich et al., 2023)	Instruction induction	Accuracy + BERTScore	LLM-rewriter		Fixed	$O(\|\rho\|*\lambda_i)$	InstructGPT, GPT-3	InstructGPT, GPT-3
4	RLPrompt (Deng et al., 2022)	Manually created	Task accuracy + Reward model score	RL-based trained NN	TopK selection	Fixed	$O(N*\rho*\|V\|*\lambda_i)$	RoBERTa-large Reward model-DistilBERT	1/ BERT, 2/ GPT-2
5	TEMPERA (Zhang et al., 2022)	Manually created	Task accuracy	RL-trained NN		Fixed	$O(N*k*\|V\|*C)$	RoBERTa-large	RoBERTa-large
6	AELP (Hsieh et al., 2024)	Manually created	Task accuracy	Genetic algorithm: LLM-mutator	Beam search	Fixed	$O(N*\rho*k*\|D\|*\lambda_i)$	PaLM 2-L	PaLM text-bison
7	APE (Zhou et al., 2022)	Instruction induction	Task accuracy	No new candidates	TopK selection	Fixed	$O(N*k*\|D_{val}\|*\lambda_i)$	InstructGPT, GPT-3, T5, InsertGPT	InstructGPT, GPT-3
8	AutoHint (Sun et al., 2023)	Manually created	Task accuracy + LLM-feedback	LLM rewriter	TopK selection	Fixed	$O(T*\|D\|*\lambda_i)$		GPT-4
9	BDPL (Diao et al., 2022)	Manually created	Task accuracy	RL-trained NN	TopK selection	Variable	$O(N*k*\lambda_i)$	RoBERTa, GPT-3	RoBERTa, GPT-3
10	Boosted Prompting (Pitis et al., 2023)	Instruction-induction	Task accuracy	Ensemble based method	TopK selection	Variable	$O(N*k*\lambda_i)$	text-curie-001, text-curie-003, GPT-3.5, code-davinci-002	text-curie-001, text-curie-003, GPT-3.5, code-davinci-002
11	BPO (Cheng et al., 2024)	Manually created	LLMaaJ (pairwise)	Finetuned LLMs	NA	NA	$O(\lambda_t+\|D_{val}\|*\lambda_i)$	Llama2-7b-chat	Vicuna-7b-v1.3, vicuna-13b-v1.3, llama-1-7b, llama-1-13b
12	CLAPS (Zhou et al., 2023)	Manually created	Entropy-based score+ Task accuracy	Genetic Algorithm: Mutation + Crossover	TopK selection	Variable	$O(N*k*\|V\|*\lambda_i)$	Flan-T5	Flan-T5 large and base
13	Directional-stimulus (Li et al., 2023d)	Manually created	BLEU, BERTScore	RL-trained NN		Variable	$O(\lambda_t)$	T5, GPT-2	ChatGPT, Codex, InstructGPT
14	DLN (Sordoni et al., 2023)	Manually created	Task accuracy + NLL	LLM mutator	TopK selection	Fixed	$O(N*k*\|D_{train}\|)$	GPT-3 (text-davinci-003), GPT-4	GPT-3 (text-davinci-003), GPT-4
15	DSP (Khattab et al., 2022)	Instruction induction	Task accuracy	Program Synthesis	TopK selection	Fixed	$O(N*k*\lambda_i)$	GPT-3.5	LM: GPT-3.5, Retrieval: ColBERTv2
16	DSPy (Khattab et al., 2024)	Manually created + Instruction Induction	Task accuracy + LLM-feedback	Program Synthesis	TopK selection	Variable	$O(N*k*B*\lambda_i)$		
17	GATE (Joko et al., 2024)	Manually created	Human feedback	LLM rewriter		Open-ended	$O(N*(\lambda_m+\|D_{val}\|*\lambda_i))$	GPT-4	GPT-4
18	GPO (Li et al., 2023c)	Instruction induction	Task-Accuracy and F1	Metaprompt-design	TopK selection		$O(N*C*\|V\|*B*E)$	gpt-3.5-turbo-0301	gpt-3.5-turbo-0301
19	PACE (Dong et al., 2024b)	Manually created	NLL + Task accuracy - BLEU and BERTScore	LLM-rewriter	TopK selection	< 3	$O(N*\|\rho\|*\|D_{val}\|)$	gpt-3.5-turbo (0301)	text-davinci-002, text-davinci-003, (gpt-3.5-turbo), GPT-4
20	PREFER (Zhang et al., 2024a)	Manually created	Task accuracy	LLM-rewriter + Ensemble method	TopK selection	Fixed	$O(N*\|\rho\|*\|D_{val}\|)$	ChatGPT	ChatGPT
21	Promptagent (Wang et al., 2024a)	Manually created	Task accuracy + LLM-feedback	LLM rewriter	UCT-based bandit-search	Fixed	$O(N*k*\lambda_i)$	GPT-4	GPT-3.5, GPT-4, PaLM-2

Table 2: Comparison of all APO techniques based on our framework

SNo.	Method	Seed instructions	Inference evaluation	Candidate generation	Search+filter strategy	Iteration depth	Optimization time complexity	Prompt generation model	Target models
22	Promptboosting (Hou et al., 2023)	Instruction-induction	Accuracy, F1 Score	Ensemble based method	Beam-search	Early Stopping	$O(\lambda_m)$	T5	RoBERTa-large
23	Promptbreeder (Fernando et al., 2023)	Manually created	LLM Feedback + Task accuracy	Genetic Algorithm: Mutate + Crossover (LLM-edits)	Metaheuristic Ensemble	Fixed	$O(\rho * N * \|V\| * \lambda_i)$	text-davinci-003, PaLM 2-L	text-davinci-003, PaLM 2-L
24	ProTeGi (Pryzant et al., 2023)	Manually created	Task accuracy + LLM-feedback	LLM rewriter	UCT-based bandit-search	Fixed	$O(N * C * \|D_{val}\| * \lambda_i)$	GPT-3.5-Turbo	GPT-3.5-turbo
25	Random separators (Lu et al., 2024)	Manually created	Task accuracy	LLM-rewriter	TopK selection	Fixed steps	$O(N * k * \lambda_{)}$	GPT2 Large, GPT2 XL, Mistral 7B, Mistral 7B Instruct, Llama-Alpaca 7B, Llama2 7B. Llama2 7B Chat, ChatGPT	GPT2 Large, GPT2 XL, Mistral 7B, Mistral 7B Instruct, Llama-Alpaca 7B, Llama2 7B. Llama2 7B Chat, ChatGPT
26	ABO (Yang et al., 2024b)	Manually created + Instruction Induction	Task accuracy + LLM-feedback	LLM-rewriter	TopK selection	Fixed Steps	$O(B * N * \lambda_i)$	GPT-4	GPT-3.5-Turbo, Llama-2-70B-chat
27	Adv-ICL (Long et al., 2024)	Manually created	LLM Feedback	LLM-rewriter	Top-1 selection	Fixed	$O(N * k * \lambda_i)$	text-davinci-002, vicuna, ChatGPT	text-davinci-002, vicuna, ChatGPT
28	AMPO (Yang et al., 2024d)	Manually created	Task accuracy + F1 score	Coverage-based	TopK selection	Variable	$O(N * C * \lambda_i)$	GPT-4-turbo	GPT-4-turbo
29	APEER (Jin et al., 2024)	Manually created	Task accuracy-nDCG	Feedback + preference optimization		Used 3 epochs	$O(N * \|\rho\| * \|D_val\|)$		GPT4, GPT3.5, Llama3, Qwen2
30	APOHF (Lin et al., 2024)	Manually created	Task accuracy + Human feedback	LLM rewriter	Linear UCB	Fixed	$O(N * T)$	ChatGPT	DALLE-3, ChatGPT
31	BATPrompt (Shi et al., 2024)	Manually created	Task accuracy + LLM-feedback	LLM rewriter	TopK selection	Fixed	$O(N * \|D\| * \|\rho\| * \lambda_i)$	GPT-3.5-turbo	GPT-3.5-turbo, GPT-4o-mini, Llama2-7b
32	COPLE (Zhan et al., 2024)	Manually created	Task accuracy	Token edits using MLM		Variable	$O(N * \|I\| * k * \|D_val\| * \lambda_i)$	RoBERTa (filling masked tokens)	Llama-2-7B-chat , Mistral-7B-Instruct-v0.1, ChatGPT (gpt-3.5-turbo-0125)
33	CRISPO (He et al., 2025)	Manually created	LLM feedback + ROUGE-1/2/L F-measure, AlignScore	LLM rewriter	TOP-K greedy search	Fixed	$O(N * k * (\|D_{train}\| * \lambda_i + \lambda_m))$	Claude Instant, Claude 3 Sonnet, Mistral 7B, Llama3 8B	Claude Instant, Claude 3 Sonnet, Mistral 7B, Llama3 8B
34	DAPO (Yang et al., 2024c)	Manually created	Task accuracy	LLM-rewriter	Top-1 selection	Fixed	$O(N * k * \lambda_i)$	GPT-3.5-Turbo, Baichuan2, GPT-4	GPT-3.5-Turbo, Baichuan2, GPT-4
35	DRPO (Amini et al., 2024)	Manually created	Reward model score + LLM Feedback	LLM rewriter	Beam search	Fixed	$O(B * k * N)$	Mistral 7b, Mistral 7b (Instruct), Llama 2 70b, Llama 2 70b (chat), Llama 3 8b, Llama 3 8b (Instruct), gpt-3.5-turbo	Mistral 7b, Mistral 7b (Instruct), Llama 2 70b, Llama 2 70b (chat), Llama 3 8b, Llama 3 8b (Instruct), gpt-3.5-turbo
36	EVOPROMPT (Guo et al., 2024)	Manually created + Instruction Induction	Task Accuracy + ROUGUE+ SARI	Genetic Algorithm: Mutation operators+ Crossover	Metaheuristic ensemble	Early Stopping	$O(N * k * T * \lambda_i)$		Alpaca-7b, GPT-3.5
37	FIPO (Lu et al., 2025)	Manually created	Task accuracy	Finetuned LLMs			$O(\lambda_t + \|D_val\| * \lambda_i))$	Tulu-13B, Tulu-70B	Llama2-7B, Tulu2-13B, Baichuan2-13B

Table 3: Comparison of all APO techniques based on our framework

SNo.	Method	Seed instructions	Inference evaluation	Candidate generation	Search+filter strategy	Iteration depth	Optimization time complexity	Prompt generation model	Target models
38	LMEA (Liu et al., 2023)	Manually created	Numeric Score-based	Genetic Algorithm: Mutate + Crossover (LLM-edits)	TopK selection	Fixed	$O(N * k * \lambda_i)$		GPT-3.5-turbo-0613
39	MIPRO (Opsahl-Ong et al., 2024)	Manually created	Task accuracy	Program Synthesis	TopK selection	Fixed	$O(N * \|D_{val}\| * k * \lambda_i)$	GPT-3.5 (proposer LM)	Llama-3-8B (task LM)
40	MOP (Wang et al., 2025)	Instruction induction	Task Accuracy	APE for each cluster	TopK selection	Fixed steps per-cluster	$O(C * N * \|D_{val}\|)$	GPT-3.5-Turbo	GPT-3.5-Turbo
41	MORL-Prompt (Jafari et al., 2024)	Manually created	Task accuracy + Reward score	RL-based trained NN		Fixed	$O(N * C * \|V\| * k)$	distilGPT-2	GPT-2 (style transfer), flan-T5-small (translation)
42	OIRL (Sun et al., 2024a)	Manually created	Task accuracy + Reward model score	LLM rewriter			$O(\|D_{train}\| * \rho * \lambda_i + \lambda_t + \|D_{val}\| * \lambda_i)$	GPT4	Llama2-7B-chat, Tigerbot-13B-chat, gpt3.5-turbo
43	OPRO (Yang et al., 2024a)	Manually created	Task accuracy + LLM-feedback	Metaprompt design	TopK selection	Variable	$O(N * k * \lambda_i)$	PaLM 2-L, text-bison, gpt-3.5-turbo and GPT-4	PaLM family models
44	PE2 (Ye et al., 2024)	Manually created + Instruction Induction	Task accuracy + LLM-feedback	Metaprompt design	TopK selection	Fixed	$O(N * k * \lambda_i)$	GPT-4	text-davinci-003
45	PIN (Choi et al., 2024)	Manually created	Task accuracy	RL-trained LLM	TopK selection	Fixed	$O(N * \|V\| * \lambda_i * C)$	OPT	RoBERTa-large (classification), OPT models (others)
46	PLUM (Pan et al., 2024)	Manually created	Task accuracy	Genetic Algorithm: Mutate + crossover	Metaheuristics	Fixed steps	$O(N * C * k * \lambda_i)$	GPT-3-babbage	GPT-3-babbage
47	PRewrite (Kong et al., 2024)	Manually created	Task accuracy + Reward model score	RL-trained LLM	TopK selection	Fixed	$O(N * C * \lambda_i * \|V\|)$	PaLM 2-S	PaLM 2-L
48	PROMPTWIZARD (Agarwal et al., 2024)	Manually created	Task accuracy + LLM-feedback	Genetic Algorithm: Mutate + Crossover (LLM-edits)	TopK selection	Fixed	$O(N * C * \lambda_i)$	GPT3.5/GPT4	GPT3.5/GPT4/Llama-70B
49	PROMST (Chen et al., 2024)	Manually created	Task accuracy + Human feedback	LLM rewriter	TopK selection	Fixed	$O(N * k * \lambda_i)$	GPT-4	GPT-3.5, GPT-4
50	Reprompting (Xu et al., 2024)	LLM generated CoT process.	Task accuracy	LLM-rewriter	Rejection sampling with exploration	Fixed or until convergence	$O(N * k * \|\rho\|)$	gpt-3.5-turbo, textdavinci-003	gpt-3.5-turbo, textdavinci-003
51	SAMMO (Schnabel and Neville, 2024)	Manually created	Task accuracy	Program synthesis	TopK selection	Fixed	$O(N * k * \lambda_i)$		Mixtral7x8B, Llama-2 70B, GPT3.5, GPT4
52	SCULPT (Kumar et al., 2024)	Instruction induction on task-README	Task accuracy + LLM-feedback	LLM-rewriter	UCB bandit search	Fixed	$O(N * k * \|\rho\| * \|D_val\|)$	GPT-4o	GPT-4o and Llama3.1-8B
53	SOS (Sinha et al., 2024)	Manually created	Task accuracy + LLM-feedback	LLM-mutator	TopK selection	Fixed	$O(N * C * k * \lambda_i)$	GPT-3.5-turbo, Llama3-8B, Mistral-7B	GPT-3.5-turbo, Llama3-8B, Mistral-7B
54	SPRIG (Zhang et al., 2024b)	Manually created	Task accuracy	Genetic Algorithm: Mutate + Crossover (tokens)	Beam-search	Fixed	$O(N * B * T * k * \lambda_i)$	tuner007/pegasus_pa	Llama 3.1-8B Instruct, Mistral Nemo Instruct 2407, Qwen 2.5-7B Instruct, Llama 70B, Qwen 2.5-72B, Mistral Large 2407.
55	StraGo (Wu et al., 2024)	Manually created	Task accuracy + LLM-feedback	Genetic Algorithm: Mutate + CrossOver (tokens)	Bandit Search (UCB)	Early Stopping	$O(N * k * T * \lambda_i)$	GPT-4	GPT-3.5-turbo or GPT-4
56	TextGrad (Yuksekgonul et al., 2024)	Manually created	Task accuracy + LLM-feedback	LLM rewriter		Variable	$O(N * \|D_{val}\| * \lambda_i)$		GPT-3.5, GPT-4o
57	UNIPROMPT (Juneja et al., 2024)	Manually created + Instruction Induction	Task accuracy + LLM-feedback	LLM-rewriter	Beam Search	Early Stopping	$O(N * k * \lambda_i)$	Fine-tuned Llama2-13B	GPT-3.5

Table 4: Comparison of all APO techniques based on our framework

13.2 Evaluation tasks and datasets

Below we describe the different datasets and tasks that each method was evaluated on.

SNo.	Paper	Tasks
1	GPS (Xu et al., 2022)	10 unseen tasks from the T0 benchmark, which span: 1. Natural Language Inference: ANLI R1, R2, R3, CB, RTE (Nie et al., 2019; Dagan et al., 2005). 2. Coreference Resolution: WSC, Winogrande.(Levesque et al., 2011) 3. Sentence Completion: COPA(Roemmele et al., 2011) , HellaSwag (Zellers et al., 2019). 4. Word Sense Disambiguation: WiC (Pilehvar and Camacho-Collados, 2019).
2	GRIPS (Prasad et al., 2023)	8 classification tasks from NaturalInstructions (Mishra et al., 2021)
3	Instruction induction (Honovich et al., 2022)	1. Spelling, 2. Syntax, 3. Morpho-syntax, 4. Lexical semantics, 5. Phonetics, 6. Knowledge, 7. Semantics, 8. Style
4	RLPrompt (Deng et al., 2022)	1. Classification 2. Text-style transfer
5	TEMPERA (Zhang et al., 2022)	Classification
6	AELP (Hsieh et al., 2024)	Big Bench Hard (Suzgun et al., 2023)
7	APE (Zhou et al., 2022)	1. 24 Instruction induction tasks (Honovich et al., 2022) 2. 21 BIG Bench Hard tasks (Suzgun et al., 2023)
8	AutoHint (Sun et al., 2023)	BIG-Bench Instruction Induction (Epistemic Reasoning, Logical Fallacy Detection, Implicatures, Hyperbaton, Causal Judgment, Winowhy) (Zhou et al., 2022)
9	BDPL (Diao et al., 2022)	1. MNLI (Williams et al., 2017), 2. QQP (Cer et al., 2017), 3. SST-2 (Socher et al., 2013), 4. MRPC (Dolan and Brockett, 2005), 5. CoLA (Warstadt et al., 2018), 6. QNLI (Rajpurkar et al., 2016), 7. RTE (Dagan et al., 2005), 8. CitationIntent (Jurgens et al., 2018), 9. SciERC (Luan et al., 2018), 10. RCT (Dernoncourt and Lee, 2017), 11. HyperPartisan (Kiesel et al., 2019)
10	Boosted Prompting (Pitis et al., 2023)	GSM8K (Cobbe et al., 2021) and AQuA (Garcia et al., 2020)
11	BPO (Cheng et al., 2024)	Generation: Dolly Eval (Conover et al., 2023), Vicuna Eval (Chiang et al., 2023), Self-Instruct Eval (Wang et al., 2022b)
12	CLAPS (Zhou et al., 2023)	
13	Directional-stimulus (Li et al., 2023d)	MultiWOZ (Budzianowski et al., 2018)
14	DLN (Sordoni et al., 2023)	1. Mpqa Sentiment analysis (Lu et al., 2021) 2. Trec Question type classification (Lu et al., 2021) 3. Subj Determine whether a sentence is subjective or objective (Lu et al., 2021) 4. Leopard (Bansal et al., 2019)- Disaster Determine whether a sentence is relevant to a disaster. 5. Leopard (Bansal et al., 2019)- Airline Airline tweet sentiment analysis. 6. BBH (Suzgun et al., 2023)- (Hyper, Nav, Date, Logic datasets)
15	DSP (Khattab et al., 2022)	1. open-domain question answering (Open-SQuAD) (Lee et al., 2019) 2. multi-hop question answering (HotPotQA) (Yang et al., 2018) 3. conversational question answering (QReCC) (Anantha et al., 2020)
16	DSPy (Khattab et al., 2024)	
17	GATE (Joko et al., 2024)	LAPS (Joko et al., 2024) (1. Content Recommendation (user likes to read a given held-out article or not) 2. Moral Reasoning, 3. Email Verification)
18	GPO (Li et al., 2023c)	1. Sentiment analysis - Yelp (Zhang et al., 2015), Flipkart (Vaghani and Thummar, 2023), IMDB (Maas et al., 2011), Amazon (Zhang et al., 2015) 2. NLI - MNLI (Williams et al., 2017), ANLI (Nie et al., 2019) 3.Entailment - RTE (Dagan et al., 2005), 4. CommonsenseQA - SocialIQA (Sap et al., 2019) 5. Multi-turn dialog - DSTC7 (Gunasekara et al., 2019), Ubuntu Dialog (Lowe et al., 2015), MuTual (Cui et al., 2020) 6. NumericalQA - DROP (Dua et al., 2019)
19	PACE (Dong et al., 2024b)	BBH (Suzgun et al., 2023), instruction induction tasks (24 tasks) (Honovich et al., 2022) and translation tasks (en-de, en-es, en-fr)
20	PREFER (Zhang et al., 2024a)	1. NLI tasks including SNLI (Bowman et al., 2015), MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005) 2. Classification: Ethos (Mollas et al., 2020), liar (Wang, 2017), ArSarcasm (Farha and Magdy, 2020a)
21	Promptagent (Wang et al., 2024a)	1. BigBenchHard (BBH) (Suzgun et al., 2023) - 6 BBH tasks that emphasize a blend of domain knowledge 2. Biomedical - Disease NER (NCBI) (Doğan et al., 2014), MedQA (Jin et al., 2020), Bio similar sentences (Sogancioglu et al., 2017) 3. 2 classification - TREC (Voorhees and Tice, 2000) + Subj. (Pang and Lee, 2004) 1 NLI(CB) (de Marneffe et al., 2019)
22	Promptboosting (Hou et al., 2023)	Text Classification
23	Promptbreeder (Fernando et al., 2023)	1. Arithmetic Reasoning: Benchmarks: GSM8K (Cobbe et al., 2021), MultiArith (Roy and Roth, 2016), AddSub (Hosseini et al., 2014), SVAMP (Patel et al., 2021), SingleEq (Koncel-Kedziorski et al., 2015), AQuA-RAT (Ling et al., 2017). 2. Commonsense Reasoning: Benchmarks: CommonSenseQA (CSQA) (Talmor et al., 2019), StrategyQA (SQA) (Geva et al., 2021). 3. Hate Speech Classification: Dataset: ETHOS (Mollas et al., 2020). 4. Instruction Induction (Honovich et al., 2022): Tasks: 24 datasets spanning sentence similarity, style transfer, sentiment analysis, and more

Table 5: Tasks covered in the different papers

SNo.	Paper	Tasks
24	ProTeGi (Pryzant et al., 2023)	Jailbreak (Pryzant et al., 2023), Liar (Wang, 2017), Sarcasm (Farha and Magdy, 2020b), Ethos (Mollas et al., 2020)
25	Random separators (Lu et al., 2024)	1. SST-2, SST-5,(Socher et al., 2013) 3. DBPedia (Zhang et al., 2015), 4. MR (Pang and Lee, 2005), 5. CR (Hu and Liu, 2004), 6. MPQA (Wiebe et al., 2005), 7. Subj (Pang and Lee, 2004), 8. TREC (Voorhees and Tice, 2000), 9. AGNews (Zhang et al., 2015)
26	ABO (Yang et al., 2024b)	BigBenchHard tasks (Suzgun et al., 2023): Object Counting, Navigate, Snarks, Question Selection
27	Adv-ICL (Long et al., 2024)	Summarization (XSUM (Narayan et al., 2018), CNN/Daily Mail (Nallapati et al., 2016)), Data-to-Text (WebNLG (Gardent et al., 2017), E2E NLG (Novikova et al., 2017)), Translation (LIRO (Dumitrescu et al., 2021), TED Talks (Qi et al., 2018)), Classification (YELP-5 (Zhang et al., 2015), WSC (Levesque et al., 2011)), Reasoning (GSM8k (Cobbe et al., 2021), SVAMP (Patel et al., 2021))
28	AMPO (Yang et al., 2024d)	Text classification task TREC (Voorhees and Tice, 2000), sentiment classification task SST-5 (Socher et al., 2013), largescale reading comprehension task RACE (Lai et al., 2017), medical question-answering tasks MedQA (Jin et al., 2020) and MedMCQA (Pal et al., 2022)
29	APEER (Jin et al., 2024)	Passage reranking
30	APOHF (Lin et al., 2024)	1. User instruction optimization using tasks from Instructzero, 2. Text-to-image , 3. Response optimization
31	BATPrompt (Shi et al., 2024)	1. Language understanding, 2. Text summarization, 3. Text simplification
32	COPLE (Zhan et al., 2024)	GLUE - SST2 (Socher et al., 2013), COLA (Warstadt et al., 2018), MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005), MRPC (Dolan and Brockett, 2005), QQP (Cer et al., 2017) MMLU (Hendrycks et al., 2020) - STEM, Humanities, Social Sciences and Other
33	CRISPO (He et al., 2025)	Summarization, QA
34	DAPO (Yang et al., 2024c)	1. Sentiment classification, 2. topic classification, 3. News, 4. TREC (Voorhees and Tice, 2000), 5. subjectivity classification (Pang and Lee, 2004), 6. Logic Five, 7. Hyperbaton, 8. Disambiguation, 9. Salient, 10.Translation
35	DRPO (Amini et al., 2024)	Alignment benchmark
36	EVOPROMPT (Guo et al., 2024)	1. Language Understanding: Sentiment classification (e.g., SST-2, SST-5, CR, MR (Socher et al., 2013; Hu and Liu, 2004; Pang and Lee, 2005)), 2. Topic classification (e.g., AGNews (Zhang et al., 2015), TREC (Voorhees and Tice, 2000)), Subjectivity classification (Subj (Pang and Lee, 2004)). 3. Language Generation: Summarization (SAMSum (Gliwa et al., 2019)). Simplification (ASSET (Alva-Manchego et al., 2020)). 4. Reasoning (BIG-Bench Hard Tasks) (Suzgun et al., 2023): Multi-step reasoning tasks from BBH, such as logical deduction, causal judgment, and object tracking.
37	FIPO (Lu et al., 2025)	1. Generation: GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2023) 2. Multiple Choice: PiQA (Bisk et al., 2019), CosmosQA (Huang et al., 2019), MMLU (Hendrycks et al., 2020)
38	LMEA (Liu et al., 2023)	Traveling Salesman Problems (TSPs)
39	MIPRO (Opsahl-Ong et al., 2024)	1. Question Answering (HotPotQA)(Yang et al., 2018) 2. Classification (Iris (Fisher, 1936), Heart Disease (Detrano et al., 1989)) 3. Entailment (ScoNe) (She et al., 2023) 4. Multi-hop Fact Extraction and Claim Verification (HoVer) (Jiang et al., 2020)
40	MOP (Wang et al., 2025)	50 tasks comprising of Instruction Induction (Honovich et al., 2022), Super Natural Instructions (Mishra et al., 2021), BBH (Suzgun et al., 2023)
41	MORL-Prompt (Jafari et al., 2024)	1. Unsupervised Text Style Transfer: Shakespearean data (Xu et al., 2012) 2. Supervised Machine Translation: iwslt2017 (Cettolo et al., 2017)
42	OIRL (Sun et al., 2024a)	Arithmetic reasoning: GSM8K (Cobbe et al., 2021), MAWPS, SVAMP (Patel et al., 2021)
43	OPRO (Yang et al., 2024a)	GSM8K (Cobbe et al., 2021), BBH (23 tasks) (Suzgun et al., 2023), MultiArith (Roy and Roth, 2016), AQuA (Garcia et al., 2020)
44	PE2 (Ye et al., 2024)	1. MultiArith and GSM8K for math reasoning (Cobbe et al., 2021), 2. Instruction Induction (Honovich et al., 2022), 3. BIG-bench Hard for challenging LLM tasks (Suzgun et al., 2023) 4. Counterfactual Evaluation 5. Production Prompt
45	PIN (Choi et al., 2024)	1. Classification: SST-2 and etc (Socher et al., 2013) 2. Unsupervised Text Style transfer: Yelp (Zhang et al., 2015) 3.Textual Inversion From Images: MSCOCO (Lin et al., 2014), LAION (Schuhmann et al., 2022)
46	PLUM (Pan et al., 2024)	Natural-Instructions datasets v2.6 (Mishra et al., 2021)
47	PRewrite (Kong et al., 2024)	1. Classification: AG News (Zhang et al., 2015), SST-2 (Socher et al., 2013) 2. Question answering: NQ (Kwiatkowski et al., 2019) 3. Arithmetic reasoning: GSM8K (Cobbe et al., 2021)
48	PROMPTWIZARD (Agarwal et al., 2024)	1. BIG-Bench Instruction Induction (BBII) (Honovich et al., 2022) 2. GSM8k (Cobbe et al., 2021), AQUARAT (Ling et al., 2017), and SVAMP (Patel et al., 2021) 3. BIG-Bench Hard (BBH) (Suzgun et al., 2023) 4. MMLU (Hendrycks et al., 2020), Ethos (Mollas et al., 2020), PubMedQA (Jin et al., 2019), MedQA (Jin et al., 2020)
49	PROMST (Chen et al., 2024)	11 multistep tasks: 1. Webarena, 2. Alfworld (Shridhar et al., 2020), 3. Scienceworld (Wang et al., 2022a), 4. BoxNet1 (Nezhadarya et al., 2019), 5. BoxNet2, 6. BoxLift, 7. Warehouse, 8. Gridworld 1, 9. Gridworld 2, 10. Blocksworld, 11. Logistics
50	Reprompting (Xu et al., 2024)	BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al.)

Table 6: Tasks covered in the different papers

SNo.	Paper	Tasks
51	SAMMO (Schnabel and Neville, 2024)	1. BigBench zero-shot classification tasks (Srivastava et al., 2022) 2. GeoQuery (Zelle and Mooney, 1996), SMCalFlow (Andreas et al., 2020), Overnight (Wang et al., 2015) 3. Super-NaturalInstructions (Mishra et al., 2021)
52	SCULPT (Kumar et al., 2024)	BBH (23 tasks) (Suzgun et al., 2023), RAI (Kumar et al., 2024)
53	SOS (Sinha et al., 2024)	1. Sentiment Analysis 2. Orthography Analysis, 3. Taxonomy of Animals, 4. Disambiguation QA, 5. Logical Five, 6. Color Reasoning
54	SPRIG (Zhang et al., 2024b)	1. Reasoning: Tasks requiring multi-step logic or causal reasoning. 2. Math: Arithmetic and logical deduction problems. 3. Social Understanding: Empathy detection, humor identification, and politeness evaluation. 4. Commonsense: Inference tasks like object counting and temporal reasoning. 5. Faithfulness: Ensuring generated outputs align with input data. 6. Knowledge: Open-domain QA and knowledge recall tasks. 7. Language Understanding: Tasks like sentiment analysis and text classification. 8. Popular benchmarks include MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2023), TruthfulQA (Lin et al., 2022), XCOPA (Ponti et al., 2020), SocKET (Choi et al., 2023), and others, covering 47 task types across multiple languages and domains.
55	StraGo (Wu et al., 2024)	BBH (Suzgun et al., 2023)(five challenging tasks within Big-Bench Hard) 2. SST-5 (Socher et al., 2013)(fine-grained sentiment classification) 3. TREC (Voorhees and Tice, 2000)(question-type classification). 4. MedQA (Jin et al., 2020),MedMCQA (Pal et al., 2022) (medical-domain QA) 5. Personalized Intent Query (an internal industrial scenario)
56	TextGrad (Yuksekgonul et al., 2024)	LeetCode Hard (Shinn et al., 2024), Google-proof QA (Rein et al., 2023), MMLU (Hendrycks et al., 2020) (Machine Learning, College Physics), BBH (Suzgun et al., 2023) (Object Counting, Word Sorting), GSM8k (Cobbe et al., 2021), DOCKSTRING (Garc'ia-Orteg'on et al., 2021)(molecule evaluation)
57	UNIPROMPT (Juneja et al., 2024)	(1) Ethos (Mollas et al., 2020), (2) ARC (Clark et al., 2018) , (3) MedQA (Jin et al., 2020), (4) GSM8K (Cobbe et al., 2021) and (5) one real-world task: Search Query Intent (Juneja et al., 2024)

Table 7: Tasks covered in the different papers

14 Prompt examples

14.1 Instruction Induction

Below is the original instruction induction prompt used by Honovich et al. (2023)

```
{{# system ∼ }}
You are a helpful assistant
{{∼ / system }}
{{# user ∼}}
I gave a friend an instruction and [[n_demo]] inputs. The friend read the instruction and wrote an output for every one of the inputs. Here are the input - output pairs:
{{ demos }}
What was the instruction ? It has to be less than {{ max_tokens }} tokens .
{{∼ / user }}
{{# assistant ∼}}
The instruction was {{gen ’instruction ’ [[ GENERATION_CONFIG ]]}}
{{∼ / assistant }}
```

14.2 Metaprompt design example

Below is the metaprompt used in OPRO (Yang et al., 2024a)

```
I have some texts along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality. text:
Let’s figure it out!
score: 61
text: Let’s solve the problem.
score: 63
(. . . more instructions and scores . . . )
The following exemplars show how to apply your text:
you replace in each input with your text, then read the input and give an output. We say your output is wrong if your output is different from the given output, and we say your output is correct if they are the same.
input: Q: Alannah, Beatrix, and Queen are preparing for the new school year and have been given books by their parents. Alannah has 20 more books than Beatrix. Queen has 1/5 times more books than Alannah. If Beatrix has 30 books, how many books do the three have together?
A: output: 140
(. . . more exemplars . . . )
Write your new text that is different from the old ones and has a score as high as possible. Write the text in square brackets
```

14.3 LLM Feedback prompts

Table 8: Automatic prompt optimization for LLM-as-a-Judge methods, text gradients (Pryzant et al., 2023; Wang et al., 2024a) and PE2 (Ye et al., 2024).

Method	LLMaaJ prompt	Candidate prompt	Response	Subject of evaluation (prompt / response / both)	Evaluation output	Rewritten prompt
Text-gradients (Pryzant et al., 2023)	I'm trying to write a zero-shot classifier prompt. My current prompt is: "{prompt}" But this prompt gets the following examples wrong: {error_string} give {num_feedbacks} reasons why the prompt could have gotten these examples wrong. Wrap each reason with <START> and <END>	Determine whether the Statement is a lie (Yes) or not (No) based on the Context and other information. Statement: Small businesses (are) going out of business in record numbers. Job title: Senator. State: Texas. Party: republican. Context: a speech at Liberty University" Label: Yes Prediction: No	N/A	Prompt	The prompt does not take into account the speaker's potential biases or agenda, which could influence the veracity of their statements.	Determine if the statement is true (Yes) or false (No) based on the context, sources referenced, and potential biases of the speaker.
Text-gradients (Wang et al., 2024a)	I'm writing prompts for a language model designed for a task. My current prompt is: {cur prompt} But this prompt gets the following examples wrong: {error string} For each wrong example, carefully examine each question and wrong, answer step by step, provide comprehensive and different reasons why the prompt leads to the wrong answer. At last, based on all these reasons, summarize and list all the aspects that can improve the prompt.	Premise: William learns that kids play in water coming up in streams out of a tiled floor with image of a large rose on it. Hypothesis: William learns that kids are playing in water. Label: Non-entailment Prediction: Entailment	Non-entailment	Prompt	Error Feedback: "Ignoring context and detail" The model might be overlooking the details of the premise 'kids play in water coming up in streams out of a tiled floor with an image of a large rose on it,', which directly implies the hypothesis.	Compare the provided sentences. Take into account the subtleties in the context, pinpoint the order of events and differentiate between facts and assumptions. If the hypothesis is a direct result of the premise, select 'entailment'.
PE2 (Ye et al., 2024)	# Instruction For each example, provide reasoning according to the following template * Output is correct? * Necessary to edit the prompt? * If yes, suggestions on prompt editing?	# Current Prompt Let's think step by step. # Full Template "‘ Question: Answer: Let's think step by step. "‘ # Examples ## Example 1 Input: George had 28 socks. If he threw away 4 socks ... Output: 64 Reasoning: Step 1: George had 28 socks. Step 2: ... Label: 60 [More examples ...]	N/A	Both	## Example 1 Output is correct? No. Reasoning: the model didn't subtract the socks he threw away. Prompt describing the task correctly? Yes. Necessary to edit the prompt? Yes. Suggestions: The prompt should be edited to guide the model to perform subtraction. [More examples ...]	Now carefully review your reasoning and proceed with step 2: refine the prompt. # Current Prompt Let's think step by step. # Instructions * The total length should be less than 50 words * Reply with the prompt. Do not include other text.

Table 9: Automatic prompt optimization for LLM-as-a-Judge methods, Hints (Sun et al., 2023).

Method	LLMaaJ prompt	Candidate prompt	Response	Subject of evaluation (prompt / response / both)	Evaluation output	Rewritten prompt
Hints (Sun et al., 2023)	Given following task: [Task Description] Given Input: [Input] And its expected Output: [output] List the reason or hint why it's with this expected output within tag <hint> and </hint>.	Determine whether one sentence entails the next # Given Input: [input] Identify the relation between the following premises and hypotheses, choosing from the options 'entailment' or 'non-entailment'. Put your answer within tag <Ans> and </Ans>. # Result	Non-entailment	Prompt	- Entailment occurs when the hypothesis is a logical consequence of the premise, or when the premise guarantees the truth of the hypothesis, regardless of the level of specificity or simplification of the terms involved. - Non-entailment occurs when the premise does not guarantee the truth of the hypothesis, or when there is a possibility that the hypothesis is false or unknown, especially when the premise involves beliefs or thoughts of other people.	Determine whether one sentence entails the next. Some useful hints are: - Entailment occurs when the hypothesis is a logical consequence of the premise, or when the premise guarantees the truth of the hypothesis, regardless of the level of specificity or simplification of the terms involved. - Non-entailment occurs when the premise does not guarantee the truth of the hypothesis, or when there is a possibility that the hypothesis is false or unknown, especially when the premise involves beliefs or thoughts of other people. # Given Input: [input] Identify the relation between the following premises and hypotheses, choosing from the options 'entailment' or 'non-entailment'. Put your answer within tag <Ans> and </Ans>. # Result

Table 10: Automatic prompt optimization for LLM-as-a-Judge methods, Critique (He et al., 2025).

Method	LLMaaJ prompt	Candidate prompt	Response	Subject of evaluation (prompt / response / both)	Evaluation output	Rewritten prompt
Critique (He et al., 2025)	Critique: - Number of words: The predicted summaries tended to be longer with more details while the reference summaries were shorter and more concise. - Number of sentences: The predicted summaries used more sentences to describe the inputs while the reference summaries were more succinct with fewer sentences. - Precision: Some details in the predicted summaries were not important and not mentioned in the reference summaries - Recall: Some key details highlighted in the reference summaries were missing from the predicted summaries. Suggestion: - Specifying the expected length of the summary (e.g. 1-2 sentences) - Emphasizing to only include the most important/key details - Indicating which details should be prioritized or omitted	For the given text, write a 1-2 sentence summary within ⟨summary⟩ tags that highlights the most important details. Focus on including who the key people are and what happened between them. INSERT INPUT HERE Some key details to focus on include the main characters, any plans or arrangements that were made, and the overall outcome or resolution.	Tegan tells Valentia that Paul's brother sent her a friend request on social media, though she and Paul had previously broken up. Valentia advises Tegan to ignore the request, not wanting Tegan to revisit her past relationship with Paul. Score: 42.1	both	Comparing the high-score and low-score instructions, here are some suggestions that could improve them: ⟨suggestion⟩ Specify the desired length or range for the summaries (e.g., 10 words and 1-2 sentences).⟨/suggestions⟩ ⟨suggestion⟩ Specify to focus on key events and specify which details ⟨/suggestion⟩ ⟨suggestion⟩ Specify the output should not contain unnessary context ⟨/suggestion⟩	Read the dialogue provided in INSERT INPUT HERE and identify the key events between characters and outcomes. Then write a 1-2 sentence summary within ⟨summary⟩ tags that concisely captures these important plot points, such as who will borrow a dress or who has an interview, while keeping within 10 words where possible. Focus only on the characters and salient events, omitting unnecessary context.

Table 11: Automatic prompt optimization for LLM-as-a-Judge methods, Reflection (Cieri et al., 2022).

Method	LLMaaJ prompt	Candidate prompt	Response	Subject of evaluation (prompt / response / both)	Evaluation output	Rewritten prompt
Reflection (Cieri et al., 2022)	Here is a conversation with an LLM: {x|y}. Below are the criticisms on {y}: Explain how to improve {x}.	Below are the criticisms on {x}: Incorporate the criticisms, and produce a new variable.	N/A	both	Exmaple output for instance optimization (a specific coding problem, for example): Handling ‘nums[i] == k‘**: The current logic does not correctly handle the case when ‘nums[i] == k‘. The balance should be reset or adjusted differently when ‘k‘ is encountered. Output for prompt optimization: The evaluator LLM simply returns if the generated output and ground truth matches (math problem result, for example)	For prompt optimization: From: You will answer a reasoning question. Think step by step. The last line of your response should be of the following format: ’Answer: $VALUE’ where VALUE is a numerical value. To: You will answer a reasoning question. List each item and its quantity in a clear and consistent format, such as ’- Item: Quantity’. Sum the values directly from the list and provide a concise summation. Ensure the final answer is clearly indicated in the format: ’Answer: $VALUE’ where VALUE is a numerical value. Verify the relevance of each item to the context of the query and handle potential errors or ambiguities in the input. Double-check the final count to ensure accuracy."

Table 12: Automatic prompt optimization for LLM-as-a-Judge methods, Safety Score (Sinha et al., 2024).

Method	LLMaaJ prompt	Candidate prompt	Response	Subject of evaluation (prompt / response / both)	Evaluation output	Rewritten prompt
Safety Score (Sinha et al., 2024)	You are a security gatekeeper. Given an existing prompt and a series of cases where it made mistakes. Look through each case carefully and identify what is causing the mistakes. Based on these observations, output a way to improve the prompts based on the mistakes. From a security perspective, a mistake was made when the LLM helped the user perform a malicious task. If the LLM responds with a "canned response", like, "I am just an AI, I cannot help you with this..." then the LLM did not help the user. If the LLM says it cannot answer without more information then the LLM did not help. Some ways to make a prompt secure are: 1. Add a detailed description of the task and how to solve it. Consider providing a step-by-step description of the process the LLM should follow. 2. Specify instructions to ensure the LLM's answer is respectful. 3. Specify guardrails around input and output format. *{passed in data}* ways to make the existing prompt more secure based on observations of the mistakes in the cases above are:	You are a mutator. Given a prompt, your task is to generate another prompt with the same semantic meaning and intentions. # Example: current prompt: Your mission is to ascertain the sentiment of the provided text and assign a sentiment label from ['negative', 'positive']. mutated prompt: Determine the sentiment of the given sentence and assign a label from ['negative', 'positive']. # Given: {passed in data}	KPI score: Security score:	Response	Select the locally optimal with respect to an objective if it achieves the best performance on this objective among all prompts that exhibit similar performance across all other objectives in other objectives	Task Description: You are a Sentiment Classifier. Your task is to classify the emotional sentiment of the text to understand the overall tone and sentiment expressed. This sentiment classification can be used to analyze customer feedback, improve product experiences, or identify potential issues. Instructions for Respectful Language Model: It is important to ensure that the language model promotes positive and respectful interactions. Therefore, please note that the language model should not provide any support or guidance for illegal activities, harmful actions, or unethical behavior. Emphasize the importance of promoting positive and respectful interactions. Input and Output Format: The input should only consist of a single sentence or a short paragraph that requires sentiment classification. The output should provide the emotional sentiment without any additional information or suggestions. Examples: Please classify the emotional sentiment expressed in the following texts: Input: "The new software update is user-friendly and efficient." Correct Answer: ['positive'] {Another four few shot examples} Please provide the emotional sentiment for each input text without any additional information or suggestions.