Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 22-23, 2025

Enhancing the Reasoning Capabilities of a Domain-Specific Language Model for Nuclear Applications via Knowledge Distillation

Solji Park [a], Byeongjae Kim [a], Yonggyun Yu [bc*]

[a] *Korea University of Technology and Education,*
[b] *Korea National University of Science and Technology,*
[c] *Korea Atomic Energy Research Institute*
[*] *Corresponding author: ygyu@kaeri.re.kr*

Keywords : Distillation, Reasoning, LLM(large language model), Nuclear AI, AtomicGPT

1. Introduction

Recent advancements in reasoning-focused large language models (LLMs) have demonstrated remarkable improvements in structured reasoning and complex problem-solving capabilities. Specifically, models like DeepSeek R1 [1] have advanced significantly compared to earlier versions like DeepSeek V3[2]. The newer R1 version can handle challenging logical and mathematical tasks more effectively. This improvement underscores the importance of reasoning capabilities beyond general language skills, especially for specialized tasks.

For AtomicGPT [3], a specialized LLM for nuclear applications, requires advanced reasoning skills to handle complex nuclear tasks effectively. Improved reasoning allows AtomicGPT to make better-informed decisions and predictions, which are vital in sensitive and critical nuclear operations.

We enhance the reasoning capabilities of AtomicGPT through knowledge distillation, transferring structured reasoning patterns from a larger teacher model. This allows AtomicGPT to perform complex tasks more effectively despite limited model size.

Our method leverages a rigorously filtered reasoning dataset, termed s1K-1.1 [4], which is primarily derived from the reasoning traces of advanced models. In particular, we focus on using reasoning traces generated by the DeepSeek R1 model (supplemented by Google's Gemini model [5] where necessary) as the "teacher" data for distillation. By transferring the advanced reasoning patterns of these powerful teacher models into AtomicGPT, we seek to imbue the smaller 9B-parameter AtomicGPT with improved structured reasoning skills. AtomicGPT is a specialized LLM for applications in the nuclear domain, and strengthening its reasoning ability is expected to benefit AI-driven analysis and decision support in nuclear research. The following sections detail our dataset curation and distillation methodology, experimental results, and conclusions drawn from this work.

2. Methods

2.1 Knowledge Distillation Framework

Knowledge distillation is a technique where a smaller model (student model) learns structured reasoning skills from a larger, more advanced model (teacher model). Traditional knowledge distillation methods, as introduced by Hinton et al. (2015), typically rely on soft-label training, where the student model learns from the probability distributions of the teacher model by using KL divergence loss to match those distributions [6].

However, in our study the teacher model (DeepSeek R1) was not directly loaded, so soft labels (logits-based probability distributions) were not available when using the s1K-1.1 dataset for distillation. Instead, we employ a hard-label distillation approach, where the teacher models' responses serve as direct supervision for the student model. To optimize this process, we used cross-entropy loss (in place of KL divergence), which measures the difference between the predicted probability distribution and the actual label.

2.2 Applied Hard-Label Knowledge Distillation Process

Figure 1 illustrates our hard-label distillation workflow, which consists of three core stages: generating reasoning traces from teacher models, constructing a hard-labeled dataset, and training the student model with distillation. In this unified training step, the student model is optimized using hard-label supervision, and Low-Rank Adaptation (LoRA) is applied to improve memory and compute efficiency.

For example, a reasoning trace such as "If x > y and y > z, then x > z." is used as the target output for training. These teacher-generated outputs are treated as final labels during distillation.

This approach allows the student model to learn structured reasoning patterns directly from expert outputs without relying on access to the teacher's soft probability distributions. The use of cross-entropy loss, instead of KL divergence, enables effective training under hard-label supervision.

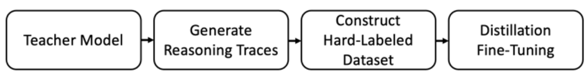

Fig. 1. Applied Hard-Label Distillation Workflow

- **Reasoning Trace Generation**: A teacher model (DeepSeek R1) is prompted to generate reasoning traces in response to mathematical problem-solving questions.

- **Hard-Labeled Dataset Construction**: The teacher's final outputs are curated into a training dataset by treating each response as a hard label. No soft probability distributions are preserved.

- **Distillation Fine-Tuning**: The student model is fine-tuned using the hard-labeled dataset with cross-entropy loss. This enables it to imitate the teacher's structured reasoning patterns.

- **LoRA Optimization**: Low-Rank Adaptation (LoRA [7]) is applied during fine-tuning to reduce memory usage and improve training efficiency while maintaining reasoning quality.

Unlike traditional distillation approaches that rely on soft labels or logits, our method enables structured reasoning transfer solely from finalized textual outputs. This hard-label distillation framework is particularly suitable when teacher logits are inaccessible.

2.3 Fine-Tuning Procedure

We applied a fine-tuning strategy using LoRA (Low-Rank Adaptation) to enhance model adaptability while minimizing memory and computational costs. Our fine-tuning process involved the following key steps:

2.3.1 Training Environment

The fine-tuning process was conducted using a hybrid GPU setup consisting of two NVIDIA A100 GPUs with 40GB of VRAM and six TITAN RTX GPUs with 24GB of VRAM each. The training was executed using PyTorch and Hugging Face Transformers.

2.3.2 Data Processing & Preparation

Structured reasoning traces were converted into a model input format aligned with our fine-tuning framework. We applied a data collator during preprocessing to maintain padding consistency and enable efficient batch processing. The dataset was then split into training and validation subsets to allow systematic evaluation of performance throughout fine-tuning. DeepSeek R1 served as the primary teacher model. However, if a problem instance was marked 'No' (indicating DeepSeek R1 did not produce a solution), we used the Gemini model's response instead. For instance, a reasoning trace like "If the derivative of a function is positive, then the function is increasing" is tokenized and used as the correct output for model training.

As part of the preprocessing pipeline, we transformed the s1K-1.1 dataset to match the input format expected by the Gemma2-based AtomicGPT model. Specifically, the `deepseek_thinking_trajectory` and `deepseek_attempt` fields were tokenized using special instruction-style tokens, `<start_of_turn>think` and `<start_of_turn>answer`, respectively. These were then concatenated into a single text field, forming a structured input sequence that mimics multi-turn reasoning dialogues. This approach allowed the model to learn the progression from thought formulation to final answer within a single context, improving alignment with its tokenization schema and enhancing reasoning coherence.

An example of such a training instance is as follows:

```
<bos><start_of_turn>user
Given a rational number, write it as a fraction in lowest terms and
calculate the product of the resulting numerator and denominator.
For how many rational numbers between 0 and 1 will $20_{}^{}!$ be
the resulting product?<end_of_turn>
```

```
<start_of_turn>model
<start_of_turn>think
Alright, so I need to figure out how many rational numbers between
0 and 1 have a product of the numerator and denominator (when
written in lowest terms) equal to 20 factorial. Let me start by
understanding the problem.
...
```

```
<start_of_turn>answer
Answer: Given a rational number between 0 and 1, we need to
write it as a fraction in lowest terms \(\frac{a}{b}\) and calculate
the product of the resulting numerator and denominator. We need
to determine how many such rational numbers exist where the
product \(a \times b = 20!\).
...
Thus, the number of rational numbers between 0 and 1 for which
the product of the numerator and denominator in lowest terms is
\(20!\) is \(\boxed{128}\).<end_of_turn><eos>
```

Fig. 2. Example of Preprocessed Multi-Turn Input for AtomicGPT Fine-Tuning

2.3.3 Fine-Tuning Configuration

- LoRA settings:
  - Rank (r) = 16
  - LoRA alpha = 32
  - Dropout rate = 0.1

- Optimization settings:
  - Optimizer: Paged AdamW
  - Learning rate: 2e-5 with a cosine decay scheduler
  - Warmup ratio: 0.03 for training stability
  - Maximum sequence length: 8192 tokens (Based on a maximum context window of 32,768 tokens to prevent truncation of long reasoning sequences. [8])

- Batch size: 2, with gradient accumulation steps set to 8

2.3.4 Training

The model was fine-tuned for 5 epochs—a setting determined through multiple tests to be optimal. This duration ensured that the student model effectively captured the structured reasoning patterns from the teacher's responses.

We used cross-entropy loss as the primary objective to align the student model's output with the teacher model's output, thereby reinforcing structured reasoning. To enhance computational efficiency while maintaining accuracy, we conducted fine-tuning in FP16 (half-precision) mode. Mixed precision techniques were leveraged to reduce memory consumption and accelerate computation. The entire fine-tuning process took approximately one hour to complete.

3. Experimental Evaluation and Results

3.1 Distilled Model Evaluation

We evaluated the distilled AtomicGPT model on two challenging reasoning benchmarks: MATH-500 (a set of 500 multi-step mathematics problems [9]) and GPQA Diamond (a graduate-level scientific Q&A benchmark [10] designed to resist simple lookups and therefore require reasoning). The results are summarized in Table 1, comparing the original AtomicGPT-gemma2-9B with our distilled AtomicGPT-Distill-gemma2-9B on these benchmarks. We report performance using the pass@1, pass@5, and self-consistency metrics. Pass@1 and pass@5 denote the percentage of problems the model solves correctly with its top answer and within its top five answers, respectively. The self-consistency score reflects the model's internal consistency in reasoning, with higher values indicating more reliable reasoning paths.

Benchmark	**Model**	**pass@1**	**pass@5**
MATH-500	AtomicGPT-gemma2-9B	24.0	37.4
	AtomicGPT-**Distill**-gemma2-9B	23.4	58.2
GPQA Diamond	AtomicGPT-gemma2-9B	22.2	66.7
	AtomicGPT-**Distill**-gemma2-9B	29.8	68.1

Table 1: Performance of Base vs. Distilled AtomicGPT on Reasoning Benchmarks.

Overall, the distilled AtomicGPT model demonstrated notable improvements on these reasoning tasks. On the MATH-500 benchmark, the distilled model's **pass@5 score jumped to 58.2% (from 37.4% in the base model)**.

A similar trend was observed on the GPQA Diamond benchmark: the distilled model's **pass@1 score jumped to 29.8% (versus 22.2% for the base model)**, representing a major improvement in single-answer accuracy. It also attained a higher pass@5 score (68.1% vs. 66.7%), further indicating that it follows more stable and structured reasoning paths. These results highlight the effectiveness of our knowledge distillation and dataset refinement approach in enhancing AtomicGPT's complex problem-solving capabilities compared to the original model. In particular, the distilled model exhibits greater robustness and deeper reasoning, underscoring the practicality of this distillation strategy.

While this approach effectively transfers reasoning patterns, it has some limitations. Without soft labels, the student model lacks insight into the teacher model's confidence in its answers, which may affect nuanced decision-making. In addition, emphasizing structured multi-step reasoning during training may lead the model to favor deeper problem-solving over direct answer retrieval, slightly reducing its performance on straightforward questions. The heavy filtering of simpler problems in our dataset limited the model's exposure to direct-answer scenarios, which likely impacted its pass@1 accuracy. Nonetheless, the notable improvements in MATH-500 pass@5 and GPQA Diamond pass@1 demonstrate that the model has become more reliable at handling complex multi-step reasoning tasks.

3.2 Ongoing and Future Work

While the distilled AtomicGPT has shown improved reasoning capabilities, we are exploring further refinements. One direction involves evaluating alternative teacher models beyond DeepSeek R1 to provide more diverse and generalized reasoning examples.

We also recognize that not employing test-time scaling [8] may have limited our performance gains. Test-time scaling is a training-time calibration technique introduced in the s1 paper for the s1K-1.1 dataset. This method optimizes model performance during inference without increasing the training data volume and has demonstrated significant improvements in low-data regimes. We plan to experiment with test-time scaling to assess its potential impact, particularly under limited-data conditions.

Another limitation of our current framework is the absence of soft labels, which prevents the student model from leveraging the confidence levels encoded in the teacher model's predictions. This lack of nuance may hinder the diversity and adaptability of reasoning strategies acquired during distillation. To address this, we aim to integrate soft labels into our training pipeline by expanding our data extraction process, enabling a more effective balance between knowledge fidelity and model generalization.

4. Conclusions

This study proposed a knowledge distillation strategy to enhance the multi-step reasoning capabilities of AtomicGPT-gemma2-9B, a domain-specific large language model. By leveraging a high-quality reasoning dataset generated from advanced teacher models, we fine-tuned the student model using hard-label distillation techniques. As a result, the distilled model demonstrated notable performance improvements—achieving a significant increase in pass@5 accuracy on the MATH-500 benchmark and a higher pass@1 score on the GPQA Diamond benchmark. These results validate the effectiveness of our distillation approach in transferring structured reasoning patterns.

Improving the reasoning capacity of AtomicGPT has significant implications for nuclear engineering and safety-critical domains, where complex decision-making requires robust, interpretable AI assistance. Enhanced reasoning performance enables more accurate safety assessments, anomaly detection, and operational diagnostics, thus promoting greater trust in AI-driven systems. This advancement supports the broader adoption of foundation models in high-stakes applications where reliability and domain-specific expertise are paramount.

Future work will explore diverse teacher models, semi-supervised learning with soft labels, and optimization of inference-time performance. Additionally, we aim to scale the distilled model to larger architectures and apply it to more intricate reasoning tasks, including symbolic computation and mathematical theorem proving. As the model continues to be refined, AtomicGPT is expected to evolve into a powerful and trustworthy tool for advanced reasoning in both nuclear and general scientific applications.

REFERENCES

[1] DeepSeek-AI, “DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning,” arXiv:2501.12948, 2025.

[2] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., ... & Piao, Y. (2024). Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

[3] Yeom Seung Don, ChangSu Choi, Lim KyungTae, & Yu Yong Gyun (2024-11-20). Development and Performance Evaluation of a Domain-Specific Language Model for Nuclear: A Comparative Study Using a Custom-Built Dataset. Proceedings of Symposium of the Korean Institute of communications and Information Sciences, 경북.

[4] Muennighoff, N., et al. (2025). s1K-1.1: A structured reasoning dataset (Version 1.1) [Dataset]. Hugging Face. https://huggingface.co/datasets/s1K-1.1

[5] Google. (2025). Gemini 2.0 Flash Thinking | Gemini API | Google AI for Developers. Retrieved from https://ai.google.dev/gemini-api/docs/thinking

[6] G. Hinton et al., “Distilling the Knowledge in a Neural Network,” NeurIPS Workshop, 2015.

[7] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). Lora: Low-rank adaptation of large language models. ICLR, 1(2), 3.

[8] Muennighoff, Niklas, et al. "s1: Simple test-time scaling." arXiv preprint arXiv:2501.19393 (2025).

[9] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., ... & Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.

[10] Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., & Bowman, S. R. (2023). GPQA: A Graduate-Level Google-Proof Q&A Benchmark. arXiv preprint arXiv:2311.12022.