# An Automated Fine-Tuning Pipeline for Nuclear Domain-Specific LLMs using a Conversational Human-in-the-Loop Agent Framework

Seungdon Yeom [a], Hogeon Seo [ab*]
[a]*Korea Atomic Energy Research Institute, 111 Daedeok-daero 989beon-gil, Yuseong-gu, Daejeon, Republic of Korea*
[b]*University of Science and Technology, 217 Gajeong-ro, Yuseong-gu, Daejeon, Republic of Korea*
[*]*Corresponding author: hogeony@kaeri.re.kr*

***Keywords :*** large language model, automated fine-tuning, LLM agent, on-premises LLM, nuclear domain

## 1. Introduction

Large language models (LLMs) adapted to the nuclear domain can support regulatory document question answering, operating-procedure retrieval, and knowledge management. Our previous work on AtomicGPT [1] demonstrated the value of a nuclear domain-specific LLM; however, constructing such models remains labor-intensive: machine-learning specialists must profile the computing environment, select a parameter-efficient training method and hyperparameters, supervise multi-hour training jobs against failures such as out-of-memory (OOM) errors, and iterate through evaluation rounds. This expertise barrier is compounded in the nuclear field by a domain-specific constraint: sensitive corpora cannot leave institutional boundaries, which excludes cloud-based AutoML and fine-tuning services and mandates a fully on-premises workflow, often on air-gapped networks.

This paper presents an agent-based framework that automates the end-to-end fine-tuning workflow—environment profiling, hardware-aware training-plan generation, training execution and failure recovery, evaluation, and iterative optimization—while keeping the human expert in control through five approval gates operated from a conversational web console. The contributions are threefold: (i) a crash-safe, resumable state-machine pipeline in which each major proposal can be approved, rejected, or modified by the researcher in natural language; (ii) an agent decision layer in which a locally hosted LLM, invoked through bounded ReAct-style tool-use loops [2], designs training plans from the data and hardware profile, diagnoses failures by inspecting logs and data, and proposes training adjustments based on error analysis. Agent outputs are checked against deterministic rules, with rule-based fallbacks used when the LLM is unavailable; and (iii) a five-model case study of nuclear-knowledge injection (2B–31B; Gemma 4 and Qwen3.5/3.6 families) executed end-to-end with locally served open-weight models only, in which agent-designed plans injected measurable domain knowledge into the compact models while preserving performance on the other evaluation tasks.

## 2. Framework Architecture and Methods

In this section, we describe the pipeline and its human approval gates (2.1), the agent decision layer that designs and validates every training plan (2.2), and the design of a five-model case study on nuclear knowledge injection (2.3).

### *2.1 Pipeline and Approval Gates*

Fig. 1 shows the overall architecture. A run proceeds through a deterministic state machine: environment profiling, a data-plan gate (G1), preprocessing, a training-plan gate (G2), training, evaluation, a review gate (G4), and final model selection (G5). For minor training failures (e.g., out-of-memory), the framework automatically adjusts the training configuration and retries the run, whereas major failures trigger a recovery gate (G3) for agent-based diagnosis. At each gate, the researcher may approve or reject the proposal, or request a modification in natural language (e.g., “reduce epochs to 2”), which the agent parses into a revised proposal. The framework stores all proposals, decisions, and execution traces in an append-only audit log, allowing a run to resume safely after a process crash.

### *2.2 Agent Decision Layer*

The agent generates a training plan by inspecting the dataset statistics and GPU profile, including the training method, LoRA rank, learning rate, epochs, batch configuration, and sequence cutoff. A deterministic layer then validates and adjusts each proposal to fit the available hardware, switching to a more memory-efficient method (full fine-tuning → LoRA [3] → QLoRA) when necessary, and falls back to a rule-based plan when no LLM is available; training execution is delegated to an established fine-tuning library [4]. For failure diagnosis and follow-up training decisions, the agent runs a bounded ReAct loop over read-only inspection tools with a fixed maximum number of steps; state-changing actions are never exposed to the LLM. The evaluation stage also evaluates the corresponding base model, enabling direct comparison with the fine-tuned model.

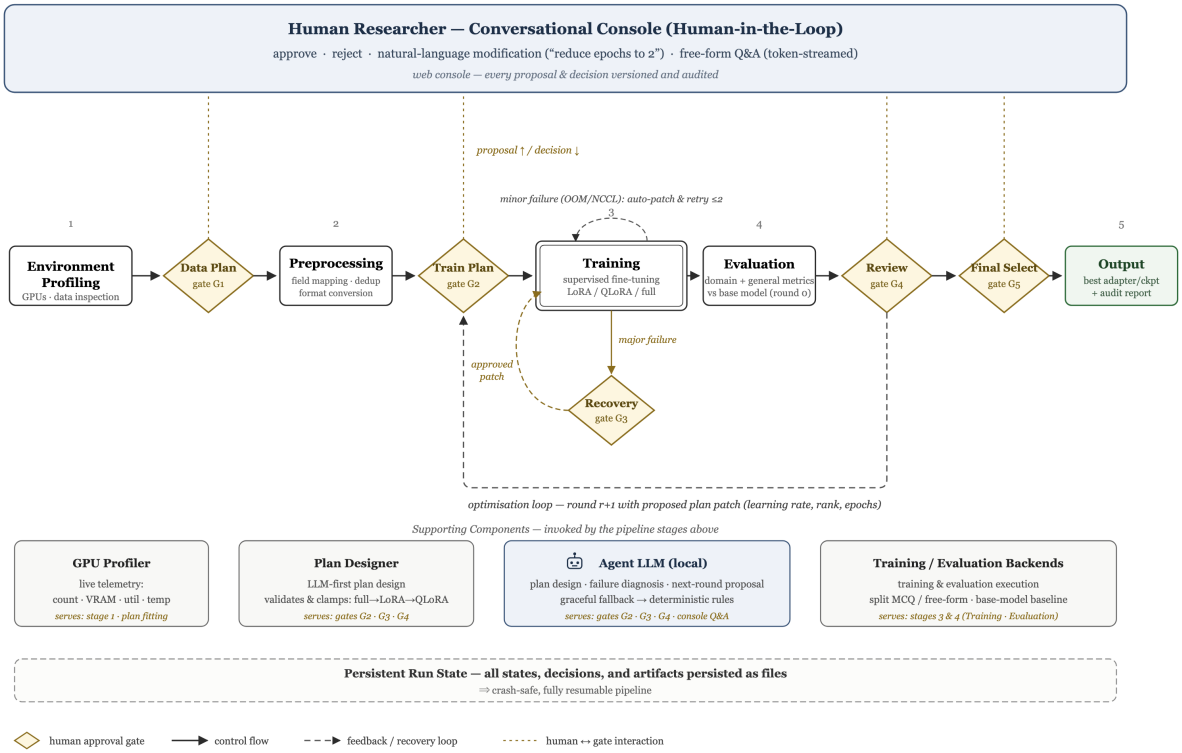



Fig. 1. Overview of the proposed framework: pipeline stages, five human approval gates (G1–G5), failure-recovery loops, and the error-analysis-driven optimization loop.

### *2.3 Case Study Design*

We deployed the framework on an 8-GPU NVIDIA B200 server and fine-tuned five recent open-weight instruction-tuned models: Gemma-4-E2B-it, Gemma-4-E4B-it, Gemma-4-31B-it, Qwen3.5-4B and Qwen3.6-27B. The training corpus consists of 1,000 Korean short-form QA pairs generated from a public nuclear-education wiki (Atomic Wiki, Seoul National University) by a locally served open-weight 120B model (Nemotron-120B-Super), which also serves as the framework's agent LLM. To reduce train–evaluation contamination, generated questions with a token-set Jaccard similarity greater than 0.5 to any evaluation item were discarded; the maximum token-set Jaccard similarity between any retained training question and an evaluation item was 0.11. Evaluation assesses domain knowledge and response quality using three evaluation tasks: (a) 100 four-choice items from the U.S. NRC Generic Fundamentals Examination (GFE) bank, scored by answer-choice extraction accuracy; (b) the 100-question NuclearQA free-form benchmark [5], scored on a 0–10 scale by an independent judge model (EXAONE-4.0-32B), which was different from the corpus-generation model to reduce potential evaluation bias; and (c) an auxiliary set of 100 Korean licensing-exam-style four-choice questions. Because free-form judging is still model-based, we treat the multiple-choice metrics, which do not require an LLM judge, as the primary evidence.

The five runs were launched concurrently, one GPU each, from the conversational console, with human involvement limited to gate approvals. The agent designed a distinct plan for each model after inspecting the dataset statistics and the GPU profile; representative plans for the three compact models are shown in Table I.

Table I. Training plans designed by the agent for the three compact models. batch = per-device batch × gradient accumulation; steps = resulting optimizer steps (1,000 samples × epochs ÷ effective batch).

| Model | Rank | Epochs | LR | Batch | Cutoff | Steps |
|---|---|---|---|---|---|---|
| Gemma-4-E2B-it | 16 | 1.0 | 2e-4 | 4×4 | 512 | 62 |
| Gemma-4-E4B-it | 32 | 0.48 | 2e-4 | 4×4 | 512 | 30 |
| Qwen3.5-4B | 32 | 1.0 | 1e-4 | 8×4 | 512 | 31 |

## 3. Quantitative Evaluation of Nuclear Knowledge Injection

All five runs completed training and evaluation on all three tasks without any hand-written configuration files. Under the agent-designed plans (Table I), the multiple-choice results in Table II show measurable domain-knowledge gains: GFE accuracy rose by 11 percentage points for E2B and 7 percentage points for E4B, and remained unchanged for Qwen3.5-4B, while the auxiliary Korean set showed small gains of one to four percentage points for all three models. On free-form answers, the independent judge score improved for Qwen3.5-4B (7.77 → 8.59), while the other models' scores were essentially unchanged. In a separate follow-up run on Gemma-4-E4B-it, an additional agent-proposed training round yielded no further multiple-choice improvement, suggesting that additional training provided little benefit under this setting.

Table II. Evaluation results (base → fine-tuned). GFE MCQ = answer-extraction accuracy on the GFE bank; judge = mean score (0–10) on NuclearQA from an independent judge model (EXAONE-4.0-32B); K-MCQ = auxiliary Korean four-choice set. The two largest models (Gemma-4-31B-it, Qwen3.6-27B) are discussed in the Limitations section.

| Model | GFE MCQ (%) | Judge (0–10) | K-MCQ (%) |
|---|---|---|---|
| Gemma-4-E2B-it | 31 → 42 | 7.42 → 7.38 | 34 → 35 |
| Gemma-4-E4B-it | 35 → 42 | 7.78 → 7.90 | 35 → 39 |
| Qwen3.5-4B | 47 → 47 | 7.77 → 8.59 | 43 → 45 |

## 4. Limitations

The same experiment also included the two largest models, Gemma-4-31B-it and Qwen3.6-27B, which showed little measurable improvement: multiple-choice accuracy was essentially unchanged (GFE 65 → 65% and 61 → 60%, with the Korean set similarly stable), and the independent judge scores were also essentially unchanged (8.15 → 8.32 and 7.93 → 7.92). The larger models started from substantially higher multiple-choice baselines, and the 1,000-pair distilled corpus may have been insufficient to produce measurable gains at this scale. The framework operated successfully with the 27–31B models, but demonstrating measurable knowledge gains at this scale may require a substantially larger domain corpus. Two further caveats apply: free-form quality assessment remains model-based even with an independent judge; and all findings derive from a single Korean-language corpus and a single method family (LoRA), so generality across corpora and training methods remains to be established.

## 5. Conclusions

We presented an agent-based framework for fully on-premises domain-specific LLM fine-tuning, in which the agent handles the fine-tuning workflow while the researcher retains control through approval gates. In the five-model case study, all runs completed training and evaluation without hand-written configuration files, with human involvement limited to gate decisions. Agent-designed plans also produced measurable domain-knowledge gains in the compact-model experiments. These results also indicate that no single metric fully captures the effects of domain adaptation, as multiple-choice accuracy and free-form response quality changed differently across models.

Future work will evaluate the framework with larger nuclear-domain corpora, including data from AtomicGPT [1], and additional training methods such as continued pre-training and preference-based learning. We also plan to add general-capability benchmarks such as KMMLU and support automated hyperparameter optimization. In addition, we plan to develop deterministic validators for agent-generated training plans and a multi-agent verification layer in which an independent critic agent reviews the primary agent's decisions before they are presented for human approval.

## ACKNOWLEDGMENTS

This research was supported by a grant from the Korea Atomic Energy Research Institute (KAERI) (No. 526140-26), and the Advanced GPU Utilization Support Program, funded by the Government of the Republic of Korea (Ministry of Science and ICT) (No. 02-26-01-0492).

## REFERENCES

[1] Yeom, S., Lim, K., & Yu, Y. (2026). AtomicGPT: A Domain-Adapted Large Language Model for Secure On-Premises Applications in Nuclear Engineering. Nuclear Engineering and Technology, 104565.
[2] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.
[3] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
[4] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., & Luo, Z. (2024, August). 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) (pp. 400-410).
[5] Acharya, A., Munikoti, S., Hellinger, A., Smith, S., Wagle, S., & Horawalavithana, S. (2023). Nuclearqa: A human-made benchmark for language models for the nuclear domain. arXiv preprint arXiv:2310.10920.