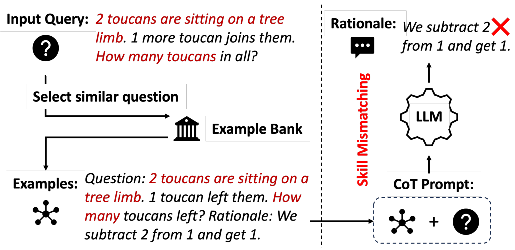
LaRS: Latent Reasoning Skills for Chain-of-Thought Reasoning

Zifan Xu[1*], Haozhu Wang[2], Dmitriy Bespalov[2], Xian Wu[2], Peter Stone[1,3], Yanjun Qi[2]

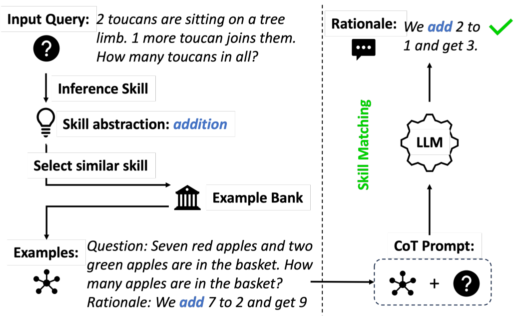
[1]The University of Texas at Austin, [2]Amazon Web Service, [3]Sony AI

Abstract

Chain-of-thought (CoT) prompting is a popular in-context learning (ICL) approach for large language models (LLMs), especially when tackling complex reasoning tasks. Traditional ICL approaches construct prompts using examples that contain questions similar to the input question. However, CoT prompting, which includes crucial intermediate reasoning steps (rationales) within its examples, necessitates selecting examples based on these rationales rather than the questions themselves. Existing methods require human experts or pre-trained LLMs to describe the skill, a high-level abstraction of rationales, to guide the selection. These methods, however, are often costly and difficult to scale. Instead, this paper introduces a new approach named **La**tent **R**easoning **S**kills (`LaRS`) that employs unsupervised learning to create a latent space representation of rationales, with a latent variable called a *reasoning skill*. Concurrently, `LaRS` learns a *reasoning policy* to determine the required reasoning skill for a given question. Then the ICL examples are selected by aligning the reasoning skills between past examples and the question. This approach is theoretically grounded and compute-efficient, eliminating the need for auxiliary LLM inference or manual prompt design. Empirical results demonstrate that LaRS consistently outperforms SOTA skill-based selection methods, processing example banks four times faster, reducing LLM inferences during the selection stage by half, and showing greater robustness to sub-optimal example banks.



(a) Question-similarity-based selection.



(b) Skill-based selection.

Figure 1: CoT prompting with examples selected by (a) similar questions and (b) similar skills that (mis)match the skills in their rationales.

1 Introduction

Large Language Models (LLMs) exhibit remarkable capabilities in solving various downstream tasks through in-context learning (ICL) (Brown et al., 2020), even without being explicitly trained on the distribution of in-context examples (Vaswani et al., 2017; Devlin et al., 2019; Rae et al., 2021; Chowdhery et al., 2022; Wei et al., 2022a). Using in-context learning, LLMs generate output for an input query by conditioning on a prompt that contains a few input-output *demonstrations*.

Reasoning tasks have proven to be particularly difficult for language models and NLP in general (Rae et al., 2021; Bommasani et al., 2021; Nye et al., 2021). In the recent literature, chain-of-thought (CoT) prompting, an ICL method, has

*Majority of this work was done when Zifan Xu was an intern at Amazon Web Service during the summer 2023. A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (FAIN-2019844, NRT-2125858), ONR (N00014-18-2243), ARO (W911NF-23-2-0004, W911NF-17-2-0181), Lockheed Martin, and UT Austin's Good Systems grand challenge. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3624–3643
November 12-16, 2024 ©2024 Association for Computational Linguistics

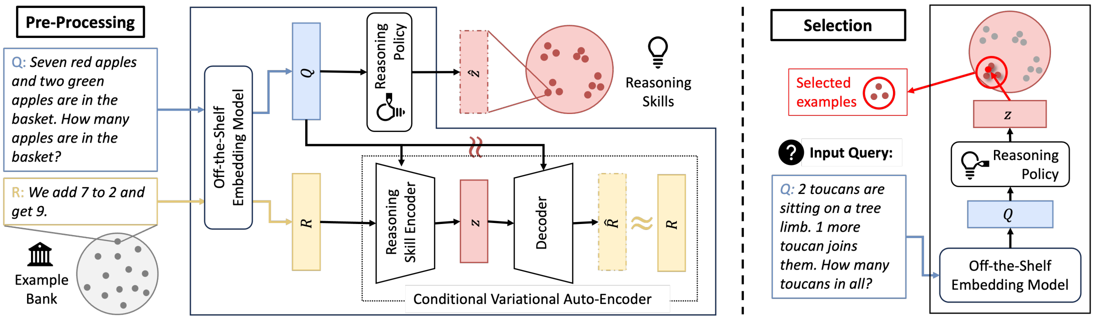


Figure 2: An overview of LaRS including a pre-processing stage (left) and a selection stage (right).

been proposed to improve LLMs on a wide spectrum of reasoning tasks by guiding LLMs to produce a sequence of intermediate steps (rationale) for generating a (better) final answer (Cobbe et al., 2021a; Wei et al., 2022b; Suzgun et al., 2022). The prompts for CoT are composed of *demonstrations* that contain not only input and output, but also the rationales for why the output holds.

The core challenge for ICL lies in designing effective demonstrations to prompt LLMs. Much evidence has indicated the significant impact of demonstrations on the performance of ICL (Lu et al., 2021; Liu et al., 2021). To form a prompt, one important setting considers selecting demonstrations from an existing example bank, termed demonstration selection (Dong et al., 2022). While a variety of methods exist in the ICL literature for automating this process, CoT prompts are distinct in that they include not only questions and answers but also specially-designed rationales. *This distinction highlights the importance of rationales in selecting demonstrations for CoT prompting.* Specifically, CoT prompting should select demonstrations that illustrate relevant skills within their rationales to effectively address a given question. For instance, in solving math word problems (as depicted in Fig. 1), a useful rationale involves computing addition to get the correct answer. Selecting few-shot examples based on the question similarity (Fig. 1a) might lead to examples showcasing subtraction and generate incorrect rationales. However, skill-based selection (Fig. 1b) can align the skills between examples and the given question, which leads to correct answers guided by relevant rationales.

To achieve such a skill-based demonstration selection, An et al. (2023b) introduces `Skill-KNN`, which employs pre-trained LLMs to generate skill descriptions. Then, the few-shot examples are selected based on the embedding of the skill descriptions computed by another pre-trained embedding model. Although this approach is straightforward, the LLM-generated skill descriptions can be somewhat arbitrary, heavily relying on the manually crafted prompts. This reliance constrains its wider applicability across diverse reasoning tasks. Moreover, the approach requires to generate a unique skill description for each example, which limits its scalability to larger example banks.

Rather than relying on LLMs, we introduce **La**tent **R**easoning **S**kill Discovery (LaRS), a new skill-based demonstration selection method. This approach learns skills as latent space representations of rationales through unsupervised learning. The essence of LaRS lies in a unique formulation for the generation of rationales, which we term the latent *skill model*. This model, inspired by the principles of topic models (Xie et al., 2021a), conditions the generation of a rationale on both a given question and a latent variable, called a *reasoning skill*. This latent variable embodies a high-level abstraction of the rationales, such as formats, equations, or knowledge.

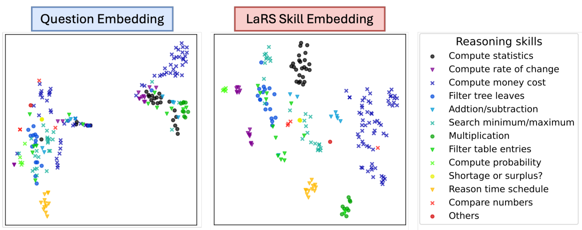


Figure 3: t-SNE projections of question embedding and LaRS reasoning skill embedding of the exmaples from TabMWP (Lu et al., 2022) dataset. The 12 different colors correspond to 12 skill labels annotated by human.

Under the skill model formulation, LaRS utilizes a Conditional Variational Auto-encoder (CVAE) to approximate the generation of rationales on a small dataset from the example bank. As a result, two probabilistic models can be learned concurrently: (1) a *reasoning skill encoder* that maps an example to the actual reasoning skills demonstrated in the

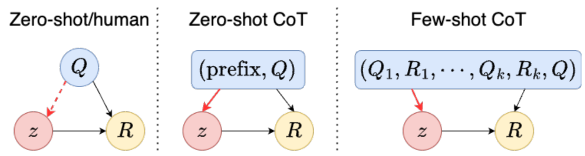


Figure 4: Causal graphs for prompting with zero-shot/human (left), zero-shot CoT (middle), and few-shot CoT (right) for generating rationales via skills. The dashed arrow from Q to z indicates possible sub-optimal inference of the reasoning skills from both human and zero-shot LLM generations.

rationale; and (2) a *reasoning policy* that predicts the reasoning skills required for a particular question. This method of learning through a CVAE, especially when applied to a small dataset from the example bank, is both cost-efficient and fast compared to `Skill-KNN`. Fig. 2 presents an overview of `LaRS`. In addition, Figure 3 shows the learned reasoning skill embedding (right) that effectively separates examples with different skill labels, while the off-the-shelf question embedding does not.

The efficacy of `LaRS` is evaluated on four different benchmarks based on five backbone LLMs with varying scales. The method is also compared with baseline approaches, including an oracle method that assumes access to ground truth rationales. `LaRS` consistently outperforms `Skill-KNN` and also matches the oracle performance in almost half of the experiments. In addition, `LaRS` reduces half of the LLM inference, eliminates the need of human prompt design, and maintains better robustness to sub-optimal example banks. A summary of this paper's contribution is as follows:

- We propose `LaRS`, a novel unsupervised demonstration selection approach for CoT prompting, and empirically verify its effectiveness through large scale experiments.
- We introduce the latent skill model, a plausible formulation for CoT reasoning, which has illuminated a deeper understanding of CoT prompting.
- We present theoretical analyses of the optimality of the latent-skill-based selection method.

2 Related Work

2.1 CoT Reasoning

CoT prompting is a special prompt design technique that encourages LLMs to generate intermediate rationales that guide them towards providing accurate final answers. These rationales can exhibit remarkable flexibility in their styles. For instance, the original work by (Wei et al., 2022b) specially designs rationales in the in-context demonstrations to suit different reasoning tasks. Moreover, novel prompt designs that highlight diverse formats of the rationales have emerged to enhance CoT prompting. For example, (Kojima et al., 2022) proposed Program of Thoughts (PoT) that disentangles textual reasoning from computation, with the latter specially handled through program generation.

In contrast to manual design, our method `LaRS` can be thought of as automatic discovery of diverse rationale styles from an example bank. This method can also dynamically select reasoning skills based on the specific questions. Worth noting, (Chen et al., 2023) introduces SKills-in-Context (SKiC), which confines rationale generation to predefined "skills" within the prompt. Although sharing a similar motivation to `LaRS`, we emphasize two crucial distinctions: (1) while SKiC relies on manual "skills" design, `LaRS` automatically discovers them, (2) SKiC presents a full list of "skills" in the prompt, allowing LLMs to select from them, whereas `LaRS` learns the skill selection from the example bank, explicitly instructing LLMs on which skill to employ through in-context examples.

2.2 Demonstration Selection

Demonstration selection refers to a special setting, where the prompts are constructed by selecting examples from an example bank. In this context, our `LaRS` aligns with the paradigm of unsupervised demonstration selection, which involves designing heuristics for this selection process. A variety of heuristics have been explored, including similarity (Gao et al., 2021; Hu et al., 2022), diversity (Zhang et al., 2022), coverage (Gupta et al., 2023), and uncertainty (Diao et al., 2023). Among these, `Skill-KNN` ((An et al., 2023b)) shares the closest resemblance to our approach. However, `Skill-KNN` relies on pre-trained LLMs to provide "skill" annotations, which could be arbitrary and resource-intensive, requiring extensive inferences of LLMs and human prompt design. In contrast, `LaRS` automatically discovers reasoning skills by learning a lightweight CVAE represented by two-layer MLPs and standard loss function. In addition, the selections based on these discovered reasoning skills are theoretically-grounded based on the latent skill model and the theoretical analyses presented in this paper.

3 Formulation

In this section, we formally describe the *skill model*, a new formulation for explaining the generation of rationales in CoT reasoning. In Section 3.1, the skill model is first introduced to describe the human-generated rationales. Then, Section 3.2 illustrates how the skill model can be adapted to LLM-generated rationales. Finally, leveraging the concept of reasoning skill as outlined in the skill model, a new latent-skill-based demonstration selection method is formally described in Section 3.3.

3.1 Skill Model

Let $\mathcal{X}$ be the set of all sequences of tokens, $\mathcal{Z}$ be the continuous vector space of latent reasoning skills, and P_H denotes the probability distribution of real-world natural language. CoT reasoning is to generate a rationale $R \in \mathcal{X}$ given a question $Q \in \mathcal{X}$, whose correctness[1] can be verified by an indicator function $\mathbb{1}(R, Q) := \mathbb{1}(R \text{ is the correct rationale for } Q)$.

The skill model assumes that the real-world conditional distribution of R given Q can be described as follows:

$$P_H(R \mid Q) = \int_{\mathcal{Z}} P_H(R \mid z, Q) P_H(z \mid Q) dz \quad (1)$$

where, $P_H(z \mid Q)$ is the posterior of selecting latent reasoning skills in human reasoning, called a reasoning policy. $P_H(R \mid z, Q)$ is the posterior distribution of generating R given a question Q and a reasoning skill z. A causal graph illustrating such a generation process involving a latent reasoning skill z is presented in Fig. 4 on the left.

Unlike (Wang et al., 2023), this formulation considers a dependency of z on Q reflecting a preference for selecting particular reasoning skills to solve a given question. We justify this formulation as follows:

1. Rationales can exhibit remarkable flexibility, manifesting diverse formats, topics, and knowledge, which can naturally be abstracted into the high-level concepts of reasoning skills.
2. The selection of these skills is not bound by strict determinism. For instance, diverse reasoning paths and formats could all contribute toward finding the correct final answer. Therefore, real-world data is a mixture of diverse skills captured by a stochastic reasoning policy $P_H(z \mid Q)$.

[1]For math word problems, whose answers are discrete labels, the correct rationale should contain the correct answer label as the final step. For code generation, the correct rationale should be the correct code.

3.2 CoT prompting

LLMs are pre-trained conditional generators. Given an input query $X \in \mathcal{X}$, the conditional distribution of an output $Y \in \mathcal{X}$ generated by LLMs can be written as $P_M(Y \mid X)$. LLMs are usually trained on generic real-world data distribution such that $P_M(Y \mid X) \approx P_H(Y \mid X)$.

Prior studies have presented an implicit topic model formulation in explaining the in-context learning mechanisms of LLMs (Wang et al., 2023; Xie et al., 2021a). Similarly, we posit that LLMs can be viewed as implicit skill models for generating rationales. To elaborate, when generating rationales, LLMs' conditional distribution $P_M(R \mid Q)$ can be extended as follows (with illustrations in Fig. 4 on the left):

$$P_M(R \mid Q) = \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) dz \quad (2)$$

This implicit skill model assumes that LLMs also infer reasoning skills z, which resembles the real-world generation of rationales.

The above formulation only encompasses the zero-shot generation of rationales. In practice, prompts are commonly provided to guide LLMs' generation. In general, two CoT prompting strategies exist: zero-shot CoT, employing a prompt comprising a short prefix and a test question, and few-shot CoT, employing a prompt containing pairs of questions and rationales. Denoting $pt \in \mathcal{X}$ as a prompt, a unified formulation for both prompting strategies can be derived as follows:

$$P_M(R \mid pt) = \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid pt) dz \quad (3)$$

$$\text{0-shot CoT: } pt = (\text{prefix}, Q) \text{ or } (Q, \text{prefix})$$

$$k\text{-shot CoT: } pt = (Q_1, R_1, \cdots, Q_k, R_k, Q)$$

Here, the formulation is simplified such that the use of prompts only influences the probability distribution of z. For instance, a prefix specifying the generation's format can be interpreted as specifying the reasoning skill z by shaping the distribution from $P_M(z \mid Q)$ to $P_M(z \mid pt)$. This simplification aligns with empirical evidence suggesting that in-context examples serve as mere pointers to retrieve already-learned knowledge within LLMs (Shin et al., 2020; Min et al., 2022; Wang et al., 2022).

Drawing upon this formulation, we can gain insight into the failure of zero-shot generation. *In general, real-world data is inherently noisy, indicating that the reasoning policy $P_H(z \mid Q)$ may be sub-optimal, and the reasoning skills are not chosen to maximize the accuracy of answering a test question.* Trained on this generic real-world data distribution, $P_M(z \mid Q)$ could also be sub-optimal, leading to the failure of zero-shot generation. On the other hand, CoT prompting improves the reasoning performance by shaping the distribution of reasoning skills using carefully-designed prompts that contain either prefix or few-shot examples.

3.3 Skill-Based Demonstration Selection

The analysis above suggests that the key to the success of CoT prompting is to design an effective prompt that improve upon the posterior distribution of human's preference of reasoning skills $P_H(z \mid Q)$. To design an effective prompt, the demonstration selection problem assumes access to an example bank of question-rationale pairs, denoted as $\mathcal{D}_E = \{(R, Q)\}$. This example bank is usually specially-crafted and has a distribution different from the real-world distribution. Denoting P_E as the distribution of the example bank, R is distributed according to $P_E(R \mid Q)$ for all $(R, Q) \in \mathcal{D}_E$.

Given $\mathcal{D}_E$, the demonstration selection is to select a few question-rationale pairs from $\mathcal{D}_E$. Assuming that each selected demonstration is i.i.d, a demonstration selection method can be uniquely defined as a probabilistic model $g(Q, R|Q_{\text{test}}) := \mathcal{X} \mapsto \Delta(\mathcal{X})$ that maps a test question Q_{test} to a probability distribution of demonstrations. Then, we can formally define the skill-based demonstration selection method as follows:

Definition 1 *Skill-based demonstration selection is given by*

$$g_{skill}(Q, R \mid Q_{test}) = \int_{\mathcal{Z}} P_E(Q, R \mid z) P_E(z \mid Q_{test}) dz$$

Intuitively, this selection method maximizes the probability of a selected demonstration showcasing the reasoning skill that is likely to be chosen according to $P_E(z \mid Q)$. Since the example bank is usually specially-crafted and contains rationales showcasing "better" reasoning skills, the in-context examples that align with $P_E(z \mid Q)$ are intuitively more effective. In Section 4.3, we provide theoretical analysis of the optimality of this skill-based selection when conditioned on certain ideal assumptions of the example bank and LLMs.

4 Method

To enable the skill-based demonstration selection (Definition 1), we introduce our approach `LaRS` , which involves learning a conditional variational autoencoder (CVAE) to approximate P_E using the data from the example bank $\mathcal{D}_E$. We then outline a practical demonstration selection process aligning with the skill-based selection. The schematic overview of `LaRS` (right) and the corresponding demonstration selection process (left) are illustrated in Figure 2.

4.1 Latent Reasoning Skill Discovery

The conditional variational autoencoder (CVAE) has emerged as a popular approach for modeling probabilistic conditional generation. As one specific case, the skill model, introduced in this paper, can effectively be represented as a CVAE. Therefore, we introduce `LaRS` that employs a CVAE to approximate the generation of rationales using the data from the example bank $\mathcal{D}_E = \{(Q, R)\}$.

In particular, this CVAE includes three coupled models: an encoder model, a decoder model, and a reasoning policy model, independently parameterized by ω, ψ, and ϕ respectively. Drawing from the notations introduced in the skill model, the reasoning policy model is a conditional Bayesian network $\pi_\phi(z \mid Q)$, determining the posterior distribution of latent reasoning skill z given a question Q. The decoder model is also a conditional Bayesian network $p_\psi(R \mid z, Q)$ that generates a rationale R, conditioned on both Q and z, where z is sampled from $\pi_\phi(z \mid Q)$. Finally, the encoder model $q_\omega(z \mid Q, R)$ is another conditional Bayesian network, mapping a question-rationale pair to z. In this paper, we train this CVAE using classical variational expectation maximization and the reparameterization trick.

Specifically, the classical variational expectation maximization optimizes a loss function as follows:

$$\begin{aligned} &\mathcal{L}_{\text{CVAE}}(\phi, \omega, \psi) = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} \qquad (4) \\ &\mathcal{L}_{\text{recon}} = -\mathbb{E}_{(Q,R)\sim\mathcal{D}_E, z\sim q_\omega(|Q,R)}[\log p_\psi(R|z, Q)] \\ &\mathcal{L}_{\text{KL}} = \mathbb{E}_{(Q,R)\sim\mathcal{D}_E}[\mathrm{D}_{\text{KL}}(q_\omega(z \mid Q, R) \parallel \pi_\phi(z \mid Q))] \end{aligned}$$

By training to minimize this loss function, q_ω and π_ϕ can be learned to effectively approximate the conditional distributions $P_E(z \mid Q, R)$ and

$P_E(z \mid Q)$. It is worth noting that the decoder model acts an auxiliary model that only roughly reconstructs rationales for the purpose of training the encoder and the reasoning policy model, and is not deployed to generate rationales in the downstream tasks.

Ideally, all three models would be represented by language models, processing token sequences as input and generating token sequences as output. However, training full language models for demonstration selections can be computationally expensive. Instead, we adopt a pre-trained embedding model denoted as $f : \mathcal{X} \mapsto \Theta$, which maps the token space $\mathcal{X}$ to an embedding space Θ. Consequently, the decoder model, encoder model, and reasoning policy model transform into $p_\psi(f(R)|z, f(Q))$, $q_\omega(z|f(Q, R))$, and $\pi_\phi(z|f(Q))$, respectively. They now condition on and generate the embeddings instead of the original tokens. In the actual implementation, we use the same feed-forward neural network to represent both π_ϕ and q_ω, predicting the mean and variance of Gaussian distributions of latent reasoning skills. On the other hand, p_ψ is a feed-forward neural network that deterministically predicts a value in the embedding space.

4.2 Demonstration Selection

Since the distribution $P_E(Q, R \mid z)$ in Definition 1 is practically intractable, we propose a selection process that effectively aligns with the skill-based selection using the learned π_ϕ and q_ω. For a given test question Q_{test}, the desirable reasoning skill $z_{\text{test}} = \arg\max_z[\pi_\phi(z|f(Q_{\text{test}}))]$ can be computed using the reasoning policy. Subsequently, each example from the example bank can be scored based on the cosine similarity between z_{test} and z_{post}, where $z_{\text{post}} = \arg\max_z[q_\omega(z|Q, R))]$ represents the maximum likelihood skill of the current example. Finally, a CoT prompt can be constructed by selecting the top-k examples according to the computed scores. The step-by-step procedure is outlined in Algorithm 1.

4.3 Theoretical Analysis

In this section, we provide a theoretical analysis of the optimality of the skill-based selection by Definition 1.

Let $P_M(R \mid Q, g)$ denotes LLMs' conditional distribution of a rationale R given a test question Q under a demonstration selection method g. $P_M(R \mid Q, g)$ can be extended as follows:

$$P_M(R \mid Q, g) = \int_{\mathcal{X}^k} P_M(R \mid pt)\Pi_{i=1}^k[g(Q_i, R_i \mid Q)d(Q_i, R_i)]$$

Here, each demonstrations (Q_i, R_i) is independently sampled from $g(Q_i, R_i \mid Q), \forall i = 1, \cdots, k$. These k demonstrations form a prompt $pt = (Q_1, R_1, \cdots, Q_k, R_k, Q)$.

We want to show that $P_M(R \mid Q, g)$ is the optimal conditional distribution that maximizes the accuracy of rationales if the selection follows skill-based selection method or $g = g_{skill}$. We begin by defining the optimal conditional distribution as follows:

Definition 2 *Optimal conditional distribution of rationales given questions $P^*(R \mid Q)$ is given by:*

$$P^*(R \mid Q) = \underset{P(\cdot|Q)\in\Delta(\mathcal{X})}{\arg\max} \int_{\mathcal{X}} \mathbb{1}(R, Q)P(R \mid Q)dR$$

Here $\mathbb{1}(R, Q)$ is the indicator function of the correctness of R given a question Q (see Section 3.1).

Then, we state two major assumptions as follows:

Assumption 1 *Example bank is sampled from the optimal conditional distribution, or $P_E(R \mid Q) = P^*(R \mid Q)$.*

Assumption 2 *Humans and LLMs are expert rationale generators given reasoning skills and questions, meaning that $P_H(R \mid z, Q) = P_E(R \mid z, Q) = P_M(R \mid z, Q)$.*

Assumption 1 is rooted in the fact that example banks are human-crafted that contains the most useful rationales for answering the questions. In Assumption 2, P_M capturing P_H is a common assumption in the literature studying LLMs (Xie et al., 2021b; Saunshi et al., 2020; Wei et al., 2021). $P_E(R \mid z, Q) = P_H(R \mid z, Q)$ is based on the assumption that reasoning skills are shared across humans, and the generation of rationales is identical given the same reasoning skills and questions.

Based on the above definiton and two assumptions, we prove the following theorem.

Theorem 1 *A LLM gives the optimal conditional distribution of rationales given questions:*

$$P_M(R \mid Q, g_{skill}) = P^*(R \mid Q)$$

If (1) it is prompted by $k \to \infty$ in-context examples selected by the skill-based selection g_{skill} defined by Definition 1, (2) Assumption 2 and Assumption 1 hold.

Appendix E presents the proof for Theorem 1.

5 Experiments

This section describes the experimental settings, baselines, metrics, and main results.

5.1 Dataset

For benchmarking, the selection methods are evaluated on four challenging datasets, including two datasets of Math Word Problem (MWP): **TabMWP**, **GSM8K**, one text-to-SQL dataset: **Spider**, and one semantic parsing dataset: **COGS**.

Each dataset is split into a training set used to learn `LaRS` models and a test set used to evaluate the selection methods. While the training sets may potentially be large, we use randomly sampled 1K examples from the training set as the example bank, from which, the examples can be selected for CoT prompting. Detailed descriptions of the datasets and splitting are presented in Appendix B.

To measure the performances, we use the answer accuracy for **TabMWP** and **GSM8K**, with the answers extracted by searching the texts right after a prefix `The answer is`. For **Spider**, we use the official execution-with-values accuracy[2]. For **COGS**, we report the exact-match accuracy for semantic parsing.

5.2 Selection Methods

Our method `LaRS` is compared with the following four baselines. All the hyper-parameters related to these methods are listed in Appendix B.

`Skill-KNN` This baseline represents a state-of-the-art (SOTA) skill-based selection method. It employs pre-trained LLMs to generate skill descriptions for both the questions in the example bank and the test question. Then, the method selects examples whose skill descriptions most closely match that of the test question to form the prompt, using cosine similarity computed with a pre-trained embedding model. To examine the dependency on the LLMs' ability to generate skill descriptions, we introduce two variations: `Skill-KNN-large`, which uses the larger LLM gpt-3.5-turbo, and `Skill-KNN-small`, which uses the smaller LLM Falcon-40B-instruct. Additionally, to evaluate the effect of human-annotated skill descriptions prompting the LLMs to generate new skills, we introduce `Skill-KNN-zero`, which uses gpt-3.5-turbo to generate skill descriptions in a zero-shot fashion. `Skill-KNN-zero` closely resembles the setting of `LaRS` , as it does not rely on human prompt design. Therefore, `LaRS` is primarily compared with `Skill-KNN-zero`.

Random This baseline randomly selects k in-context examples from the example bank. For each test question, the accuracy is reported as an average over three independent random selections.

Retrieval-Q This baseline employs a pre-trained embedding model to encode a test question, and selects in-context examples based on the cosine similarity between embeddings from examples' questions and the test question.

Retrieval-R (oracle) This baseline employs a pre-trained embedding model to encode the ground-truth rationale of a test question, and selects in-context examples based on the cosine similarity between examples' rationales and the ground-truth rationale.

5.3 Backbones and Hyper-parameters

In terms of the backbone models, the ICL is conducted by two OpenAI language models: gpt-4o and gpt-3.5-turbo, two Anthropic model: claude-3-sonnet and claude-3-haiku, and one smaller-scale Falcon-40B-Instruct (Xu et al., 2023). All the embedding is computed by a pre-trained embedding model, Deberta-v2-xlarge (He et al., 2021). We also investigate different choices of embedding model in Section C.

During inference, the temperature is set to 0 (i.e., greedy decoding) to reduce the variance. The CoT prompts contain $k = 2, 4, 4, 8$ in-context examples for **TabMWP**, **GSM8K**, **Spider**, and **COGS**, respectively.

5.4 Performance comparison results

Table 7 presents experiment result summary. Detailed descriptions are as follows:

`LaRS` matches SOTA skill-based selection methods with superior computational efficiency. As shown in Table 7, across all four benchmarks and five backbone models tested, `LaRS` outperforms `Skill-KNN-zero` in 18 out of 20 experi-

[2]We use the official evaluation scripts for Spider in https://github.com/taoyds/test-suite-sql-eval.

Method	TabMWP	GSM8K	Spider	COGS
Backbone: gpt-3.5-turbo				
Random	62.4 $_{+0.0}$	75.7 $_{+0.0}$	46.8 $_{+0.0}$	67.5 $_{+0.0}$
Retrieval-Q	72.3 $_{+9.9}$	75.6 $_{-0.1}$	49.9 $_{+3.1}$	88.5 $_{+21.0}$
`Skill-KNN-zero`	77.7 $_{+15.3}$	75.0 $_{-0.7}$	49.0 $_{+2.2}$	77.9 $_{+10.8}$
`LaRS` (ours)	**78.1** $_{+15.7}$	**76.8** $_{+1.1}$	**53.0** $_{+6.2}$	**94.8** $_{+27.2}$
Retrieval-R (oracle)	77.4 $_{+15.0}$	75.5 $_{-0.2}$	64.4 $_{+17.6}$	95.7 $_{+28.2}$
Backbone: gpt-4o				
Random	87.6 $_{+0.0}$	78.1 $_{+0.0}$	74.1 $_{+0.0}$	73.0 $_{+0.0}$
Retrieval-Q	85.9 $_{-1.7}$	78.1 $_{+0.0}$	75.9 $_{+1.8}$	86.9 $_{+16.9}$
`Skill-KNN-zero`	87.7 $_{+0.1}$	**78.6** $_{-0.5}$	76.6 $_{+2.5}$	78.1 $_{+5.1}$
`LaRS` (ours)	**87.9** $_{+0.3}$	78.3 $_{+0.2}$	**77.2** $_{+3.1}$	**90.2** $_{+17.2}$
Retrieval-R (oracle)	88.8 $_{+1.2}$	77.1 $_{-1.0}$	78.1 $_{+4.0}$	92.8 $_{+19.8}$
Backbone: claude-3-sonnet				
Random	92.6 $_{+0.0}$	93.3 $_{+0.0}$	61.7 $_{+0.0}$	79.2 $_{+0.0}$
Retrieval-Q	93.1 $_{+0.5}$	92.4 $_{-0.9}$	61.8 $_{+0.1}$	94.6 $_{+15.4}$
`Skill-KNN-zero`	93.1 $_{+0.5}$	92.1 $_{-1.2}$	61.9 $_{+0.2}$	86.6 $_{+7.4}$
`LaRS` (ours)	**93.7** $_{+1.1}$	**93.6** $_{+0.3}$	**62.2** $_{+0.5}$	**96.9** $_{+17.7}$
Retrieval-R (oracle)	94.1 $_{+1.5}$	92.8 $_{-0.5}$	62.4 $_{+0.7}$	97.6 $_{+18.4}$
Backbone: claude-3-haiku				
Random	88.6 $_{+0.0}$	88.6 $_{+0.0}$	60.2 $_{+0.0}$	66.2 $_{+0.0}$
Retrieval-Q	92.2 $_{+3.6}$	88.6 $_{+0.0}$	60.0 $_{-0.2}$	88.5 $_{+22.3}$
`Skill-KNN-zero`	93.3 $_{+4.7}$	**88.8** $_{+0.2}$	61.0 $_{+0.8}$	79.7 $_{+13.5}$
`LaRS` (ours)	**93.3** $_{+4.7}$	87.6 $_{-1.0}$	**61.3** $_{+1.1}$	**89.9** $_{+23.7}$
Retrieval-R (oracle)	92.4 $_{+3.8}$	88.9 $_{+0.3}$	61.2 $_{+1.0}$	96.5 $_{+30.3}$
Backbone: Falcon-40B-Instruct				
Random	45.7 $_{+0.0}$	38.8 $_{+0.0}$	20.6 $_{+0.0}$	45.1 $_{+0.0}$
Retrieval-Q	51.9 $_{+6.2}$	37.3 $_{-1.5}$	22.1 $_{+1.5}$	73.9 $_{+28.8}$
`Skill-KNN-small`	51.4 $_{+5.7}$	36.5 $_{-2.3}$	20.3 $_{-0.3}$	59.4 $_{+14.3}$
`Skill-KNN-zero`	55.2 $_{+9.5}$	38.7 $_{-0.1}$	23.3 $_{+2.7}$	82.1 $_{+37.0}$
`LaRS` (ours)	**57.7** $_{+12.0}$	**39.1** $_{+0.3}$	**24.8** $_{+4.2}$	**89.5** $_{+44.4}$
Retrieval-R (oracle)	61.2 $_{+15.5}$	40.4 $_{+1.6}$	39.9 $_{+19.3}$	90.3 $_{+45.2}$

Table 1: Main results (%) across all backbone models and datasets. Numbers in **bold** represent the best results for each backbone model across all selection methods. The subscripted gray values indicate the relative improvement over Random selection.

ments. This result highlights the effectiveness of the latent reasoning skills learned through unsupervised learning with small CVAE models, achieving comparable performance to the skill descriptions crafted by extensively pre-trained LLMs. Notably, `Skill-KNN-zero` uses the powerful LLM gpt-3.5-turbo for skill generations. However, in scenarios where only less capable LLMs are available, such as lacking an internet connection and requiring local inference, `Skill-KNN-small`, which uses the less capable LLM Falcon-40B-instruct, suffers significant performance drops across all four benchmarks. In contrast, `LaRS` does not require powerful LLMs and achieves similar performance boosts for smaller backbone models like Falcon-40B-Instruct compared to `Skill-KNN-zero`.

Furthermore, in Table 3, we present a comparison of computational overhead, including computing time, estimated cost for pre-processing the example bank, and cost for each input query during selection, among Retrieval-Q, `LaRS`, `Skill-KNN-zero`, and a supervised selection method PromptPG (Lu et al., 2022). Our method achieves accuracy comparable to `Skill-KNN-zero`, requiring no LLM inferences (approximately $30 savings per 1k examples) and reducing computing time by 1.5 hours per 1k examples during pre-processing, along with more than 100% less cost per input query. Detailed experimental settings for estimating these costs can be found in Appendix B.

Benchmark	TabMWP			
Replace Rate (%)	0	0.1	0.2	0.3
`Skill-KNN-zero`	77.7	77.0 $_{-0.9\%}$	76.2 $_{-1.9\%}$	75.4 $_{-3.0\%}$
`LaRS`	**78.1**	**78.1** $_{-0.0\%}$	**78.0** $_{-0.1\%}$	**77.9** $_{-0.1\%}$
Benchmark	COGS			
Replace Rate (%)	0	0.1	0.2	0.3
`Skill-KNN-zero`	77.9	75.8 $_{-2.7\%}$	73.8 $_{-5.3\%}$	68.8 $_{-11.7\%}$
`LaRS`	**94.8**	**94.7** $_{-0.1\%}$	**93.3** $_{-1.6\%}$	**93.0** $_{-1.9\%}$

Table 2: Answer accuracy (%) of `Skill-KNN-zero` and `LaRS` on TabMWP and COGS benchmark with 0%, 10%, 20%, and 30% of the rationales in the example bank being replaced with random rationales. The subscripted gray values indicate the percentage drop relative to optimal example banks.

	Accuracy (%)↑	Pre-processing Time (h/1k)↓	Pre-processing Cost ($/1k)↓	Selection Cost per query ($)↓
`LaRS` (ours)	**78.1**	**0.5** $_{+0\%}$	**$0**	**$0.02** $_{+\%0}$
`Skill-KNN-zero`	77.7	2 $_{+300\%}$	$30	$0.05 $_{+\%150}$
PromptPG	74.2	6 $_{+1100\%}$	$50	$0.02 $_{+\%0}$
Retrieval-Q	72.3	0 $_{-100\%}$	$0	$0.02 $_{+\%0}$

Table 3: Comparison of accuracy and computational overhead, including computing time, estimated cost for pre-processing an example bank of 1k, and average cost per input query during selection, among four selection methods on the **TabMWP** dataset. The grey percentages represent the increased cost ratio associated with each selection method.

`LaRS` is more robust to sub-optimal example banks. `Skill-KNN` selects examples based solely on the questions. For example, it selects examples whose questions require the same skills as the given question. However, sub-optimal example banks may include examples with incorrect or sub-optimal rationales, which should be avoided. In contrast, `LaRS` considers both questions and rationales when computing the reasoning skill embedding, enhancing its robustness to sub-optimality. Table 2 presents the answer accuracy of `Skill-KNN-zero` and `LaRS` on the TabMWP and COGS benchmark with sub-optimal example banks, where 10%, 20% and 30% of rationales are replaced by random rationales from the same example banks. `Skill-KNN-zero` suffers from a 3% and 11.7% performance drop at the replacement rate of 30%, while `LaRS` experiences only a 0.1% and 1.9% performance drop under the same conditions.

6 Conclusions

This paper introduces LaRS, a novel demonstration selection method designed for CoT prompting. LaRS bases the selection on reasoning skills, which are latent representations discovered by unsupervised learning from rationales via a CVAE. Based on the experiments conducted across four LLMs and over four different reasoning tasks, LaRS manifests comparable performance on selecting effective few-shot examples for CoT reasoning while requiring no extra LLM inference and saving hours in pre-processing the example bank.

7 Limitations

Despite the success of LaRS, a few limitations and potential future directions are worth noting. First, the impact of the order of examples in the prompts is not considered. Introducing additional heuristics to sort the examples could potentially lead to better performances. Second, in the CVAE, the decoder is represented by an MLP neural network. However, it would be ideal to represent the decoder as a prompt-tuning module, which aligns better with the implicit skill model assumption. Finally, one single reasoning skill might not be sufficient to represent the entire rationale that might contain multiple steps of reasoning. Learning and selecting reasoning skills for each individual reasoning step is an interesting direction to explore.

References

Shengnan An, Zeqi Lin, Qiang Fu, B. Chen, Nanning Zheng, Jian-Guang Lou, and D. Zhang. 2023a. How do in-context examples affect compositional generalization? *ArXiv*, abs/2305.04835.

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. 2023b. Skill-based few-shot selection for in-context learning. *arXiv preprint arXiv:2305.14210*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel J. Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258.

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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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. *ArXiv*, abs/2005.14165.

Jiaao Chen, Xiaoman Pan, Dian Yu, Kaiqiang Song, Xiaoyang Wang, Dong Yu, and Jianshu Chen. 2023.

Skills-in-context prompting: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *ArXiv*, abs/2302.12246.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. *ArXiv*, abs/2012.15723.

Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2023. Coverage-based example selection for in-context learning. *ArXiv*, abs/2305.14907.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. *ArXiv*, abs/2203.08568.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. *ArXiv*, abs/2010.05465.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? In *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and A. Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *ArXiv*, abs/2209.14610.

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. *ArXiv*, abs/2104.08786.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Maxwell Nye, Anders Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *ArXiv*, abs/2112.00114.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John F. J. Mellor, Irina Higgins, Antonia

Creswell, Nathan McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446.

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

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. 2020. A mathematical exploration of why language models help solve downstream tasks. *ArXiv*, abs/2010.03648.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. In *Conference on Empirical Methods in Natural Language Processing*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Huai hsin Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*.

Colin Wei, Sang Michael Xie, and Tengyu Ma. 2021. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *ArXiv*, abs/2106.09226.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021a. An explanation of in-context learning as implicit bayesian inference. *ArXiv*, abs/2111.02080.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021b. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Tao Yu, Rui Zhang, Kai-Chou Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Z Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *ArXiv*, abs/1809.08887.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. 2022. Automatic chain of thought prompting in large language models. *ArXiv*, abs/2210.03493.

Appendix: LaRS: Latent Reasoning Skill for Chain-of-Thought Reasoning

A LaRS Demonstration Selection

A practical desmonstration selection process for LaRS that tackle the difficulty of sampling from an unknown distribution $P_E(Q, R \mid z)$ is described as follows. To begin with, LaRS learns reasoning skill encoder π_ϕ and reasoning policy q_ω. For a given test question Q_{test}, the desirable reasoning skill $z_{\text{test}} = \arg\max_z[\pi_\phi(z|f(Q_{\text{test}}))]$ can be computed using the reasoning policy. Subsequently, each example from the example bank can be scored based on the cosine similarity between z_{test} and z_{post}, where $z_{\text{post}} = \arg\max_z[q_\omega(z|Q, R))]$ represents the maximum likelihood skill of the current example. Finally, a CoT prompt can be constructed by selecting the top-k examples according to the computed scores. The step-by-step procedure is outlined in Algorithm 1.

Algorithm 1 Demonstration selection

Input: Test question Q_{test}, a pre-trained embedding model f, a reasoning policy $\pi_\phi(z|f(Q))$, a reasoning skill encoder $q_\omega(z|f(Q, R))$, and an example bank $\mathcal{D}_E = \{(Q^j, R^j)\}_j$.
Parameter: shot number k
Output: $(Q_1, R_1, Q_2, R_2, \cdots, Q_k, R_k)$

1: Compute $z_{\text{test}} \leftarrow$ mean of $\pi(z|f(Q_{\text{test}}))$
2: **for** each (Q^j, R^j) in $\mathcal{D}_E$ **do**
3: Compute $z^j_{\text{post}} \leftarrow$ mean of $q_\omega(z|f(Q^j, R^j))$
4: Compute $r^j = \frac{z_{\text{test}} \cdot {z^j_{\text{post}}}^\intercal}{|z_{\text{test}}| \cdot |z^j_{\text{post}}|}$
5: **end for**
6: Select top-k demonstrations with the largest r^j and sort them in ascending order, denoted as $(Q_1, R_1, Q_2, R_2, \cdots, Q_k, R_k)$.
7: **return** $(Q_1, R_1, Q_2, R_2, \cdots, Q_k, R_k)$ =0

B Experimental Details

B.1 Dataset

We provide a detailed description of the dataset and the split of train and test set as follows:

TabMWP (Lu et al., 2022) This dataset consists of semi-structured mathematical reasoning problems, comprising 38,431 open-domain grade-level problems that require mathematical reasoning on both textual and tabular data. We use the train set, containing 23,059 examples, to train our LaRS models, and test1k set containing 1K examples to evaluate the selection methods.

Spider (Yu et al., 2018) Spider is a large-scale text-to-SQL dataset. It includes a train set with 7,000 examples and a dev set with 1,034 examples. We use the train set to train our LaRS models, and the dev set as the test set to evaluate the selection methods.

COGS (Kim and Linzen, 2020) is a synthetic benchmark for testing compositional generalization in semantic parsing. We transform the output format in the same way as An et al. (2023a), and consider a mixture of two sub-tasks: primitive substitution (P.S.) and primitive structural alternation (P.A.). This results in a train set of 6916 examples to train our LaRS models and a test set of 1000 examples to evaluate the selection method.

GSM8k (Cobbe et al., 2021b) GSM8k is a dataset containing 8.5K high-quality, linguistically diverse grade school math word problems. It includes a train set of 7.5K problems and a test set of 1319 problems. We use the train set to train our LaRS models, and the test set to evaluate the selection methods.

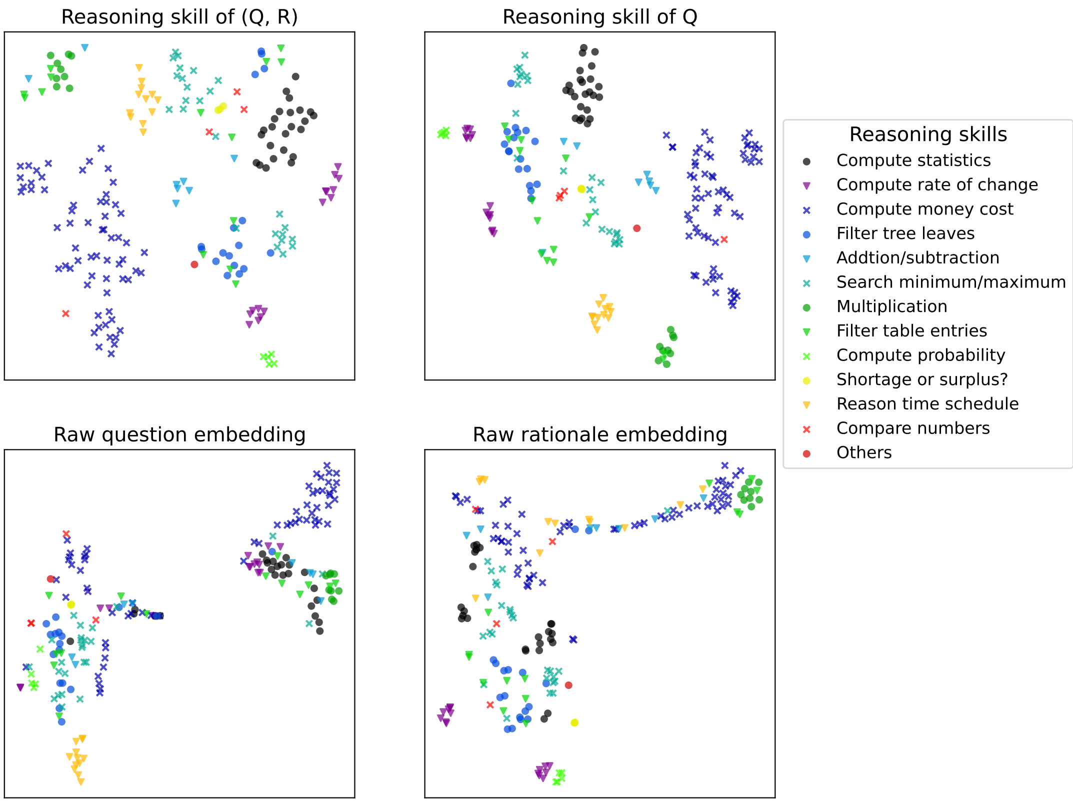


Figure 5: t-SNE projections of reasoning skills predicted from (Q, R) (top-left), reasoning skills predicted from Q (top-right), raw question embedding (bottom-left), and raw rationale embedding (bottom-right). The 12 different colors correspond to 12 skill labels provided by human.

B.2 LaRS Implementation Details

LaRS contains a encoder, a decoder, and a reasoning policy model. The reasoning skill is represented as a 128-dimensional continuous space. Both the encoder and the reasoning policy model are represented as a feed-forward multiple layer perception (MLP) with two 256-unit hidden layers, predicting the mean and variance of a multivariate Gaussian distribution in the latent space of reasoning skills. The decoder is a MLP with two 256-unit hidden layers that predicts a value in the embedding space deterministically. The dimension of the embedding space depends on the choice of pre-trained embedding models. The models are trained using the loss function in Equation 4 with a batch size of 256 and a learning rate of 0.0001 for 1000 epochs on a machine with 48 CPU cores and a Nvidia A40 GPU. Those hyper-parameters apply for all four datasets.

B.3 Skill-KNN Implementation Details

We used the same skill annotations as the original Skill-KNN implementation for COGS and Spider dataset. For TabMWP and GSM8K, we manually create skill annotations for 8 questions for each dataset. The new skill annotations are shown in Table 4 and 5.

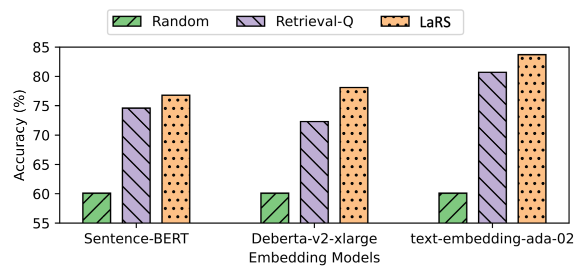
For Skill-KNN-zero with zero-shot generation of the skill description, the prompts used for the four datasets are shown in Table 6.

C Analysis and Ablation

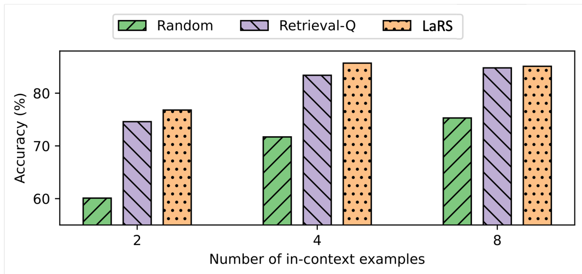
This section provides in-depth analysis and explains the reasoning of the success of LaRS .

Why reasoning skill is a better guidance for demonstration selection?

In **TabMWP** dataset, 200 examples are labeled based on the skills being showcased out of 12 manually-crafted skills labels, including “compute statistics”, “compute rate of change”, “Reason time schedule”, “Compute probability”, et. al. We investigate how the unsupervisedly discovered reasoning skills by LaRS align with human's understanding of skills. More specifically, a visualization of how human-labeled skills distribute based on the t-SNE projections of four different types of embedding is shown in Fig. 5. Both the reasoning skill encoder (reasoning skill of (Q, R)) and the reasoning policy (reasoning skill of Q) trained by LaRS demonstrate clear separation of the labeled 12 skills. At the mean time, the human-labeled skills are not well-separated by raw question embedding, and even raw rationale embeddings. This indicates that the discovered reasoning skills aligns well with human-labeled skills even without explicit labels being provided during the training. This sheds the light on why the demonstration selection based on similar reasoning skills can improve the CoT prompting.



(a) The accuracy of Random, Retrieval-Q, and, LaRS based on three different pre-trained embedding models.



(b) The accuracy of Random, Retrieval-Q, and LaRS using different number of in-context examples.

Figure 6: Performances of three different selection methods under (a) different pre-trained embedding models, and (b) different number of in-context examples.

Robustness to different pre-trained embedding models. Fig. 6a compares the performances of Random, Retrieval-Q, and LaRS based on three pre-trained embedding models, including Sentence-BERT (Reimers and Gurevych, 2019), Deberta-v2-xlarge, and, text-embedding-ada-02 (Neelakantan et al., 2022) from OpenAI. We observe that the performances of retrieval-based selection methods monotonously improve with more capable pre-trained embedding models. However, our LaRS shows consistent improvements over Retrieval-Q given the same embedding models.

Robustness to k: the number of in-context examples. This study compares three selection methods, including Random, Retrieval-Q, and LaRS under three different number of in-context examples 2, 4, and 8. The results are summarized in Fig. 6b. While the accuracy monotonously improves with the increasing number of in-context examples, LaRS consistently outperforms Retrieval-Q.

How does Skill-KNN perform under stricter conditions?

D Case Study

To explore the examples categorized as distinct skills within the learned latent reasoning skill representation, we employed K-means clustering on the latent reasoning skills of 1,000 examples from the **TabMWP** dataset. The centroids of these clusters are detailed in Table 8. The analysis presented in this table reveals that our method effectively discerns examples showcasing specific skills, such as "Searching minimum/maximum" and "Computing rate change".

E Theoretical Analysis

To prove Theorem 1, we start with the equation of rationale generation via CoT prompting, employing the skill-based demonstration selection method denoted as g_{skill}. The process can be formalized as follows:

$$P_M(R \mid Q, g_{skill}) = \int_{\mathcal{X}^k} P_M(R \mid pt)\Pi_{i=1}^{k}[g_{skill}(Q_i, R_i \mid Q)d(Q_i, R_i)] \tag{5}$$

where Equation 5 is integrated by substituting $pt = (Q_1, R_1, \cdots, Q_k, R_k, Q)$ as outlined in Equation 3, leading to:

$$P_M(R \mid Q, g_{skill}) = \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) \Pi_{i=1}^{k} [P_{skill}(z \mid Q)] dz \tag{6}$$

In this context, $P_{skill}(z \mid Q)$ is defined as:

$$P_{skill}(z \mid Q) = \int_{(Q',R') \in \mathcal{X}} P_M(z \mid Q', R') g_{skill}(Q', R' \mid Q) d(Q', R') dz' \tag{7}$$

Substituting the Definition 1 into Equation 7, leading to:

$$P_{skill}(z \mid Q) = \int_{(Q',R') \in \mathcal{X}} \int_{z' \in \mathcal{Z}} P_M(z \mid Q', R') P_E(Q', R' \mid z') P_E(z' \mid Q) dz' \tag{8}$$

Applying Assumption 2 into the above equation, replacing $P_M(z \mid Q', R')$ with $P_E(z \mid Q', R')$:

$$\begin{aligned} P_{skill}(z \mid Q) &= \int_{(Q',R') \in \mathcal{X}} \int_{z' \in \mathcal{Z}} P_E(z \mid Q', R') P_E(Q', R' \mid z') P_E(z' \mid Q) dz' \\ &= \int_{z' \in \mathcal{Z}} \delta(z = z') P_E(z' \mid Q) dz' \\ &= P_E(z \mid Q) \end{aligned} \tag{9}$$

By reintegrating the derived expression for $P_{skill}(z \mid Q)$ back into Equation 6, we arrive at:

$$P_M(R \mid Q, g_{skill}) = \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) \Pi_{i=1}^{k} [P_E(z \mid Q)] dz \tag{10}$$

Take the limit of $k \to \infty$, above equation siplifies to:

$$P_M(R \mid Q, g_{skill}) = \int_{\mathcal{Z}} P_M(R \mid z, Q) P_E(z \mid Q) dz \tag{11}$$

Applying Assumption 2 into the above equation, replacing $P_M(R \mid z, Q)$ with $P_E(R \mid z, Q)$:

$$P_M(R \mid Q, g_{skill}) = \int_{\mathcal{Z}} P_E(R \mid z, Q) P_E(z \mid Q) dz = P_E(R \mid Q) \tag{12}$$

According to Assumption 1, the example bank can approximate expert rationale generation, or $P_E(R \mid Q) = P^*(R \mid Q)$, we then conclude:

$$P_M(R \mid Q, g_{skill}) = P^*(R \mid Q) \tag{13}$$

Equation 13 means that the CoT prompting under the skill-based demonstration selection method give the optimal conditional distribution of rationales given questions by Definition 2. This proves the Theorem 1 under Assumption 1 and Assumption 2.

ID	Table	Question	Skill Description
1	Name | Score Jackson | 32 Madelyn | 31 Gary | 36 Suzie | 33 Edgar | 31 Ben | 32 Felipe | 29	Some friends played miniature golf and wrote down their scores. What is the range of the numbers?	To solve this problem, we need to find the greatest number and the least number. Then, subtract the least number from the greatest number.
2	x | y 17 | 13 18 | 6 19 | 2	The table shows a function. Is the function linear or nonlinear?	To solve this problem, we need to compare the rate of change between any two rows of the table.
3	box of tissues | $0.90of hand lotion | $0.94 tube of toothpaste | $0.84 package of dental floss | $0.85 box of bandages | $0.87 bottle of nail polish | $0.99	Sophie has $1.50. Does she have enough to buy a box of tissues and a package of dental floss?	To solve this problem, we need to compute the total cost and compare it with the budget.
4	Day | Number of fan letters Monday | 3,985 Tuesday | 1,207 Wednesday | 6,479 Thursday | 2,715 Friday | 8,078	An actor was informed how many fan letters he received each day. How many more fan letters were received on Friday than on Tuesday?	To solve the problem, we need to locate the two values in the table and do subtraction.
5	Stem | Leaf 3 | 1, 5, 7, 8 4 | 0, 3, 5, 5, 8 5 | 2, 4, 5, 7, 9 6 | 4, 5, 6 7 | 1, 1, 7, 8 8 | 9 | 0	Daniel counted the number of silver beads on each bracelet at Lowell Jewelry, the store where he works. What is the largest number of silver beads?	To solve this problem, we need to locate the largest number from a stem-and-leaf plot.
6	Number of tanks | Number of tadpoles 1 | 10 2 | 20 3 | 30 4 | 40 5 | ?	Each tank has 10 tadpoles. How many tadpoles are in 5 tanks?	To solve this problem, we need to complete the table according to the tendency of the columns.
7	| Blue sticker | Green sticker Front door of the house | 2 | 4 Back door of the house | 3 | 3	Lester keeps all his spare keys in a box under his bed. Recently, Lester decided the box was becoming unmanageable, as none of the keys were labeled. He set about labeling them with colored stickers that indicated what each key opened. What is the probability that a randomly selected key opens the front door of the house and is labeled with a green sticker? Simplify any fractions.	To solve this problem, we need to find the number of outcomes in the event and the total number of outcomes. Then compute the probability.
8	Sparrowtown | 8:00 A.M. | 2:00 P.M. | 4:45 P.M. Danville | 9:15 A.M. | 3:15 P.M. | 6:00 P.M. Princeton | 10:30 A.M. | 4:30 P.M. | 7:15 P.M. Westminster | 11:45 A.M. | 5:45 P.M. | 8:30 P.M. Oakdale | 1:30 P.M. | 7:30 P.M. | 10:15 P.M.	Look at the following schedule. Lee just missed the 4.30 P.M. train at Princeton. What time is the next train?	To solve this problem, we need to locate the entry from the table and read the next entry.

Table 4: Skill description annotation for TabMWP dataset.

ID	Question	Skill Description
1	Angela slept 6.5 hours every night in December. She decided she should get more sleep and began sleeping 8.5 hours a night in January. How much more sleep did Angela get in January?	To solve this question, we need to do subtraction, inference the total number of days in a month, and do multiplication.
2	Edith is a receptionist at a local office and is organizing files into cabinets. She had 60 files and finished organizing half of them this morning. She has another 15 files to organize in the afternoon and the rest of the files are missing. How many files are missing?	To solve this question, we need to do division, addition, and subtraction.
3	Rosalina receives gifts from three people on her wedding day. How many gifts did she get if Emilio gave 11 gifts, Jorge gave 6 gifts, and Pedro gave 4 gifts?	To solve this question, we need to do addition.
4	A store puts out a product sample every Saturday. The last Saturday, the sample product came in boxes of 20. If they had to open 12 boxes, and they had five samples left over at the end of the day, how many customers tried a sample if the samples were limited to one per person?	To solve this question, we need to do multiplication and subtraction.
5	Billy is counting the rings in two trees. Weather fluctuations in this area mean that each tree's rings are in groups of two fat rings and four thin rings. If Billy counts 70 ring groups in the first tree and 40 ring groups in the second tree, how much older is the first tree? (Trees grow 1 ring per year.)	To solve this question, we need to do addition, subtraction, and multiplication.
6	A group of six friends planned to buy a car. The cost of the car is $1700 and they plan to share the cost equally. They had a car wash to help raise funds, which would be taken out of the total cost. The remaining cost would be split between the six friends. At the car wash, they earn $500. However, Brad decided not to join in the purchase of the car. How much more does each friend have to pay now that Brad isn't participating?	To solve this question, we need to do subtraction, division, and multiplication.
7	In Fifi's closet, she hangs all of her clothes on colored plastic hangers. She has clothes hanging on 7 pink hangers, 4 green hangers, one less blue hanger than there are green hangers, and one less yellow hanger than there are blue hangers. What is the total number of colored hangers in Fifi's closet?	To solve this question, we need to do subtraction and addition.
8	At the family reunion, everyone ate too much food and gained weight. Orlando gained 5 pounds. Jose gained two pounds more than twice what Orlando gained. Fernando gained 3 pounds less than half of what Jose gained. How much weight, in pounds, did the three family members gain at their reunion?	To solve this question, we need to do multiplication, addition, and subtraction.

Table 5: Skill description annotation for GSM8K dataset.

Dataset	Prompt
TabMWP	Describe the required skills to solve the following problems based on the data from the tables in one sentence
GSM8K	Describe the required skills to solve the following questions in one sentence
Spider	Describe the needed skills to solve the task on the database schema in one sentence.
COGS	Describe the required skills to parse the following sentences in one sentence.

Table 6: Prompts for zero-shot skill generation.

Method	TabMWP	GSM8K	Spider	COGS
Backbone: gpt-3.5-turbo				
Skill-KNN-large	**78.3** +15.9	75.0 −0.7	**58.4** +11.6	94.6 +27.2
Skill-KNN-small	75.5 +13.2	74.9 −0.8	37.3 −9.5	79.9 +12.7
Skill-KNN-zero	77.7 +15.3	75.0 −0.7	49.0 +2.2	77.9 +10.8
LaRS (ours)	**78.1** +15.7	**76.8** +1.1	**53.0** +6.2	**94.8** +27.2
Backbone: gpt-4o				
Skill-KNN-large	80.6 +11.3	62.0 −0.2	56.3 +9.8	**96.8** +23.4
Skill-KNN-small	77.4 +8.1	62.3 +0.1	47.4 +0.3	79.4 +6.0
Skill-KNN-zero	87.7 +0.1	**78.6** −0.5	76.6 +2.5	78.1 +5.1
LaRS (ours)	**87.9** +0.3	78.3 +0.2	**77.2** +3.1	**90.2** +17.2
Backbone: claude-3-sonnet				
Skill-KNN-large		93.2 −0.1	**25.9** +7.6	96.2 +17.0
Skill-KNN-small		92.3 −1.0	18.2 −0.1	86.6 +7.4
Skill-KNN-zero	93.1 +0.5	92.1 −1.2	61.9 +0.2	86.6 +7.4
LaRS (ours)	**93.7** +1.1	**93.6** +0.3	**62.2** +0.5	**96.9** +17.7
Backbone: claude-3-haiku				
Skill-KNN-zero	93.3 +4.7	**88.8** +0.2	61.0 +0.8	79.7 +13.5
LaRS (ours)	**93.3** +4.7	87.6 −1.0	**61.3** +1.1	**89.9** +23.7
Backbone: Falcon-40B-Instruct				
Skill-KNN-large	55.9 +10.2	**40.3** +1.5	23.7 +2.9	81.0 +35.9
Skill-KNN-small	51.4 +5.7	36.5 −2.3	20.3 −0.3	59.4 +14.3
Skill-KNN-zero	55.2 +9.5	38.7 −0.1	23.3 +2.7	82.1 +37.0
LaRS (ours)	**57.7** +12.0	**39.1** +0.3	**24.8** +4.2	**89.5** +44.4

Table 7: Skill-KNN-large, Skill-KNN-small, and Skill-KNN-zero compare with LaRS .

Cluster ID	Table	Question	Skill
0	[TITLE]: School play committees Committee | Boys | Girls Casting | 17 | 5 Set design | 14 | 17 Lighting | 20 | 20 Costume | 7 | 4 Music | 2 | 13	Some students at Dayton Middle School signed up to help out with the school play. Which committee has the most boys? Options: (A) set design (B) lighting (C) casting (D) costume	Search minimum/maximum
1	[TITLE]: Pairs of shoes per store Stem | Leaf 1 | 9 2 | 3, 3 3 | 0, 2 4 | 2, 4 5 | 5, 7 6 | 2, 5 7 | 7 8 | 0, 2, 4, 4 9 | 0, 0	Ivan counted the number of pairs of shoes for sale at each of the shoe stores in the mall. How many stores have exactly 23 pairs of shoes?	Search tree leaves
2	[TITLE]: None piece of licorice | $0.07 gum drop | $0.05 gumball | $0.08 cinnamon candy | $0.01 peppermint candy | $0.08 lemon drop | $0.07	Derek has $0.06. Does he have enough to buy a piece of licorice and a cinnamon candy? Options: (A) yes (B) no	Compute money cost
3	[TITLE]: None Number of offices | Number of chairs 1 | 2 2 | 4 3 | 6 4 | 8 5 | ?	Each office has 2 chairs. How many chairs are in 5 offices?	Multiplication
4	[TITLE]: None popcorn balls | $1/kilogram coffee cake | $3/kilogram blueberry bars | $2/kilogram cream cheese bars | $2/kilogram lemon bars | $3/kilogram	Sarah went to the store and bought 2 kilograms of blueberry bars. How much did she spend? (Unit: $)	Compute money cost
5	[TITLE]: None x | y 12 | 19 13 | 9 14 | 2	The table shows a function. Is the function linear or nonlinear? Options: (A) linear (B) nonlinear	Compute rate of change
6	[TITLE]: Tractors Farmer | Number of tractors Farmer Judy | 4 Farmer Joe | 7 Farmer Megan | 7 Farmer Rick | 4 Farmer Jane | 4	Some farmers compared how many tractors they own. What is the mode of the numbers?	Compute statistics
7	[TITLE]: None pink sweater | $6.69 pair of brown pants | $9.66 plaid scarf | $2.45 pair of sandals | $7.69 white polo shirt | $4.86	How much money does Heather need to buy a pair of brown pants and a plaid scarf? (Unit: $)	Compute money cost
8	[TITLE]: Tour bus schedule Location | Arrive | Depart the riverfront | 9:55 A.M. | 10:20 A.M. the zoo | 10:35 A.M. | 11:30 A.M. art museum | 12:05 P.M. | 12:30 P.M. science museum | 1:00 P.M. | 1:45 P.M. skyscraper | 1:50 P.M. | 2:20 P.M. governor's mansion | 2:50 P.M. | 3:45 P.M. old building | 4:00 P.M. | 4:45 P.M. famous bridge | 5:15 P.M. | 5:40 P.M. the aquarium | 6:20 P.M. | 7:00 P.M. landmark sculpture | 7:45 P.M. | 8:20 P.M.	Look at the following schedule. Which stop does the bus depart from at 11.30 A.M.? Options: (A) zoo (B) riverfront (C) old building (D) science museum	Reason time schedule

Cluster ID	Table	Question	Skill
9	[TITLE]: None poppyseed muffin | $2.31 bowl of yogurt | $1.35 blueberry pancakes | $7.28 hash browns | $4.56 bowl of granola | $2.97 bagel with cream cheese | $2.56	Max has $13.33. How much money will Max have left if he buys a bagel with cream cheese and blueberry pancakes? (Unit: $)	Compute money cost
10	[TITLE]: Balloons sold Day | Number of balloons Wednesday | 568 Thursday | 586 Friday | 558 Saturday | 565	The manager of a party supply store researched how many balloons it sold in the past 4 days. On which day did the store sell the most balloons? Options: (A) Wednesday (B) Thursday (C) Friday (D) Saturday	Search minimum/maximum
11	[TITLE]: None forklift | $9,987.00 dump truck | $9,543.00 race car | $8,370.00 crane | $6,996.00 bulldozer | $7,547.00 hydrofoil | $8,047.00	How much more does a forklift cost than a dump truck? (Unit: $)	Compute money cost

Table 8: The closest examples to the 12 cluster centers computed by K-Means clustering method on reasoning skill latent variables.