Generalizing Numerical Reasoning in table data through utilization of Primitive Descriptions and self-supervised learning

Hanjun Cho[1], Hanseong Kim[2] and Jay-Yoon Lee[1]

[1] Graduate School of Datascience, Seoul National University, Seoul, Korea, {gkswns0531, lee.jayyoon}@snu.ac.kr
[2] Industrial and Information System Engineering, Soongsil University, Seoul, Korea, gkstjd1201@soongsil.ac.kr

Abstract

As NLP technologies advance, the demand for automatic processing of expert domain documents has surged, driven by the need to manage such documents' increasing volume and complexity. This paper addresses the challenge of performing effective question answering (QA) on numerical tabluar data on expert domain documents. Our study shows that fine-tuning models on specific domains often impede general reasoning capability showing dependencies on specific table header terms or jargon in question. To address this problem we propose **N**umerical **R**easoning utilizing **P**rimitive **D**escription (NRPD) framework that enhances the generalization capability of models to improve numerical reasoning performance. Additionally, our approach incorporates high-level descriptive information to create robust models capable of handling novel tables and operations. Our contributions include identifying and mitigating the issue of header dependency through anonymization, leveraging self-supervised learning to automatically generate numerical reasoning data, and proposing a new system that can leverage high-level descriptions to disentangle domain-specific terminology. This results in a model that is better equipped to handle real-world numerical reasoning scenarios, addressing overlooked aspects in existing research.

Keywords— *NLP, QA, Numerical Reasoning, Self-supervised learning*

I. INTRODUCTION

Recent advancements in Natural Language Processing (NLP) have led to significant improvements in complex understanding and generation tasks. However, language models still face challenges in numerical reasoning tasks [12]. Numerical reasoning involves finding relevant *documents* based on a given *query* and generating the correct *answer* by creating a *program*. Numerical reasoning is especially challenging in expert domains, where it requires not only generating accurate arithmetic operations but also reflecting specific domain knowledge. Nevertheless, efforts to automate the processing of expert domain documents have been ongoing as processing these documents are expensive due to requiring skilled professionals.

	Test set		
	FinQA	Financial jargon	Anonymized
FinQA	61.54	11.49	41.02

Table 1. Comparison of performance metrics on different test sets. The "Financial jargon" includes terms not present in the training set of FinQA, aiming to analyze Jargon dependency. The "Anonymized" examines header dependency by replacing table headers with anonymized tokens on FinQA (B.)

Particularly in finance, previous studies [5, 6, 15, 16] have attempted to enhance numerical reasoning ability in expert domains by fine-tuning language models toward specific domains. However, we unveil that this approach often deters the generalization capability of a model (Table 2) which makes a numerical reasoner trained on one dataset inapplicable to other table datasets.

We further identify that two key factors contribute to the limited generalization capability of numerical reasoning for Table QA with our analysis in (Table 1). (1) ***Header dependency***: Table headers may differ from those encountered during model training, we observe that the models fine-tuned in certain dataset (e.g.FinQA) struggle to process information from tables with unfamiliar headers. (2) ***Domain-Specific Jargon***: In numerical reasoning tasks, questions typically contain tokens that represent the arguments (e.g. sales, revenue) and operations (e.g. difference, ratio) needed to generate the correct answer. However, domain-specific jargons that encapsulate both arguments and operations within a single term will not perform well during test-time if the model has not been properly exposed to these terms in training.

To address these problems, we present the **N**umerical **R**easoning utilizing **P**rimitive **D**escription (**NRPD**) framework, consisting of the following main components:

First, unlike traditional numerical reasoning approaches

Auxiliary information

Question: What was the percent of the change in the stock price performance for hum from 2010 to 2011?

Description: [hum - hum] / hum

	12/31/2009	12/31/2010	12/31/2011	12/31/2012	12/31/2013	12/31/2014
hum	$ 100	$ 125	$ 201	$ 160	$ 244	$ 342
s&p 500	$ 100	$ 115	$ 117	$ 136	$ 180	$ 205
peer group	$ 100	$ 112	$ 123	$ 144	$ 198	$ 252

program: subtract(201,125),divide(#0,125)

✔ **OURS predicted**: subtract(201,125),divide(#0,125)

✘ **Baseline predicted**: divide(201,125),divide(#0,125)

Domain - specific Jargon

Question: what was the 2006 tax expense?
Description: the provision for income taxes * the effective tax rate

	2006	2005	2004
computed expected tax	$ 987	$ 633	$ 129
state taxes net of federal effect	86	-19	-5
non deductible executive compensation	11	14	12
provision for income taxes	$ 829	$ 480	$ 104
effective tax rate	29%	27%	28%

program: multiply(829, 29)

✔ **OURS predicted**: multiply(829, 29)

✘ **Baseline predicted**: subtract(987, 829)

Fig. 1. The examples illustrate cases where the baseline method fails, but NRPD provides correct results. In the first case, The description is utilized as auxiliary information to generate the correct answer. In the second case, This is an effective way to deal with domain-specific jargon where baseline models fail.

Train	Test set	
	FinQA	NumReason-500
FinQA	61.54	5.62
NumReason-500	1.48	81.45

Table 2. **NumReason-500** is a dataset that we automatically generated based on S&P500 SEC reports, while **FinQA** is a human-annotated dataset. Both datasets originate from the same domain and similar sources. However, when the same model is trained on one dataset and tested on the other, significant performance degradation is observed.

that rely solely on the query, the NRPD framework leverages primitive descriptions to aid in generating answers. Experimental results demonstrate that incorporating descriptions into the query can improve performance, with average performance increasing from 27.10 to 64.59 across various datasets.

Second, we propose anonymizing headers during training to reduce header dependency. This approach prevents the model from memorizing specific headers to solve problems, thereby enhancing its ability to properly learn numerical reasoning skills effectively.

Third, we propose a self-supervised learning (SSL) approach to train the model to that can effectively utilize descriptions and anonymization. This technique generates datasets entirely automatically, eliminating the need for human annotation. During the self-supervised learning process, we applied anonymization to mitigate header dependency and jargon dependency, ensuring the model learns under unbiased conditions in expectation for it to be applicable to a wide variety of table numerical reasoning problems.

We examine the effectiveness of our SSL approach by further fine-tuning the SSL model on small samples from the expert-domain dataset. This approach improves the average performance by 37%-point (more than double) compared to models that did not utilize SSL. This proposed approach exceeds the performance of models trained on the full expert-domain dataset by a large margin even when utilizing a small portion (10%) of the dataset and is more robust to domain shift. Furthermore, with sufficient expert-domain data on the order of thousands of examples, the NRPD framework can increase the document processing accuracy of non-experts to as high as 79.61%, within 8%-point of the experts accuracy, by improving the baseline with the same dataset by 20%-point.

Our contributions are fourfold:

- **Highlight the Lack of Generalization in Traditional Fine-tuning Approach** We identified that the traditional fine-tuning approach frequently results in the model losing its generalization ability due to header dependency and domain-specific jargon.
- **Propose Methodologies to Enhance Generalization Ability**: we introduced Self-Supervised Learning (SSL) and Anonymization. SSL allows the generation of diverse numerical reasoning data without human labor. During SSL, anonymization can be applied to prevent the model from building spurious dependencies to headers and jargon.
- **Enhanced Transferability for low-resource environments** Our methodology demonstrated superior performance compared to traditional models trained on thousands of expert-domain data samples, even when trained on only a few hundred such samples.

- **NRPD Framework for Bridging the Expertise Gap** With the assistance of NRPD, we can significantly enhance the ability of non-experts to process documents in specialized domains.

II. Related Works

Numerical Reasoning has been extensively studied and is recognized as one of the limitations of recent large language models (LLMs) [12]. Datasets like DROP [8] and MathQA [1] are prominent examples that focus on numerical reasoning. These datasets involve locating supporting information from text or tables and extracting answers based on the relevant information. Furthermore, the HybridQA [4], which combines information from both text and tables, is also being explored. Tables contain structured information that requires comprehensive understanding. This has led to the development of encoders such as TaBERT [14] and TAPAS [10], which incorporate embeddings that capture the relationships within tables. Recently, there has been a shift towards leveraging large language models (LLMs) for this task, given their superior capacity for understanding and generating text [2, 11, 3].

Finance Numerical Reasoning Research on numerical reasoning is also actively conducted in the finance domain. FinQA [5] and TATQA [16] are typical numerical reasoning tasks that address both text and table contexts. Additionally, existing studies have critiqued the simplicity of tables, leading to the introduction of MultiHiertt [15], which handles more complex tables. Furthermore, real-world scenarios necessitate remembering previous questions, as required in conversational contexts like ConvFinQA [6].

Self-supervised learning in reasoing task In the realm of reasoning tasks, self-supervised learning is extensively researched to enhance reasoning capabilities by generating or augmenting datasets. [9] aims at mathematical reasoning by separately generating numeric data and textual data, thus promoting improvements in comprehension and numerical reasoning. [13] boosts performance by using external sources to find missing information in questions and tables through SQL queries.

III. Task Definition

In this work, we address the challenge of generating reasoning programs to answer complex numerical questions over financial data. Our approach builds on the methodology introduced in the FinQANet framework[5], specifically focusing on the program generator component. FinQANet is a comprehensive framework consisting of two main components: a retriever and a program generator. For the purpose of our paper, we concentrate exclusively on the program generator.

A. Program Generator

The program generator is tasked with generating executable reasoning programs to answer financial questions. Given a financial report F consisting of textual content E and structured tables T, along with a question Q, the goal is to generate a sequence of operations $G = \{w_0, w_1, \ldots, w_n\}$ that can be executed to produce the correct answer A.

The program G is then executed to obtain the answer A. This can be expressed as:

$$P(A \mid T, E, Q) = \sum P(G_i \mid T, E, Q)$$

where $\{G_i\}$ represents all the correct programs that can yield the answer. The generation of G involves selecting each token w_t in the sequence based on the previous tokens and the given inputs. This process is modeled as:

$$P(G_t \mid T, E, Q) = \prod_{t=0}^{n} P(w_t \mid w_0, \ldots, w_{t-1}, T, E, Q)$$

Each step w_t is chosen to maximize the conditional probability given the context up to step $t-1$.

The programs generated by the FinQANet's program generator utilize a Domain Specific Language (DSL) comprising various mathematical and table operations. In our work, we use the same DSL as defined in [5].

B. Evaluation Metrics

To evaluate the performance of the program generator, we use two primary metrics:

Execution Accuracy: This measures the accuracy of the final results obtained by executing the generated programs.

Program Accuracy: This evaluates the correctness of the generated programs by comparing them to the annotated gold programs. Two programs are considered equivalent if they perform the same operations in a mathematically equivalent manner.

Execution accuracy can sometimes result in correct answers despite incorrect formulas due to coincidental matches. Since our goal is to measure the model's numerical reasoning abilities, we use Program Accuracy as the evaluation metric, which requires the entire correct answer program to be matched for it to be considered correct.

IV. Numerical Reasoning utilizing Primitive Description (NRPD)

We propose the NRPD framework, which aims to enhance the model's reasoning capability by utilizing primitive descriptions and self-supervised learning. The NRPD framework generates numerical reasoning datasets with descriptions without the need for human annotation and employs anonymization techniques during training. This

approach enables the model to achieve more robust numerical reasoning abilities. We will explain each component in detail.

A. Description

Descriptions serve two purposes. First, they define jargon, enabling the model to handle previously unseen terminology encountered during training. For existing jargon, they explain the arguments and operations involved. Second, descriptions provide additional information to help model solve numerical reasoning tasks. When a user queries the model, it is assumed they have a basic understanding of the necessary computations. In this context, a description acts as an instruction containing high-level sketch about the answer, typically combining row headers and operations in a table. The model is trained to generate a comprehensive and sophisticated answer from this simple description. An example of the description is illustrated in Figure 1.

Concatenating Descriptions In traditional numerical reasoning, problems were solved based on C, T given Q. However, we expanded this approach by incorporating descriptions (Des), applying T, C given Q, Des.

$$P(A|T,C,Q,Des) = \sum P(G_i|T,C,Q,Des)$$

B. Anonymization

Anonymization means replacing the tokens in the row and column headers with arbitrary tokens. This technique aims to reduce header dependency and enhance the model's generalization capability. During the learning process, the model tends to focus on memorizing specific table headers, which prevents it from developing the ability to perform actual numerical reasoning. To avoid this, in the training phase, we replaced the both row and column headers in the tables with arbitrary tokens from the BERT vocabulary [7][1], excluding special tokens such as [CLS], [PAD], and [UNK]. Through this approach, the model learns not just to match based on the surface form of the headers, but to understand which column and row to extract the desired data from. We used same tokens to replace headers in the questions, context, and description, ensuring consistency.

C. Automatic Data Generation

Automatic data generation enables us to generate a numerical reasoning dataset from tables without human labor, as illustrated in Figure 2. the typical method for generating such a dataset involves a human looking at a table, generating questions, and then manually annotating them with the appropriate answer program. Our approach eliminates the need for human intervention by automating the data generation process.

In this process, we collect table and text context from source data (SEC reports of S&P 500) and randomly choose a sequence of operators and select corresponding arguments from cell values in a table. Using these, we execute them to derive the answer.

Subsequently, this passes through the Question Module to generate the final data instance. The Question Module consists of two components: NumReason-500 and DescJargon-500.

NumReason-500 aims to generate questions in natural form, to enhance the dataset's overall numerical reasoning capabilities.

DescJargon-500 focuses on generating questions that cannot be answered without a description, thereby preventing answer extraction solely from questions.[2]

D. Self-supervised learning

Following Section C., we created DescJargon-500. Furthermore, to ensure generality, we applied anonymization to DescJargon-500 as described in Section B.. We conducted self-supervised learning using DescJargon-500 following the Section iii..

After that, it proceeds to further training on the target domain dataset.

In the **further training stage**, the model assimilates domain-specific knowledge, including the questioning style and expertise relevant to the domain, to optimize its performance. In real-world scenarios, experts in each domain will need to manually construct datasets to create further training, which is a costly process. Therefore, it is crucial that the target domain training performs effectively with low-resource data.

In summary, SSL enhances the model's numerical reasoning ability through the effective utilization of descriptions, while anonymization reduces header dependency and improves the model's generalization performance.

Finally, further training on the target domain, which requires only a small amount of training data, optimizes the model for that specific domain.

V. Experiment

A. Dataset

Financial Table QA Dataset To benchmark our approach, we also include a subset of the FinQA[5] dataset that requires table-only data for generating answers. From

[1]Tokens are selected from a pre-defined range (e.g., token indices 2000-22000). However, during inference time, we applied tokens ranging from 22,000 to 25,000.

[2]For instance, we crafted questions like *"What is 2012 AAPL's Q1 Formula1?"* such that extracting the answer requires understanding the description of *"Formula1"*, which includes domain-specific jargon such as *"tax expense"*

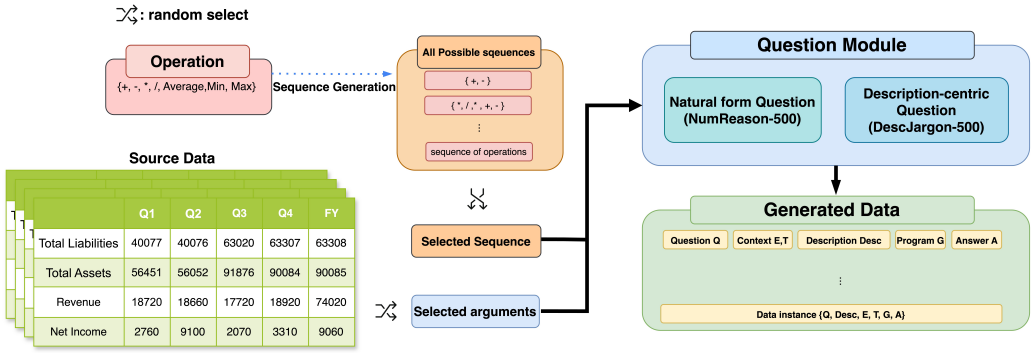

Fig. 2. This is illustration of **automated data generation**. The automated data generation technique, shown in the D., produces results numerical reasoining data. This approach demonstrates effective performance while reducing the need for extensive human involvement.

FinQA, we extracted 4,340 training examples, 615 dev examples, and 780 test examples, focusing on table-only examples where descriptions can be generated automatically.

Domain shift Dataset To evaluate the model's cross-domain performance, we constsructed domain shift datasets. These datasets modify the table headers from the extracted FinQA dataset, replacing them with terms specific to various expert domains. In this process, it is necessary to also replace the table headers that appear in the context or questions. Therefore, domain shift is only possible for data samples where table headers are present in the question or context. Based on this criterion, we constructed two versions of domain shift datasets: one for the **Mechanical** domain and one for the **Biology** domain, each consisting of 3,944 training samples, 550 development samples, and 716 test samples from the FinQA dataset. For consistent comparison, the original FinQA data, derived from the same source as the domain shift datasets, is designated as the **Finance** domain data.

Automatic Generated Datasets This dataset is constructed for self-supervised learning (§ D.). It is based on table information (income statements, balance sheets, cash flows) and stock prices extracted from SEC reports of S&P 500 companies from 1983 to 2023[3]. The **DescJargon-500 dataset** consists of 74,480 training samples, 14,896 development samples, and 7,448 test samples. The **NumReason-500 dataset** comprises 70,000 training samples, 20,000 development samples, and 10,000 test samples.

B. *Experimental Models*

To verify the effectiveness of our NRPD framework, we applied the model architecture of FinQANet [5] directly. All hyperparameters, including learning rate, batch size, and optimizer settings, were kept identical to those used in the original FinQANet experiments to ensure a fair comparison. Additionally, we used the same retriever results and trained only the program generator to compare reasoning abilities.

We utilized the RoBERTa-large[7][4], and conducted experiments in the same environment as the baseline, using a batch size of 16. The model was trained on the automatic generated data for 5 epochs and on the further train data for 100 epochs.

VI. Experiment Result

A. *Domain Shift Performance*

Table 3 illustrates the variations in in-domain and cross-domain performance based on self-supervised learning and description conditions. The SSL was trained on the DescJargon-500 dataset with anonymization applied. The results of experiments where anonymization was not applied during the SSL process are detailed in Appendix C. This experiment includes datasets that each represent a different domain. The anonymized test set removes the influence of specific words, allowing us to evaluate the pure numerical reasoning ability of the model. Additionally, as highlighted in the further training stage, the further train set must perform well with limited data. To achieve this, we sampled 10% of the training set for each domain to construct the further training set. The models were trained on these further train sets, and their performance was evaluated on the corresponding test sets for each domain. The bottom of the table presents a comparison of results obtained by training with 100% of the FinQA further train set. The Baseline refers to the existing FinQANet model trained with the original FinQA dataset.

Baseline Performance When the baseline model is trained using 10% of the FinQA dataset, it achieves an average performance of 27.10. While the in-domain performance on the same FinQA domain is 36.73, the cross-domain performance on other expert domains drops by nearly 10%. Notably, on the anonymized test set, the performance further declines to 19.27, almost half of the in-domain performance. This trend is maintained, albeit to

[3]SEC reports of S&P 500.

[4]we use huggingface transformers library roberta-large

Methods (Sample size)	Train domain (Target domain)	Test domain				
		Avg.	Finance	Mechanical	Biology	Anonymized
Baseline (10%)	Finance	27.10	36.73	27.10	25.28	19.27
	Mechanical	27.90	28.77	31.15	29.19	22.49
	Biology	27.69	29.75	29.61	31.29	20.11
NRPD w/o Des (10%)	Finance	28.35	38.69	27.09	25.70	21.93
	Mechanical	33.24	34.36	36.31	33.24	29.05
	Biology	31.60	30.31	32.54	34.08	29.47
NRPD w/o SSL (10%)	Finance	43.01	50.00	44.27	44.27	33.52
	Mechanical	47.19	48.32	52.01	49.58	38.83
	Biology	40.71	38.82	43.71	44.41	35.89
NRPD (10%)	Finance	64.59	66.20	67.18	63.27	61.73
	Mechanical	64.28	64.94	67.45	62.01	62.71
	Biology	63.79	63.69	64.25	64.52	62.70
Baseline (100%)	Finance	54.61	63.83	51.68	55.73	47.21
NRPD (100%)	Finance	75.98	79.61	77.37	73.32	73.60

Table 3. Prformance comparison across different train domains and training methods. The table includes results for baseline models and NRPD models with and without SSL and descriptions, evaluated on Finance, Mechanical, Biology, and Anonymized test sets. The underline indicates that the train domain and test domain are the same, representing an in-domain.

varying degrees, when the model is trained on other expert domains. These results indicate that the baseline approach lacks sufficient generalization ability.

Baseline with SSL (NRPD w/o Des) After training with SSL and then fine-tuning without description, the model showed in-domain performance gains of 1.96%p in Finance, 5.16%p in Mechanical, and 2.79%p in Biology over the baseline. For the anonymized test set, performance improvements of 2.66%p, 6.56%p, and 9.36%p in each domain over the baseline indicated that the anonymization applied during the SSL process improved generalization. However, the cross-domain performance improvements over the baseline were not significant when tested on other expert domains, suggesting that fine-tuning for specific expert domains may cause the model to lose generalization capabilities faster across different expert domains.

Baseline with Description (NRPD w/o SSL) When trained on the train domain with descriptions but without SSL, there were noticeable performance improvements compared to the baseline. In the Mechanical domain, the average performance improvement was close to 20%p, while in the Biology domain, the improvement was approximately 13%p. This difference is likely due to the complexity of domain header. The Biology domain includes relatively difficult terms like "erythropoiesis" whereas the Mechanical domain includes simpler terms like "bolt". These results indicate that, despite the use of descriptions, there remains a dependency on specific terms.

NRPD Framework Performance Using our NRPD framework, both in-domain and cross-domain performance significantly improved compared to the baseline. In the Finance domain, the in-domain performance increased by 29.47%p, reaching 66.20%, which is higher than the baseline trained on 100% of the data. Additionally, the difference between anonymized test performance and in-domain performance was within 5%p across all domains. This demonstrates that NRPD not only substantially enhances in-domain performance but also improves cross-domain performance, significantly boosting the model's generalization capabilities.

Train	Test	
	Original	Anonymized
Original	98.2	68.2
Anonymized	97.2	97.1

Table 4. Performance of the DescJargon-500 dataset under different anonymization conditions.

NRPD Framework Compared to Baseline on Full Training Dataset Comparing NRPD to the baseline, In-domain performance increases by 15.78%p, reaching 79.61. Given that the performance of experts was 87.49 and non-experts was 48.17 [5], NRPD significantly narrows the performance gap between these groups. Additionally, cross-domain performance sees a substantial improvement, with a 26.39 increase on an anonymized test that measures numerical reasoning skills. This demonstrates that NRPD can effectively enhance the model's numerical reasoning capabilities and improve its generalization ability.

B. Effectivness of Anonymization

Experiments on the DescJargon-500 dataset were conducted under different anonymization conditions

Impact of Header Dependency When the model was trained without any anonymization and tested on the same condition, it achieved a high accuracy of 98.2. However, when anonymization was applied during testing, the performance significantly dropped to 68.2. This indicates a

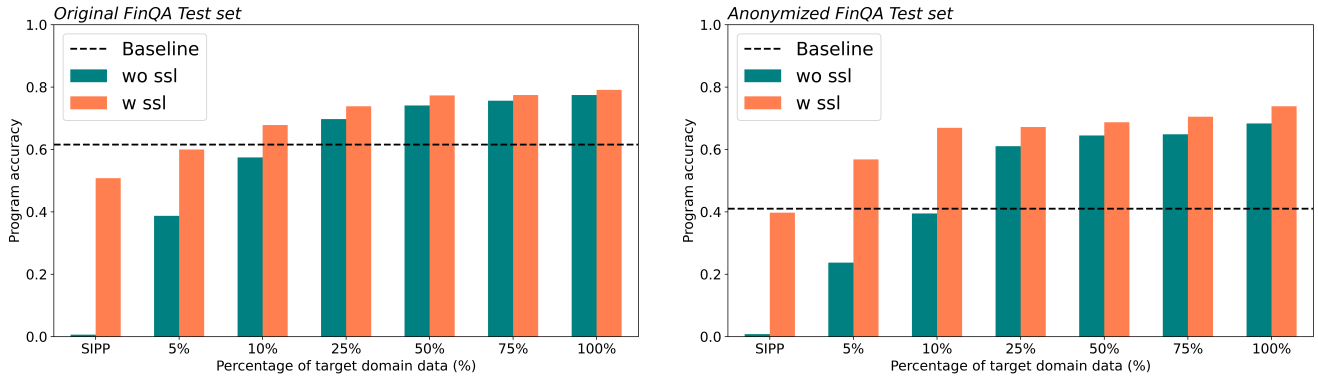

Fig. 3. Comparison of model performance trained with varying sizes of the FinQA training data and evaluated on both the original and anonymized FinQA test sets. The figure shows performance with and without self-supervised learning (SSL), with the dashed line indicating FinQANet baseline performance. Models with SSL consistently outperform those without SSL, especially with limited data, highlighting SSL's effectiveness.

strong dependency on the specific table headers encountered during training, even in settings where descriptions were provided.

Robustness through Anonymization Models trained with anonymization and provided description showed robust performance across various test settings, maintaining accuracies between 96 and 97.2. This indicates that training with anonymized headers helps the model to generalize better and reduce dependency on specific headers, thereby enhancing its ability to perform numerical reasoning tasks more robustly.

Overall, these results underscore the importance of reducing header dependency through anonymization techniques and highlight the potential of using descriptions to improve the model's performance in financial document QA tasks.

C. *Performance variation with further-train data size*

Figure 3 compares the performance of the model trained with varying sizes of the original FinQA training data (total of 4340 examples) and evaluated on both the original FinQA test set and the anonymized FinQA test set. This comparison aims to assess the impact of different sizes of further training data on the model's performance, specifically focusing on the effect of self-supervised learning.

Effectiveness of SSL on low-resources The model trained with SSL consistently outperforms the model without SSL across all sample sizes. This performance difference becomes particularly pronounced in low-resource scenarios. For instance, with the SIPP configurations, the impact of SSL is dramatic, resulting in significantly higher program accuracy compared to models without SSL. Given the high cost and effort required to construct thousands of expert-domain datasets in real-world applications, achieving reasonable performance with low-resource environment is notably impressive. This highlights the critical role of SSL in enhancing model performance when data is scarce, making it a valuable approach for practical applications.

Generalization Impact of the NRPD Framework When evaluated on an anonymized test set, the baseline model's performance drops sharply. However, NRPD demonstrate relative robustness to the anonymized test set. Notably, the SIPP configuration performs close to the baseline even when constructed with only one type of correct answer program. Furthermore, when using 5% of the data, the performance significantly surpasses the baseline. This robustness suggests that NRPD helps the model to generalize better by focusing on the reasoning process rather than memorizing specific headers.

VII. Conclusion

Our study tackles two main challenges in numerical reasoning within expert domains: the need for skilled human experts to label the data and the lack of generalization capability due to dependencies in specific table header and domain-specific jargons.

We introduced the NRPD framework to address these issues. By incorporating high-level sketch from query posers and utilizing Self-Supervised Learning (SSL), the model effectively learns numerical reasoning in an unbiased environment. Anonymization during training helps the model focus on reasoning abilities rather than dependencies.

The NRPD framework improves model transferability, enabling strong performance even with minimal target domain data. Additionally, it enhances non-experts' ability to process specialized documents, reducing associated costs. Overall, NRPD mitigates dependency issues, boosts model generalization, and lowers practical costs in expert-domain document processing.

VIII. Limitations and Future Work

Despite the contributions of our research, there are several limitations that need to be addressed. First, our method

requires additional information in the form of descriptions from humans, which introduces a dependency on human input. To mitigate this, future work should focus on generating descriptions using large language models (LLMs) instead of relying on human input.

Moreover, our use of self-supervised learning primarily aimed at enhancing the model's ability to utilize descriptions. However, the self-supervised learning technique we proposed is highly flexible and can be applied to any table data. Future research should explore extending and applying this method in different contexts to fully leverage its potential. These future directions aim to overcome the current limitations and further improve the efficiency and applicability of our framework.

IX. Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) grant (RS-2023-00280883, RS-2023-00222663), by the National Super computing Center with super computing resources including technical support (KSC-2023-CRE-0176), and partially supported by New Faculty Startup Fund from Seoul National University.

References

[1] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

[2] Wenhu Chen. Large language models are few (1)-shot table reasoners. *arXiv preprint arXiv:2210.06710*, 2022.

[3] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

[4] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*, 2020.

[5] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*, 2021.

[6] Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*, 2022.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

[9] Mor Geva, Ankit Gupta, and Jonathan Berant. Injecting numerical reasoning skills into language models. *arXiv preprint arXiv:2004.04487*, 2020.

[10] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020.

[11] Xianzhi Li, Xiaodan Zhu, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? an examination on several typical tasks. *arXiv preprint arXiv:2305.05862*, 2023.

[12] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*, 2023.

[13] Yujian Liu, Jiabao Ji, Tong Yu, Ryan Rossi, Sungchul Kim, Handong Zhao, Ritwik Sinha, Yang Zhang, and Shiyu Chang. Augment before you try: Knowledge-enhanced table question answering via table expansion. *arXiv preprint arXiv:2401.15555*, 2024.

[14] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[15] Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. *arXiv preprint arXiv:2206.01347*, 2022.

[16] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*, 2021.

SUMMARY OF THIS PAPER

A. *Problem Setup*

This paper addresses the challenge of performing effective question answering (QA) on numerical tabular data in expert domain documents. Traditional fine-tuning approaches often lead to models that depend on specific table headers and jargon, resulting in poor generalization across different domains. To overcome this, we propose the Numerical Reasoning utilizing Primitive Description (NRPD) framework, which enhances the generalization capability of models for numerical reasoning tasks by mitigating header dependency and utilizing self-supervised learning (SSL).

B. *Novelty*

The NRPD framework introduces several novel components. First, it incorporates anonymization by replacing tokens in row and column headers with arbitrary tokens during training. This reduces dependency on specific headers and improves generalization. Second, it uses self-supervised learning (SSL) to generate diverse numerical reasoning data without human labor and applies anonymization to prevent models from forming spurious dependencies on headers and jargon. Third, it employs high-level descriptions to define jargon and provide additional information, thereby enhancing the model's ability to handle novel tables and operations.

C. *Algorithms*

the NRPD framework employs anonymization during training. By replacing tokens in row and column headers with arbitrary tokens, the model is prevented from memorizing specific headers, thus focusing on understanding the underlying structure and relationships within the table data. This reduces header dependency and ensures that the model learns true numerical reasoning skills.

In addition, the NRPD framework leverages automatic data generation to create numerical reasoning datasets without the need for human annotation. This process involves collecting table and text contexts from source data, selecting sequences of operators and corresponding arguments from table cell values, and deriving answers.

The final component of the NRPD framework is self-supervised learning (SSL), which utilizes the dataset with applied anonymization to train the model under unbiased conditions. This is followed by further training on the target domain dataset, allowing the model to assimilate domain-specific knowledge, including questioning styles and expertise relevant to the domain. This process optimizes the model's performance, even with limited training data, ensuring robust numerical reasoning and improved cross-domain generalization.

D. *Experiments*

The experiments demonstrate the effectiveness of the NRPD framework in improving both in-domain and cross-domain performance. The baseline model trained on 10% of the FinQA dataset achieved an average performance of 27.10, with significant drops in cross-domain performance and on anonymized test sets, indicating poor generalization. Using descriptions without SSL (NRPD w/o SSL) led to noticeable performance gains, although dependency on specific terms persisted. Combining descriptions and SSL in the full NRPD framework resulted in substantial improvements. In the Finance domain, in-domain performance increased by 29.47 percentage points, outperforming the baseline trained on the full dataset. The framework also maintained robust performance across anonymized tests, showing enhanced generalization. Models trained with anonymization maintained high accuracy across various test settings, reducing header dependency and improving generalization. Additionally, SSL significantly boosted performance, especially in low-resource environments, highlighting its critical role in practical applications.