$\mathcal{V}$-SYNTHESIS: Task-Agnostic Synthesis of Consistent and Diverse In-Context Demonstrations from Scratch via $\mathcal{V}$-Entropy

Anonymous ACL submission

Abstract

High labeling cost for in-context learning (ICL) demonstrations motivates using large language models (LLMs) for synthesis to reduce overhead. However, existing synthesis methods are mainly task-specific or rely on pre-existing demonstrations. So this paper focuses on synthesizing demonstrations from scratch for arbitrary tasks. A major challenge in synthesizing from scratch is ensuring consistency with the target task, as the lack of labeling guidance could lead to synthesis bias. We first propose a consistency metric called $\mathcal{V}$-SCORE, which has higher performance and lower computation cost compared with the metrics based on grams or embedding vectors. Furthermore, we introduce $\mathcal{V}$-SYNTHESIS, which leverages $\mathcal{V}$-SCORE for proportional sampling to ensure both high consistency and diversity of synthesized demonstrations. Experimental results demonstrate that $\mathcal{V}$-SYNTHESIS yields an average performance improvement of 2.0% compared to existing synthesis methods confirming the effectiveness of $\mathcal{V}$-SYNTHESIS[1].

1 Introduction

In-context learning (ICL) is an effective approach to enhancing the performance of large language models (LLMs) (Brown et al., 2020; Dong et al., 2024). By providing task-relevant demonstrations within the input, ICL guides the reasoning process for the given user question, thereby improving the performance. However, the reliance on human-labeled demonstrations limits the applicability of ICL under the data-insufficient scenario. To address this limitation, many works propose synthesizing demonstrations for the target task (Long et al., 2024). Some works design the synthesis procedure for the given tasks (He et al., 2024; Chang and Fosler-Lussier, 2023). Other works enrich existing demonstrations based on labeled data (Wang et al., 2024, 2025a; Su et al., 2024).

[1] Our code and data will be released upon acceptance.

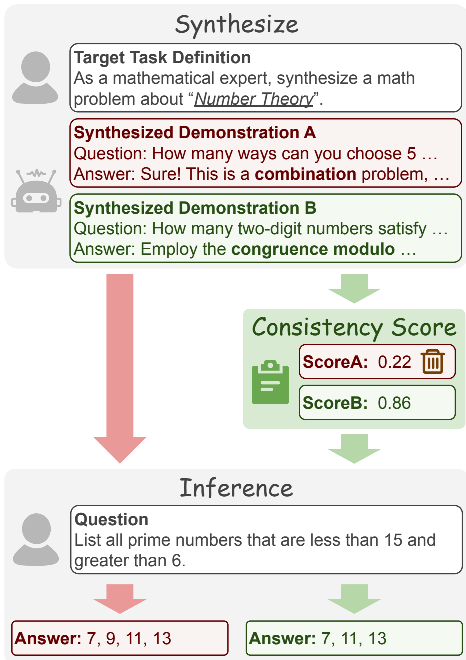

Figure 1: The previous work (left) compared with our method (right). The previous work directly uses the synthesized demonstrations inconsistency with the target task, leading to the incorrect answer. Our method calculates the consistency score of synthesized demonstrations, filtering the results of low scores to ensure high consistency, leading to the correct answer.

However, the above works depend on the existing labeled data or can only be applied to specific tasks, limiting their application. While *how to synthesize demonstrations from scratch for arbitrary tasks* is still under discovery. Although some studies propose to synthesize demonstrations directly based on the task definition, such methods have only been evaluated on relatively simple tasks (e.g., coin flip, causal judgement) (Chen et al., 2023a). For more complex tasks, the lack of guidance with demonstrations could lead to the generation irrelevant to the target task, which negatively impact the performance of ICL (Liu et al., 2022; Dong et al., 2024), as shown in Figure 1. Therefore, ***enhancing the consistency between the synthesized demonstrations and the target task*** is one of the key to improving the quality of synthesized demonstrations. Furthermore, prior work has shown that the demonstration diversity also significantly impacts ICL performance (Levy et al., 2023; Wang et al., 2024). Therefore, in this paper, we primarily discuss ***how to synthesize demonstrations from scratch with high consistency and diversity***.

In this paper, we mainly discuss how to enhance the consistency betweem synthesized demonstrations and the target task. Existing metrics based on grams (Broder et al., 1997) or embeedings (Singhal and Google, 2001) suffer from suboptimal performance and computational inefficiency due to their reliance on external models, leading to additional calculation and the embedding space gap (Biś et al., 2021). Therefore, we first propose a novel consistency metric called $\mathcal{V}$-SCORE, which measures how much information in demonstrations is learned from the given task definition. Since $\mathcal{V}$-SCORE can be calculated using the synthesis model, it alleviates additional calculation and embedding gap, having better performance and efficiency. Then, we propose $\mathcal{V}$-SYNTHESIS, which synthesizes demonstrations iteratively based on $\mathcal{V}$-SCORE. During the synthesis of each iteration, we sample demonstrations proportionally to their $\mathcal{V}$-SCORE, ensuring consistency while promoting diversity.

To validate the effectiveness of $\mathcal{V}$-SYNTHESIS, we conduct experiments on four mainstream datasets covering different tasks and domains. Our experimental results demonstrate that $\mathcal{V}$-SYNTHESIS yields an 2.0% performance gain over previous methods and a 3.4% average gain using alternative consistency metrics, proving the effectiveness of $\mathcal{V}$-SCORE and $\mathcal{V}$-SYNTHESIS. Additionally, analysis using demonstrations with different consistency and diversity shows that $\mathcal{V}$-SYNTHESIS successfully synthesizes demonstrations with high consistency and diversity.

Our contributions are as follows:

1. We propose $\mathcal{V}$-SYNTHESIS, which can better reflect the consistency between the synthesized demonstration and the target task with lower computation cost compared to existing metrics.
2. Based on the metric $\mathcal{V}$-SCORE, we introduce $\mathcal{V}$-SYNTHESIS, which is a consistency-weighted sampling method that ensures consistency while enhancing the diversity of demonstrations.
3. On four mainstream datasets, $\mathcal{V}$-SYNTHESIS achieves an average improvement of 2.0% compared to previous synthesis methods, demonstrating the effectiveness of $\mathcal{V}$-SCORE.

2 Related Works

Demonstration Synthesis Considering that previous ICL works rely on human-labeled demonstrations (Dong et al., 2024), which limits the application of ICL in low-resource scenarios, many researchers propose using LLMs to synthesize demonstrations (Long et al., 2024). These approaches can generally be divided into two categories: demonstrations synthesis based on existing labeled data and synthesis for specific tasks. Methods based on existing labeled data mainly focus on enhancing the quality of the demonstrations, such as increasing the diversity of demonstrations (Su et al., 2024) or modifying existing demonstrations based on user questions (He et al., 2024; Sarukkai et al., 2025). Task-specific synthesis designs the demonstration synthesis according to the characteristics of the task, such as executing the synthesized SQL for the text-to-SQL task (Chang and Fosler-Lussier, 2023; Wang et al., 2024) or transferring the existing demonstrations from similar tasks for the target task (Wang et al., 2025a).

However, current demonstration synthesis methods are primarily based on existing demonstrations or are task-specific, lacking methods for synthesizing demonstrations from scratch for arbitrary tasks. Although preliminary research exists, it mainly focuses on simple tasks (Chen et al., 2023a). For more complex tasks, due to the lack of guidance of demonstrations, models could misunderstand the task definition, leading to synthesized demonstrations not consistent with the target task (Dong et al., 2024). Therefore, in this paper, we discuss how to enhance the consistency between synthesized demonstrations and the target task to improve the performance of ICL.

Consistency Measurement The consistency metric is used to measure the degree of consistency between two texts, which is widely applied in tasks such as retrieval (Zhu et al., 2024; Shrivastava and Li, 2014) and deduplication. Early research primarily focused on gram-based methods to measure consistency, including algorithms like n-gram (Broder et al., 1997) and BM25 (Robertson and Zaragoza, 2009; Li et al., 2023). To address the limitation of gram-based approaches in capturing deep semantic information, many methods have been proposed that encode texts into semantic vectors, using the similarity between these vectors as a consistency metric (Singhal and Google, 2001; Mikolov et al., 2013; Yang et al., 2023; Luo et al., 2023). More recently, with the advent of powerful LLMs, there has been significant research into how LLMs can be directly utilized to assess the consistency between given texts (Wan et al., 2025).

However, the aforementioned metrics are hard to apply directly to the task of assessing the consistency between the synthesized demonstrations and the task definition. Gram-based methods exhibit poor performance, methods based on embedding vectors suffer from the gap between embedding models and reasoning models, and LLM-based methods incur high computational costs. Therefore, in this paper, we propose a novel consistency metric based on $\mathcal{V}$-entropy (Ethayarajh et al., 2022) for evaluating the consistency between the synthesized demonstrations and the task definition with low computational overhead and high performance.

3 Methodology

In this section, we discuss how to synthesize demonstrations with high consistent and diversity from scratch. We first introduce the consistency metric $\mathcal{V}$-SCORE based on the $\mathcal{V}$-entropy. Based on $\mathcal{V}$-SCORE, we propose $\mathcal{V}$-SYNTHESIS, which synthesizes demonstrations with high diversity and consistency through multiple iterations.

3.1 Consistency Metric with $\mathcal{V}$-Entropy

As the discussion in §2, current consistency metrics are limited by the problems of the embedding space gap and the low efficiency. To solve this problem for the demonstration synthesis, we propose to measure consistency based on $\mathcal{V}$-entropy ($H_{\mathcal{V}}$) (Ethayarajh et al., 2022). Specifically, let X, Y denote random variables with sample spaces $\mathcal{X}, \mathcal{Y}$ respectively. Let $\emptyset$ denote a null input without information about Y. Given predictive family $\mathcal{V} \subseteq \Omega = \{f : \mathcal{X} \cup \emptyset \to P(\mathcal{Y})\}$, the definition of the $\mathcal{V}$ entropy is:

$$H_{\mathcal{V}}(Y|X) = \inf_{f \in \mathcal{V}} \mathbb{E}[-\log f[X](Y)] \qquad (1)$$

Intuitively, Equation 1 represents how much information Y can obtain from X when using the optimal predictor f. Let T denote the random variable on the sample space $\mathcal{T}$, representing the target task. We use the measure $I_{\mathcal{V}}(T \to (X, Y))$ to indicate the consistency between the given demonstration and the target task:

$$\begin{aligned} I_{\mathcal{V}}(T \to (X, Y)) = H_{\mathcal{V}}((X, Y)|T) \\ - H_{\mathcal{V}}((X, Y)|\emptyset) \end{aligned} \qquad (2)$$

We call Equation 2 as **$\mathcal{V}$-SCORE**, which measures the information gain learned by the model in comparison to the case where no task definition is provided. In practical computation, since we adapt inference using ICL without fine-tuning, we consider Ω as the same LLM using different demonstrations, thereby calculating Equation 2 by selecting the demonstrations synthesized that are most similar to (X, Y). Intuitively, using the synthesized demonstrations most similar to (X, Y) can be seen to filter demonstrations similar to already synthesized ones, ensuring consistency while maintaining diversity in the synthesized results.

Compared to the consistency metrics discussed in §2, the advantages of $\mathcal{V}$-SCORE are as follows: *(i)* It can directly utilize the model of the synthesis for computation, avoiding the errors caused by the gap of the embedding spaces using additional embedding models. *(ii)* It allows for direct computation using the probability likelihood calculated during the synthesis without additional models, reducing the additional computational overhead.

3.2 $\mathcal{V}$-SYNTHESIS

In this section, we introduce $\mathcal{V}$-SYNTHESIS, which synthesizes demonstrations based on Equation 2, ensuring high consistency and diversity of the synthesized results. The overview of $\mathcal{V}$-SYNTHESIS is shown in Figure 2. $\mathcal{V}$-SYNTHESIS synthesizes demonstrations through multiple iterations. In the first iteration, demonstrations can be synthesized from scratch or labeled by humans. In each subsequent iteration, the synthesis results from the previous iteration are used as input to guide the synthesis as the discussion in §3.1.

Each iteration of $\mathcal{V}$-SYNTHESIS consists of two steps: Synthesize and Sample. The Synthesize step synthesizes demonstrations based on the provided demonstrations and the target task definition. In the Sample step, demonstrations are sampled from the synthesized results using Equation 2 to ensure that the sampled demonstrations have high consistency and diversity. The prompt used in our method is provided in Appendix A.

3.2.1 Synthesize

We use LLMs to synthesize demonstrations for a given target task. The input consists of a task definition and demonstrations labeled or synthesized from the previous iterations, and the output is a set of synthesized demonstrations. To enhance diversity, we sample multiple synthetic demonstrations

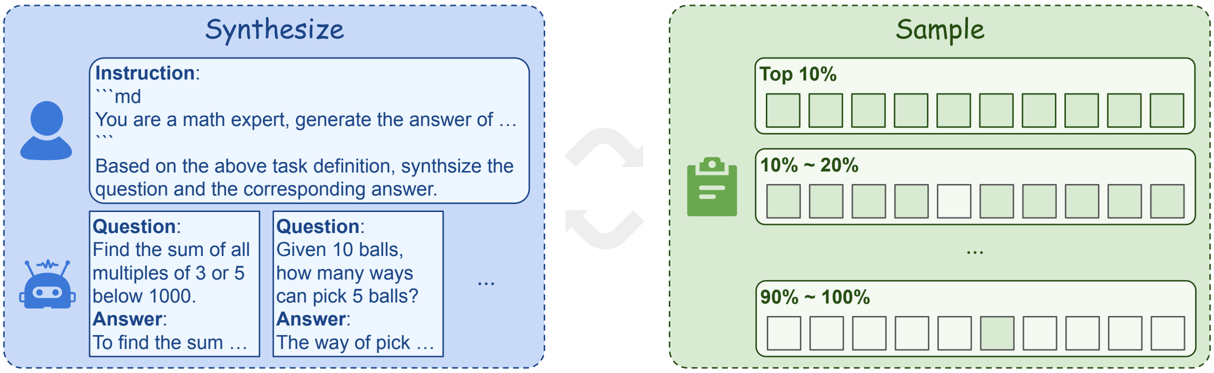

Figure 2: The overview of $\mathcal{V}$-SYNTHESIS, which consists of two steps: *(i)* Synthesize: synthesize the demonstrations with the given task definition and the demonstrations of the previous iterations; *(ii)* Sample: sample the synthesized demonstrations proportionally based on $\mathcal{V}$-SCORE, where the green squares denote the sampled demonstrations.

for the same input. Additionally, following He et al. (2024), if the task specifies different question types, we generate demonstrations separately for each type. For instance, if a mathmetical task includes question types algebra and geometry, we generate algebraic and geometric questions separately.

To ensure the accuracy of the synthesized demonstrations, we ask the model to reason through the questions in the synthesized demonstrations and check whether the reasoned answers match the generated answers. Given that some questions could not be answered correctly on the first attempt, we sample multiple times for each question and consider the synthesized demonstration correct as long as the model answers correctly at least once. Only those synthesized demonstrations whose answers match the questions are retained, ensuring the quality of the synthesized results.

3.2.2 Sample

Following the generation of synthetic results, we perform sampling to ensure consistency with the target task. We first compute the consistency score for each generated demonstration using Equation 2. The likelihood $p(X|Y)$ obtained from the model serves as $f[X](Y)$ within $I_{\mathcal{V}}$. Consistent with the discussion in Section §3.1, for each demonstration synthesized in the current iteration, we select the most similar existing demonstrations for inference. The LLM with selected demonstrations acts as f that makes Equation 2 reach its infimum.

Upon obtaining the consistency score for each demonstration, we sample demonstrations proportionally to their $\mathcal{V}$-SCORE to ensure diversity. Specifically, we first rank the demonstrations based on their $\mathcal{V}$-SCORE and then divide them into deciles (10% intervals, which we call the sample ratio). For the top 10%, we sample 100% of them; for the next 10%, we randomly sample 90%; this pattern continues, with the last 10% being randomly sampled at a rate of only 10%. We do not directly sample the highest-scoring results because demonstrations with high scores tend to exhibit similarity, leading to reduced diversity, which is further discussed in §4.4. By employing proportional sampling, we aim to ensure consistency while simultaneously enhancing diversity, thereby improving the performance of ICL. It can be considered that as the sample ratio increases, the diversity of the sampled demonstrations gradually increases, while the consistency gradually decreases. We discuss in detail the impact of different sample ratios on performance in §4.4 and Appendix C, while also elaborate on why the high consistency of synthetic data harms the diversity in Appendix D.

3.2.3 Efficiency of $\mathcal{V}$-SYNTHESIS

Although $\mathcal{V}$-SYNTHESIS synthesizes demonstrations relying on multi-iteration, the synthesis process is performed ***offline***. During inference, users can directly utilize the synthesized results without additional computation, ensuring efficiency in practical applications. Besides, even with limited computational resources during synthesis, as demonstrated in §4.5, $\mathcal{V}$-SYNTHESIS yields significant performance improvements with a small amount of synthesized demonstrations, proving its effectiveness in low-resource scenarios.

4 Experiment

4.1 Experiment Setting

Dataset We adapt experiments on four mainstream datasets: MATH (Hendrycks et al., 2021), MetaTool (Huang et al., 2024), FinQA (Chen et al., 2021b), and MedQA (Jin et al., 2021), covering

Model	Method	MATH	MetaTool	FinQA	MedQA
Llama3.1-8b	w/o. Human	46.8	51.8	48.1	56.9
	+ Self-ICL	48.3	51.9	49.3	62.1
	+ $\mathcal{V}$-SYNTHESIS	**50.8**	**60.3**	**53.9**	**65.8**
	w. Human	49.0	57.2	50.7	64.3
	+ Self-ICL	48.2	58.1	51.3	64.7
	+ $\mathcal{V}$-SYNTHESIS	**51.2**	**60.7**	**54.6**	**67.2**
Llama3.1-70b	w/o. Human	63.6	59.1	58.3	77.5
	+ Self-ICL	64.0	59.3	58.6	79.2
	+ $\mathcal{V}$-SYNTHESIS	**66.0**	**62.2**	**59.0**	**81.0**
	w. Human	62.8	58.3	63.6	82.5
	+ Self-ICL	63.0	60.3	64.3	84.8
	+ $\mathcal{V}$-SYNTHESIS	**63.2**	**61.1**	**65.2**	**85.2**

Table 1: The performance of $\mathcal{V}$-SYNTHESIS compared with Self-ICL (Chen et al., 2023a). w/o. Human denotes synthesis from scratch and w. Human denotes synthesis based on the training set of each dataset. The best performance under each setting is marked in **bold**.

diverse tasks and domains. Detailed descriptions of these four datasets are provided in Appendix B. Across all datasets, we employ Exact Match (EM) for evaluation and adapt the experiments on the test sets. The results on MATH and MetaTool allow us to observe the performance of $\mathcal{V}$-SYNTHESIS on different tasks. The experiment results on FinQA and MedQA show the performance of $\mathcal{V}$-SYNTHESIS in different domains.

Model We conduct experiments on Llama3.1-8b and Llama3.1-70b (Grattafiori et al., 2024). Llama3.1 is one of the leading open-source models currently, demonstrating excellent performance across multiple mainstream tasks. By comparing performance across different scales, we can evaluate the effectiveness of $\mathcal{V}$-SYNTHESIS on models with varying capabilities.

Implementation Detail For the demonstration synthesis, following prior work (Wang et al., 2024), we employ a 2-shot setting and utilize BM25 to select similar demonstrations. We set the sampling number to 8, the temperature to 0.9, and top_p to 0.9. The synthesis scale under each setting is present in Appendix B. We present the task definition we used of each dataset in Appendix B. During inference evaluation, we adopt a 3-shot setting and use BM25 to select demonstrations similar to the user question following Wang et al. (2024). The inference prompt we use is identical to that in Chen et al. (2023b); Grattafiori et al. (2024).

4.2 Main Experiment

The main experimental results are shown in Table 1. It can be observed that $\mathcal{V}$-SYNTHESIS achieves 2.0% performance improvment on average compared with other baselines under different settings, demonstrating its effectiveness. To further verify the effectiveness of $\mathcal{V}$-SYNTHESIS, we experiment with synthetic data as training data in Appendix E. Besides, from Table 1 we can also find that:

Scale Our method consistently yields performance improvements across models of varying scales, demonstrating its effectiveness regardless of model capacity. Notably, the performance gains achieved by $\mathcal{V}$-SYNTHESIS are more pronounced on smaller-scale models compared to their larger counterparts. This discrepancy arises because smaller models possess a limited ability to tackle complex tasks and thus rely more heavily on the guidance provided by demonstrations during the inference process. Conversely, larger-scale models already exhibit strong inherent reasoning capabilities, diminishing their dependence on explicit demonstration guidance.

Label $\mathcal{V}$-SYNTHESIS consistently delivers performance gains in both synthesis from scratch and synthesis with labeling settings, demonstrating its effectiveness. Furthermore, it can be observed that the performance improvement achieved through $\mathcal{V}$-SYNTHESIS is less substantial when starting with labeled demonstrations compared to synthesis from scratch. This is because manually labeled demonstrations are inherently of higher quality and can already effectively guide ICL, thus rendering the impact of synthesized demonstrations relatively less significant. Conversely, for synthesis from scratch, the initial absence of demonstrations guidance for ICL leads to a more pronounced performance enhancement through our method.

Dataset Our method brings performance gains across datasets spanning diverse tasks and domains, demonstrating its generalizability. Furthermore, the performance improvement achieved by $\mathcal{V}$-SYNTHESIS is more pronounced on the tool-use task (MetaTool) compared to mathematical reasoning tasks (MATH, FinQA). This is because tool use is less frequent and relies more heavily on demonstration guidance than the math task. Additionally, $\mathcal{V}$-SYNTHESIS yields significant improvements on domain-specific datasets (FinQA, MedQA), suggesting that the synthesized demonstrations also encapsulate domain knowledge, effectively guiding domain-related reasoning.

Method	MATH	MetaTool	FinQA	MedQA
$\mathcal{V}$-SYNTHESIS	50.8	60.3	53.9	65.8
- Iteration	$47.9_{(-2.9)}$	$55.5_{(-4.8)}$	$51.4_{(-2.5)}$	$63.5_{(-2.3)}$
- Sampling	$49.8_{(-1.0)}$	$59.0_{(-1.3)}$	$50.5_{(-3.4)}$	$64.2_{(-1.6)}$
- Diversity	$49.8_{(-1.0)}$	$58.7_{(-1.6)}$	$49.4_{(-4.5)}$	$64.6_{(-1.2)}$

Table 2: The ablation study of $\mathcal{V}$-SYNTHESIS on Llama3.1-8b with synthesis from scratch. *(i)* Iteration: Using only the synthesis results from the first iteration. *(ii)* Sampling: Utilizing the complete set of synthesis results without sampling. *(iii)* Diversity: Sampling the top 50% of results based on $\mathcal{V}$-SCORE directly.

Metric	EM	Time Complexity
NGram	47.2	$O(NL)$
Embedding	47.4	$O(N\mathcal{M}_e(L))$
LLM-as-Judge	47.1	$O(N\mathcal{M}_l(L))$
$\mathcal{V}$-SCORE	**50.8**	$O(N\mathcal{M}_l(L))$

Table 3: The performance and time complexity of $\mathcal{V}$-SYNTHESIS with different consistency metrics on MATH using Llama3.1-8b. N represents the data scale, L denotes the average output length, and $\mathcal{M}_e$ and $\mathcal{M}_l$ represent the time required for encoding a string of length L using the embedding model and LLM, respectively. The best performance is marked in **bold**.

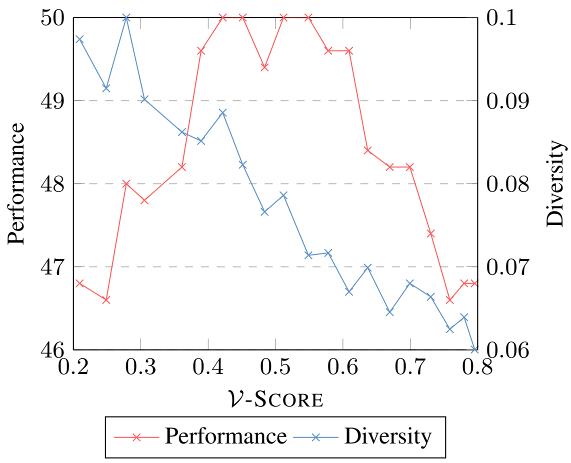

Figure 3: The performance and diversity under different $\mathcal{V}$-SYNTHESIS on MATH using Llama3.1-8b without human labeling. We randomly sample 20 groups from MATH demonstrations, with each group containing 100 demonstrations. X-axis denotes the average $\mathcal{V}$-SCORE on the test data. We employ the metric DM (Wang et al., 2024) to measure the demonstration diversity.

4.3 Ablation Study

To validate the impact of different components in $\mathcal{V}$-SYNTHESIS on effectiveness, we conduct ablation experiments. The results are shown in Table 2, from which it can be observed that ablating each component leads to a performance decrease, demonstrating the effectiveness of each part of our method. Furthermore, from the table, we can also observe that: *(i)* The ablation of the iteration (-Iteration) has the most significant impact on performance since a smaller number of iterations results in a higher proportion of task-inconsistent demonstrations and poorer diversity in the synthesis, thus failing to effectively guide ICL. *(ii)* Compared to not performing sampling (- Sampling), directly using the top 50% of data based on $\mathcal{V}$-SCORE (- Diversity) result in a more severe performance degradation in most settings, which indicates that for synthesized demonstrations, data with high consistency scores tends to have higher similarity, leading to a weaker effect on ICL.

4.4 Effect of Consistency and Diveristy

As discussed in §3.2.2, when synthesizing demonstrations, excessive consistency leads to poor diversity, while excessive diversity also results in poor consistency. Therefore, in this section, we discuss the impact of consistency and diversity on the quality of synthesized demonstrations, as well as their corresponding effects on the performance of ICL.

Does $\mathcal{V}$-SCORE Outperform Other Consistency Metrics? To validate the effectiveness of $\mathcal{V}$-SCORE compared to other consistency metrics, we compare the performance of samlping using different metrics and provide the computational complexity of the additional computational resources required to calculate each metric. The experimental results are shown in Table 3, from which we can observe that: *(i)* Compared to other metrics, $\mathcal{V}$-SCORE achieves better performance, demonstrating that our metric can better reflect the consistency between the demonstrations and the task, thereby ensuring that the sampling results can better guide the solution of the task. *(ii)* While the computational complexity of $\mathcal{V}$-SCORE is higher than that of methods like NGram and Embedding, considering that the demonstration synthesis is offline, the inference-time overhead of different metric remains the same, demonstrating the effectiveness of our method.

Can $\mathcal{V}$-SYNTHESIS Reflect the Demonstration Consistency to the Task? To validate the effectiveness of $\mathcal{V}$-SCORE in reflecting the demonstration consistency, and to demonstrate the high similarity among model-synthesized high-consistency data discussed in §3.2.2, we conduct statistical experiments. We randomly sample 20 groups from MATH demonstrations, with each group containing

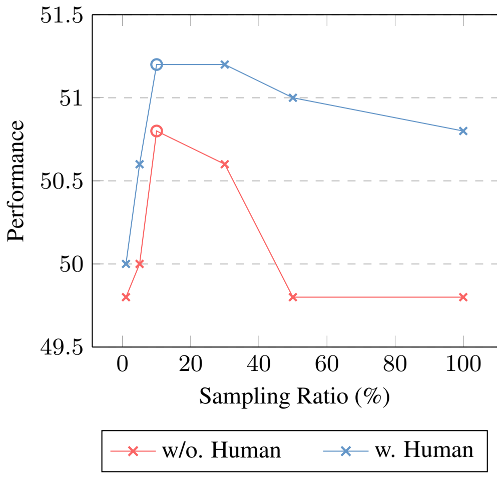

Figure 4: The performance of $\mathcal{V}$-SYNTHESIS on MATH with Llama3.1-8b under different sample ratios. For example, if the sample ratio is 5, we cut and sample the data at the rate of 5% during the proportional sample, as the discussion in §3.2.2. ∘ denotes the performance of our main experiments.

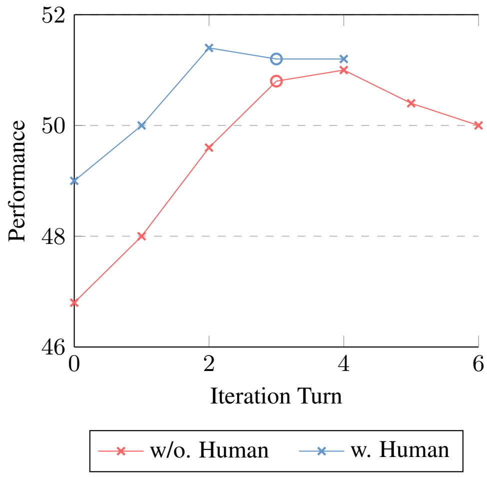

Figure 5: The performance of MATH using Llama3.1-8b with different iteration numbers with labeling and from scratch. 0 on the X-axis represents the zero-shot result. ∘ denotes the iteration of the main experiment.

100 demonstrations. The experimental results are shown in the figure, from which we observe: *(i)* the performance improvement brought by $\mathcal{V}$-SCORE exhibits an inverted U-shaped trend since when consistency is low, the demonstrations struggle to effectively guide ICL due to the low relevance to the target task. Conversely, when consistency is high, the high similarity among synthesized data leads to poor diversity. *(ii)* The diversity results support the above observation, showing a gradual decrease in diversity as the consistency increases, indicating high similarity among the demonstrations synthesized with high consistency.

How to Balance the Diversity and the Consistency? To further observe the impact of demonstration consistency and diversity on ICL performance, we adjust the sampling ratio in §3.2.2 to evaluate their effects. Specifically, when the sample ratio is 1%, demonstrations are filtered directly based on the consistency score, reflecting the highest consistency. Conversely, when the sample ratio is 100%, all synthesized demonstrations used for inference, reflecting the highest diversity. Therefore, it can be considered that as the sample ratio increases, consistency gradually decreases while diversity gradually increases, which is further discussed in Appendix C.

The experimental results are shown in the figure, from which we can observe: *(i)* With 10% as a dividing point, the model performance shows a trend of increasing first and then decreasing as the sample ratio increases, indicating that both low consistency and low diversity lead to poor ICL performance. *(ii)* Compared to reducing consistency (increasing sample ratio), reducing diversity has a greater impact on performance (decreasing sample ratio), with a more significant downward trend in performance, suggesting that diversity affects ICL performance more significantly than consistency.

4.5 Effect of Different Factors

How Does the Iteration Number Affect the Performance? To evaluate the effectiveness of $\mathcal{V}$-SYNTHESIS under varying computational resource constraints, we assess the impact of different numbers of synthesis iterations on performance. The experimental results are illustrated in Figure 5, from which we can observe the following: *(i)* When the number of iterations is relatively small (< 4), the performance of our method consistently increases with more iterations, which is attributed to the model synthesizing a more diverse set of task-relevant demonstrations. *(ii)* However, once the number of iterations reaches a certain threshold (≥ 4), the model performance begins to fluctuate, suggesting that continuously increasing the number of iterations does not guarantee sustained performance enhancement since the number of task-consistent demonstrations the model can synthesize is finite. *(iii)* Notably, even with a minimal number of iterations ($= 1$), the performance of $\mathcal{V}$-SYNTHESIS surpasses that of the baseline without any synthesized demonstrations, demonstrating the effectiveness under low computational resource.

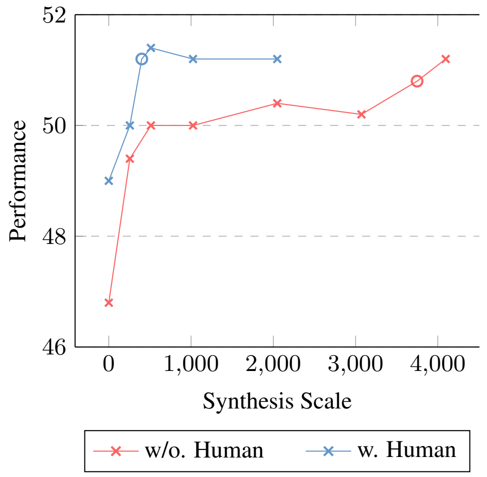

Figure 6: The performance of MATH using Llama3.1-8b with different synthesis scales with labeling and from scratch. 0 on the X-axis represents the zero-shot result. ∘ denotes the synthesis scale of the main experiment.

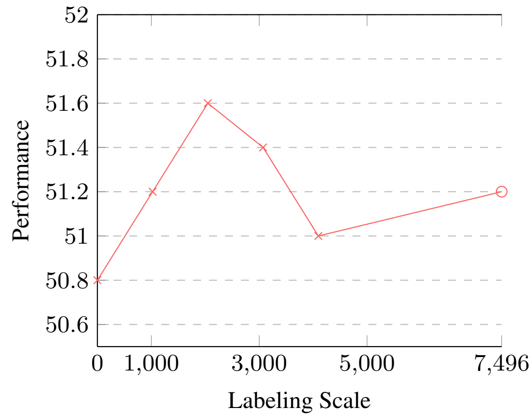

Figure 7: The performance of MATH using Llama3.1-8b with different initial labeling scales. 0 on the X-axis represents the result of zero-shot. ∘ denotes the synthesis scale of the main experiment.

How Does the Synthesis Scale Affect the Performance? To evaluate the effectiveness of $\mathcal{V}$-SYNTHESIS under varying computational resources, we analyze its performance across different synthesis scales. The experimental results are illustrated in Figure 6. From the figure, we can observe the following: *(i)* When the synthesis scale is relatively small, the performance significantly improves as the synthesis scale increases, demonstrating the effectiveness of our method in synthesizing demonstrations. Particularly, even with a limited synthesis scale, $\mathcal{V}$-SYNTHESIS yields substantial performance gains, proving its efficacy under low computational resource constraints. *(ii)* As the synthesis scale continues to expand, the performance improvement brought by $\mathcal{V}$-SYNTHESIS gradually plateaus. This suggests that continually increasing the synthesis scale does not lead to sustained performance enhancement, indicating that the diversity of demonstrations relevant to the target task that the model can synthesize is finite.

How Does the Initial Labeling Scale Affect the Performance? To evaluate the effectiveness of $\mathcal{V}$-SYNTHESIS under varying labeling resources, we experiment with different scales of labeled data, which is randomly sampled. The results, as shown in Figure 7, indicate that: *(i)* With smaller labeling scales, performance gradually improves as the scale of labeled data increases, demonstrating the complementary information between human-labeled and synthetic data. *(ii)* As the labeling scale grows larger, performance starts to fluctuate, suggesting that the diversity of human-labeled data is also limited, and continuously increasing the labeling scale cannot consistently enhance model performance. *(iii)* Even with a small labeling scale, $\mathcal{V}$-SYNTHESIS still brings significant performance improvement, proving the effectiveness of our method in the labeling-insufficient scenarios.

5 Conclusion

The existing demonstration synthesis works mainly focus on specific tasks, or is synthesized based on existing demonstrations. Therefore, in this paper, we discuss how to synthesize highly consistent and diverse demonstrations from scratch for arbitrary task. We first propose $\mathcal{V}$-SCORE, a metric for measuring the consistency between demonstrations and the target task, which shows better performance and lower computational cost compared to previous metrics based on grams or embedding vectors. Based on $\mathcal{V}$-SCORE, we propose $\mathcal{V}$-SYNTHESIS, which samples synthesized results proportionally according to their consistency scores to ensure both high diversity and high consistency of the synthesized demonstrations. We experiment with $\mathcal{V}$-SYNTHESIS on four mainstream datasets, where $\mathcal{V}$-SYNTHESIS achieves a 2.0% performance improvement compared to previous demonstration synthesis methods and an average of 3.4% performance improvement compared to other consistency metrics, demonstrating its effectiveness. Furthermore, additional analysis experiments show that $\mathcal{V}$-SYNTHESIS effectively balances the consistency and diversity of synthesized demonstrations, thus effectively guiding ICL performance.

Limitations

(i) The number of experimental datasets and models is limited, where future work will include more datasets and models to further validate the effectiveness of $\mathcal{V}$-SYNTHESIS. *(ii)* There is a lack of mechanistic analysis regarding the balance between the consistency and diversity of the synthesized demonstrations, where future work will involve further analyzing and explaining why the consistency and diversity of synthesized demonstrations cannot be simultaneously improved.

Ethics Statement

All datasets and models used in this paper are publicly available, and our usage follows their licenses and terms.

References

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 596–609, Dublin, Ireland.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online. Association for Computational Linguistics.

Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. 1997. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8):1157–1166. Papers from the Sixth International World Wide Web Conference.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Laurie Burchell, Alexandra Birch, and Kenneth Heafield. 2022. Exploring diversity in back translation for low-resource machine translation. In *Proceedings of DeepLo*, Seattle, USA.

Zak Buzzard. 2025. Understanding the quality-diversity trade-off in diffusion language models. *Preprint*, arXiv:2503.10683.

Shuaichen Chang and Eric Fosler-Lussier. 2023. Selective demonstrations for cross-domain text-to-SQL. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14174–14189, Singapore. Association for Computational Linguistics.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. 2024. On the diversity of synthetic data and its impact on training large language models. *Preprint*, arXiv:2410.15226.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021a. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 9:1–50.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023a. Self-ICL: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662, Singapore. Association for Computational Linguistics.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023b. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. Finqa: A dataset of numerical reasoning over financial data. *Proceedings of EMNLP 2021*.

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

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur

Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha

White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Alex Havrilla, Andrew Dai, Laura O'Mahony, et al. 2024. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *Preprint*, arXiv:2412.02980.

Wei He, Shichun Liu, Jun Zhao, Yiwen Ding, Yi Lu, Zhiheng Xi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Self-demos: Eliciting out-of-demonstration generalizability in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3829–3845, Mexico City, Mexico. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2024. Metatool benchmark for large language models: Deciding whether to use tools and which to use. In *The Twelfth International Conference on Learning Representations*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, et al. 2024. Understanding the effects of rlhf on llm generalisation and diversity. *Preprint*, arXiv:2310.06452.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Haotian Luo, Yixin Liu, Peidong Liu, and Xianggen Liu. 2024. Vector-quantized prompt learning for paraphrase generation. In *Findings of EMNLP*, pages 13389–13398, Singapore.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *Preprint*, arXiv:2305.14128.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Vishnu Sarukkai, Zhiqiang Xie, and Kayvon Fatahalian. 2025. Self-generated in-context examples improve llm agents for sequential decision-making tasks. *Preprint*, arXiv:2505.00234.

Anshumali Shrivastava and Ping Li. 2014. In defense of minhash over simhash. *Preprint*, arXiv:1407.4416.

Amit Singhal and I. Google. 2001. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24.

Yi Su, Yunpeng Tai, Yixin Ji, Juntao Li, Yan Bowen, and Min Zhang. 2024. Demonstration augmentation for zero-shot in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14232–14244, Bangkok, Thailand. Association for Computational Linguistics.

Xingchen Wan, Han Zhou, Ruoxi Sun, and Sercan O Arik. 2025. From few to many: Self-improving many-shot reasoners through iterative optimization and generation. In *The Thirteenth International Conference on Learning Representations*.

Dingzirui Wang, Longxu Dou, Xuanliang Zhang, Qingfu Zhu, and Wanxiang Che. 2024. Improving demonstration diversity by human-free fusing for text-to-SQL. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1193–1207, Miami, Florida, USA. Association for Computational Linguistics.

Dingzirui Wang, Xuanliang Zhang, Qiguang Chen, Longxu Dou, Xiao Xu, Rongyu Cao, YINGWEI MA, Qingfu Zhu, Wanxiang Che, Binhua Li, Fei Huang, and Yongbin Li. 2025a. In-context transfer learning: Demonstration synthesis by transferring similar tasks.

Zaitian Wang, Jinghan Zhang, Xinhao Zhang, et al. 2025b. Diversity-oriented data augmentation with large language models. *Preprint*, arXiv:2502.11671.

Cedric Deslandes Whitney and Justin Norman. 2024. Real risks of fake data: Synthetic data, diversity-washing and consent circumvention. *Preprint*, arXiv:2405.01820.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456, Singapore. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey. *Preprint*, arXiv:2308.07107.

Prompt of Synthesis

````
```md
{task_definition}
```
Given Question: {question}

Based on the above task definition and the given question, synthesize a question and the corresponding answer that is similar to the given question of the task.
````

Table 4: The prompt of the demonstration synthesis.

A Prompt of $\mathcal{V}$-SYNTHESIS

In this section, we present the prompt of $\mathcal{V}$-SYNTHESIS, as shown in Table 4. The inference prompts we used are some with the previous works (Grattafiori et al., 2024; Chen et al., 2023b; Jin et al., 2021).

B Datasets

In this section, we introduce the experimental datasets we used in detail. The task definitions used by $\mathcal{V}$-SYNTHESIS are shown in Table 5. The synthesis scale of each dataset is shown in Table 6.

MATH MATH is a dataset for evaluating mathematical problem-solving capabilities of machine learning models. It comprises challenging mathematics problems sourced from high school competitions, covering topics such as algebra, geometry, number theory, calculus, and probability. A key feature of the MATH dataset is the inclusion of detailed, step-by-step solutions for each problem, facilitating the training of models to generate derivations and explanations, not just final answers. This dataset serves as a rigorous testbed for assessing and advancing the reasoning abilities of advanced AI systems in the domain of mathematics.

MetaTool MetaTool is designed to evaluate the capabilities of Large Language Models (LLMs) in understanding and selecting appropriate tools. It comprises diverse user queries that trigger LLMs to use tools in both single-tool and multi-tool scenarios. These queries are generated through various methods to ensure diversity, aimming to assess tool usage awareness and the proficiency in tool selection across different contexts, including scenarios with similar tool choices, specific situational needs, potential tool reliability issues, and the necessity for multiple tools. This dataset facilitates a systematic evaluation of LLMs as intelligent agents.

FinQA FinQA is a large-scale question answering dataset designed for numerical reasoning over financial reports. It comprises approximately 8,000 expert-annotated question-answer pairs grounded in 2,800 financial documents, which include both textual and tabular data. A key feature is the provision of gold reasoning programs, offering step-by-step operations required to derive the answers. This dataset specifically targets the challenges of complex numerical understanding and multi-step reasoning inherent in the financial domain, aiming to drive research beyond general-purpose QA systems towards more domain-specific analytical capabilities.

MedQA The MedQA dataset is a prominent benchmark in the field of medical question answering, designed to evaluate the ability of models to comprehend and answer medical questions. It is constructed from professional medical licensing examinations in the United States, mainland China, and Taiwan, providing a diverse and challenging set of multiple-choice questions. The dataset encompasses English, Simplified Chinese, and Traditional Chinese, with a total of over 60,000 questions across these languages. MedQA is widely used for training and evaluating natural language processing models, particularly large language models, on their medical knowledge and reasoning capabilities.

C The Effect of Sample Ratio

In this section, we analyze the effect of the sample ratio on diversity and consistency. Assume the scale of the synthetic example set to be sampled is M, and the sample ratio is $r \in (0, 1)$. The scale of the sampled example set is then:

$$\begin{aligned} &\sum_{i=1}^{1/r} M \times (1 - (i-1) \times r) \\ &= M \times (1/r - r \times \sum_{i=1}^{1/r} (i-1)) \\ &= M \times (1/r - (\frac{1}{2} + \frac{1}{2r})) \\ &= M \times (\frac{1}{2r} - \frac{1}{2}) \end{aligned} \tag{3}$$

It can be observed that increasing the sample ratio samples more demonstrations, thereby increasing demonstration diversity. Conversely, as the sample ratio decreases, sampling tends to retain demon-

Dataset	Task Definition
MATH	Suppose you are a **mathematical** expert, based on the above task definition, generate mathematical problems and the corresponding solution.
MetaTool	Suppose you are a **tool using** data annotator, based on the above task definition, generate the user question and the corresponding tool to be used. Select a tool to be used from the following list: *TOOL_LIST*
FinQA	Suppose you are a **financial** data annotator, based on the above task definition and the given table and paragraphs, you should generate **calculation** question about the table and paragraphs, and then generate the solution of the question.
MedQA	Suppose you are a **medical** expert, based on the above task definition, you should first generate a medical question under the patient's medical scenario and five options marked from A to E, and then generate the solution of the question.

Table 5: The hand-written task definition of the experimental datasets we used. *TOOL_LIST* of MetaTool is the list of all possible tools can be used, which is discussed in Huang et al. (2024).

Dataset	w/o. Human 8b	w/o. Human 70b	w. Human 8b	w. Human 70b
MATH	2300	1899	2021	1736
MetaTool	1789	2675	1850	4443
FinQA	1629	4806	1558	3233
MedQA	200	242	432	504

Table 6: The synthesis scale on each dataset and model with and without labeling.

strations with higher consistency scores, leading to increased demonstration consistency.

D Relationship between Consistency and Diversity of Synthesized Data

In the section, we discuss the relationship of the consistency and the diversity of the synthesized data. **Consistency** (how faithfully a new demonstration preserves the meaning or class of the source) keeps the model on-track, while **diversity** (how different the demonstration is in words, grammar, or topic) helps the model generalise. Large empirical surveys show that whenever researchers tighten the similarity threshold used to filter synthetic demonstrations, vocabulary richness and structural variety drop in lock-step, a symptom sometimes called *mode collapse* (Havrilla et al., 2024; Chen et al., 2021a).

Concrete experiments make the trade-off visible. In **Quality-Controlled Paraphrase Generation**, forcing high semantic overlap lowers Distinct-n diversity scores (Bandel et al., 2022). **Vector-Quantised Prompt Learning** repeats the pattern with a small code-book of rewrite styles (Luo et al., 2024). At a larger scale, Chen et al. compute a cluster-based metric across millions of model-written instructions and find that aggressive label-correctness filtering removes rare topical clusters (Chen et al., 2024). Raising the *guidance scale* (a knob that enforces quality) in a diffusion language model similarly cuts diversity (Buzzard, 2025), and reinforcement learning from human feedback improves average preference scores but compresses syntactic variety (Kirk et al., 2024). Even classic **back-translation** for low-resource machine translation shows richer phrasing only when round-trip similarity checks are relaxed (Burchell et al., 2022).

Because pushing consistency too far can hurt robustness, recent work searches for *Pareto* compromises. One practical recipe is to sample with a higher **temperature** (which flattens the probability distribution) or a larger **top-p** cutoff (keeping the most probable tokens whose cumulative probability is p), then discard only the worst semantic outliers. **DoAug** follows this two-stage idea and reports double-digit accuracy gains while keeping labels intact (Wang et al., 2025b). At the same time, ethicists warn that headline diversity numbers can mask repeated cultural biases, a problem dubbed "diversity-washing" (Whitney and Norman, 2024). In practice, mixing real and synthetic samples, monitoring simple statistics such as type–token ratio, and occasionally inspecting the data by hand remain the safest way to balance the two goals.

E Performance of $\mathcal{V}$-SYNTHESIS using SFT

To further assess the efficacy of the demonstrations synthesized by $\mathcal{V}$-SYNTHESIS, we perform Supervised Fine-Tuning (SFT) (Zhang et al., 2024) utilizing the synthesized demonstrations as training data. Table 7 presents the experimental results, which show that the data synthesized by $\mathcal{V}$-SYNTHESIS

Method	Data	MATH	MetaTool	FinQA	MedQA
ICL	w/o. Human	46.8	51.8	48.1	56.9
	+ $\mathcal{V}$-SYNTHESIS	**50.8**	**60.3**	**53.9**	**65.8**
SFT	w/o. Human	46.8	51.8	48.1	56.9
	+ $\mathcal{V}$-SYNTHESIS	**51.6**	**62.1**	**55.0**	**69.2**
ICL	w. Human	49.0	57.2	50.7	64.3
	+ $\mathcal{V}$-SYNTHESIS	**51.2**	**60.7**	**54.6**	**67.2**
SFT	w. Human	49.0	57.2	50.7	64.3
	+ $\mathcal{V}$-SYNTHESIS	**53.0**	**63.3**	**56.7**	**68.0**

Table 7: The performance using ICL and SFT with the data synthesized by $\mathcal{V}$-SYNTHESIS on Llama3.1-8b. We adapt SFT using LLaMA-Factory (Zheng et al., 2024).

yields performance improvement across all experimental settings. This confirms the effectiveness of our approach even with SFT, thus validating the quality of our synthesized data.