Antero Vuorela

Enhancing Large Language Models Through Post-Training, External Tool Integration and External Information Integration

School of Technology and Innovations
Bachelor's thesis
Automation and information technology

Vaasa 2025

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Antero Vuorela
Title of the thesis: Enhancing Large Language Models Through Post-Training, External Tool Integration and External Information Integration
Degree: Bachelor of Technology
Discipline: Automation and Information Technology
Supervisor: Janne Koljonen
Year: 2025 **Pages:** 30

ABSTRACT:

Suuret kielimallit (Large Language Models, LLMs) ovat nousseet keskeiseksi osaksi nykyaikaista tekoälyä, mahdollistaen sovelluksia keskusteluboteista sisällöntuotantoon ja tutkimusavustajiin. Vaikka laajamittainen esikoulutus tekstiaineistoilla on tuottanut vaikuttavia yleiskielisiä kyvykkyyksiä, se on myös jättänyt merkittäviä luontaisia rajoituksia kuten taipumus hallusinaatioihin, vanhentunut tieto ja rajallinen päättelykyky. Tämä kandidaatintyö tarkastelee keinoja parantaa suurien kielimallien suorituskykyä jälkikoulutuksella sekä ulkoisten työkalujen ja informaation integrointia osaksi mallien toimintaa.

Työssä tutkitaan useita jälkikoulutusmenetelmiä, mukaan lukien valvottu hienosäätö (supervised fine-tuning), vahvistusoppiminen ihmisen tai tekoälyn palautteen avulla (RLHF, RLAIF), suora preferenssien optimointi (DPO) sekä päättelytehoa kasvattava testiaikaskaalaus (Test-time scaling). Lisäksi käsitellään ulkoisten tietolähteiden hyödyntämistä, kuten hakuun perustuvaa tekstintuottoa (Retrieval-augmented generation) ja työkalujen sekä sovellusliittymien integrointia, mukaan lukien Model Context Protocol. Analyysi osoittaa, että nämä menetelmät täydentävät toisiaan, ja parhaan lopputuloksen saavuttamiseksi niitä kannattaa yhdistää järjestelmällisesti.

Tulokset osoittavat, että mikään yksittäinen menetelmä ei riitä ratkaisemaan kaikkia LLM-mallien haasteita. Sen sijaan suorituskyky ja luotettavuus paranevat selvästi, kun eri tekniikat yhdistetään harkitusti osaksi kokonaisuutta. Tämä työ tarjoaa katsauksen suurten kielimallien jälkikoulutus menetelmistä ja suosituksia älykkäämpien, kontekstitietoisempien sovellusten rakentamiseen.

KEYWORDS: large language models, post-training, fine-tuning, reinforcement learning, retrieval-augmented generation, tool integration, AI alignment, model context protocol

Contents

1 Introduction 5

2 Large Language Models 7

2.1 Overview of Large Language Models 7

2.2 Foundational technology 7

2.2.1 Machine learning and deep learning 7

2.2.2 Natural language processing and text representation 8

2.2.3 Language modelling and transformer architecture 9

2.2.4 Foundation models 11

2.3 The function of large language models 11

2.4 The importance of post-training improvement of large language models 12

3 External tool usage and post-training enhancement of LLMs 14

3.1 Post-training techniques 14

3.2 External tool and information integration 19

4 Discussion 23

4.1 Post-training techniques analysis 23

4.2 External tool and information integration 23

4.3 Integration strategies 24

4.4 Practical considerations and trade-offs 25

5 Conclusion 26

References 28

Figures

Figure 1. The transformer model architecture. 10

Figure 2. The supervised fine-tuning process. 15

Figure 3. Comparison of RLHF and RLAIF for preference alignment. 16

Figure 4. RLHF and DPO compared. 16

Figure 5. The process of OAIF. 17

Figure 6. Best-of-N sampling and sequential revisions compared. 18

Figure 7. The retrieval-augmented generation process. 20

Figure 8. Tool use with and without MCP compared. 21

Tables

Table 1. Enhancement techniques divided in layers based on their function. 24

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CoT	Chain-of-Thought
DAP	Direct Alignment from Preferences
DL	Deep Learning
DPO	Direct Preference Optimization
LLM	Large Language Model
MCP	Model Context Protocol
ML	Machine Learning
NLP	Natural Language Processing
OAIF	Online AI Feedback
RAG	Retrieval-Augmented Generation
RL	Reinforcement Learning
RLHF	Reinforcement Learning with Human Feedback
RLAIF	Reinforcement Learning with AI Feedback
SFT	Supervised Fine-Tuning
TTC	Test-Time Computation
UI	User Interface

1 Introduction

Large language models (LLMs) have transformed the landscape of natural language processing, yet they often show limitations in areas such as reasoning capabilities, ethical understanding, and domain specific performance. These issues largely stem from their pre-trained architectures (Tie et al., 2025). While pre-training on vast corpora has laid the foundations for these models, research is shifting toward post-training techniques to achieve further improvements (Kumar et al., 2025).

Post-training is a way of specializing a language model after the foundations have been laid in pre-training. Current techniques include fine-tuning, which uses curated data to update the parameters and specialize the model, reinforcement learning, which uses feedback to alter the response of the model and test-time scaling which allows the model to allocate computational resources during inference to allow for more reasoning (Kumar et al., 2025). For further improving the accuracy of LLMs, techniques such as retrieval-augmented generation to allow models to access external data (Gao et al., 2023), and external tool integration for better accuracy in answering questions (Zhuang et al. 2023) should be considered.

The central question this thesis examines is: How can we effectively build on top of large language models through post-training training techniques as well as external tool and information integration. By investigating these interconnected strategies, this bachelor's thesis aims to provide insights for alternative approaches for significantly improving the performance of LLMs post pre-training.

The purpose of this bachelor's thesis is to examine the different ways in which large language models can be improved post-training using different tools and techniques. The study will analyse how different post-training techniques can improve model performance, assess the benefits of contextual data integration, and explore the role of exter-

nal tool integration in expanding the functionality of LLMs. The goal is to build a comprehensive understanding of post-training enhancement techniques and look at how they complement each other.

In chapter 2, a general overview of LLMs will be given along with an examination of why post-training enhancement techniques of LLMs are important. Chapter 3 will give a more in-depth look into the function and potential of different post-training techniques. Chapter 4 focuses on analysis of the techniques discussed in the previous chapter, with the aim of finding the best ways of integrating post-training techniques. Chapter 5 will conclude the thesis and summarize the findings.

2 Large Language Models

Large language models (LLMs) are a relatively new AI technology and the fast development of them has outpaced education about it. This section aims to explain the foundational AI technology that makes modern LLMs possible, explain what LLMs are and how they work and explain why post-training improvements of LLMs are important.

2.1 Overview of Large Language Models

Large language models are a significant breakthrough in AI and natural language processing. They are a type of foundation model that can generate and understand language and other content to accomplish diverse tasks. An LLM is a machine learning model with a very large number of parameters, often billions or trillions, which is trained on massive text datasets. LLMs can infer from context, generate logical human like responses, translate between languages, summarize, answer questions and assist with coding and they are already revolutionizing many fields such as chatbots, virtual assistants, content generation, research assistance and language translation (IMB, 2023).

2.2 Foundational technology

Large Language Models (LLMs) are a recent development in the field of generative AI, and they are only possible due to big innovations in subfields of AI such as deep learning, natural language processing and transformer models. This section aims to give context on what technologies LLMs rely on.

2.2.1 Machine learning and deep learning

Most current AI systems use machine learning. In AI predictive machine learning, models trained on historic data are used to make future predictions. The rise of machine learning

changed the way AI systems were built previously, rather than giving the program instructions on how to solve the task, machine learning would learn it from the data (Bommasani et al., 2022). Deep learning is a subset of machine learning that simulates the decision-making power of the human brain by using neural networks with multiple layers (Holdsworth & Scapicchio, 2024). Deep learning differs from conventional machine learning by its ability to effectively process natural unannotated data (LeCun et al., 2015).

2.2.2 Natural language processing and text representation

Natural language processing (NLP) is a subfield of computer science and artificial intelligence that uses machine learning to enable computers to understand and communicate using human language. Natural language processing merges rule-based computational linguistics with statistical modelling, machine-learning algorithms, and deep learning, enabling computers and digital systems to detect, interpret and generate text and speech (Stryker & Holdsworth, 2024).

In pre-training LLMs tokenize text, meaning the process of converting text into sequences of tokens. The tokens can be words, sub-words, or characters. A token is the smallest unit of meaning in text that a language model can understand. Each token is assigned its own unique integer after tokenization. (Gondal, 2024). For language models to incorporate contexts of ever larger scope, embeddings were implemented to effectively use unlabelled data in a self-supervised manner. In language modelling embeddings are divided into static- and dynamic embeddings. Static embeddings are pre-trained into the model and capture the meaning of the word based its use in the training dataset. Later deep neural network models were able to derive dynamic embeddings, which allowed for the vector representation of a token to change based on the context of the surrounding words (Patil & Gudivada, 2024).

2.2.3 Language modelling and transformer architecture

Language modelling is a key task to NLP and Language Understanding. Language modelling works by learning a probability distribution over sets of characters relating to a language. A language model (LM) can inherit grammatical structure and distil information from corpora (Jozefowicz et al., 2016). In practice, an LM learns to predict the next word given previous context.

Transformer models are a form of neural network architecture that are great at processing sequential data. Transformer models are most prominently used for natural language processing tasks for LLMs but are also good at other AI fields such as computer vision and speech recognition.

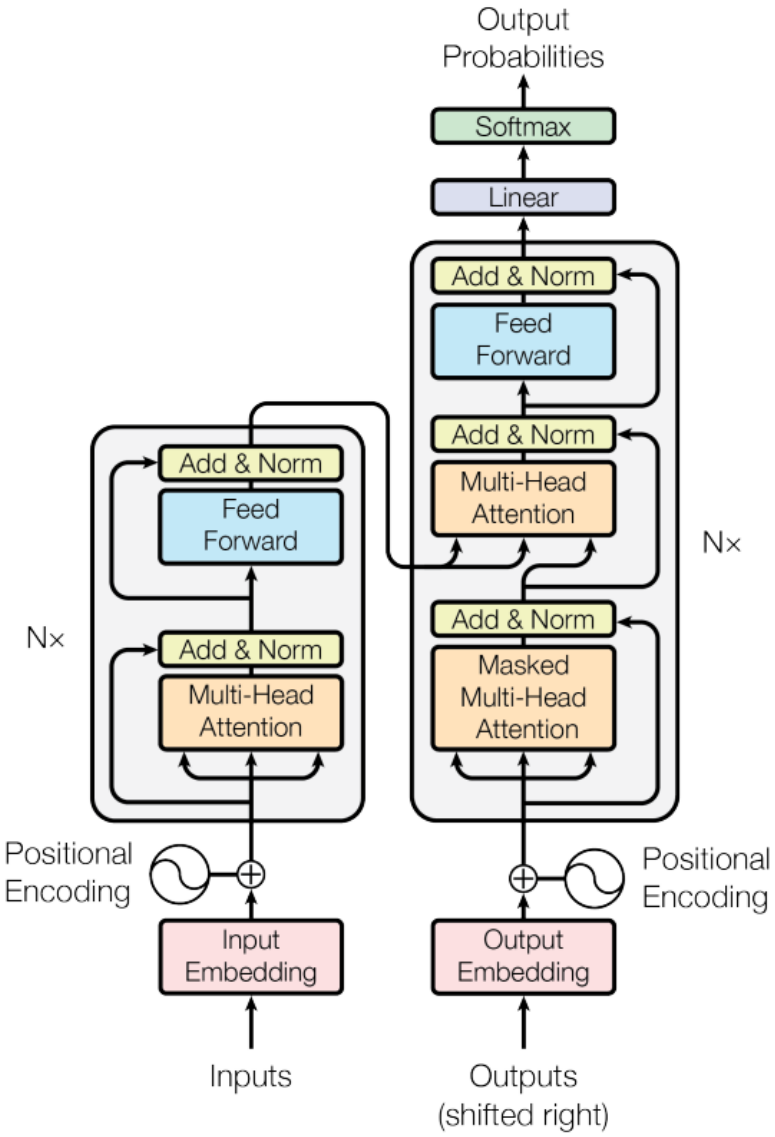

Figure 1. The transformer model architecture (Avaswani et al., 2017).

What makes transformer models powerful is their inherent self-attention mechanism as seen in Figure 1. The attention layer gives transformer models a much greater ability to discern the relationships and dependencies between different parts and words of the

imputed sentence compared to the preceding architectures such as recurrent neural networks and convolutional neural networks. (Stryker & Bergmann, 2025)

2.2.4 Foundation models

Foundational models are models trained on a wide range of unlabelled data that can be used for different tasks, in contrast to the previous task-specific AI models (Murphy, 2022). Foundational models are a central but incomplete part of AI, prominently done with natural language processing. Foundational models are enabled by scale and transfer learning. Transfer learning is what makes foundational models possible, its purpose is to use the comprehension gained from one task to apply it to another. Scale is what makes foundational models powerful (Bommasani et al., 2022). Massive training datasets are what makes modern LLMs possible.

2.3 The function of large language models

LLMs work by using deep learning with massive amounts of textual data. They are generally based on a transformer architecture, which are great at understanding sequential text data. LLMs are multilayered neural networks and each layer has parameters that can be fine-tuned in training. There is also an attention mechanism, which focuses on specific parts of the data sets (IMB, 2023).

The development of LLMs is divided into two main parts, pre- and post-training. The purpose of pre-training is to use massive amounts of data to train a foundational model. A foundational model allows easy fine-tuning for different use cases in the post-training phase.

In the training phase, LLMs train predicting the following word in a sentence using the context given by the earlier words. This is done by giving a probability score to the previous tokens, which are then transformed into embeddings which together numerically represent the context of the sentences and allows the LLM to predict the following words (IMB, 2023).

For LLMs to be accurate, they need to train on massive amounts of data, allowing them to learn grammar, semantics, and conceptual relationships. After training, LLMs can generate text by predicting the next word based on the user input and utilizing the patterns and information they have gained (IMB, 2023).

2.4 The importance of post-training improvement of large language models

LLMs have revolutionized NLP allowing for diverse applications across various domains. However, they remain constrained by their inherent limitations such as limited context lengths, tendencies to hallucinate meaning to make up information, suboptimal reasoning proficiency and inherited biases (Tie et al., 2025). Performance improvements of language models have mainly relied on the scaling of train-time compute using self-supervised pre-training (Kaplan et al., 2020; Hoffmann et al., 2022). Pre-training with massive amounts of data has set the foundation for LLMs, but research is increasingly shifting towards post-training techniques for further improvements (Kumar et al., 2025).

Metz et al. (2024) reveal that some of the largest companies in the LLM base such as OpenAI, Google and Anthropic have been struggling to meaningfully improve on their previous models. They claim that part of the reason for this is the difficulty of finding untapped sources of high-quality data. This emerging plateau in model performance has shifted the focus to the exploration of alternative enhancement strategies beyond raw scaling. According to Metz et al. (2024), rather than relying solely on ever-larger datasets, researchers are increasingly investigating methods that enable models to improve post-

training. One thing they highlight is OpenAI's o1 model, which spends more time computing the answer before giving an output, a process which OpenAI calls reasoning.

3 External tool usage and post-training enhancement of LLMs

The reasoning, ethical and domain specific performance of pre-trained LLM architectures remain limited necessitating a move into advanced post-training language models (Tie et al., 2025). For further improving the accuracy of LLMs, techniques such as retrieval-augmented generation (Gao et al., 2023) and external tool integration (Zhuang et al. 2023) should be considered. While this chapter focuses on the individual enhancement techniques, chapter 4 will give a more comprehensive overview of the different roles and integration strategies of post-training enhancement techniques.

3.1 Post-training techniques

While pre-training provides a broad lingual foundational model, post-training is an important step to refine the knowledge, reasoning capabilities, factual accuracy, alignment with user intent and ethical considerations of the model (Kumar et al., 2025). In short, the purpose of post-training is to specialize the model in a chosen direction.

Fine-tuning is a post-training technique with the purpose of specializing a model using curated data. Many different fine-tuning techniques exist for different purposes. Supervised fine-tuning (SFT) is used to refine a pretrained language model on a supervised dataset containing high-quality human-made examples. The purpose of SFT is ensuring the model attains to compliance with style and format guidelines (Kumar et al., 2025). Put simply, SFT teaches the model to mimic ideal responses.

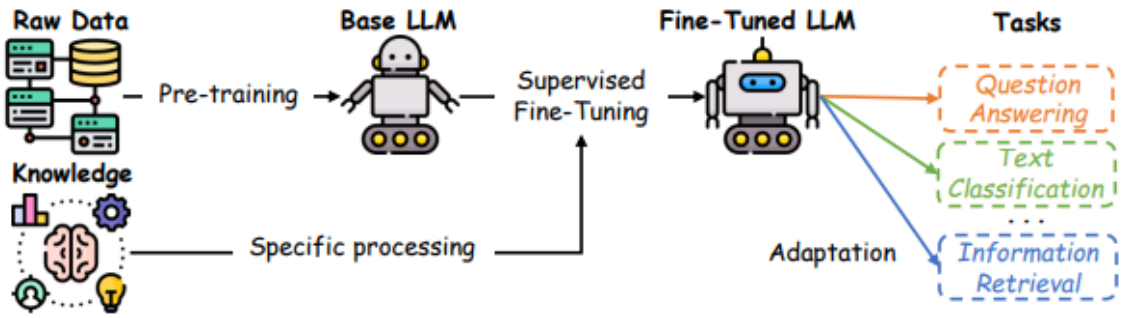

Figure 2. The supervised fine-tuning process (Tie et al., 2025).

As depicted in Figure 2, the SFT process begins with a pre-trained model. During fine-tuning, the model is aligned with the requirements of the given application. This is done by adjusting the parameters using task-specific annotated data (Tie et al., 2025). Fine-tuning forms the foundation for post-training, and it comes in many forms. Instruction fine-tuning, for example, guides models to follow user instructions accurately and helpfully. In contrast, dialogue fine-tuning uses chat transcripts to improve multi-turn conversational ability. Chain-of-thought (CoT) reasoning fine-tuning is used to teach models to produce reasoning traces by training on supervised reasoning annotations. This can improve both the explainability and accuracy of the model on complex tasks. (Kumar et al., 2025).

Reinforcement learning (RL) allows an agent to learn behaviour through trial-and-error interactions with a dynamic environment (Kaelbling, Littman & Moore, 1996). A common purpose of using reinforcement learning for LLMs is to align the model's behaviour in a desired direction based on feedback, but it can also be used to increase the reasoning capabilities of a model. For aligning the model, reinforcement learning with human feedback (RLHF) is an essential fine-tuning method. This approach works by using a reward model that explicitly takes human input, allowing the model to adapt more closely to human preferences (Tie et al., 2025). Unlike SFT, RLHF trains a model to generalize past specific examples to gain a nuanced understanding of the preferences of humans. For a more scalable and low-cost solution, reinforcement learning with AI feedback (RLAIF) is an alternative solution (Tie et al., 2025).

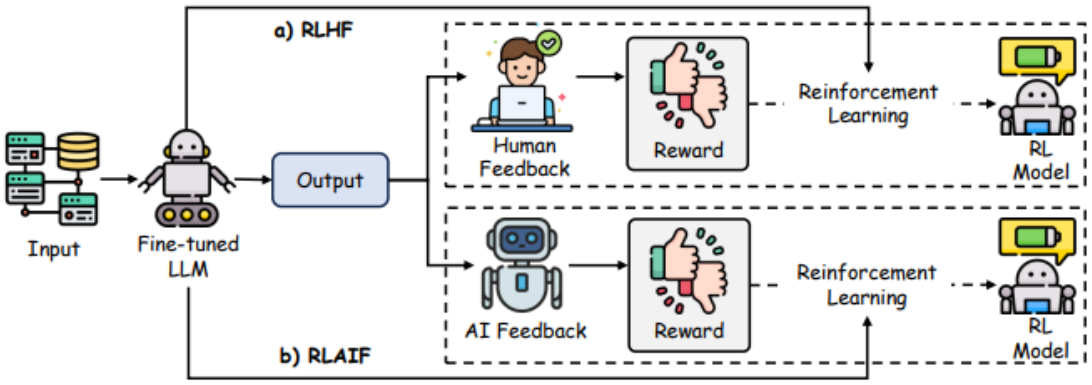

Figure 3. Comparison of RLHF and RLAIF for preference alignment (Tie et al., 2025).

Direct alignment from preferences (DAP) methods are a recent development for alternative methods in preference alignment (Guo et al., 2024). Direct preference optimization (DPO) was first presented by Rafailov et al. (2023), as a more computationally efficient and stable alternative to RLHF. DPO makes the reward optimization process simpler by linking the reward function to the optimal policy directly. Instead of training a reward model and fine-tuning a language model through reinforcement learning, DPO trains the LM directly with human preference data thus removing the need for fitting a reward model (Tie et al., 2025).

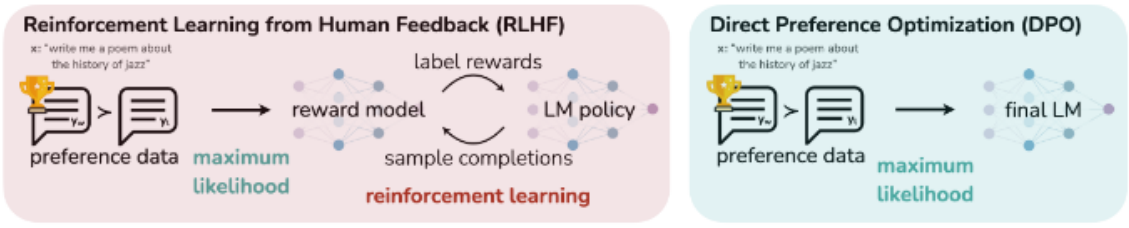

Figure 4. RLHF and DPO compared (Rafailov et al., 2025).

As seen in Figure 4, without explicit reinforcement learning or reward modelling DPO simplifies the process. For all the benefits of DAP methods, they tend to suffer from a lack of online feedback. The preference datasets for DAP methods are generally gathered pre training phase and the responses in the datasets generated by different LLMs. This

leads to the feedback of DAP methods being completely offline as the LLM policy in training is unable to get feedback for its own outputs The end result is a distribution between the policy being aligned, and the one that generated the dataset. This is in contrast to RLHF, where the reward model gives online feedback to the LLM policy from its outputs during reinforcement learning (Guo et al., 2024).

As the next step from RLHF and DAP methods, Guo et al. (2024) propose online AI feedback (OAIF). OAIF promises to include both the practicality of DAP methods and the online nature of RLHF. OAIF, specifically the alignment of the LLM policy, works by sampling two responses to a prompt from the current policy, obtaining online feedback by prompting an LLM to give preference annotations and using the feedback to update the model policy trough standard DAP losses (Guo et al., 2024). This approach is shown in Figure 5.

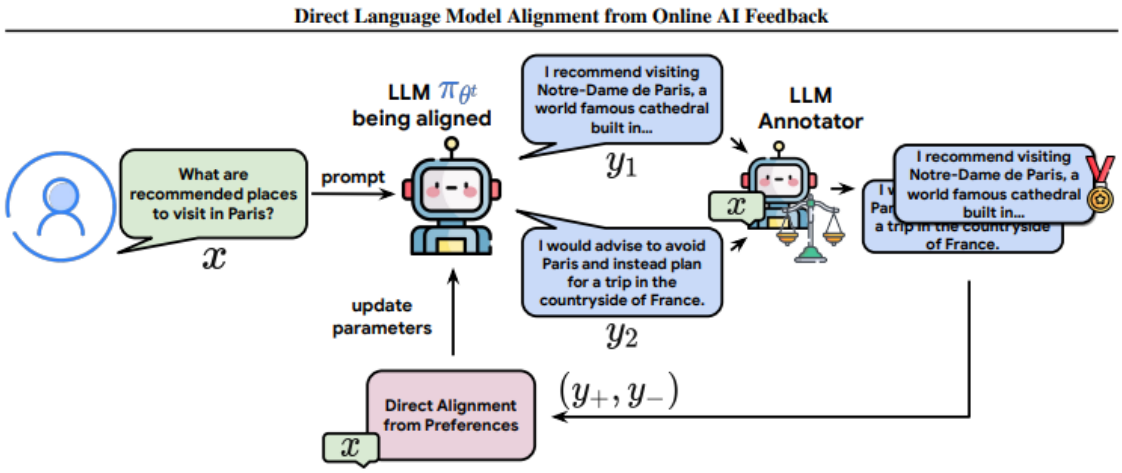

Figure 5. The process of OAIF (Guo et al., 2024).

Reinforcement learning for reasoning entails the use of a reward-based optimization to improve the model's chain-of-thought capabilities (Tie et al., 2025). In much the same way that humans spend more time thinking on complex problems, reinforcing a model to think tasks through step-by-step can help them with long-term reasoning tasks (Tie et al., 2025).

Test-time computation (TTC), or inference, is the process during which the model takes input data and uses its learned parameters to generate an output. Test-time scaling, as proposed by Snell et al. (2024), allows a model to allocate additional computational resources during the inference phase of the model's operation. This approach allows the model to process challenging prompts more thoroughly, similarly how humans spend more time thinking about complex problems to improve their answer. Best-of-N sampling is perhaps the simplest approach to scaling TTC (Snell et al., 2024). As seen in Figure 5, the premise of best-of-N is generating multiple outputs in ¨parallel¨ and selecting the one that scores highest with a learned verifier (Cobbe et al., 2021) or a reward model (Lightman et al., 2023). As seen in Figure 5, another option is to use sequential revisions, each generated in sequence and conditioned on previous attempts, or the combination of the two methods (Snell et al., 2024).

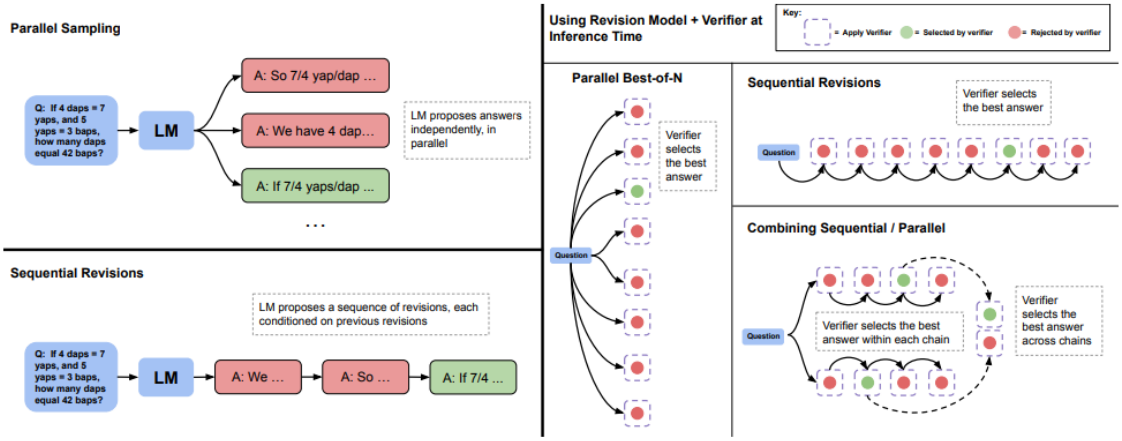

Figure 6. Best-of-N sampling and sequential revisions compared (Snell et all., 2024).

The viability of scaling test-time compute was recently demonstrated by OpenAI with their o1 model, the approach of which they describe as using large-scale RL (OpenAI, 2024; Muennighoff et al., 2025). Soon after o1´s release, DeepSeek R1 has been able to replicate o1-level performance similarly using RL with multiple training stages and millions of samples (DeepSeek-AI et al., 2025; Muennighoff et al., 2025). However, OpenAI´s and DeepSeek´s advancements in test time scaling lack transparency and limit research progress. In an effort to create a simple and open test-time scaling solution, Muennighoff

et al. (2025) propose forcing a maximum and/or minimum number of thinking tokens as a decoding-time intervention mechanism called budget forcing. Budget forcing can extend inference by suppressing the end-of-token delimiter and optionally injecting a token such as wait into the reasoning trace to help the model reflect on what it has generated. While budget forcing did improve model performance, the performance gains were found to flatten out when scaling further and it is limited by the model's context window (Muennighoff et al., 2025). Despite these limitations, test-time scaling has already been proven viable and further research will bring it to open models.

3.2 External tool and information integration

As LLMs have a propensity to hallucinate and their numerical reasoning capabilities remain limited, external tools and data can be used to enhance the capability of LLMs to answer questions (Zhuang et al., 2023).

Without an ability to retrieve external data, basic pre-trained LLMs tend to suffer from a lack of context when faced with request dealing with information that is not included in the training data, is highly specific or is up to date. For this reason, techniques like retrieval-augmented generation (RAG) have been created to allow LLMs to access external information. Retrieval-augmented generation allows language models to integrate external knowledge sources into the generation process. This addresses inherent limitations of LLMs, such as hallucinations, meaning the making up of information, outdated information and as the source of the information provided by the LLM is usually unclear, a lack of transparency in answers. As depicted in Figure 6, RAG works by retrieving relevant information from external sources before generating a response. This allows LLMs to access up-to-date and domain-specific information that might not be present in the training data. By combining the general knowledge of the model learned in the training phase with external up-to-date and domain specific information, RAG enhances the reliability and accuracy of the generated text (Gao et al., 2023). While RAG addresses the

problems of knowledge cutoff and suboptimal model accuracy, it remains limited to passive retrieval of information, meaning it a RAG based system cannot take further action than providing textual responses (Huo et al., 2025).

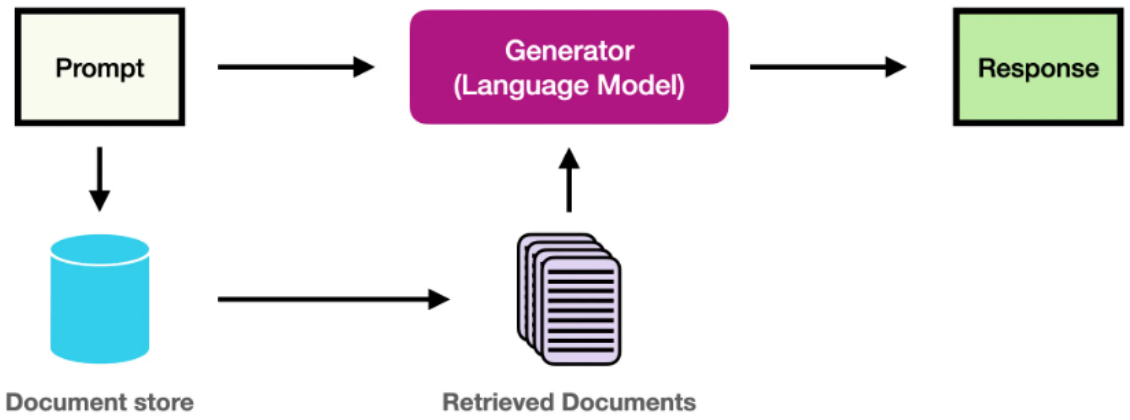

Figure 7. The retrieval-augmented generation process (Prompt Engineering Guide, n.d.).

Although RAG introduces dynamic context retrieval, its capabilities remain passive and limited to information fetching without the capacity for action or computation. To evolve past these limitations, external tool integration equips LLMs with the ability to interact with external programs through a variety of means. Zhuang et al. (2023) present their ToolQA benchmark, which they use to compare both standard and tool-augmented LLMs. The benchmark contains questions from multiple different contextual dimensions such as mathematical, scientific, and social, and they have integrated multiple tools into the tool augmented LLMs to help with these tasks. The tool-augmented LLMs had a significantly higher success rate in the benchmark as expected due to their access to external data but also tended to use incorrect data sources and tool calls, an issue which could potentially be addressed with fine-tuning.

AI agents capable autonomous of tool use and interaction with data sources have recently gained significant traction, further accelerated by the function calling ability of OpenAI allowing LLMs to call external APIs (Huo et al., 2025; OpenAI, 2023). The ability to retrieve real time data, perform computations and interact with external systems ex-

panded the capabilities of LLMs. The next step was a general-purpose protocol for standardized AI-tool interactions; model context protocol (MCP), introduced by anthropic late 2024 (Huo et al., 2025; Anthropic, 2024). MCP allows AI applications and external tools to communicate dynamically and enables AI agents to autonomously choose and operate tools based on the task at hand. This is in contrast to traditional implementations of wiring APIs manually for each service and tool that the AI application interacted with as seen in Figure 7. MCP also improves on the abilities of RAG by enabling models to interact with tools and external data actively, allowing for both retrieval and action in a unified workflow (Huo et al., 2025).

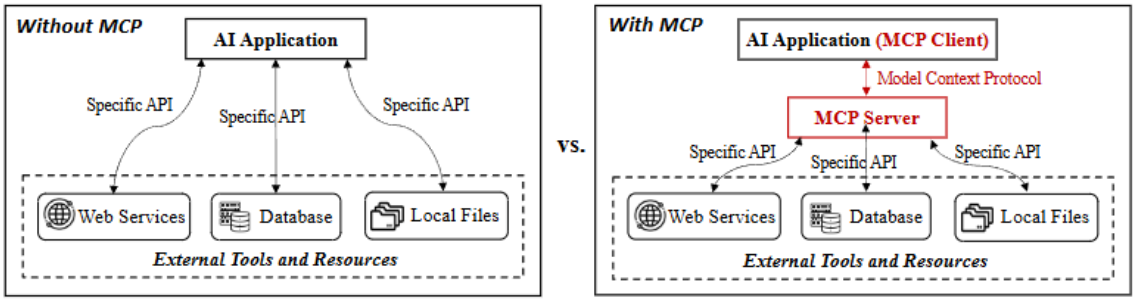

Figure 8. Tool use with and without MCP compared (Huo et al., 2025).

Alongside function calling increasing work is put into creating autonomous AI agents that can complete human tasks by navigating and using user interfaces (UIs). While a lot of research has focused on enabling AI agents to navigate and understand UI structures, there has been a gap in understanding the real-world impacts of their actions. By studying and modelling the impacts of UI operations, developers can better anticipate and mitigate potential issues that come from AI interactions with UIs (Zhang et al., 2024). In their study Zhang et al. (2024) concluded that current datasets do not adequately represent the complexities of UI action interactions, especially ones with major consequences, nor can top LLMs consistently understand the impacts of UI actions. While AI capable of navigating UI made for humans is a big leap for truly autonomous AI systems, function calls and sandboxes might be the safer option for now.

As the scale and costs grow, the enhancements in LLMs start showing diminishing results (Metz et al. 2024). This has driven toward alternative ways to improve model performance, such as self-refinement methods that give LLMs the ability to review and enhance their own outputs post training. Gou et al. (2023) demonstrate how integrating interactive tools within the LLM framework creates a feedback loop in which the model can identify and correct errors in its own responses. By integrating tool-based critiques, the LLM not only fixes inaccuracies but also refines its own generation process, leading to more reliable and context accurate outcomes. In a similar strategy Madaan et al. (2023) present an alternative approach, where the LLM systemically re-evaluates and revises its responses. This self-feedback loop allows the mode to enhance the clarity and accuracy of its output through each iteration. By setting its own initial output as the baseline for improvement, the method mitigates the issues that come from the inherent limitations of dataset training and compensates for the diminishing returns observed during pre-training.

4 Discussion

This chapter critically evaluates the post-training enhancement techniques and external tool integration approaches presented in chapter 3. The analysis focuses on identifying the strengths and weaknesses of each method, the issues that they solve and how they should be implemented.

4.1 Post-training techniques analysis

Supervised fine-tuning is useful for improving task specific performance by fine-tuning a base model on annotated data. This is essential for initial alignment and performance optimization. Reinforcement learning, especially RLHF, allows models to be safer and aligned with human preferences. However, RL requires carefully designed reward functions and is costly to implement. RLAIF is a low-cost alternative to RLHF, with ever increasing potential as LLMs evolve allowing for better AI feedback. DPO further simplifies alignment by training directly on preference data without an explicit reward model, improving computational efficiency and stability. Direct Language Model Alignment from Online AI Feedback extends these ideas by generating feedback for the models' own outputs in an online loop, combining both the benefits of RLHF and DAP methods. Test-time scaling boost performance by allocating more compute for more difficult prompts that need it. This is good for increasing efficiency, but it doesn't correct the inherent issues of LLMs. Retrieval-augmented generation pulls documents from datasets, helping reduce hallucination and improve factual grounding.

4.2 External tool and information integration

Connecting LLMs to external tools (calculators, APIs, databases) extends capabilities beyond text generation, enabling precise computation, real-time data access and actions in software environments (Zhuang et al., 2023). Ensuring robust tool use requires careful

fine-tuning on tool calling patterns and monitoring execution results. Standardized protocols such as model context protocol can streamline integration. While external tools greatly expand functionality and accuracy for specific tasks, they demand engineering effort to build reliable pipelines and guardrails against misuse or unexpected behaviour.

4.3 Integration strategies

The main insight from analysing these enhancement techniques is that they address different limitations of LLMs and are most effective when combined. A comprehensive AI system might include multiple techniques layered on top of each other as seen in Table 1.

Table 1. Enhancement techniques divided in layers based on their function.

Layers	Techniques	Function
Foundation	SFT	Provides task specific alignment
Alignment	RLHF, RLAIF, DPO, OAIF	Ensures safety and human preference alignment
Knowledge	RAG	Incorporates external information when relevant
Capability	Tool integration, MCP	Enables computation and action beyond text generation
Quality	Self-refinement	Provides ongoing output refinement
Efficiency	Test-time scaling	Allocates computational resources based on task complexity

An advanced system might use supervised fine-tuning for specializing the model to a task, align the model with RLHF, RLAIF, OAIF or DPO, use RAG to add context to the prompt if external data is needed, generate an answer and finally apply a self-refinement loop to verify the output. Frameworks such as MCP might also be integrated so that the model can better process and retrieve data trough tools. In this example pipeline each layer addresses a different weakness of LLMs. Fine-tuning aligns the model to the desired

purpose, the alignment techniques ensure the answer is safe and in the expected format, RAG fills the missing context, MCP allows tool calls and the self-refine loop refines the output.

4.4 Practical considerations and trade-offs

Implementing these techniques in production systems requires careful consideration of various trade-offs. While techniques like test-time scaling and self-refinement can improve output quality, they significantly increase computational costs and latency. Organizations must balance quality improvements against operational constraints. Integrating multiple enhancement techniques increases system complexity, potentially introducing new failure modes and making debugging more challenging. Many techniques require high-quality training data or external knowledge bases. The cost and effort of maintaining these resources must be considered for implementation decisions. External tool integration and autonomous capabilities introduce new security considerations. Proper sandboxing, access controls, and monitoring mechanisms are essential. As systems become more complex, evaluating their performance becomes more difficult. Traditional benchmarks may not capture the full range of capabilities and potential failure modes.

5 Conclusion

As large language models (LLMs) reach the limits of what can be achieved through pre-training, post-training techniques and external tool integration have emerged as vital strategies for continued advancement. This thesis has explored several methods for enhancing LLMs past their initial training, highlighting the importance of fine-tuning, reinforcement learning, test-time scaling and retrieval-augmented generation (RAG). Additionally, it examined how external tools and protocols like Model Context Protocol (MCP) can enhance LLMs with capabilities that go beyond static knowledge, enabling dynamic interaction with data and software environments.

Each of the explored techniques targets different weaknesses inherent in pre-trained LLMs. Fine-tuning increases task specific alignment, reinforcement learning ensures alignment with human preferences, RAG improves factual grounding and test-time scaling allows for more efficient resource management. External tools significantly extend the possibilities of LLMs by enabling real-time computation, information retrieval and interaction with the outside world. These approaches are most effective when used in combination, creating multi-layered systems that can compensate for individual limitations and reinforce each other's strengths. However, multiple challenges remain. The computational cost of combining several enhancement techniques raises questions about the practical scalability and cost effectiveness. Security and safety considerations raise in importance as models gain autonomy trough tool integration.

Large language models are a relatively new development in the field of NLP, and thus research on the post-training enhancement of LLMs has substantial room for development. Several open questions remain for future research. For one, research is required on how different enhancement techniques can be combined most effectively for specific applications. Research is also needed on finding the practical scalability limits of techniques such as test-time scaling and self-refinement. As LLM systems become more capable, especially with the introduction of API calling and agentic capabilities, how can it be ensured that they remain aligned and within safe boundaries? Lastly, re-search is

needed on how the computational cost of these enhancement techniques can be minimized while preserving their benefits. The fast-moving development of this field suggests that new technologies and approaches will continue to emerge, requiring ongoing research and development efforts to understand their implications and optimal application methods.

The path forward for LLM development lies not only in scaling up model sizes, but in developing smarter post-training systems that integrate alignment, self-refinement, test-time scaling, external information, external tools, and agency. By understanding and combining these techniques, researchers and developers can push the performance and reliability of LLMs to new heights making them more efficient, accurate, adaptable, and aligned with human goals.

References

Anthropic. (2024?). Introducing the Model Context Protocol. *Anthropic News*. Retrieved June 9, 2025, from https://www.anthropic.com/news/model-context-protocol

Bommasani, R., et al. (2021, August 16). On the Opportunities and Risks of Foundation Models. *arXiv*. Retrieved May 4, 2025, from https://arxiv.org/abs/2108.07258

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021, October 26). Training Verifiers to Solve Math Word Problems. *arXiv*. Retrieved June 12, 2025, from https://arxiv.org/abs/2110.14168

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023, December 18). Retrieval-Augmented Generation for Large Language Models: A Survey. *Shanghai Research Institute for Intelligent Autonomous Systems*. Retrieved May 7, 2025, from https://simg.baai.ac.cn/paperfile/25a43194-c74c-4cd3-b60f-0a1f27f8b8af.pdf

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., & Chen, W. (2023, May 19). CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. *arXiv*. Retrieved May 4, 2025, from https://arxiv.org/abs/2305.11738

Guo, S., Zhang, B., Liu, T., Liu, T., Khalman, M., Llinares, F., Ramé, A., Mesnard, T., Zhao, Y., Piot, B., et al. (2024, February 29). Direct language model alignment from online AI feedback. *arXiv*. Retrieved June 9, 2025, from https://arxiv.org/pdf/2402.04792

Holdsworth, J., & Scapicchio, M. (2024, June 17). What is deep learning? *IBM THINK*. Retrieved May 4, 2025, from https://www.ibm.com/think/topics/deep-learning

Hou, X., Zhao, Y., Wang, S., & Wang, H. (2025, March). Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. *arXiv*. Retrieved June 9, 2025, from https://arxiv.org/pdf/2503.23278

IBM. (2023, November 2). What are large language models (LLMs)? *IBM THINK*. Retrieved May 7, 2025, from https://www.ibm.com/think/topics/large-language-models

Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016, February 7). Exploring the limits of language modeling. *arXiv*. Retrieved May 7, 2025, from https://arxiv.org/abs/1602.02410

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996, May 1). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*. Retrieved May 7, 2025, from https://www.jair.org/index.php/jair/article/view/10166

Komeili, M., Shuster, K., & Weston, J. (2021, July 15). Internet-Augmented Dialogue Generation. *arXiv*. Retrieved May 7, 2025, from https://arxiv.org/abs/2107.07566

Kumar, K., Ashraf, T., Thawakar, O., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H., Torr, P. H. S., Khan, F. S., & Khan, S. (2025, March 24). LLM Post-Training: A Deep Dive into Reasoning Large Language Models. *arXiv*. Retrieved May 7, 2025, from https://arxiv.org/abs/2502.21321

LeCun, Y., Bengio, Y., & Hinton, G. (2015, May 27). Deep learning. *Nature*. Retrieved May 4, 2025, from https://www.nature.com/articles/nature14539

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., & Cobbe, K. (2023, May 31). Let's Verify Step by Step. *arXiv*. Retrieved June 12, 2025, from https://arxiv.org/abs/2305.20050

Madaan, A., et al. (2023, March 30). Self-Refine: Iterative Refinement with Self-Feedback. *arXiv*. Retrieved May 4, 2025, from https://arxiv.org/abs/2303.17651

Metz, R., Ghaffary, S., Bass, D., & Love, J. (2024, November 13). OpenAI, Google and Anthropic Struggle to Build More Advanced AI. *Bloomberg Law*. Retrieved May 4, 2025, from https://www.bloomberg.com/news/articles/2024-11-13/openai-google-and-anthropic-are-struggling-to-build-more-advanced-ai

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candes, E., & Hashimoto, T. (2025, March 1). s1: Simple test-time scaling. *arXiv*. Retrieved June 9, 2025, from https://arxiv.org/pdf/2501.19393

Murphy, M. (2022, May 9). What are foundation models? *IBM Research Blog*. Retrieved May 7, 2025, from https://research.ibm.com/blog/what-are-foundation-models

OpenAI. (2024, September?). Learning to reason with LLMs. *OpenAI*. Retrieved June 9, 2025, from https://openai.com/index/learning-to-reason-with-llms/

OpenAI. (n.d.). Function calling. *OpenAI Developer Documentation*. Retrieved June 9, 2025, from https://platform.openai.com/docs/guides/function-calling?api-mode=responses

Prompting Guide. (2025, May?). Retrieval Augmented Generation (RAG). *Prompt Engineering Guide*. Retrieved June 9, 2025, from https://www.promptingguide.ai/research/rag

Raschka, S. (2024, September 15). dpo-from-scratch.ipynb. *GitHub*. Retrieved May 7, 2025, from https://github.com/rasbt/LLMs-from-scratch/blob/main/ch07/04_preference-tuning-with-dpo/dpo-from-scratch.ipynb

Snell, C., Lee, J., Xu, K., & Kumar, A. (2024, August 6). Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *arXiv*. Retrieved May 7, 2025, from https://arxiv.org/abs/2408.03314

Stryker, C., & Bergmann, D. (2025, March 28). What is a transformer model? *IBM THINK*. Retrieved May 7, 2025, from https://www.ibm.com/think/topics/transformer-model

Stryker, C., & Holdsworth, J. (2024, August 11). What is NLP (natural language processing)? *IBM THINK*. Retrieved May 4, 2025, from https://www.ibm.com/think/topics/natural-language-processing

Zhang, Z. J., Schoop, E., Nichols, J., Mahajan, A., & Swearngin, A. (2025, March 22). From Interaction to Impact: Towards Safer AI Agents Through Understanding and Evaluating Mobile UI Operation Impacts. *arXiv*. Retrieved May 7, 2025, from https://arxiv.org/abs/2410.09006

Zhuang, Y., Yu, Y., Wang, K., Sun, H., & Zhang, C. (2023). "ToolQA: A Dataset for LLM Question Answering with External Tools." ("ToolQA | Proceedings of the 37th International Conference on Neural ...") *NeurIPS Proceedings*. Retrieved May 7, 2025, from https://proceedings.neurips.cc/paper_files/paper/2023/hash/9cb2a7495900f8b602cb10159246a016-Abstract-Datasets_and_Benchmarks.html