Latent Skill Discovery for Chain-of-Thought Reasoning

Zifan Xu
University of Texas at Austin
zfxu@utexas.edu

Haozhu Wang
Amazon Web Service
haozhuw@amazon.com

Dmitriy Bespalov
Amazon Web Service
dbespai@amazon.com

Peter Stone
University of Texas at Austin
pstone@cs.utexas.edu

Yanjun Qi
Amazon Web Service
yanjunqi@amazon.com

Abstract

Recent advances in Large Language Models (LLMs) have led to an emergent ability of chain-of-thought (CoT) prompting, a prompt reasoning strategy that adds intermediate rationale steps between questions and answers to construct prompts. Conditioned on these prompts, LLMs can effectively learn in context to generate rationales that lead to more accurate answers than when answering the same question directly. To design LLM prompts, one important setting, called *demonstration selection*, considers selecting demonstrations from an example bank. Existing methods use various heuristics for this selection, but for CoT prompting, which involves unique rationales, it is essential to base the selection upon the intrinsic skills that CoT rationales need, for instance, the skills of addition or subtraction for math word problems.

To address this requirement, we introduce a novel approach named **R**easoning **S**kill **D**iscovery (`RSD`) that uses unsupervised learning to create a latent space representation of rationales, called a *reasoning skill*. Simultaneously, `RSD` learns a *reasoning policy* to determine the required reasoning skill for a given question. This can then guide the selection of examples that demonstrate the required reasoning skills. Our approach offers several desirable properties: it is (1) theoretically grounded, (2) sample-efficient, requiring no LLM inference or manual prompt design, and (3) LLM-agnostic. Empirically, `RSD` outperforms existing methods by up to 6% in terms of the answer accuracy across multiple reasoning tasks.

1 Introduction

Large Language Models (LLMs) exhibit remarkable capabilities in solving various downstream tasks through in-context learning (ICL) [4], even without being explicitly trained on the distribution of in-context examples [29, 10, 24, 7, 32]. 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 [24, 3, 23]. In the recent literature, chain-of-thought (CoT) prompting has 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 [9, 33, 28]. To achieve this, CoT prompts are composed of *demonstrations* that contains not only input-output pairs, but also rationales.

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 [20, 18].

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.

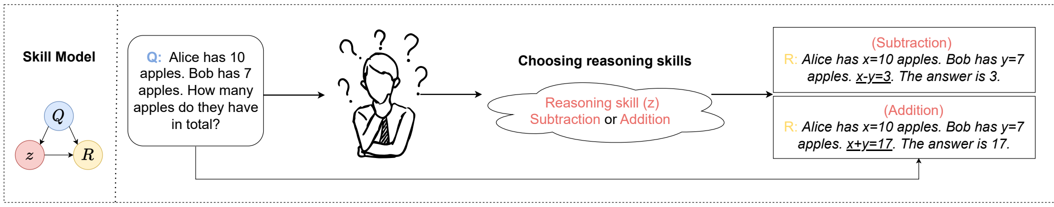

Figure 1: A diagram showing the generation of a rationale for a simple math word problem. The causal graph on the left shows the dependencies between question Q, reasoning skill z, and rationale R. Candidate rationales are shown as italicized font in the two rectangle panels.

To form a prompt, one important setting considers selecting demonstrations from an existing example bank, termed demonstration selection. 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.*

Motivated similarly, An et al. [2] propose to select examples with rationales showcasing requisite *skills* for solving specific reasoning problems. For instance, the skills for simple math word problems could be basic numerical operations, including addition, subtraction, multiplication, and division. These skill labels are automatically annotated by pre-trained LLMs. While intuitively appealing, it is unclear why the selection based on these skill labels can be effective for CoT prompting. Furthermore, their skill annotations tend to be somewhat arbitrary, heavily reliant on human prompt design for LLM annotations.

To address these challenges, this paper introduces *skill model*, a plausible framework for explaining the generation of rationales, drawing inspiration from a modified topic model [30]. By representing a rationale as R, a reasoning question as Q and introducing a latent variable $z \in \mathcal{Z}$ representing "skills" with $\mathcal{Z}$ being the latent space, the generation of R can be formulated as follows:

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

This generation process involves sampling a latent variable z based on Q through a posterior distribution denoted as $P(z \mid Q)$. Subsequently, the probability of generating a rationale R is conditioned on both z and Q. Within this framework, the "skills" can be formally defined as the latent variable z. Conditioned on z and Q, the rationales can be generated. We call z a *reasoning skill* and the posterior $P(z \mid Q)$ a *reasoning policy*. This reasoning policy encodes a strategic preference of reasoning skills based on a particular question. Fig. 1 shows an illustration of this generation process. Further justification for this formulation is provided in Section 2.1.

Under this formulation, a skill-based selection method, akin to An et al. [2], can be precisely described as selecting examples showcasing reasoning skills that align with an optimal choice of reasoning skills that maximize answer accuracy for a given question. A formal definition of skill-based selection is provided in Section 2.3. Moreover, Appendix E presents a theoretical analysis underscoring the optimality of this skill-based demonstration selection method.

Prior methods annotate reasoning skills by humans or LLMs. For example, Chen et al. [5] use a few discrete labels of known reasoning skills, from which humans or LLMs can use to label examples. However, such annotations can be resource-intensive and vary across different tasks. To mitigate the challenge, in this paper, we propose a novel framework called `RSD` that uses unsupervised learning to discover reasoning skills in an *expert-generated* example bank of question-rationale pairs (with no skill labels). Under our formulation of skill model, `RSD` learns the generation of rationales with a conditional variational autoencoder (CVAE). As a result, two probabilistic models can be learned concurrently: (1) a *reasoning skill encoder*, approximating $P(z \mid Q, R)$ to map question-rationale pairs to reasoning skills; (2) a *reasoning policy*, approximating $P(z \mid Q)$ that reflects an expert's preference of selecting reasoning skills. These discovered reasoning skills can be utilized to perform the skill-based example selection for CoT prompting.

The effectiveness of `RSD` is evaluated on four different benchmarks based on four backbone LLMs with varying scales. The method is also compared with reasonable baselines, including an oracle method that assumes access to ground truth rationales. `RSD` achieves improvements of up to 6% over non-oracle baselines with similar computational cost and matches the oracle performance in almost half of the experiments. In summary, the paper presents three major contributions: (1) We introduce the *skill model*, a plausible formulation for CoT reasoning, and empirically verify its effectiveness through four sets of experiments; (2) We propose `RSD`, a novel unsupervised demonstration selection approach for CoT prompting, which is both sample-efficient and LLM-agnostic; (3) We introduce theoretical analysis that grounds the skill-based selection method.

2 Formulation

In this section, we formally describe the *skill model*, a new formulation for explaining the generation of rationales in CoT reasoning. In Section 2.1, the skill model is first introduced to describe the human-generated rationales. Then, Section 2.2 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 skill-based demonstration selection method is formally described in Section 2.3.

2.1 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. The 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 is given as follows:

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

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. 1 on the left.

Unlike Wang et al. [30], this formulation considers a dependency of z on Q reflecting a preference for selecting specific reasoning skills given a question. We justify this formulation as follows. First, rationales can exhibit remarkable flexibility, manifesting diverse formats, topics, and knowledge, which can naturally be abstracted into a high-level concept of skills. Second, 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 $P_H(z \mid Q)$.

2.2 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 [30, 34]. 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. 2):

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

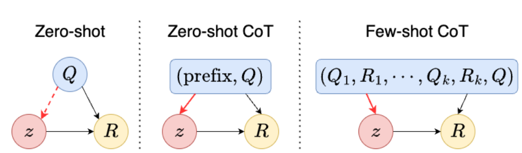

Figure 2: Causal graphs for prompting with zero-shot (left), zero-shot CoT (middle), and few-shot CoT (right) for generating rationales via skills.

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

[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.

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:

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

$$\text{0-shot CoT: } pt = (\text{prefix}, Q) \text{ or } (Q, \text{prefix}); \quad k\text{-shot CoT: } pt = (Q_1, R_1, \cdots, Q_k, R_k, Q) \tag{4}$$

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 [27, 21].

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 instructions or few-shot examples.

2.3 Skill-Based Demonstration Selection

The analysis above suggests that the key to the success of CoT prompting is to design an effective prompt that shapes the posterior distribution of reasoning skills, assuming that the real-world distribution $P_H(z \mid Q)$ is potentially sub-optimal. In contrast to the real-world distribution, 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 example selection is given by*

$$g_{RSD}(Q, R \mid Q_{test}) = \int_{\mathcal{Z}} P_E(Q, R \mid z) P_E(z \mid Q_{test}) dz \tag{5}$$

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

3 Method

To enable the skill-based demonstration selection (Definition 1), we introduce our approach RSD, 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 RSD (right) and the corresponding demonstration selection process (left) are illustrated in Figure 3.

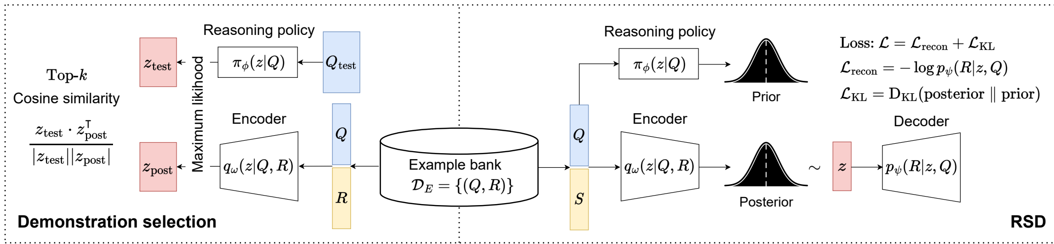

Figure 3: An overview of RSD and the demonstration selection process.

3.1 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 RSD 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. Detailed description of this training process is presented in Appendix B.1.

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. Once the encoder and the reasoning policy are learned, the demonstration can effectively be selected by the procedure described in Appendix B.2.

4 Experiments

Section 4.1 describes the experiment settings including benchmarks, selection methods, and backbone models. Then the main results of these experiments are presented in Section 4.2.

4.1 Selection Methods and Datasets

We refer to the skill-based selection by our RSD approach as Retrieval-RSD , which is compared with three baselines detailed as follows: (1) **Random**, which randomly selects k in-context examples from the example bank; (2) **Retrieval-Q**, which selects in-context examples based on the cosine similarity between embeddings from examples' questions and the test question; (3) **Retrieval-R (oracle)**, which selects in-context examples based on the cosine similarity between embeddings from examples' rationales and the ground-truth rationale. Detailed description and all the hyper-parameters related to these methods are listed in Appendix C.

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 RSD 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 C.

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.

In terms of the backbone models, the ICL is conducted by two OpenAI language models: text-davinci-003 and gpt-3.5-turbo, one Anthropic model: Claude-v2, and one smaller-scale Falcon-40B-Instruct [36]. All the embedding is computed by a pre-trained embedding model, Deberta-v2-xlarge [14]. We also investigate different choices of embedding model in Appendix D.

4.2 Main results

Table 1 presents a summary of the results. Detailed descriptions are provided as follows. In-depth analysis and ablation studies are presented in Appendix D

Backbone	Method	TabMWP	GSM8K	Spider	COGS
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 $_{+21.0}$
	Retrieval-RSD (ours)	**78.1** $_{+15.7}$	**76.8** $_{+1.1}$	**53.0** $_{+6.2}$	**94.6** $_{+27.1}$
	Retrieval-R (oracle)	77.4 $_{+15.0}$	75.5 $_{-0.2}$	64.4 $_{+17.6}$	95.7 $_{+28.2}$
text-davinci-003	Random	69.3 $_{+0.0}$	62.2 $_{+0.0}$	47.1 $_{+0.0}$	73.4 $_{+0.0}$
	Retrieval-Q	76.5 $_{+7.2}$	62.7 $_{+0.5}$	**50.2** $_{+2.9}$	92.1 $_{+18.7}$
	Retrieval-RSD (ours)	**80.8** $_{+11.5}$	**62.7** $_{+0.5}$	48.6 $_{+1.5}$	**96.6** $_{+23.2}$
	Retrieval-R (oracle)	80.4 $_{+11.1}$	63.8 $_{+1.6}$	67.3 $_{+20.2}$	97.3 $_{+23.9}$
Claude-v2	Random	77.7 $_{+0.0}$	86.9 $_{+0.0}$	40.2 $_{+0.0}$	77.6 $_{+0.0}$
	Retrieval-Q	80.1 $_{+2.4}$	88.2 $_{+1.3}$	45.5 $_{+5.3}$	93.5 $_{+15.9}$
	Retrieval-RSD (ours)	**80.9** $_{+3.2}$	**88.3** $_{+1.4}$	**47.7** $_{+7.5}$	**96.6** $_{+19.0}$
	Retrieval-R (oracle)	80.3 $_{+2.6}$	88.4 $_{+1.5}$	60.8 $_{+20.6}$	97.3 $_{+19.7}$
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 $_{+28.8}$
	Retrieval-RSD (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.

Retrieval-RSD outperforms Retrieval-Q. Across all four benchmarks and four backbone models tested, our proposed Retrieval-RSD consistently outperforms Retrieval-Q, which searches nearest neighbors based on the raw embedding of questions. This observation suggests that the success of demonstration selection lies in the learned reasoning skill representation rather than relying solely on the raw information provided by the question.

Retrieval-RSD is LM-agnostic. The superiority of RSD is consistent across four different LMs, including a small-scale Falcon-40B-Instruct, while not trained specifically for any of these LMs. This finding highlights the universality of the learned reasoning skill representation, allowing any LMs to benefit from it.

5 Conclusions

This paper introduces RSD, a novel demonstration selection method designed for CoT prompting. RSD bases the selection on reasoning skills, which are latent representations discovered by unsupervised learning via a CVAE. The effectiveness of RSD is empirically supported by the experiments conducted across four LLMs and over four different reasoning tasks. We discuss limitations of this work in Appendix F.

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

References

[1] An, S., Lin, Z., Fu, Q., Chen, B., Zheng, N., Lou, J.-G., and Zhang, D. (2023a). How do in-context examples affect compositional generalization? *ArXiv*, abs/2305.04835.

[2] An, S., Zhou, B., Lin, Z., Fu, Q., Chen, B., Zheng, N., Chen, W., and Lou, J.-G. (2023b). Skill-based few-shot selection for in-context learning. *arXiv preprint arXiv:2305.14210*.

[3] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T. F., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J. F., Ogut, G., Orr, L. J., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y. H., Ruiz, C., Ryan, J., R'e, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K. P., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M. A., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258.

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T. J., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *ArXiv*, abs/2005.14165.

[5] Chen, J., Pan, X., Yu, D., Song, K., Wang, X., Yu, D., and Chen, J. (2023a). Skills-in-context prompting: Unlocking compositionality in large language models.

[6] Chen, J., Pan, X., Yu, D., Song, K., Wang, X., Yu, D., and Chen, J. (2023b). Skills-in-context prompting: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*.

[7] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N. M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B. C., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., García, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Díaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K. S., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2022). Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

[8] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021a). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

[9] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. (2021b). Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

[11] Diao, S., Wang, P., Lin, Y., and Zhang, T. (2023). Active prompting with chain-of-thought for large language models. *ArXiv*, abs/2302.12246.

[12] Gao, T., Fisch, A., and Chen, D. (2021). Making pre-trained language models better few-shot learners. *ArXiv*, abs/2012.15723.

[13] Gupta, S., Singh, S., and Gardner, M. (2023). Coverage-based example selection for in-context learning. *ArXiv*, abs/2305.14907.

[14] He, P., Liu, X., Gao, J., and Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

[15] Hu, Y., Lee, C.-H., Xie, T., Yu, T., Smith, N. A., and Ostendorf, M. (2022). In-context learning for few-shot dialogue state tracking. *ArXiv*, abs/2203.08568.

[16] Kim, N. and Linzen, T. (2020). Cogs: A compositional generalization challenge based on semantic interpretation. *ArXiv*, abs/2010.05465.

[17] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

[18] Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. (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*.

[19] Lu, P., Qiu, L., Chang, K.-W., Wu, Y. N., Zhu, S.-C., Rajpurohit, T., Clark, P., and Kalyan, A. (2022). Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *ArXiv*, abs/2209.14610.

[20] Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. (2021). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *ArXiv*, abs/2104.08786.

[21] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

[22] Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy, C., et al. (2022). Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

[23] Nye, M., Andreassen, A., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., Sutton, C., and Odena, A. (2021). Show your work: Scratchpads for intermediate computation with language models. *ArXiv*, abs/2112.00114.

[24] Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P.-S., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J. F. J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S. M., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J.-B., Tsimpoukelli, M., Grigorev, N. K., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., de Masson d'Autume, C., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., de Las Casas, D., Guy, A., Jones, C., Bradbury, J., Johnson, M. G., Hechtman, B. A., Weidinger, L., Gabriel, I., Isaac, W. S., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K. W., Stanway, J., Bennett, L. L., Hassabis, D., Kavukcuoglu, K., and Irving, G. (2021). Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446.

[25] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[26] Saunshi, N., Malladi, S., and Arora, S. (2020). A mathematical exploration of why language models help solve downstream tasks. *ArXiv*, abs/2010.03648.

[27] Shin, T., Razeghi, Y., IV, R. L. L., Wallace, E., and Singh, S. (2020). Eliciting knowledge from language models using automatically generated prompts. In *Conference on Empirical Methods in Natural Language Processing*.

[28] Suzgun, M., Scales, N., Scharli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., hsin Chi, E. H., Zhou, D., and Wei, J. (2022). Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*.

[29] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS*.

[30] Wang, X., Zhu, W., and Wang, W. Y. (2023). Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*.

[31] Wei, C., Xie, S. M., and Ma, T. (2021). Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *ArXiv*, abs/2106.09226.

[32] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., hsin Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022a). Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

[33] Wei, J., Wang, X., Schuurmans, D., Bosma, M., hsin Chi, E. H., Xia, F., Le, Q., and Zhou, D. (2022b). Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

[34] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021a). An explanation of in-context learning as implicit bayesian inference. *ArXiv*, abs/2111.02080.

[35] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021b). An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

[36] Xu, C., Guo, D., Duan, N., and McAuley, J. (2023). Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

[37] Yu, T., Zhang, R., Yang, K.-C., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I. Z., Yao, Q., Roman, S., Zhang, Z., and Radev, D. R. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *ArXiv*, abs/1809.08887.

[38] Zhang, Z., Zhang, A., Li, M., and Smola, A. J. (2022). Automatic chain of thought prompting in large language models. *ArXiv*, abs/2210.03493.

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

A Related Work

A.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. [33] 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. [17] 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 `RSD` can be thought of as automatic discovery of diverse rationale styles, termed as *reasoning skills*, from an example bank. This method can also dynamically select reasoning skills based on the specific questions. Worth noting, Chen et al. [6] introduces SKills-in-Context (SKiC), which confines rationale generation to predefined "skills" within the prompt. Although sharing a similar motivation to `RSD`, we emphasize two crucial distinctions: (1) while SKiC relies on manual "skills" design, `RSD` automatically discovers them, (2) SKiC presents a full list of "skills" in the prompt, allowing LLMs to select from them, whereas `RSD` learns the skill selection from the example bank, explicitly instructing LLMs on which skill to employ through in-context examples.

A.2 Demonstration Selection for Prompt Design

Demonstration selection refers to a special setting, where the prompts are constructed by selecting examples from an example bank. In this context, our `RSD` 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 [12, 15], diversity [38], coverage [13], and uncertainty [11]. Among these, Skill-KNN (An et al. [2]) 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, `RSD` automatically discovers reasoning skills by learning a very lightweight CVAE. In addition, the selections based on these discovered reasoning skills are theoretically-grounded by a plausible explanation for ICL.

B Details of Reasoning Skill Discovery

B.1 Training Loss

The CVAE in `RSD` is trained by the classical variational expectation maximization that optimizes a loss function as follows:

$$\mathcal{L}_{\text{CVAE}}(\phi, \omega, \psi) = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} \tag{6}$$

$$\mathcal{L}_{\text{recon}} = -\mathbb{E}_{(Q,R)\sim\mathcal{D}_E, z\sim q_\omega(|Q,R)}[\log p_\psi(R|z, Q)] \tag{7}$$

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{(Q,R)\sim\mathcal{D}_E}[\mathrm{D}_{\text{KL}}(q_\omega(z \mid Q, R) \parallel \pi_\phi(z \mid Q))] \tag{8}$$

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.

B.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 Appendix B.2, with a summarized visual representation provided in Fig. 3 (left).

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: $(Q_1, R_1, Q_2, R_2, \cdots, Q_k, R_k)$

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

C Experimental Details

C.1 Selection Methods

We refer to the skill-based selection by our `RSD` approach as Retrieval-`RSD` , which is compared with the following three baselines.

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.

C.2 Dataset

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

TabMWP [19] 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 `RSD` models, and test1k set containing 1K examples to evaluate the selection methods.

Spider [37] 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 `RSD` models, and the dev set as the test set to evaluate the selection methods.

COGS [16] is a synthetic benchmark for testing compositional generalization in semantic parsing. We transform the output format in the same way as An et al. [1], 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 `RSD` models and a test set of 1000 examples to evaluate the selection method.

GSM8k [8] 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 `RSD` models, and the test set to evaluate the selection methods.

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[3]. For **COGS**, we report the exact-match accuracy for semantic parsing.

C.3 Hyper-parameters

`RSD` 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 6 with a batch size of 256 and a learning rate of 0.0001 for 1000 epochs. Those hyper-parameters apply for all four datasets.

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

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

D Analysis and Ablation

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

What skills does RSD discover? 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 RSD 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. 1. Both the reasoning skill encoder (reasoning skill of (Q, R)) and the reasoning policy (reasoning skill of Q) trained by RSD 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.

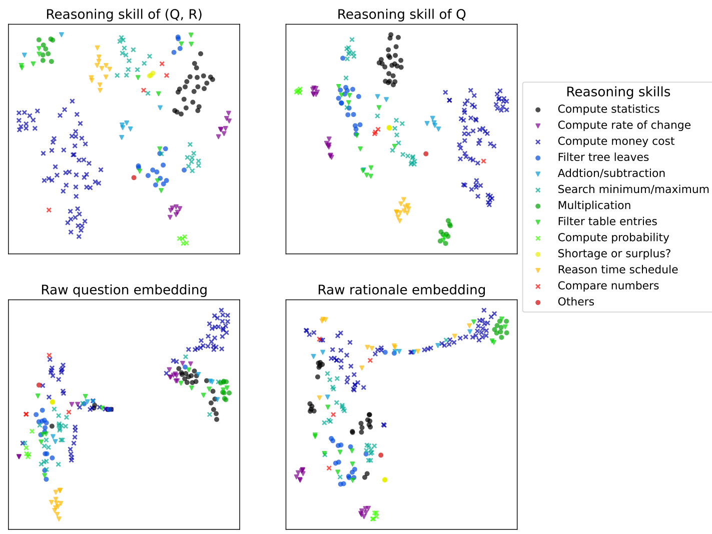

Figure 1: t-SNE projections of reasoning skills predicted from (Q, R) (top-left), reasoning skills predicted from Q (top-right), raw question embedding (bottom-left), and raw rationale embedding (bottom-right). The 12 different colors correspond to 12 skill labels provided by human.

Robustness to different pre-trained embedding models. Fig. 2 compares the performances of Random, Retrieval-Q, and Retrieval-RSD based on three pre-trained embedding models, including Sentence-BERT [25], Deberta-v2-xlarge, and, text-embedding-ada-02 [22] from OpenAI. We observe that the performances of retrieval-based selection methods monotonously improve with more capable pre-trained embedding models. However, our Retrieval-RSD 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 Retrieval-RSD under three different number of in-context examples 2, 4, and 8. The results are summarized in Fig. 3. While the accuracy monotonously improves with the increasing number of in-context examples, Retrieval-RSD consistently outperforms Retrieval-Q.

E Theoretical Analysis

In this section, we provide theoretical analysis on 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:

$$P_M(R \mid Q, g) = \int_{\mathcal{X}^k} P_M(R \mid pt)\Pi_{i=1}^{k}[g(Q_i, R_i \mid Q)d(Q_i, R_i)] \tag{9}$$

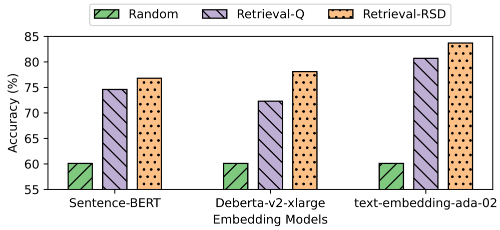

Figure 2: The accuracy of Random, Retrieval-Q, and, Retrieval-RSD based on three different pre-trained embedding models.

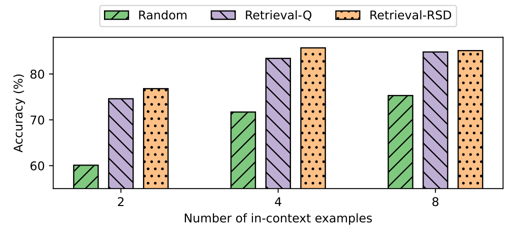

Figure 3: The accuracy of Random, Retrieval-Q, and Retrieval-RSD using different number of in-context examples.

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 $g = g_{\texttt{RSD}}$. 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:*

$$P^*(R \mid Q) = \underset{P(\cdot|Q)\in\Delta(\mathcal{X})}{\arg\max} \int_{\mathcal{X}} \mathbb{1}(R, Q) P(R \mid Q) dR \tag{10}$$

Here $\mathbb{1}(R, Q)$ is the indicator function of the correctness of R given a question Q (see Section 2.1).

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 $P_H(R \mid z, Q) = P_E(R \mid z, Q) = P_M(R \mid z, Q)$ and $P_H(z \mid Q, R) = P_E(z \mid Q, R) = P_M(z \mid Q, R)$.*

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 [35, 26, 31]. $P_E(R \mid z, Q) = P_H(R \mid z, Q)$ 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:*

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

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

To prove Theorem 1, we start by extending CoT prompting under the skill-based demonstration selection method $g_{\texttt{RSD}}$ as follows:

$$\begin{aligned}
&P_M(R \mid Q, g_{\texttt{RSD}}) \\
&= \int_{\mathcal{X}^k} P_M(R \mid pt) \Pi_{i=1}^k [g_{\texttt{RSD}}(Q_i, R_i \mid Q) d(Q_i, R_i)] \\
&\text{\# Plug in Equation 4 and } pt = (Q_1, R_1, \cdots, Q_k, R_k, Q) \\
&= \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) \Pi_{i=1}^k [\int_{(Q_i,R_i)\in\mathcal{X}} P_M(z \mid Q_i, R_i) g_{\texttt{RSD}}(Q_i, R_i \mid Q) d(Q_i, R_i)] dz
\end{aligned}$$

$$= \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) \Pi_{i=1}^{k} [P_{\texttt{RSD}}(z \mid Q)] dz \tag{11}$$

Here,

$$\begin{aligned}
&P_{\texttt{RSD}}(z \mid Q) \\
&= \int_{(Q',R') \in \mathcal{X}} P_M(z \mid Q', R') g_{\texttt{RSD}}(Q', R' \mid Q) d(Q', R') \\
&= \int_{(Q',R') \in \mathcal{X}} \int_{z' \in \mathcal{Z}} P_M(z \mid Q', R') P_E(Q', R' \mid z') P_E(z' \mid Q) dz' \\
&\text{\# Plug in Assumption 2 with } P_M(z \mid Q', R') = P_E(z \mid Q', R') \\
&= \int_{(Q',R') \in \mathcal{X}} \int_{z' \in \mathcal{Z}} P_E(z \mid Q', R') P_E(Q', R' \mid z') P_E(z' \mid Q) dz' \\
&= \int_{z' \in \mathcal{Z}} \delta(z = z') P_E(z' \mid Q) dz' \\
&= P_E(z \mid Q)
\end{aligned} \tag{12}$$

Plug Equation 10 back into Equation 9, we have

$$\begin{aligned}
P_M(R \mid Q, g_{\texttt{RSD}}) &= \int_{\mathcal{Z}} P_M(R \mid z, Q) P_M(z \mid Q) \Pi_{i=1}^{k} [P_E(z \mid Q)] \\
&\# k \to +\infty \\
&= \int_{\mathcal{Z}} P_M(R \mid z, Q) P_E(z \mid Q) dz \\
&\text{\# Plug in Assumption 2 with } P_M(R \mid z, Q) = P_E(R \mid z, Q) \\
&= \int_{\mathcal{Z}} P_E(R \mid z, Q) P_E(z \mid Q) dz \\
&\text{\# Plug in Assumption 1} \\
&= P^*(R \mid Q)
\end{aligned} \tag{13}$$

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 10. This proves the Theorem 1 under Assumption 1 and Assumption 2.

F Limitations

Despite the success of RSD, 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 performance. 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 formulation that aligns better with the implicit skill model assumption. Finally, one single reasoning skill might not be sufficient to effectively 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 next.