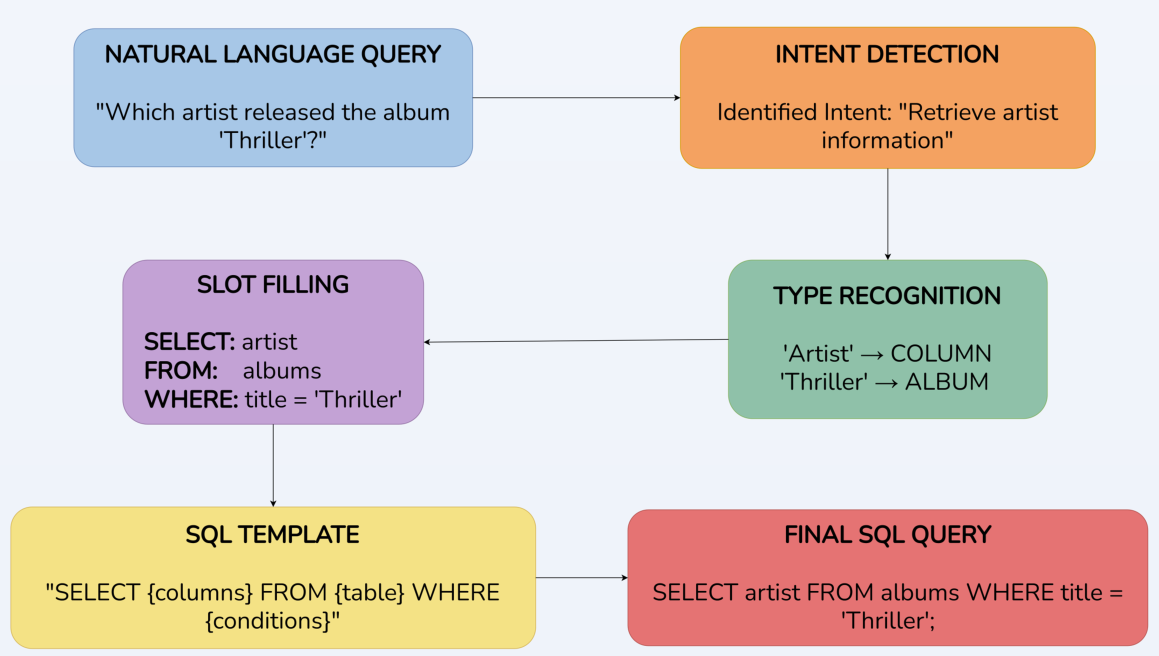
Optimizing Text-to-SQL Generation: A Dynamic Feedback Approach

Vamsi Saripudi
Department of Computer Science
Middle Tennessee State University
vs4s@mtmail.mtsu.edu

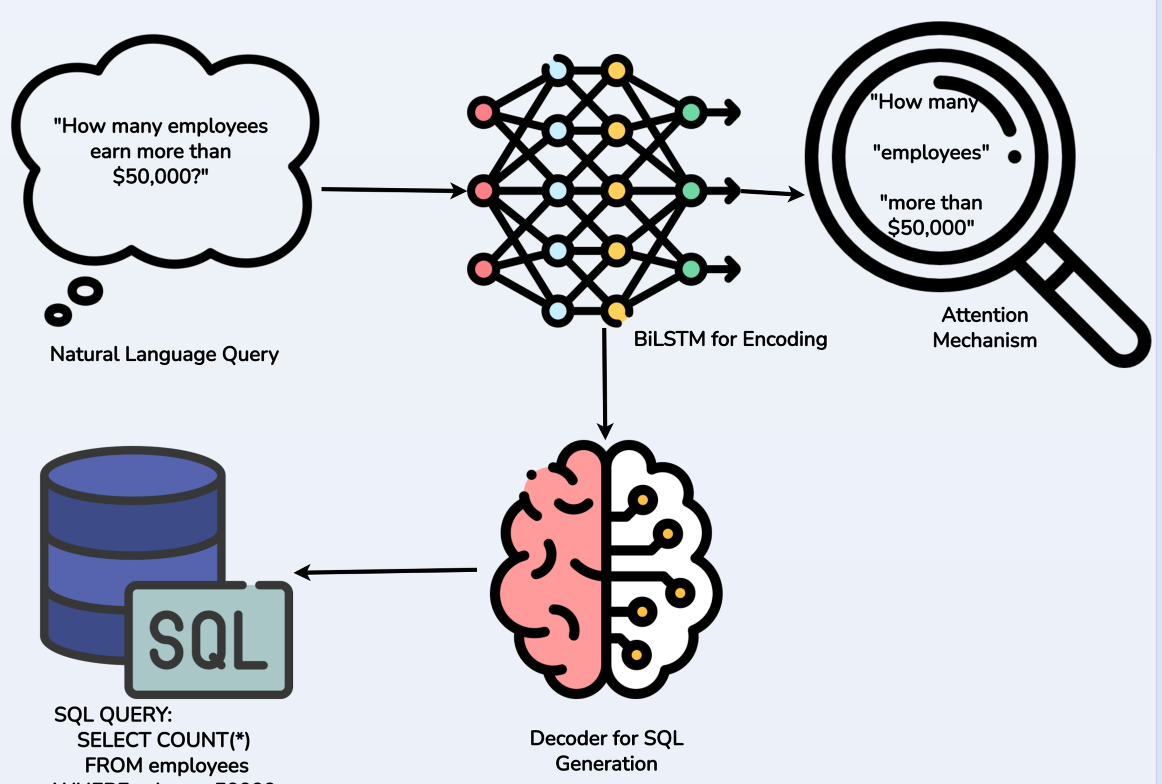
ABSTRACT

The usage of natural language interfaces for databases has gained a huge amount of traction, especially in text-to-SQL generation. This study focuses on developing a text-to-SQL generation pipeline using Google AI's Gemini 2.0 Flash Thinking Experimental 01-21 model. It aims to extend the prior methodologies by using bi-directional schema linking, context augmentation, and query selection. It integrates a dynamic feedback loop for iterative refinement of the queries for better error analysis. The proposed pipeline will be benchmarked against the Spider dataset using execution accuracy, valid efficiency score, strict recall, and non-strict recall as key metrics to test the performance of text-to-SQL generation. It explores the impact of using dynamic feedback in place of static feedback loops for iterative refinement of erroneous queries. An improvement in performance is anticipated compared to prior studies, as this dynamic feedback loop helps with better contextual understanding and more flexibility in rectifying errors. The outcome of this study may benefit organizations that are developing systems useful for non-technical users to use a database and get insights without the requirement of using SQL syntax. The study may provide useful insights to help people and organizations working on building database AI agents. It will help strengthen the collaboration between artificial intelligence and database management systems.

BACKGROUND

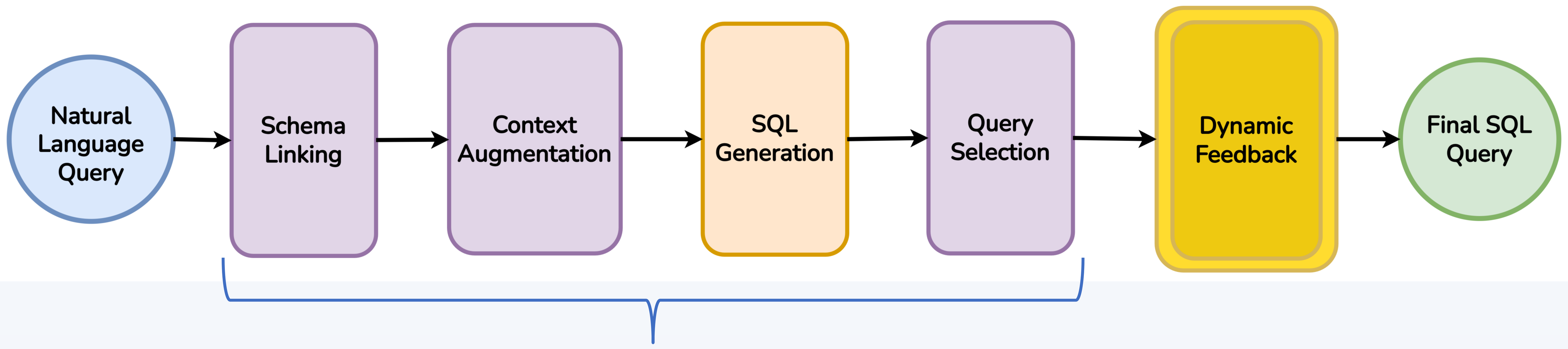


Slot-Filling Process for Text-to-SQL Generation [3]



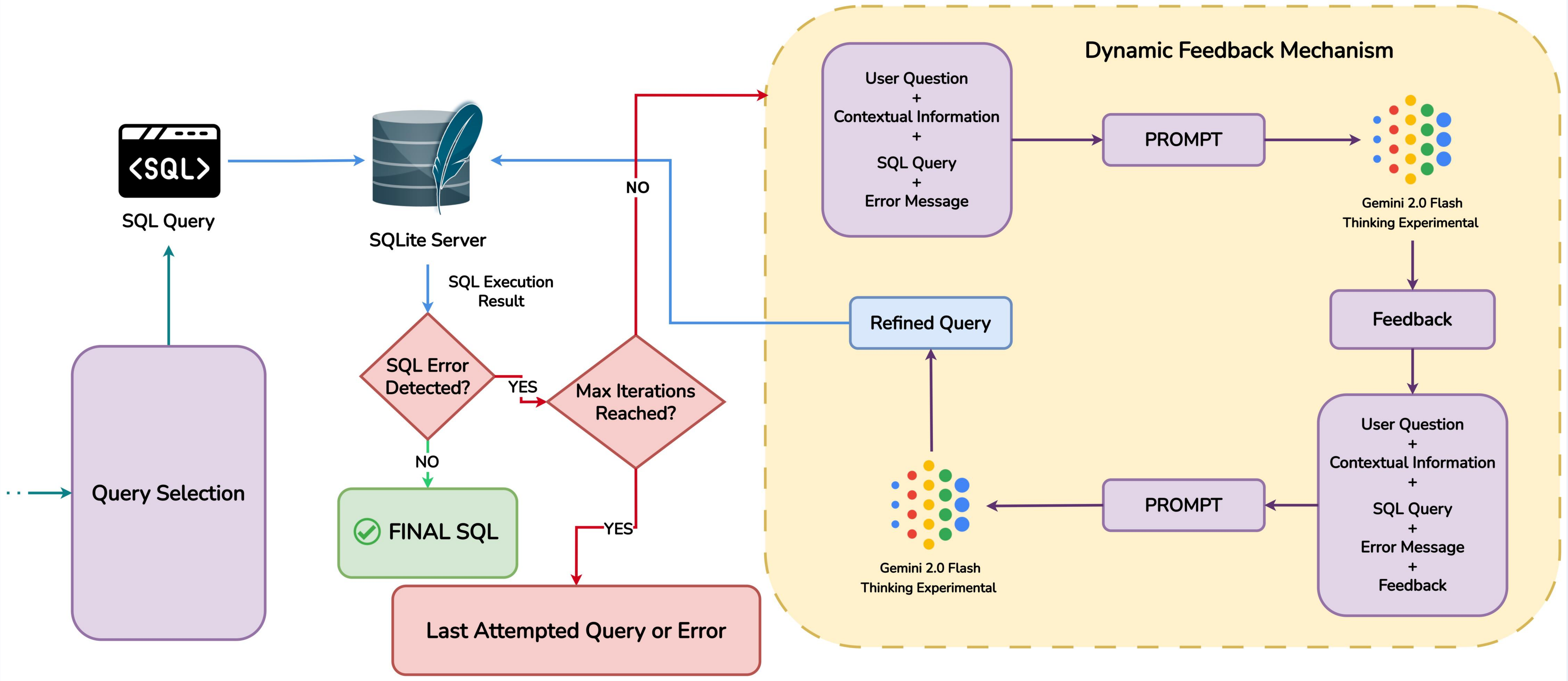
Sequence-to-Sequence Model for Text-to-SQL Generation [2]

METHODS



[4]

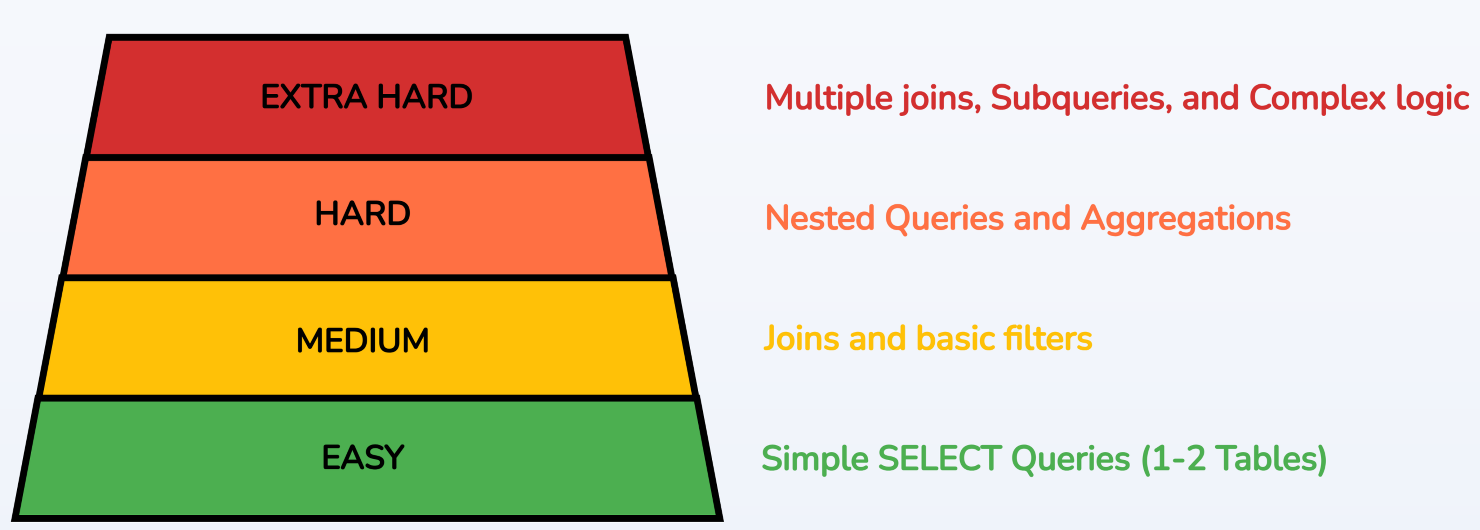
DYNAMIC FEEDBACK



PROMPT TEMPLATE STRUCTURE

- ROLE DEFINITION
- TASK OBJECTIVES
- INPUT SPECIFICATIONS
- OUTPUT FORMAT
- EXECUTION STEPS
- IMPORTANT NOTES

SPIDER DATASET QUERY CLASSIFICATION [1]



EXTRA HARD LEVEL QUERY

User Question: What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy) FROM country
WHERE name NOT IN
(SELECT T1.name FROM country AS T1
JOIN country_language AS T2
ON T1.code = T2.country_code
WHERE T2.language = "English" AND T2.is_official = "T");
```

ANTICIPATED RESULTS

Dynamic Feedback approach is expected to improve query accuracy by adapting to errors more effectively, and ensures better contextual understanding compared to static feedback.

User Query: "Find the average salary of employees working on at least 2 active projects in the Finance department.**"**

Initial SQL Query:

```
SELECT e.name, AVG(e.salary)
FROM employees e
JOIN projects p ON e.id = p.emp_id
WHERE e.department = 'Finance';
```

Lacks filtering for active projects and project count.

Static Feedback Refined Query:

```
SELECT e.name, AVG(e.salary)
FROM employees e
JOIN projects p ON e.id = p.emp_id
WHERE e.department = 'Finance'
GROUP BY e.department;
```

Groups correctly but includes all projects.

Dynamic Feedback Refined Query:

```
SELECT e.name, AVG(e.salary)
FROM employees e
JOIN projects p ON e.id = p.emp_id
WHERE e.department = 'Finance'
AND  p.status = 'active'
GROUP BY e.department
HAVING COUNT(DISTINCT p.id) >= 2;
```

Now filters employees based on active projects.

Unlike static feedback, which provides the same instructions in every iteration of refinement, the integration of dynamic feedback works by analyzing query errors, incorporating execution results, and generating adaptive refinements allowing the model to respond more effectively to errors.

CONCLUSIONS

- The anticipated results indicate improvement in ***execution accuracy***, ***valid efficiency score***, and ***recall rates***, resulting in a more robust pipeline for complex queries.
- Future work involves expanding benchmarking the pipeline across multiple datasets like **ATIS**, **WikiSQL**, **BIRD**, etc.

ACKNOWLEDGEMENTS

I extend my gratitude to Dr. Joshua Phillips and my peers for their constructive feedback and insightful suggestions, which have contributed to the improvement of this study.

REFERENCES

[1] T. Yu et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," 2018, *arXiv*. doi: 10.48550/ARXIV.1809.08887.

[2] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning," 2017, *arXiv*. doi: 10.48550/ARXIV.1709.00103.

[3] T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev, "TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation," 2018, *arXiv*. doi: 10.48550/ARXIV.1804.09769.

[4] Z. Cao, Y. Zheng, Z. Fan, X. Zhang, W. Chen, and X. Bai, "RSL-SQL: Robust Schema Linking in Text-to-SQL Generation," 2024, *arXiv*. doi: 10.48550/ARXIV.2411.00073.