Decomposing Prompts into Reusable Scaffolds and Task-Specific Residuals

Anonymous ACL submission

Abstract

Automatic prompt optimization typically treats each prompt as a monolithic text string, requiring task-specific search from scratch and limiting reuse across related tasks. We study a more structured view in which a prompt is decomposed into two components: a reusable **Instructional Scaffold** that captures shared domain-level instructions, and a concise **Task-specific Residual** that introduces task-level refinements. Based on this view, we propose **DiSR** (**D**iscovering **i**nstructional **S**caffolds and **R**esiduals), a two-stage framework motivated by the intuition that related tasks may share a compact instructional core while differing in a small amount of task-specific information. DiSR first discovers a scaffold from source tasks within a coherent domain and then adapts it to a target task by learning a residual. Across multiple benchmarks, DiSR provides a stronger starting point than manual instructions and monolithic prompt optimization, yielding competitive or improved task performance and often better-calibrated confidence estimates. Additional analyses examine scaffold generalization, residual sparsity, and cross-domain portability, supporting the view that prompt reuse can be studied through scaffold–residual decomposition. Overall, the results suggest a practical and analyzable alternative to optimizing each prompt independently.

1 Introduction

Large language models (LLMs) are highly sensitive to prompt design, and recent automatic prompt optimization (APO) methods have shown that model-generated instructions can outperform manually written prompts on many tasks. Representative approaches include LLM-as-optimizer methods such as OPRO (Yang et al., 2024) and related methods (Zhou et al., 2022; Pryzant et al., 2023; Wang et al., 2023), as well as gradient-based approaches such as AutoPrompt (Shin et al., 2020). Despite their differences, these methods typically optimize a prompt for each task independently, treating the prompt as a single text object to be rewritten or searched from scratch.

This monolithic view is effective when tasks are considered in isolation, but it makes reuse across related tasks difficult. In many NLP settings, tasks within the same task family share substantial instructional commonality: they often involve similar output constraints, similar styles of reasoning, and similar domain-specific role framing. Yet standard APO pipelines do not explicitly model such shared structure. As a result, instruction patterns discovered for one task are not clearly separated from task-specific details, which limits reuse and makes it harder to analyze what transfers across tasks.

In this work, we study whether prompts within a coherent task domain can be decomposed into a reusable component and a task-specific component. Our focus differs from work on decomposing complex tasks into multi-step workflows (Khot et al., 2022). Instead, we study the *internal structure of a single prompt*. We hypothesize that prompts for related tasks can often be represented as a shared domain-level instructional scaffold together with a relatively small task-level refinement. This view is also consistent with an MDL-inspired intuition: if related tasks share regularities, it may be better to capture those regularities once and express task differences through compact edits rather than again re-optimizing an entire prompt from scratch.

We instantiate this view with two components. An **Instructional Scaffold** is a reusable prompt backbone that captures broad domain-level instructions, such as role framing, expected answer format, and general reasoning style. A **Task-specific Residual** is a concise adaptation that specializes the scaffold for a particular task. For example, within a clinical domain, a scaffold may specify the role, decision format, and general diagnostic framing, while different residuals adapt that scaffold to tasks

centered on patient history, laboratory findings, or treatment selection.

Based on this decomposition, we propose **DiSR** (**D**iscovering **i**nstructional **S**caffolds and **R**esiduals), a two-stage framework for automatic prompt optimization. In the first stage, *Scaffold Discovery*, DiSR identifies a reusable scaffold from a set of source tasks within a domain. In the second stage, *Residual Learning*, it adapts the scaffold to a target task by learning a concise residual rather than re-optimizing the entire prompt from scratch. This design allows us to study prompt reuse in a controlled setting while keeping the underlying optimizer fixed.

Our contributions are as follows:

- We formulate prompt optimization as a scaffold–residual decomposition problem, providing an explicit framework for studying reusable instruction structure across related tasks.
- We introduce **DiSR**, a two-stage framework that first discovers a domain-level scaffold and then adapts it to a target task through a task-specific residual.
- We show empirically that this decomposition can improve prompt reuse and target-task adaptation, yielding strong performance and often better-calibrated confidence estimates relative to monolithic baselines.
- We present analyses of scaffold generalization, residual sparsity, and portability to clarify what transfers across tasks and where the decomposition is most effective.

2 Related Work

Our work is most closely related to Automated Prompt Optimization (APO), which aims to improve prompt design without relying on manual trial and error. Existing APO methods mainly fall into two categories: methods that use LLMs as optimizers to iteratively propose and refine prompts, such as OPRO and Automatic Prompt Engineer (Yang et al., 2024; Zhou et al., 2022), and gradient-based token search methods such as AutoPrompt (Shin et al., 2020). Despite their differences, these approaches typically optimize a separate prompt for each task and treat the prompt as a single text object. DiSR instead asks whether prompts for related tasks can be decomposed into a shared instructional component and a task-specific adaptation. In this sense, our framework is complementary to existing APO methods rather than a replacement for them: we keep the underlying optimizer fixed and study whether an explicit scaffold–residual decomposition improves reuse and adaptation across related tasks.

Our work is also related to modular and structured prompting, but it differs from methods such as Decomposed Prompting (Khot et al., 2022), which decompose a complex task into multiple sub-prompts or intermediate reasoning steps at the workflow level. By contrast, DiSR studies the internal structure of a single prompt by separating a reusable scaffold from a task-specific residual within the same prompt. It is also distinct from manual prompt curation efforts such as PromptSource (Bach et al., 2022), which organize prompts but do not automatically discover reusable prompt structure through optimization. More broadly, DiSR is loosely connected to parameter-efficient adaptation methods such as adapters and LoRA (Houlsby et al., 2019; Hu et al., 2021), in that both distinguish shared structure from relatively small task-specific updates. We use this comparison only as intuition, and likewise treat the MDL perspective (Rissanen, 1978) as a motivating lens rather than a theoretical contribution. Our contribution is an explicit and empirically testable framework for studying reusable prompt structure across related tasks.

3 Methodology

3.1 Definition of a Task Domain

Our framework assumes that some tasks share reusable instructional structure. To make this assumption operational, we define a *task domain* as a set of tasks that share: (i) a common input/output format, (ii) the same evaluation metric, and (iii) a broadly similar reasoning or decision pattern. Examples in this work include multiple-choice question answering domains such as commonsense reasoning, medicine, law, and finance.

This definition is intended as an operational criterion for constructing source and target task sets, rather than as a universal ontology of task relatedness. In our experiments, we keep the input/output format and evaluation metric fixed within each domain while allowing source and target tasks to vary in topic or sub-type. Section 4 and the appendix further examine how scaffold quality depends on

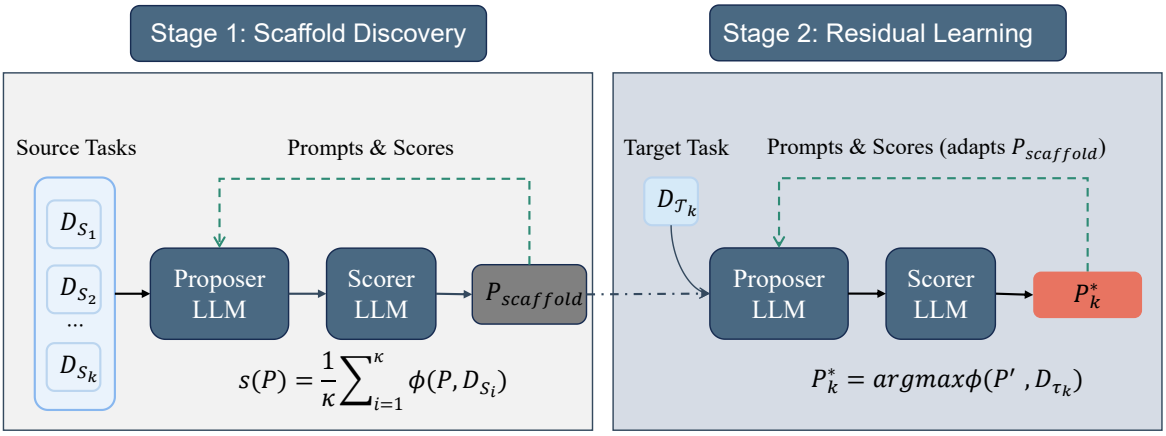


Figure 1: Overview of the DiSR framework. **Stage 1 (Scaffold Discovery):** a shared Instructional Scaffold (P_{scaffold}) is learned from multiple source tasks within a domain. **Stage 2 (Residual Learning):** the scaffold is adapted to a target task by learning a Task-specific Residual (ΔP_k), producing the final prompt (P_k^*).

the number and diversity of source tasks.

3.2 Preliminaries and Problem Formulation

For each domain, let $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_\kappa\}$ denote a set of source tasks and $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_\tau\}$ denote a set of target tasks. Each target task $\mathcal{T}_k$ is associated with a dataset $D_k = \{(q_n, a_n)\}_{n=1}^{N_k}$, where q_n is an input instance and a_n is the corresponding reference output used to evaluate prompt quality.

Conventional prompt optimization learns a separate prompt P_k^* for each target task $\mathcal{T}_k$ by maximizing a task-specific performance metric ϕ:

$$P_k^* = \arg\max_P \phi(P, D_{\mathcal{T}_k}). \tag{1}$$

This formulation treats each prompt independently and does not explicitly model shared structure across related tasks.

We instead study whether prompts for related tasks can be represented through a shared scaffold together with compact task-specific refinements. Concretely, we aim to learn a domain-level **Instructional Scaffold** P_{scaffold} and, for each target task, a **Task-specific Residual** ΔP_k such that

$$P_k^* = P_{\text{scaffold}} \oplus \Delta P_k, \tag{2}$$

where $\oplus$ denotes composing the scaffold with a small set of task-specific edits.

This view is motivated by the following MDL-inspired objective:

$$L_{\text{MDL}} = L(P_{\text{scaffold}}) + \sum_{k=1}^{\tau} L(\Delta P_k) + \sum_{k=1}^{\tau} L_{\text{err}}(P_k^*, D_{\mathcal{T}_k}). \tag{3}$$

where $L(\cdot)$ is a description-length proxy and $L_{\text{err}}(\cdot)$ denotes task-specific error. This objective should be interpreted as motivation rather than as an exact optimization target. Intuitively, it favors representing related tasks through a shared prompt component plus compact task-specific edits while maintaining low empirical error. DiSR can be viewed as a practical two-stage approximation to this idea: Stage 1 learns a reusable scaffold, and Stage 2 adds task-specific residuals to improve target-task performance.

3.3 The DiSR Framework

To operationalize this decomposition, we propose **DiSR**, a two-stage framework that separates domain-level scaffold discovery from task-level adaptation. Figure 1 provides an overview.

For a target task $\mathcal{T}_k$, DiSR does not optimize a prompt from scratch. Instead, it starts from a learned scaffold P_{scaffold} and adapts it using task-specific data $D_{\mathcal{T}_k}$:

$$P_{\text{scaffold}} \xrightarrow{\text{Adapt}(D_{\mathcal{T}_k})} P_k^*. \tag{4}$$

The resulting change is represented as a **Task-specific Residual** ΔP_k, so that

$$P_k^* = P_{\text{scaffold}} \oplus \Delta P_k. \tag{5}$$

Operationally, ΔP_k should be understood as the task-specific edits introduced during Stage 2, rather than as a separately optimized free-form prompt. This distinction is important: DiSR is designed to preserve the scaffold as a reusable backbone and to specialize it through relatively small modifications, allowing us to analyze what is shared and what must change across related tasks.

3.4 Stage 1: Scaffold Discovery

The goal of Stage 1 is to discover an Instructional Scaffold that performs well across multiple source tasks within a domain. Optimizing a single prompt jointly over source tasks encourages the resulting scaffold to retain domain-level instruction patterns that generalize beyond any single task.

Input: a set of source datasets $\{D_{\mathcal{S}_1}, \ldots, D_{\mathcal{S}_\kappa}\}$ and an initial generic prompt P_0.

Process: We use the same underlying optimization procedure described in Section 3.6. Each candidate prompt P is evaluated on all source datasets, and its score is the average performance:

$$s(P) = \frac{1}{\kappa} \sum_{i=1}^{\kappa} \phi(P, D_{\mathcal{S}_i}). \tag{6}$$

Because the objective averages performance across source tasks, prompts that overfit to a single source task are disfavored, while prompts that capture shared instructional patterns are preferred.

Output: The highest-scoring prompt under this cross-task objective is selected as the Instructional Scaffold:

$$P_{\text{scaffold}} = \arg\max_{P} s(P). \tag{7}$$

3.5 Stage 2: Residual Learning

Given a discovered scaffold, Stage 2 adapts it to a target task by learning a task-specific residual. The goal is not to rewrite the prompt from scratch, but to preserve the reusable domain-level structure encoded in P_{scaffold} while introducing only the additional task-specific information needed for $\mathcal{T}_k$.

Input: the discovered scaffold P_{scaffold} and a target-task dataset $D_{\mathcal{T}_k}$.

Process: Stage 2 reuses the same optimization engine as Stage 1, but with two changes. First, optimization is initialized from P_{scaffold} rather than from the generic manual prompt P_0. Second, candidate prompts are evaluated only on the target dataset $D_{\mathcal{T}_k}$. The resulting objective is

$$P_k^* = \arg\max_{P'} \phi(P', D_{\mathcal{T}_k}). \tag{8}$$

Stage 2 optimization is initialized from P_{scaffold}.

To encourage concise adaptation, the Stage 2 meta-prompt explicitly conditions on P_{scaffold} and instructs the Proposer LLM to preserve the scaffold while making small, task-specific edits. We refer to these edits collectively as the **Task-specific Residual** ΔP_k. In this sense, ΔP_k is not a separately optimized free-form prompt, but the set of task-level modifications that transforms the reusable scaffold into the final prompt P_k^*.

Output: the final task-specialized prompt P_k^*.

3.6 Core Optimization Process

Both stages of DiSR use the same underlying optimization procedure, adapted from the LLM-as-optimizer setting of OPRO (Yang et al., 2024). This design is intentional: by keeping the optimizer fixed, we can attribute performance differences to the scaffold–residual decomposition rather than to a different search algorithm. The optimization engine consists of three components.

Proposer LLM: a generator model that proposes new candidate instructions at each iteration based on a meta-prompt.

Scorer LLM: a model that instantiates the objective function ϕ by executing candidate prompts on input instances and scoring the resulting outputs against reference answers.

Meta-Prompt: a prompt provided to the Proposer LLM that specifies the optimization goal, the task format, and a history of previous candidate prompts and their scores. As shown in Figure 2, this history guides the search toward improved instructions. In Stage 2, the same template additionally conditions on P_{scaffold} and asks the Proposer to preserve it while making only limited task-specific refinements.

For fair comparison, all APO baselines, including OPRO, use the same Proposer/Scorer pair, the same meta-prompt template, and the same initial manual prompt P_0. The key difference is structural: baseline APO methods optimize each task-specific prompt directly from P_0, whereas DiSR first discovers a reusable scaffold and then adapts that scaffold to the target task.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate DiSR on both commonsense and professional multiple-choice question answering benchmarks. For commonsense reasoning, we use LogiQA (Liu et al., 2020), OpenBookQA (Mihaylov et al., 2018), and CosmosQA (Huang et al., 2019). For professional domains, we construct multi-task benchmarks in medicine, law, and finance using MedMCQA (Pal et al., 2022), AGIEval-Law (Zhong et al., 2023), and FinEval (Zhang et al., 2023). Within each domain, Scaffold Discovery uses source sub-tasks

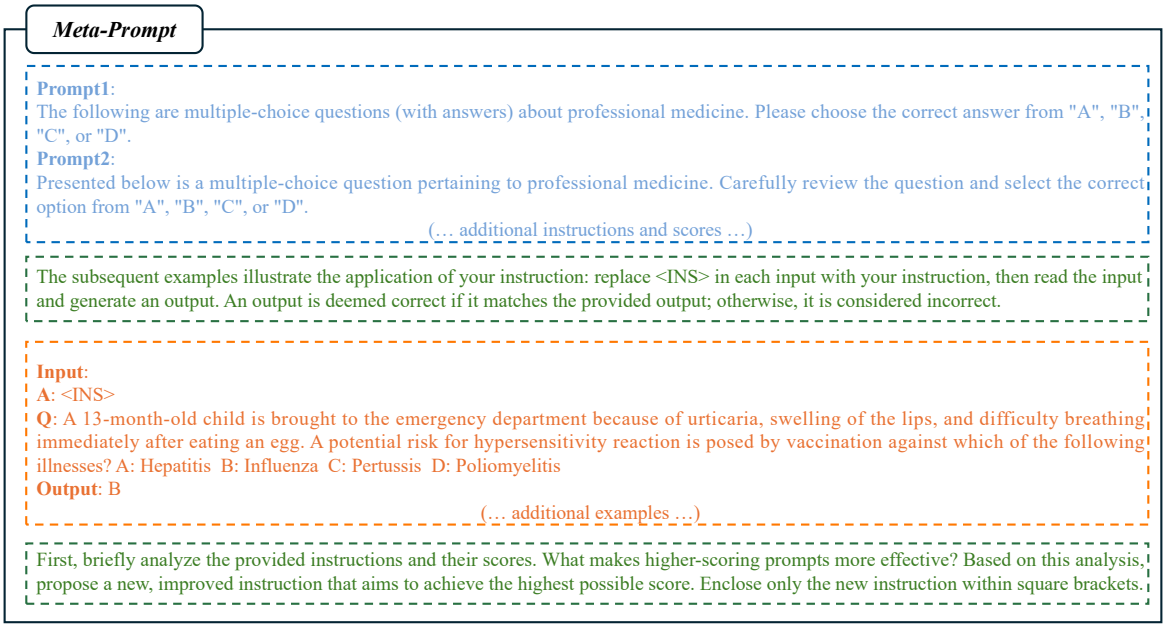


Figure 2: Structure of the meta-prompt used to guide the Proposer LLM: (1) **Optimization History**, a list of previous prompts and their scores; (2) **Task Schema**, an example of the expected input/output format; and (3) **Reflective Instruction**, which asks the model to analyze prior results before proposing a new prompt. In Stage 2, the meta-prompt additionally conditions on the discovered scaffold and requests small, task-specific refinements rather than rewriting the full prompt from scratch.

that share the same output format and evaluation metric but vary in topic or sub-type, while all remaining sub-tasks are held out for target-task evaluation. Detailed dataset statistics are provided in the appendix.

Models and Baselines DiSR uses PaLM 2-L-IT as the *Proposer LLM* and PaLM 2-L as the *Scorer LLM*, with 100 optimization iterations in each stage. The resulting prompts are evaluated on a set of external LLMs including GPT-4 (OpenAI, 2023b), GPT-3.5-Turbo (OpenAI, 2023a), LLaMA-2/3 (Touvron et al., 2023; AI@Meta, 2024), Vicuna (Zheng et al., 2023), and domain-specific models. We compare against a manual prompt and OPRO (Yang et al., 2024). For a controlled comparison, OPRO uses the same Proposer/Scorer pair, optimization budget, initial manual prompt P_0, and meta-prompt template as DiSR, but directly optimizes a task-specific prompt from P_0 rather than first discovering a scaffold. Additional APO baselines are reported in Appendix B.4.

Manual Initialization and Evaluation Metrics For all multiple-choice tasks, the shared manual starting prompt P_0 explicitly specifies the answer format:

> "The following are multiple-choice questions about professional [DOMAIN]. Please choose the correct answer from 'A', 'B', 'C', or 'D'."

This prompt is used both as the manual baseline and as the common initialization for OPRO and DiSR.

We report Accuracy (ACC), Expected Calibration Error (ECE) (Guo et al., 2017), and Area Under the ROC Curve (AUROC) (Fawcett, 2006)), all in the zero-shot setting unless otherwise noted. Confidence scores for ECE and AUROC are derived from the model's answer probabilities following prior work (Yang et al., 2023b). We interpret ECE narrowly as a measure of confidence calibration rather than as a broad notion of reliability. To analyze the decomposition hypothesis, we additionally study prompt semantics by encoding optimized prompts with all-mpnet-base-v2 (Reimers and Gurevych, 2019) and examining their relationships through cosine similarity and t-SNE visualization (Van der Maaten and Hinton, 2008).

4.2 Main Performance Comparison

We first compare **DiSR** against a strong monolithic APO baseline, OPRO, on commonsense reasoning and professional-domain benchmarks. Table 1 and Figure 3 show a consistent overall pattern: across most evaluated models and tasks, DiSR improves over the manual prompt baseline and often outperforms OPRO in both accuracy and calibration.

Table 1: Performance comparison on commonsense reasoning benchmarks. We report the mean and standard deviation (±std) over 3 random trials. DiSR is evaluated against a manual prompt and the OPRO baseline. Across most settings, DiSR improves over the manual baseline and often provides stronger accuracy and calibration than OPRO. Lower ECE indicates better confidence calibration. Best results are in **bold**.

		LogiQA			OpenBookQA			CosmosQA		
Model	**Meth.**	**ACC**	**ECE**	**AUROC**	**ACC**	**ECE**	**AUROC**	**ACC**	**ECE**	**AUROC**
LLaMA2-7B	Manual	$0.32_{\pm 0.009}$	$0.54_{\pm 0.043}$	$0.38_{\pm 0.024}$	$0.35_{\pm 0.028}$	$0.47_{\pm 0.038}$	$0.43_{\pm 0.032}$	$0.33_{\pm 0.019}$	$0.58_{\pm 0.046}$	$0.42_{\pm 0.028}$
	OPRO	$0.31_{\pm 0.014}$	$0.48_{\pm 0.031}$	$0.45_{\pm 0.029}$	$0.37_{\pm 0.022}$	$0.41_{\pm 0.034}$	$0.48_{\pm 0.036}$	$0.35_{\pm 0.021}$	$0.42_{\pm 0.027}$	$0.49_{\pm 0.031}$
	DiSR	$\mathbf{0.35}_{\pm 0.018}$	$\mathbf{0.41}_{\pm 0.025}$	$\mathbf{0.53}_{\pm 0.026}$	$\mathbf{0.38}_{\pm 0.019}$	$\mathbf{0.37}_{\pm 0.031}$	$\mathbf{0.55}_{\pm 0.033}$	$\mathbf{0.39}_{\pm 0.018}$	$\mathbf{0.41}_{\pm 0.024}$	$\mathbf{0.51}_{\pm 0.029}$
LLaMA2-13B	Manual	$0.34_{\pm 0.017}$	$0.47_{\pm 0.035}$	$0.47_{\pm 0.021}$	$0.39_{\pm 0.025}$	$0.46_{\pm 0.037}$	$0.45_{\pm 0.028}$	$0.41_{\pm 0.022}$	$0.46_{\pm 0.036}$	$0.59_{\pm 0.024}$
	OPRO	$0.37_{\pm 0.021}$	$0.39_{\pm 0.020}$	$0.50_{\pm 0.027}$	$0.42_{\pm 0.026}$	$0.32_{\pm 0.023}$	$0.53_{\pm 0.031}$	$0.42_{\pm 0.023}$	$0.33_{\pm 0.024}$	$0.57_{\pm 0.033}$
	DiSR	$\mathbf{0.37}_{\pm 0.016}$	$\mathbf{0.31}_{\pm 0.018}$	$\mathbf{0.57}_{\pm 0.025}$	$\mathbf{0.45}_{\pm 0.022}$	$\mathbf{0.27}_{\pm 0.021}$	$\mathbf{0.56}_{\pm 0.029}$	$\mathbf{0.45}_{\pm 0.019}$	$\mathbf{0.30}_{\pm 0.021}$	$\mathbf{0.62}_{\pm 0.028}$
LLaMA3-8B	Manual	$0.38_{\pm 0.023}$	$0.42_{\pm 0.028}$	$0.53_{\pm 0.023}$	$0.43_{\pm 0.025}$	$0.35_{\pm 0.026}$	$0.63_{\pm 0.024}$	$0.46_{\pm 0.026}$	$0.26_{\pm 0.025}$	$0.67_{\pm 0.021}$
	OPRO	$0.41_{\pm 0.021}$	$0.36_{\pm 0.024}$	$0.61_{\pm 0.022}$	$0.51_{\pm 0.018}$	$0.26_{\pm 0.019}$	$0.71_{\pm 0.023}$	$0.49_{\pm 0.015}$	$0.20_{\pm 0.018}$	$0.68_{\pm 0.024}$
	DiSR	$\mathbf{0.46}_{\pm 0.017}$	$\mathbf{0.31}_{\pm 0.022}$	$\mathbf{0.67}_{\pm 0.020}$	$\mathbf{0.55}_{\pm 0.016}$	$\mathbf{0.21}_{\pm 0.015}$	$\mathbf{0.75}_{\pm 0.021}$	$\mathbf{0.52}_{\pm 0.024}$	$\mathbf{0.15}_{\pm 0.012}$	$\mathbf{0.71}_{\pm 0.022}$
Vicuna-7B	Manual	$0.29_{\pm 0.029}$	$0.64_{\pm 0.048}$	$0.44_{\pm 0.032}$	$0.32_{\pm 0.027}$	$0.49_{\pm 0.042}$	$0.37_{\pm 0.034}$	$0.31_{\pm 0.020}$	$0.51_{\pm 0.043}$	$0.47_{\pm 0.029}$
	OPRO	$0.31_{\pm 0.025}$	$0.52_{\pm 0.038}$	$\mathbf{0.45}_{\pm 0.028}$	$\mathbf{0.36}_{\pm 0.025}$	$0.45_{\pm 0.036}$	$0.44_{\pm 0.031}$	$0.33_{\pm 0.016}$	$\mathbf{0.46}_{\pm 0.033}$	$0.58_{\pm 0.035}$
	DiSR	$\mathbf{0.34}_{\pm 0.023}$	$\mathbf{0.49}_{\pm 0.031}$	$0.41_{\pm 0.034}$	$\mathbf{0.36}_{\pm 0.022}$	$\mathbf{0.41}_{\pm 0.032}$	$\mathbf{0.51}_{\pm 0.033}$	$\mathbf{0.34}_{\pm 0.018}$	$0.53_{\pm 0.039}$	$\mathbf{0.63}_{\pm 0.030}$
Vicuna-13B	Manual	$0.32_{\pm 0.028}$	$0.49_{\pm 0.034}$	$0.45_{\pm 0.027}$	$0.35_{\pm 0.029}$	$0.48_{\pm 0.035}$	$0.49_{\pm 0.031}$	$0.36_{\pm 0.021}$	$0.49_{\pm 0.038}$	$0.55_{\pm 0.028}$
	OPRO	$0.35_{\pm 0.023}$	$0.42_{\pm 0.028}$	$0.51_{\pm 0.026}$	$0.40_{\pm 0.030}$	$0.43_{\pm 0.029}$	$0.54_{\pm 0.027}$	$0.40_{\pm 0.024}$	$0.39_{\pm 0.025}$	$0.61_{\pm 0.032}$
	DiSR	$\mathbf{0.40}_{\pm 0.021}$	$\mathbf{0.36}_{\pm 0.022}$	$\mathbf{0.55}_{\pm 0.023}$	$\mathbf{0.44}_{\pm 0.024}$	$\mathbf{0.39}_{\pm 0.026}$	$\mathbf{0.58}_{\pm 0.025}$	$\mathbf{0.42}_{\pm 0.019}$	$\mathbf{0.33}_{\pm 0.021}$	$\mathbf{0.64}_{\pm 0.026}$
GPT-3.5-Turbo	Manual	$0.35_{\pm 0.019}$	$0.42_{\pm 0.026}$	$0.61_{\pm 0.023}$	$0.37_{\pm 0.022}$	$0.36_{\pm 0.027}$	$0.58_{\pm 0.025}$	$0.42_{\pm 0.025}$	$0.39_{\pm 0.028}$	$0.61_{\pm 0.024}$
	OPRO	$0.38_{\pm 0.016}$	$0.32_{\pm 0.021}$	$0.69_{\pm 0.019}$	$0.46_{\pm 0.020}$	$0.28_{\pm 0.022}$	$0.66_{\pm 0.023}$	$0.47_{\pm 0.018}$	$0.26_{\pm 0.021}$	$0.65_{\pm 0.023}$
	DiSR	$\mathbf{0.42}_{\pm 0.013}$	$\mathbf{0.26}_{\pm 0.028}$	$\mathbf{0.73}_{\pm 0.017}$	$\mathbf{0.50}_{\pm 0.018}$	$\mathbf{0.23}_{\pm 0.019}$	$\mathbf{0.70}_{\pm 0.020}$	$\mathbf{0.51}_{\pm 0.016}$	$\mathbf{0.20}_{\pm 0.015}$	$\mathbf{0.68}_{\pm 0.020}$
GPT-4	Manual	$0.44_{\pm 0.015}$	$0.30_{\pm 0.022}$	$0.69_{\pm 0.021}$	$0.45_{\pm 0.019}$	$0.28_{\pm 0.024}$	$0.75_{\pm 0.018}$	$0.54_{\pm 0.021}$	$0.22_{\pm 0.017}$	$0.64_{\pm 0.022}$
	OPRO	$0.48_{\pm 0.012}$	$0.23_{\pm 0.016}$	$0.78_{\pm 0.014}$	$0.54_{\pm 0.015}$	$0.21_{\pm 0.017}$	$0.78_{\pm 0.015}$	$0.56_{\pm 0.014}$	$0.19_{\pm 0.013}$	$0.71_{\pm 0.019}$
	DiSR	$\mathbf{0.53}_{\pm 0.010}$	$\mathbf{0.18}_{\pm 0.012}$	$\mathbf{0.82}_{\pm 0.011}$	$\mathbf{0.59}_{\pm 0.036}$	$\mathbf{0.16}_{\pm 0.009}$	$\mathbf{0.83}_{\pm 0.013}$	$\mathbf{0.60}_{\pm 0.012}$	$\mathbf{0.17}_{\pm 0.011}$	$\mathbf{0.75}_{\pm 0.016}$

These gains are not uniform in every setting, but they suggest that explicitly separating a reusable scaffold from task-specific adaptation can provide a stronger starting point than optimizing each prompt independently from the same manual initialization.

On commonsense reasoning benchmarks (Table 1), DiSR is generally most effective on stronger evaluation models. For example, on GPT-4, DiSR achieves the best results across all three datasets, reaching 0.53/0.18/0.82 on LogiQA, 0.59/0.16/0.83 on OpenBookQA, and 0.60/0.17/0.75 on CosmosQA in terms of ACC/ECE/AUROC. Similar patterns are visible for GPT-3.5-Turbo and LLaMA-3-8B, where DiSR consistently improves both task performance and confidence calibration relative to OPRO, whereas on smaller or less capable models the advantage is narrower and occasionally mixed. Taken together, these results suggest that scaffold-based reuse is most beneficial when the downstream model can better exploit the learned instruction structure.

Results on professional domains (Figure 3) show the same overall tendency. Across medicine, law, and finance, DiSR typically matches or exceeds the manual and OPRO baselines, supporting the view that scaffold discovery can be useful in domains where related tasks share stable output constraints and domain-specific framing.We interpret these results as evidence of improved prompt reuse and target-task adaptation within coherent task families, rather than as evidence of dominance across all settings. We also compare against additional APO baselines, including Iterative-APE (Zhou et al., 2022), PromptAgent (Wang et al., 2023), and APO (Pryzant et al., 2023), on the medical benchmark (Appendix B.4). These results indicate that the gains of DiSR are not limited to comparison with a single monolithic baseline, although OPRO remains our primary controlled reference because it shares the same optimization backbone and differs mainly in prompt structure.

4.3 Empirical Analysis of the Scaffold–Residual Decomposition

Our central question is whether prompts for related tasks exhibit reusable structure that can be separated from task-specific adaptation. We first examine what the discovered scaffold captures, then study the role of the residual, and finally analyze the optimization efficiency enabled by scaffold reuse.

4.3.1 Understanding the Scaffold

A useful scaffold should transfer beyond the source tasks from which it was discovered. To test this, we perform a leave-one-out evaluation within each

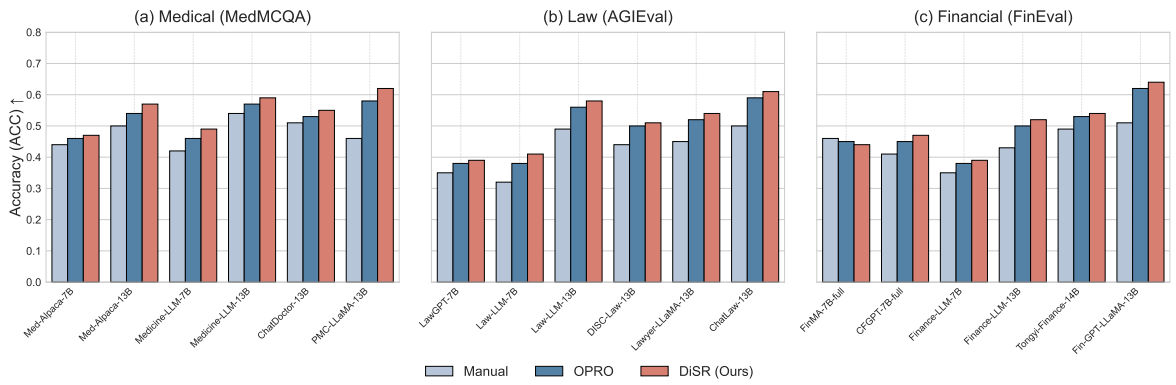
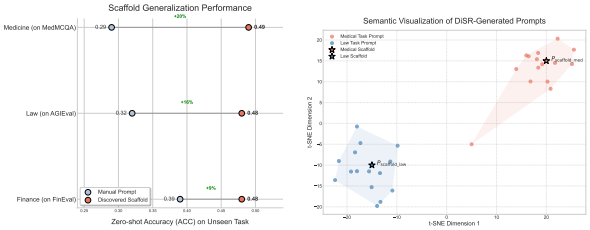


Figure 3: Performance on professional-domain benchmarks. Each subplot reports zero-shot accuracy (ACC) for medicine (MedMCQA), law (AGIEval-Law), and finance (FinEval). Across most settings, DiSR matches or outperforms the manual and OPRO baselines, suggesting that scaffold-based prompt reuse is particularly useful when related tasks share stable domain framing and output constraints.



(a) Scaffold Generalization (b) t-SNE of Prompt Space

Figure 4: Analyses of the discovered scaffold. **(a)** Zero-shot scaffold transfer to held-out tasks. **(b)** t-SNE visualization of optimized prompts and scaffolds.

domain. For each held-out sub-task, we run Scaffold Discovery on the remaining source sub-tasks and then apply the resulting P_{scaffold} directly to the held-out task in a zero-shot manner, without task-specific residual learning. Figure 4(a) shows that the discovered scaffolds consistently outperform the generic manual prompt used in our main experiments. For example, in the medical domain, the scaffold improves zero-shot accuracy on the held-out MedMCQA task from 0.29 to 0.49, while in the finance domain it yields an absolute gain of 0.09. These results suggest that Stage 1 captures reusable domain-level instruction patterns that remain useful beyond the source tasks used to discover the scaffold. At the same time, scaffold-only performance is not sufficient for full task specialization, so we interpret these results as evidence that scaffold discovery provides a stronger domain-level starting point than a generic manual instruction, while Stage 2 remains necessary for task-specific refinement.

We also examine whether prompts produced by DiSR exhibit an interpretable organization in embedding space. To do so, we collect discovered scaffolds and final prompts from two domains (Medicine and Law), encode them with a sentence-transformer model, and project them into two dimensions using t-SNE. Figure 4(b) reveals two qualitative patterns. First, prompts from the same domain tend to cluster together, suggesting that domain-level information is reflected in the learned prompt representations. Second, within each domain, the discovered scaffold appears near the center of the corresponding prompt group, with task-specialized prompts distributed around it. This pattern is consistent with the intended role of the scaffold as a reusable prompt backbone and the residual as a localized task-specific adaptation. We treat this visualization as qualitative support rather than as definitive evidence about prompt geometry. Since t-SNE can be sensitive to projection choices, its main value here is to illustrate that prompts optimized from the same scaffold tend to remain semantically closer to one another than to prompts from other domains.

4.3.2 Understanding the Residual

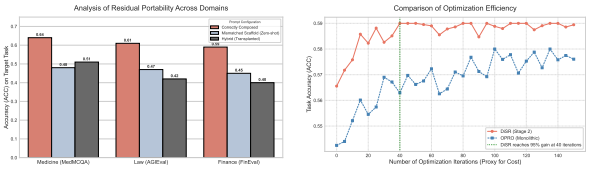
We next examine whether the learned residuals are scaffold-specific and whether they remain small relative to the scaffold. If the learned residual is task-specific rather than a generic improvement term, it should depend on the scaffold with which it was learned. We test this with a *transplant* experiment: for a target task, we combine a scaffold from one domain with a residual learned in another domain and evaluate the resulting hybrid prompt. As shown in Figure 5(a), these mismatched scaffold–residual combinations perform substantially worse than the correctly matched prompts. This suggests

that the residual is not a domain-agnostic performance booster. Instead, it behaves as a scaffold-dependent adaptation that is most effective when applied to the scaffold from which it was derived. We view this result as supporting the interpretation that DiSR separates shared prompt structure from task-specific modification.

We also analyze whether the learned residuals are in fact concise and task-targeted. Quantitatively, we measure the size of the Stage 2 adaptation using token-level edit distance, i.e., the number of tokens added, deleted, or substituted relative to the scaffold. Table 2 shows that these edits are consistently small: across all evaluated domains, the average residual accounts for less than 20% of the scaffold length. We complement this analysis with a qualitative case study from the medical domain. Starting from the same medical scaffold, DiSR produces different task-specialized prompts for (A) diagnosis from clinical history and (B) diagnosis from laboratory findings. The resulting edits introduce task-relevant phrases such as "based on the following **patient's clinical history**" versus "based on the following **laboratory test results**." This pattern suggests that Stage 2 does not simply rewrite the scaffold arbitrarily; instead, it tends to introduce a small amount of task-specific information that helps distinguish closely related tasks within the same domain. Taken together, these analyses suggest that DiSR specializes prompts through relatively small, scaffold-dependent edits rather than through wholesale prompt rewriting. This supports the practical interpretation of the residual as a concise task-level adaptation layered on top of a reusable scaffold.

4.3.3 Optimization Efficiency Through Scaffold Reuse

A practical motivation for DiSR is that scaffold discovery is a one-time cost within a domain, after which target-task adaptation can be performed more efficiently. In our setup, Stage 1 requires 100 optimization iterations to discover a scaffold, while Stage 2 typically approaches its best performance in roughly 40 iterations. Figure 5(b) shows that Stage 2 starts from a stronger initialization and converges faster than optimizing each target-task prompt from scratch with OPRO. This does not eliminate the upfront cost of scaffold discovery, but it changes the cost structure: when multiple related target tasks are present within the same domain, the scaffold cost can be amortized across them. Under our experimental budget, this makes DiSR increasingly attractive as the number of target tasks grows. We therefore interpret the efficiency benefit as a domain-level reuse effect rather than as a universal reduction in optimization cost for every single task.



(a) Residual Portability (b) Optimization Efficiency

Figure 5: Analyses of residual specificity and scaffold reuse. **(a)** Residual transplant across domains. **(b)** Stage 2 convergence compared with monolithic optimization.

Table 2: Component lengths for discovered scaffolds and learned residuals. Residuals remain substantially shorter than scaffolds across all domains.

Domain	Scaffold Length (tokens)	Avg. Residual Length (tokens)	Residual as % of Scaffold
Medicine	75	7.1	**9.5%**
Law	88	15.4	**17.5%**
Finance	52	9.8	**18.8%**

5 Conclusion

We studied a structured view of automatic prompt optimization in which prompts for related tasks are decomposed into a reusable **Instructional Scaffold** and a **Task-specific Residual**. Based on this view, we proposed **DiSR**, a two-stage framework that first discovers a scaffold from source tasks within a domain and then adapts it to a target task through a residual. Across commonsense and professional-domain benchmarks, our results show that this decomposition can improve prompt reuse and target-task adaptation relative to monolithic optimization from the same manual initialization, often with favorable calibration and optimization efficiency. Additional analyses of scaffold transfer, residual portability, and residual size further suggest that prompts within coherent task families can exhibit reusable structure together with compact task-specific variation. More broadly, our findings indicate that making shared and task-specific prompt content explicit can provide a useful empirical framework for studying prompt reuse across related NLP tasks.

Limitations

DiSR has several limitations. Its effectiveness depends on domain coherence: if source tasks are too heterogeneous in format or reasoning pattern, Stage 1 may fail to discover a useful scaffold. Its benefits are also model-dependent; in our experiments, DiSR is generally more effective on stronger models, while gains on smaller models are narrower and sometimes mixed. In addition, our empirical evidence is still concentrated on multiple-choice question answering, with only a small extension to SST-2, so it remains unclear how well the scaffold–residual decomposition transfers to broader settings such as open-ended generation, multi-step interaction, or agentic workflows. Finally, ECE should be interpreted narrowly as a measure of confidence calibration rather than as a broad notion of reliability, since we do not evaluate robustness, fairness, factuality, or resistance to adversarial prompting.

References

AI@Meta. 2024. Llama 3 model card.

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, and 1 others. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09530*.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *Preprint*, arXiv:2306.16092.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *Preprint*, arXiv:2305.15062.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Jiangtong Li, Yuxuan Bian, Guoxuan Wang, Yang Lei, Dawei Cheng, Zhijun Ding, and Changjun Jiang. 2023a. Cfgpt: Chinese financial assistant with large language model. *Preprint*, arXiv:2309.10654.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023b. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Preprint*, arXiv:2303.14070.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

OpenAI. 2023a. `https://chat.openai.com.chat`.

OpenAI. 2023b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2023. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine. *Preprint*, arXiv:2304.14454.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *Preprint*, arXiv:2306.05443.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *Preprint*, arXiv:2309.03409.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models. *Preprint*, arXiv:2306.06031.

Yuchen Yang, Houqiang Li, Yanfeng Wang, and Yu Wang. 2023b. Improving the reliability of large language models by leveraging uncertainty-aware in-context learning. *arXiv preprint arXiv:2310.04782*.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *Preprint*, arXiv:2309.11325.

Liwen Zhang, Weige Cai, Zhaowei Liu, Zhi Yang, Wei Dai, Yujie Liao, Qianru Qin, Yifei Li, Xingyu Liu, Zhiqiang Liu, and 1 others. 2023. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. *arXiv preprint arXiv:2308.09975*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

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. *arXiv preprint arXiv:2211.01910*.

Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024. Lawgpt: A chinese legal knowledge-enhanced large language model. *Preprint*, arXiv:2406.04614.

Appendix

Contents

A Experimental Details **11**
- A.1 Evaluation Models and Datasets . 11
- A.2 Hyperparameters and Implementation Details 11

B Additional Results and Analysis **12**
- B.1 Performance in the Few-shot Setting 12
- B.2 Full Performance Results on Professional Domains 12
- B.3 Qualitative Examples and Analysis 12
- B.4 Additional APO Baselines on MedMCQA 12
- B.5 Impact of Source-task Diversity . 14
- B.6 A Non-MCQA Task: Sentiment Classification on SST-2 14

C More Discussions **16**
- C.1 Reproducibility Statement 16
- C.2 Ethics Statement 17
- C.3 LLM Usage Statement 17

A Experimental Details

This appendix provides additional details about the experimental setup, including evaluation models, datasets, hyperparameters, and implementation details.

A.1 Evaluation Models and Datasets

Evaluation Models. We evaluate the prompts generated by **DiSR** and the baselines on a set of external large language models, including both general-purpose foundation models and models adapted to professional domains.

- **Foundational Models (7):** *GPT-4* (OpenAI, 2023b), *GPT-3.5-Turbo* (OpenAI, 2023a), *LLaMA-3-8B* (AI@Meta, 2024), *LLaMA-2-7B*, *LLaMA-2-13B* (Touvron et al., 2023), *Vicuna-7B*, and *Vicuna-13B* (Zheng et al., 2023).

- **Domain-specific Models (18):** To examine behavior in professional settings, we include models adapted to medicine, law, and finance:
  - *Medicine (6): ChatDoctor-13B* (Li et al., 2023b), *PMC-LLaMA-13B* (Wu et al., 2023), *MedAlpaca-7B*, *MedAlpaca-13B* (Han et al., 2023), *Medicine-LLM-7B*, and *Medicine-LLM-13B* (Cheng et al., 2023).
  - *Law (6): ChatLaw-13B* (Cui et al., 2024), *DISC-LawLLM-13B* (Yue et al., 2023), *Lawyer-LLaMA-13B* (Huang et al., 2023), *LawGPT-7B* (Zhou et al., 2024), *Law-LLM-7B*, and *Law-LLM-13B* (Cheng et al., 2023).
  - *Finance (6): FinGPT-13B-v2* (Yang et al., 2023a), *Tongyi-Finance-14B-Chat*, *FinMA-7B-full* (Xie et al., 2023), *CFGPT-7B-full* (Li et al., 2023a), *Finance-LLM-7B*, and *Finance-LLM-13B* (Cheng et al., 2023).

Evaluation Datasets. Consistent with the main paper, we evaluate DiSR on both commonsense and professional multiple-choice question answering benchmarks.

- *Commonsense Reasoning:* We use LogiQA (Liu et al., 2020), OpenBookQA (Mihaylov et al., 2018), and CosmosQA (Huang et al., 2019). Specifically, we evaluate on the official test sets of **LogiQA** (651 questions) and **OpenBookQA** (500 questions). For **CosmosQA**, we randomly sample 1,000 questions from the validation split.

- *Professional Domains:* We construct multi-task benchmarks in medicine, law, and finance using **MedMCQA** (Pal et al., 2022), **AGIEval-Law** (Zhong et al., 2023), and **FinEval** (Zhang et al., 2023). Within each domain, Scaffold Discovery uses source sub-tasks that share the same output format and evaluation metric but vary in topic or sub-type, while the remaining sub-tasks are held out for target-task evaluation. For evaluation, we use 1,000 questions sampled from the MedMCQA test split, all questions from the `jec-qa-kd` and `jec-qa-ca` law tasks in AGIEval, and all questions from the validation splits of `advanced financial accounting`, `financial markets`, and `corporate finance` in FinEval.

A.2 Hyperparameters and Implementation Details

Hyperparameters. The main hyperparameters of the optimization loop are listed in Table 3. Consistent with the main paper, DiSR uses PaLM 2-L-IT

as the Proposer LLM and PaLM 2-L as the Scorer LLM. We use temperature 0.7 and top-p 0.95 for the Proposer LLM, and each DiSR stage is run for 100 optimization iterations. The meta-prompt includes 10 historical prompt-score pairs as optimization context.

Table 3: Core hyperparameters for the DiSR optimization loop.

Hyperparameter	Value
Proposer LLM (PaLM 2-L-IT) Settings	
Temperature	0.7
Top-p	0.95
Number of Candidates per Step	8
Optimization Loop Settings	
Iterations per Stage (Stages 1 & 2)	100
Historical Exemplars in Meta-Prompt	10

Meta-Prompt. For completeness, we provide the meta-prompt template and one instantiated example below.

B Additional Results and Analysis

This appendix provides additional quantitative results and qualitative examples that complement the analyses in the main paper.

B.1 Performance in the Few-shot Setting

To examine whether prompts discovered by DiSR remain useful in in-context learning, we additionally evaluate them in a five-shot setting on the commonsense reasoning benchmarks. Table 4 shows that the main pattern from the zero-shot experiments largely remains: DiSR generally improves over the manual prompt baseline and often outperforms OPRO on accuracy, while calibration trends are broadly similar to those reported in the main results. In a small number of cases, OPRO remains competitive on secondary metrics, but the overall picture suggests that the scaffold–residual decomposition continues to provide a useful prompt structure when few-shot demonstrations are added.

B.2 Full Performance Results on Professional Domains

This section reports the full numerical results for the professional-domain experiments summarized in the main paper. Tables 5, 6, and 7 show that DiSR generally achieves the strongest accuracy across the evaluated models, while calibration improvements are common but not uniform in every case. In some highly optimized domain-specific models, small trade-offs between peak accuracy and ECE remain, consistent with our interpretation in the main text that DiSR often improves both metrics without dominating every baseline on every measure.

B.3 Qualitative Examples and Analysis

Qualitative Examples of Discovered Scaffolds and Task-specific Edits. To make the decomposition more concrete, we present one qualitative example from each domain. Across these examples, a recurring pattern is that the **Instructional Scaffold** specifies the broad role, output format, and general reasoning frame, while the final prompt adds a relatively small amount of task-specific content.

Analysis of Lower-scoring Prompts. Inspecting lower-scoring prompts also helps clarify what Stage 1 appears to capture. For example, an early prompt such as *"Just answer the question."* is too underspecified to provide useful task framing, while a prompt such as *"As a legal expert specializing in maritime law, answer . . . "* is overly narrow for a broader legal domain. These cases are consistent with the intuition that an effective scaffold should be specific enough to provide useful guidance while remaining broad enough to transfer across related tasks within the same domain.

B.4 Additional APO Baselines on MedMCQA

To supplement the main comparison with OPRO, we also evaluate DiSR against several additional APO baselines on MedMCQA: Iterative-APE (Zhou et al., 2022), PromptAgent (Wang et al., 2023), and APO (Pryzant et al., 2023). These methods use the same Proposer/Scorer LLM pair and the same meta-prompt template described in Section 4.1, but differ in search strategy. Unless otherwise noted, we report train and test accuracy on MedMCQA using Medicine-LLM-13B (Cheng et al., 2023) as the evaluation model.

Table 12 shows that DiSR achieves the highest train and test accuracy among the compared methods in this setting. We interpret this result as supplementary evidence that the gains observed for DiSR are not limited to comparison with OPRO alone. At the same time, because these methods differ in search strategy as well as prompt parameterization, we treat OPRO as the primary controlled baseline in the main paper.

Meta-Prompt Template

```
{{OPTIMIZATION_HISTORY}}

The subsequent examples illustrate the application of your instruction:
replace <INS> in each input with your instruction, then read the input
and generate an output. An output is deemed correct if it matches the
provided output; otherwise, it is considered incorrect.

{{TASK_EXAMPLES}}

First, briefly analyze the provided instructions and their scores. What
makes higher-scoring prompts more effective? Based on this analysis,
propose a new, improved instruction that aims to achieve the highest
possible score. Enclose only the new instruction within square brackets.
```

Figure 6: Template for the meta-prompt used in the optimization loop.

Table 4: Five-shot performance comparison on commonsense reasoning benchmarks. Across most settings, DiSR improves over the manual baseline and often outperforms OPRO. Best performance for each metric is in **bold**.

		LogiQA			OpenBookQA			CosmosQA		
Model	**Meth.**	**ACC**	**ECE**	**AUROC**	**ACC**	**ECE**	**AUROC**	**ACC**	**ECE**	**AUROC**
LLaMA2-7B	Manual	$0.37_{\pm 0.028}$	$0.52_{\pm 0.042}$	$0.45_{\pm 0.035}$	$0.40_{\pm 0.032}$	$0.49_{\pm 0.041}$	$0.55_{\pm 0.033}$	$0.37_{\pm 0.030}$	$0.52_{\pm 0.044}$	$0.45_{\pm 0.037}$
	OPRO	$0.40_{\pm 0.027}$	$0.44_{\pm 0.034}$	$0.51_{\pm 0.031}$	$0.41_{\pm 0.029}$	$0.46_{\pm 0.037}$	$0.57_{\pm 0.031}$	$0.40_{\pm 0.029}$	$0.49_{\pm 0.038}$	$0.52_{\pm 0.032}$
	DiSR	$\mathbf{0.41}_{\pm 0.024}$	$\mathbf{0.39}_{\pm 0.030}$	$\mathbf{0.55}_{\pm 0.027}$	$\mathbf{0.44}_{\pm 0.025}$	$\mathbf{0.42}_{\pm 0.034}$	$\mathbf{0.61}_{\pm 0.028}$	$\mathbf{0.43}_{\pm 0.026}$	$\mathbf{0.45}_{\pm 0.036}$	$\mathbf{0.56}_{\pm 0.029}$
LLaMA2-13B	Manual	$0.38_{\pm 0.025}$	$0.45_{\pm 0.039}$	$0.49_{\pm 0.033}$	$0.37_{\pm 0.032}$	$0.52_{\pm 0.040}$	$0.41_{\pm 0.036}$	$0.46_{\pm 0.028}$	$0.47_{\pm 0.035}$	$0.63_{\pm 0.027}$
	OPRO	$0.41_{\pm 0.023}$	$0.40_{\pm 0.030}$	$0.55_{\pm 0.029}$	$0.47_{\pm 0.026}$	$0.34_{\pm 0.031}$	$\mathbf{0.59}_{\pm 0.027}$	$0.45_{\pm 0.025}$	$0.38_{\pm 0.033}$	$0.62_{\pm 0.030}$
	DiSR	$\mathbf{0.43}_{\pm 0.022}$	$\mathbf{0.37}_{\pm 0.027}$	$\mathbf{0.57}_{\pm 0.025}$	$\mathbf{0.49}_{\pm 0.023}$	$\mathbf{0.31}_{\pm 0.028}$	$\mathbf{0.59}_{\pm 0.025}$	$\mathbf{0.48}_{\pm 0.024}$	$\mathbf{0.34}_{\pm 0.030}$	$\mathbf{0.67}_{\pm 0.026}$
LLaMA3-8B	Manual	$0.43_{\pm 0.026}$	$0.35_{\pm 0.032}$	$0.67_{\pm 0.023}$	$0.44_{\pm 0.028}$	$0.30_{\pm 0.026}$	$0.67_{\pm 0.024}$	$0.46_{\pm 0.027}$	$0.25_{\pm 0.024}$	$0.67_{\pm 0.025}$
	OPRO	$0.47_{\pm 0.021}$	$0.26_{\pm 0.022}$	$0.77_{\pm 0.019}$	$0.55_{\pm 0.021}$	$0.20_{\pm 0.021}$	$0.81_{\pm 0.020}$	$0.52_{\pm 0.022}$	$\mathbf{0.16}_{\pm 0.013}$	$0.75_{\pm 0.020}$
	DiSR	$\mathbf{0.50}_{\pm 0.017}$	$\mathbf{0.24}_{\pm 0.021}$	$\mathbf{0.79}_{\pm 0.016}$	$\mathbf{0.58}_{\pm 0.020}$	$\mathbf{0.17}_{\pm 0.018}$	$\mathbf{0.84}_{\pm 0.017}$	$\mathbf{0.56}_{\pm 0.019}$	$0.18_{\pm 0.019}$	$\mathbf{0.79}_{\pm 0.017}$
GPT-4	Manual	$0.47_{\pm 0.024}$	$0.26_{\pm 0.026}$	$0.75_{\pm 0.020}$	$0.50_{\pm 0.027}$	$0.21_{\pm 0.022}$	$0.78_{\pm 0.021}$	$0.55_{\pm 0.025}$	$0.18_{\pm 0.021}$	$0.70_{\pm 0.023}$
	OPRO	$0.54_{\pm 0.018}$	$0.17_{\pm 0.019}$	$0.85_{\pm 0.014}$	$0.61_{\pm 0.017}$	$0.15_{\pm 0.017}$	$0.84_{\pm 0.016}$	$0.63_{\pm 0.017}$	$0.11_{\pm 0.014}$	$0.86_{\pm 0.015}$
	DiSR	$\mathbf{0.56}_{\pm 0.015}$	$\mathbf{0.14}_{\pm 0.016}$	$\mathbf{0.89}_{\pm 0.012}$	$\mathbf{0.63}_{\pm 0.016}$	$\mathbf{0.12}_{\pm 0.014}$	$\mathbf{0.87}_{\pm 0.013}$	$\mathbf{0.65}_{\pm 0.015}$	$\mathbf{0.10}_{\pm 0.012}$	$\mathbf{0.88}_{\pm 0.011}$

Table 5: Performance comparison on the **Medicine** domain (MedMCQA).

Model	**Method**	**ACC** (↑)	**ECE** (↓)	**AUROC** (↑)
Med-Alpaca-7B	Manual	$0.36_{\pm 0.032}$	$0.51_{\pm 0.043}$	$0.47_{\pm 0.037}$
	OPRO	$0.38_{\pm 0.029}$	$0.53_{\pm 0.041}$	$0.50_{\pm 0.035}$
	DiSR (Ours)	$\mathbf{0.43}_{\pm 0.026}$	$\mathbf{0.44}_{\pm 0.034}$	$\mathbf{0.56}_{\pm 0.030}$
Medicine-LLM-13B	Manual	$0.52_{\pm 0.025}$	$0.34_{\pm 0.028}$	$0.71_{\pm 0.022}$
	OPRO	$0.55_{\pm 0.023}$	$0.30_{\pm 0.026}$	$0.73_{\pm 0.019}$
	DiSR (Ours)	$\mathbf{0.63}_{\pm 0.016}$	$\mathbf{0.17}_{\pm 0.020}$	$\mathbf{0.74}_{\pm 0.017}$
PMC-LLaMA-13B	Manual	$0.47_{\pm 0.027}$	$0.40_{\pm 0.033}$	$0.54_{\pm 0.031}$
	OPRO	$0.54_{\pm 0.024}$	$0.25_{\pm 0.025}$	$\mathbf{0.68}_{\pm 0.014}$
	DiSR (Ours)	$\mathbf{0.56}_{\pm 0.018}$	$\mathbf{0.22}_{\pm 0.021}$	$0.67_{\pm 0.019}$

Example Filled-in Meta-Prompt (Medical Domain)

```
Prompt1:
The following are multiple-choice questions (with answers) about
professional medicine. Please choose the correct answer from "A", "B",
"C", or "D".
Score: 61

Prompt2:
Presented below is a multiple-choice question pertaining to professional
medicine. Carefully review the question and select the correct option
from "A", "B", "C", or "D".
Score: 63

The subsequent examples illustrate the application of your instruction:
replace <INS> in each input with your instruction, then read the input
and generate an output. An output is deemed correct if it matches the
provided output; otherwise, it is considered incorrect.

Input:
A: <INS>
Q: A 13-month-old child is brought to the emergency department because
of urticaria, swelling of the lips, and difficulty breathing
immediately after eating an egg. A potential risk for hypersensitivity
reaction is posed by vaccination against which of the following
illnesses? A: Hepatitis B; Influenza C: Pertussis D: Poliomyelitis
Output: B

First, briefly analyze the provided instructions and their scores. What
makes higher-scoring prompts more effective? Based on this analysis,
propose a new, improved instruction that aims to achieve the highest
possible score. Enclose only the new instruction within square brackets.
```

Figure 7: An instantiated meta-prompt example for a medical optimization step.

B.5 Impact of Source-task Diversity

To study how scaffold quality depends on source-task diversity, we vary the number of source sub-tasks used in Stage 1 (2, 4, and 6) and evaluate the resulting scaffold in zero-shot transfer on a held-out target sub-task from the same domain. Table 13 reports the resulting accuracies.

Several patterns are visible. First, even with only two source sub-tasks, the discovered scaffold already outperforms the generic manual prompt in all three domains. Second, increasing the number of source sub-tasks from two to four consistently improves performance, suggesting that additional diversity helps the scaffold capture more transferable instruction structure. Third, gains diminish beyond four to six source tasks. We interpret this as evidence that scaffold discovery benefits from diversity, but does not require a large number of source tasks to become useful.

B.6 A Non-MCQA Task: Sentiment Classification on SST-2

To examine whether the scaffold–residual decomposition extends beyond multiple-choice QA, we conduct a small-scale experiment on SST-2. We cast sentiment classification as a two-label prediction task and apply the same overall DiSR procedure: Stage 1 discovers a sentiment-analysis scaffold from source subsets, and Stage 2 adapts it to

Table 6: Performance comparison on the **Law** domain (AGIEval-Law).

Model	Method	ACC (↑)	ECE (↓)	AUROC (↑)
LawGPT-7B	Manual	$0.43_{\pm 0.032}$	$0.45_{\pm 0.038}$	$0.68_{\pm 0.026}$
	OPRO	$0.44_{\pm 0.029}$	$0.41_{\pm 0.031}$	$0.67_{\pm 0.028}$
	DiSR (Ours)	$\mathbf{0.47_{\pm 0.024}}$	$\mathbf{0.39_{\pm 0.027}}$	$\mathbf{0.69_{\pm 0.022}}$
Law-LLM-13B	Manual	$0.45_{\pm 0.027}$	$0.24_{\pm 0.025}$	$0.69_{\pm 0.021}$
	OPRO	$0.51_{\pm 0.023}$	$0.25_{\pm 0.029}$	$0.71_{\pm 0.019}$
	DiSR (Ours)	$\mathbf{0.58_{\pm 0.018}}$	$\mathbf{0.18_{\pm 0.017}}$	$\mathbf{0.75_{\pm 0.015}}$
ChatLaw-13B	Manual	$0.54_{\pm 0.025}$	$0.38_{\pm 0.034}$	$0.63_{\pm 0.030}$
	OPRO	$0.62_{\pm 0.018}$	$0.15_{\pm 0.014}$	$\mathbf{0.73_{\pm 0.013}}$
	DiSR (Ours)	$\mathbf{0.64_{\pm 0.011}}$	$\mathbf{0.12_{\pm 0.010}}$	$0.72_{\pm 0.016}$

Table 7: Performance comparison on the **Finance** domain (FinEval).

Model	Method	ACC (↑)	ECE (↓)	AUROC (↑)
FinMA-7B-full	Manual	$0.44_{\pm 0.034}$	$0.51_{\pm 0.041}$	$0.56_{\pm 0.032}$
	OPRO	$0.47_{\pm 0.031}$	$0.52_{\pm 0.039}$	$0.61_{\pm 0.030}$
	DiSR (Ours)	$\mathbf{0.51_{\pm 0.027}}$	$\mathbf{0.38_{\pm 0.028}}$	$\mathbf{0.65_{\pm 0.025}}$
Finance-LLM-13B	Manual	$0.47_{\pm 0.029}$	$0.31_{\pm 0.027}$	$0.60_{\pm 0.026}$
	OPRO	$0.53_{\pm 0.023}$	$0.22_{\pm 0.022}$	$\mathbf{0.72_{\pm 0.019}}$
	DiSR (Ours)	$\mathbf{0.56_{\pm 0.021}}$	$\mathbf{0.21_{\pm 0.020}}$	$\mathbf{0.72_{\pm 0.016}}$
Tongyi-Finance-14B	Manual	$0.52_{\pm 0.026}$	$0.33_{\pm 0.024}$	$0.66_{\pm 0.023}$
	OPRO	$0.59_{\pm 0.018}$	$\mathbf{0.20_{\pm 0.014}}$	$\mathbf{0.76_{\pm 0.012}}$
	DiSR (Ours)	$\mathbf{0.61_{\pm 0.010}}$	$0.22_{\pm 0.017}$	$0.75_{\pm 0.015}$

Table 8: Qualitative example from the **Commonsense Reasoning** domain (LogiQA).

Component	Generated Prompt Text
Instructional Scaffold ($P_{\text{scaffold_commonsense}}$)	Your task is to solve the following reasoning puzzle. Analyze the provided text and question carefully. Select the most logical conclusion from the options "A", "B", "C", or "D".
Final Prompt (P_k^*) for a deductive reasoning task	Your task is to solve the following reasoning puzzle. Analyze the provided text and question carefully. **Identify the key premises and evaluate the validity of the logical chain of inference.** Select the most logical conclusion from the options "A", "B", "C", or "D".

the final evaluation split. Prompts are evaluated in a zero-shot setting, and we report Accuracy (ACC) and Expected Calibration Error (ECE).

On SST-2, DiSR follows the same overall pattern observed on the MCQA tasks: for both GPT-3.5-Turbo and LLaMA-2-13B, it improves over the manual prompt and OPRO on both accuracy and ECE. We present this experiment as a small exten-

Table 9: Qualitative example from the **Medical** domain.

Component	Generated Prompt Text
Instructional Scaffold ($P_{\text{scaffold_med}}$)	As an expert medical professional, analyze the following clinical scenario. Your task is to select the single best answer from the options "A", "B", "C", or "D". Your decision should be based on established medical guidelines and evidence-based practice.
Final Prompt (P_k^*) for "Diagnosis from History"	As an expert medical professional, analyze the following clinical scenario. Your task is to select the single best answer from the options "A", "B", "C", or "D". **Focus specifically on the patient's clinical history and symptoms to determine the most likely diagnosis.** Your decision should be based on established medical guidelines and evidence-based practice.

Table 10: Qualitative example from the **Law** domain.

Component	Generated Prompt Text
Instructional Scaffold ($P_{\text{scaffold_law}}$)	As a legal analyst, you are presented with a case. Your objective is to identify the most accurate legal conclusion from the choices "A", "B", "C", or "D". Your analysis must be grounded in relevant statutes and legal precedents.
Final Prompt (P_k^*) for "Contract Dispute"	As a legal analyst, you are presented with a case. Your objective is to identify the most accurate legal conclusion from the choices "A", "B", "C", or "D". **Evaluate the formation and validity of the contract in question, paying close attention to the elements of offer, acceptance, and consideration.** Your analysis must be grounded in relevant statutes and legal precedents.

Table 11: Qualitative example from the **Finance** domain.

Component	Generated Prompt Text
Instructional Scaffold ($P_{\text{scaffold_fin}}$)	You are a financial analyst. Based on the data provided, select the best financial recommendation from options "A", "B", "C", or "D". Your answer should align with standard principles of financial analysis and risk management.
Final Prompt (P_k^*) for "Equity Valuation"	You are a financial analyst. Based on the data provided, select the best financial recommendation from options "A", "B", "C", or "D". **Your primary task is to perform an equity valuation; consider metrics such as the P/E ratio, discounted cash flow (DCF), and market comparables.** Your answer should align with standard principles of financial analysis and risk management.

sion beyond the main task setting rather than as definitive evidence of broad task generality. Even so, the results suggest that the scaffold–residual decomposition may remain useful outside the MCQA setting when related classification tasks share a common decision format.

C More Discussions

C.1 Reproducibility Statement

We provide the main information needed to reproduce our experimental setup. All datasets are drawn from public benchmarks, and the evaluation models and data splits are listed in Appendix A.1. The optimization framework is described in Section 3, and Appendix A.2 reports the main hyperparameters together with the meta-prompt template used in

Table 12: Average train and test accuracy on MedMCQA for DiSR and additional APO baselines (evaluated with Medicine-LLM-13B).

Method	Train ACC	Test ACC
DiSR (Ours)	**0.68**	**0.63**
OPRO	0.60	0.55
PromptAgent	0.59	0.53
Iterative-APE	0.57	0.52
APO	0.54	0.50

the optimization loop. We also specify the shared manual initialization, the comparison protocol with OPRO, and the evaluation metrics used throughout the paper.

At the same time, exact reproduction may depend on access to the same proprietary optimizer models and APIs used in our experiments. Our goal is therefore to make the setup as transparent as possible so that the main comparisons and design choices can be independently verified and extended.

C.2 Ethics Statement

This work uses public benchmark datasets and evaluates prompt optimization methods in controlled experimental settings. It does not involve private user data or human-subject annotation. Because several evaluated benchmarks concern professional domains such as medicine, law, and finance, the results should not be interpreted as evidence that the resulting prompts are suitable for real-world decision-making without additional safeguards and domain-specific validation.

C.3 LLM Usage Statement

Large language models were used during manuscript preparation for language assistance and editing. All authors reviewed the final manuscript and take responsibility for its content.

Table 13: Zero-shot accuracy on a held-out target sub-task as a function of the number of source sub-tasks used during Scaffold Discovery. Results are shown for one representative domain-specific model in each domain.

Domain	Manual	2-Source	4-Source	6-Source
Medicine	0.29	0.42	0.47	**0.49**
Law	0.34	0.41	0.45	**0.47**
Finance	0.32	0.38	**0.41**	**0.41**

Table 14: Performance on SST-2 sentiment classification (zero-shot). DiSR improves over the manual prompt and OPRO in this small non-MCQA experiment.

Model	Method	ACC (↑)	ECE (↓)
GPT-3.5-Turbo	Manual	0.80	0.32
	OPRO	0.87	0.24
	DiSR (Ours)	**0.91**	**0.16**
LLaMA-2-13B	Manual	0.71	0.38
	OPRO	0.78	0.34
	DiSR (Ours)	**0.82**	**0.22**