ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

Using the Right Tool: Prompt Engineering vs. Model Tuning

Resmi Vijayan[1]; Sunish Vengathattil[2]

[1]Software Engineer, Comcast, Philadelphia PA, United States
[2]Sr. Director, Software Engineering, Clarivate Analytics, Philadelphia PA, United States

Publication Date: 2025/05/16

Abstract: The growing impact of AI on industries and human-machine relationships creates an essential question about the actual controller of AI behavioral patterns. Discussing AI control structures between prompt engineering and model tuning defines its core framework. Prompt engineering uses purposeful inputs to modify large language model results without changing the core model structure so developers and non-technical users can easily employ this approach. Model tuning requires lengthy adjustments of basic model components using fine-tuning or instruction-tuning methods and reinforcement learning. Still, it allows for strong control as a drawback of its advanced requirements and resource demands. This research analyzes the technical base frameworks, practical applications, and benefits and disadvantages of both methods which also addresses manipulative control of AI systems and general system reliability as well as ethical standards and system accessibility features. We examine the effectiveness of these approaches in practical applications through real-life situations to determine which method yields better behavioral control for AI systems. We also explore the current shifts in open-source and proprietary platforms between these control methods. The ability to control AI functions best exists on a continuum that distributes power according to specified objectives, conditions, and system capabilities. The progression of artificial intelligence technology requires us to transform our grasp of control systems, collaborative protocols and responsibility duties in AI steering. The article functions as a critical tool that helps developers, businesses, and policymakers redesign their future AI development paths.

Keywords: *Prompt Engineering; Model Tunin;, AI Control; Large Language Models; Artificial Intelligence.*

How to Cite: Resmi Vijayan; Sunish Vengathattil (2025). Using the Right Tool: Prompt Engineering vs. Model Tuning. *International Journal of Innovative Science and Research Technology,* 10(5), 274-284. https://doi.org/10.38124/ijisrt/25may255

I. INTRODUCTION

ChatGPT, beside Claude and Gemini constitute significant advancements in natural language processing because they generate human-like responses which cover tasks from tutoring through summarization to code generation to creative writing. The models operate through transformer architectures which train on massive text databases to produce coherent output based on word and concept statistical connections [3]. An LLM gains generalization power when it finishes training because it can answer endless prompts without requiring retuning for any task.

The vast flexibility of these systems leads to an essential problem of guiding their operations for consistent responsible outputs. The practice and scholarly research about LLM control presents two main methods: prompt engineering and model tuning.

Through prompt engineering practitioners develop special input designs which draw improved responses or precise results or aligned outcomes. The technique functions at the language level outside the model while drawing from training distribution content and newly discovered abilities. Model tuning describes the weight modification method within models through techniques such as fine-tuning and reinforcement learning with human feedback (RLHF) and domain-specific data embedding [11]. AI installations into critical operations and settings such as healthcare and law advising and education and military demand immediate answers about system control, accountability and data transparency. The pressures on developers, business leaders, and policymakers include maintaining model adaptability as well as protecting safety and preserving interpretability. At this point models must be tuned strategically because choosing between prompt engineering and model tuning represents a core philosophical dilemma.

Both camps debate human-AI interaction because users aim to understand their model control capabilities and effective control methods. Prompt engineering provides broad access to influence, yet it stands in contrast to model tuning, a system-level modification procedure. Nonetheless both strategies face implementation challenges.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

This article assesses AI control between prompt engineering and model tuning and examines the true manipulators of AI conduct. The article evaluates both expertise, presents the benefits and limitations of individual strategies, and suggests mixed approaches to achieving trustworthy AI control in real-life applications

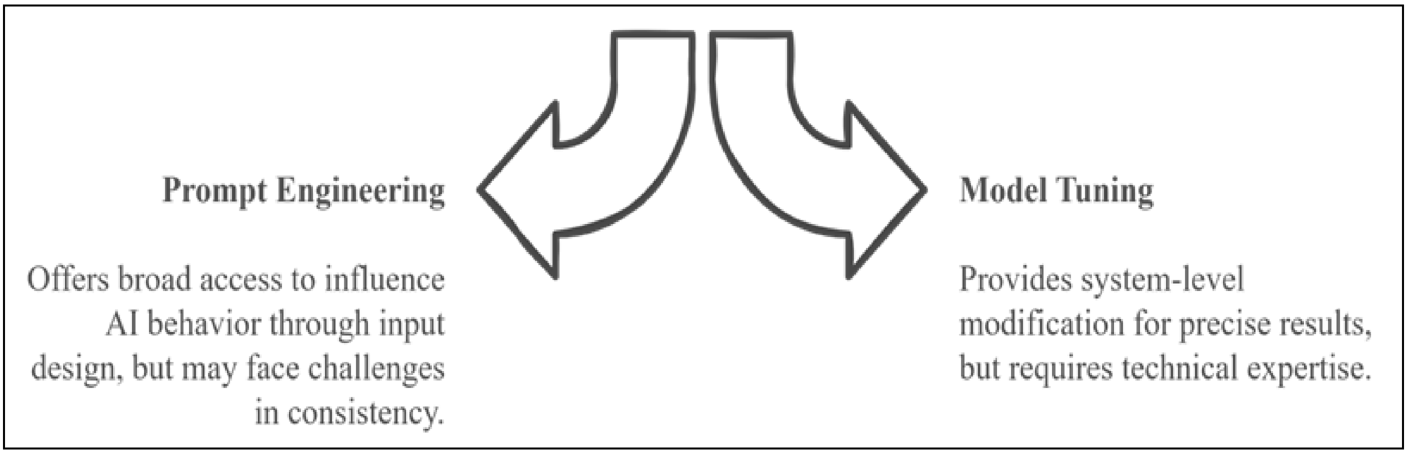

Fig 1 How to Control AI Behavior for Responsible Outputs?

II. UNDERSTANDING PROMPT ENGINEERING

Large language models need strategic input from prompt engineering to generate desired outputs. Pre-trained models reveal their latent capabilities through prompt engineering since this method alters how tasks are introduced during run time [6][8]. The LLM deployment for real-world applications across educational and healthcare domains together with law and creative writing has become possible through the emergence of this effective technique.

> *Prompt Engineering: More Than Asking Questions*

The scope of prompt engineering exceeds the practice of formulating well-worded questions. The process requires designers to create input passages which activate model neural networks and leverage their statistical response patterns. The core concept behind prompt engineering is that LLMs remain highly dependent on contextual framing, enabling users to obtain better and more creative output results [4]. Asking a question through a particular role statement ("You are a financial advisor") along with a response format instruction ("Respond in bullet points") substantially influences both the tone of the response and its relevance.

The method depends on knowledge about how LLMs handle and rank their tokens within text. The prediction system uses past patterns to anticipate the following words but small changes in verbalization lead to unpredictable output results. The method of prompt engineering serves as language-based programming which derives benefits from statistical information in the model while leaving its architectural structure and parameters untouched [9].

> *Prompt Formats: Zero-Shot, Few-Shot, Chain-of-Thought*

The following three main paradigms have developed within prompt development:

- Using zero-shot prompting the model receives the primary task form or question and depends entirely on its stored general information to generate outputs. As a basic prompt method, it remains easy to use yet produces unpredictable and erratic results when solving difficult assignments.
- Inputting example sets with their labels into the prompt constitutes Few-shot prompting. The incorporated examples serve in this context to guide the model toward following writing patterns and voice types and logical reasoning methods. The technique finds value in performing tasks related to classification alongside summarization and translation processes.
- During Chain-of-thought (CoT) prompting the model requires explicit step-by-step thought sequences which it must provide before generating the final answer. The method produces better outcomes for mathematical calculations along with abstract logical thinking and complicated problem sequences [18]

The selection of several prompting formats depends on the complexity level of tasks alongside the desired model output precision.

> *Applications in Practice*

The application of prompt engineering has spread quickly across different industries because of its straightforward setup process coupled with its operating system-independent structure.

- Bots that provide customer service now operate through adaptable prompt templates which customize responses as people make new inquiries.
- Using prompt-based systems by legal advisors helps extract vital clauses evaluate compliance needs and create draft content which reduces time-consuming manual tasks.
- Online education platforms use customized prompts as an engine to produce student materials along with quizzes as well as personalized tutorial content that suits different education levels [2][14].
- The application of LLMs in healthcare and biomedical research includes clinical note interpretation as well as patient action recommendations and medical literature digesting [23].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

The implemented applications demonstrate that prompt engineering functions as a versatile interface for building scalable, intelligent systems using low-resource infrastructure.

➢ *Strengths: Speed, Simplicity, and Adaptability*

The benefits of prompt engineering stem from its accessibility and flexibility:

- Anyone holding an LLM subscription can immediately conduct experiments through prompt engineering since model retraining is not required.
- Prompt testing and frequent modification happens in real time through which organizations can complete fast agile experiments and make rolling releases.
- The same prompts from LLMs (such as GPT-4 and Claude and PaLM) become usable after making a few simple adaptations so organizations can integrate them easily.

Startups and educational institutions together with research organizations find prompt engineering suitable because it provides high-performance outputs without requiring dedicated infrastructure or talents.

➢ *Limitations: Surface Control, Unpredictability, and Scale*

The limitations of prompt engineering emerge from its inability to achieve complete model control because the system operates without deep interventions.

- The lack of inspection into model layers combined with access limitations to its parameters prevents prompt engineers from delivering stable performance while eliminating biases from the system.
- The output variation from specific prompts emerges because of random sampling or model-specific behavior which affects the reliability of repeatable results [17].
- Maintaining quality output across various contexts becomes increasingly difficult when prompts become more complex or are used in critical settings.

System complexity leads to the complication of prompt design processes. Long prompts which include multiple levels of conditional structures alongside embedded examples make it easier to reach token limits or create reduced performance issues.

Table 1 Comparative Overview of Prompt Engineering Techniques

Technique	Description	Best Use Cases	Pros	Cons
Zero-Shot Prompting	Task is given without examples; relies solely on model's pretraining.	Simple Q&A, general advice, summarization	Easiest to implement; fast setup	May produce vague or inaccurate outputs
Few-Shot Prompting	Includes 2–5 examples in the prompt to guide response patterns.	Classification, translation, sentiment analysis	Improves output quality and format consistency	Limited by token space; sensitive to example choice
Chain-of-Thought (CoT)	Encourages step-by-step reasoning before producing the final output.	Math problems, logic puzzles, decision-making tasks	Boosts reasoning depth and task accuracy	Prompts get long; may increase latency and token usage
Role-Based Prompting	Assigns a persona or role to the model (e.g., "You are a legal expert").	Legal/medical advice, creative writing, customer service	Provides tone control and domain alignment	Can introduce hallucinations if role lacks factual context
Instruction Prompting	Provides clear task instructions (e.g., "List three pros and cons of X").	Structured output, educational content, list generation	Clear guidance improves reliability	Overly rigid prompts may reduce creative flexibility

III. EXPLORING MODEL TUNING

With large language models (LLMs) being more deeply embedded in domain-specific workflows and real-world systems, fine-grained control over their behavior is required. Prompt engineering offers a high-level way to control the outputs, but model tuning is a lower-level change to how a model 'thinks' rather than just how it is told. In this section, we explore the main categories of model tuning, discuss examples of producing such tuning, and explore the strengths and limitations of model tuning within the context of more general problems of AI alignment and deployment.

➢ *Levels of Tuning: A Spectrum of Control*

There are many different tradecrafts (of various complexities and scales of customizations) for model tuning.

- *Fine-Tuning: Rewriting the Neural Memory*

Fine-tuning refers to retraining a pre-trained language model (often large-scale, such as GPT, BERT, or T5) on a task-specific corpus. In this process, the internal weights of the model are updated as a part of all or some of the linguistic patterns, semantic priorities, and domain specific knowledge stored in the fine-tuning dataset [6][10]. Especially useful in such applications, clinical diagnosis assistants or compliance oriented legal tools for example that demand very high levels of specificity.

Fine tuning is highly effective where there is a big difference between the environment's vocabulary, syntax or interpretive nuances and general-purpose language. For example, legal case summary or biomedical research abstract has terms and logic that generalist models cannot understand without adaptation [15].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

- *Instruction Tuning: Models to Follow Human Intent*

Instruction tuning teaches a model to generalize across an extensive array of tasks by providing natural language instructions and desired outputs. Instead of training for each task, the model can interpret new instructions well. InstructGPT proved that this method both improves accuracy on previously unseen tasks along with the user satisfaction and trust of the responses [16].

Because of this user can use plain language [2], this technique works particularly well with general purpose assistants, educational tutors, and productivity applications. Instruction tuning is an effective communication bridge between the level of specificity wanted by the algorithm of the computer and the level of intuition needed by the user.

- *Reinforcement Learning from Human Feedback (RLHF): Behavioral Alignment at Scale*

Using RLHF goes beyond static datasets and trains a reward model from human preferences or judgements. Through reinforcement Learning, this reward model leads the language model to convey qualitatively good outputs, such as helpful, honest, or ethically qualified [23]. RLHF is employed in the refinement of ChatGPT 3.5 and ChatGPT 4 and has become a standard tool in the efforts to align LLMs with human values.

Though complex, RLHF combats many of the problems of supervised fine tuning by including the dynamic, true world user expectancies, which is key for use cases and requirements such as conversational AI in mental wellness, smart choice making systems, and decision-making tools [13].

➢ *Tuning on Practice: Customization at Scale*

Recently, many Organizations have been using Tuning Frameworks to adapt the foundational model for niche Applications.

- Custom GPTs created by OpenAI allows users to embed domain specific datasets and rules into their own Private GPT (P-GPT) which makes a personalized assistant for customer services, research or use in workflow automation.
- Enterprise Deployments: proprietary data is fine-tuned on LLMs to fuel internal search business tools, Fraud detection systems, highly personalized e commerce experiences [20].
- Its applications include Healthcare and security: Tuned Models explain complex input data, generate scenario-based advice and mark anomalies [19].

Model tuning reveals its versatility in these practical cases for systems where it must be as precise as possible and, by and large, context-aware and trustworthy.

➢ *Benefits of Model Tuning*

- With model tuning, developers can make model customization high fidelity and include domain specific language, logic, constraints that can be embedded directly into the architecture of the model, to produce outputs which are closer to nuances of specialized tasks [22].
- Compared to humans, it also has a more predictable and consistent output behavior which is especially critical when the system is used in high stakes areas such as in healthcare, finance, and legal decision making where the variability of response can be associated with large risks.
- Such models are better suited for multilingual and multimodal tasks when trained over diverse datasets and yield better results as they have been tuned to learn features on a broad spectrum of applications in a cross-disciplinary manner.
- Model behavior is aligned with organizational values and operational goals to reinforce ethically, communicatively or business-oriented priorities that the model is meant to implement at the business level—for instance, patient empathy in medical chatbots or compliance awareness in legal assistant.

➢ *Limitations and Challenges*

However, model tuning has significant drawbacks:

- Large Models need GPU Clusters, a long time and proprietary APIs (despite being open sourced for much time), which brings huge computational and financial costs, and most of them are inaccessible for small organizations [1].
- Tuning requires ML expertise as well as knowledge in design of NLP pipelines and data annotation. Propagations in errors in training data or hyperparameters can be displayed in the model's behavior.
- Fine Tuned Models, On Data Sensitivity and Bias Amplification: Just as the fine tuning process creates new sensitive data, we would do well to remember that fine-tuned models may replicate or amplify bias in the data they are fine-tuned on [17].
- Maintenance and Versioning: Modeling in TensorFlow Hub imposes an additional burden of maintenance: each tuned model becomes a custom artefact which must be updated in parallel with base models as they evolve and with new ethically related considerations.

Model tuning represents a highly effective methodology for changing the cognitive framework of the AI systems, and, consequently, to provide unprecedented degree of task specialization. Though this comes at a high cost, the payoff comes in the form of models that are conscious of the context, emotionally aligned with the user and behaviorally consistent across domains. Whereas prompt engineering applies to the surface level, tuning is applied directly to the substrate its intelligence is made of. It is essential for critical missions or where there are extremely strict requirements.

IV. KEY DIFFERENCES BETWEEN PROMPT ENGINEERING AND MODEL TUNING

At the basest level, it's important to understand the difference between prompt engineering and model tuning when interacting with and fine-tuning artificial intelligence systems (like large language models (LLMs)) in the first place. Table 2 shows a comparative view of key differences

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

between Prompt Engineering and Model Tuning. There is, however, a fundamental difference between both strategies in control depth, technical demands, scalability and of course context of use.

- *Superficial Input vs. Internal Optimization*

In the context of prompt engineering, the input is built or fine-tuned to ensure that it prompts the desired response from the pre-trained model. By its nature, it is a surface-level control mechanism that works over the model's existing parameters and latent representations. As Henrickson and Meroño Peñuela (2023) argue, prompts as more "interpretive cues"—interpretative help, as it were—than algorithmic instructions. On the other hand, model tuning, i.e., fine-tuning and reinforcement learning with human feedback (RLHF), alters the model's internal weights and the architecture to improve performance on a domain-specific task [1]. This draws greater influence but requires much more technical access and infrastructure.

- *Resource Requirements and Cost Implications*

As such, given the low barrier to entry, only needing minimal computational power and no training data or model internals [14], prompt engineering is a feasible option for the amateur who wants to leverage such engines for content generation. This is an ideal solution for rapid deployment in non-technical user environments. On the contrary, model tuning, for example, requires extensive resources in large datasets, high performance computing environments, and the need for ML experts [13]. However, the cost is higher overall in terms of efficiency and control.

- *Scalability and Consistency*

Agility and fast iteration are what prompt systems give us. However, these usually cannot scale well because the individual performance of prompt systems is inconsistent when applied to different inputs [4]. Prompt sensitivity is the problem of slight changes in phrasing being able to yield significantly different outputs. While model tuning is ahead of time, it leads to low-performance variability at scale, especially on complex or repetitive enterprise-type tasks [19].

- *Generalization vs. Specialization*

For broad, application to a generalizable task, prompt engineering is advantageous. In contrast, model tuning is more specialized to narrow areas of specialization such as legal advice, biomedicine, the industrial automation [15][16].

- *Use-Case Alignment: Startups vs. Industry Giants*

Because of its low cost and fast deployment, startups and educational platforms use prompt engineering [9]. In contrast, model tuning is deployed by large corporations, government institutions, and high stakes sectors to get the complete control of the performance and audit compliance [10].

Table 2 Comparative Table Showing Key Differences between Prompt Engineering and Model Tuning

Aspect	Prompt Engineering	Model Tuning
Control Level	Surface-level input manipulation	Foundational changes to model internals
Technical Barrier	Low; accessible to non-experts	High; requires deep ML expertise and access to model architecture
Resource Intensity	Minimal computing resources needed	High computational and dataset requirements
Scalability	Quick to iterate but hard to scale consistently	Highly scalable after tuning phase
Reliability	Prone to prompt sensitivity and unpredictability	More stable once tuned for a domain
Specialization	General-purpose flexibility	High domain-specific precision
Generalization	Performs well across varied tasks	May underperform outside of its fine-tuned task or domain

V. REAL WORLD CASES STUDIES

The work of AI professionals with prompts and model adjustments has increased fast as they test these methods across multiple fields and execution areas. Real-life experiments reveal distinctions between prompt engineering and model tuning through their outcomes. The following examples from OpenAI DeepMind and Meta demonstrate varied ways to use prompting and model tuning in practice.

- *Case 1: GPT-4 Prompt Engineering Playground*

OpenAI uses GPT-4 Playground as an easy-to-understand platform to teach prompt engineering to users. The Playground lets people with and without technical skills adjust their prompts quickly through its simple input tools so they can improve AI outputs immediately. The system provides options to change environmental and output functions preceding the model for better response control [24].

Through prompt engineering the GPT-4 model shows its ability to operate effectively across multiple subject areas. People can develop impressive chatbots and automatic content creation tools or teaching materials through the model interface without adjusting its technical parameters. The easy setup disadvantages GPT-4 since it produces outputs that need careful attention. When changes are made to the prompt design, GPT-4 shows strong sensitivity in its production, according to [12]. The system's quick development process makes finding ideal solutions hard to control.

GPT-4 prompt engineering speed makes it an appealing feature for users. The system lets developers change test options instantly and test multiple scenarios plus settings within a few minutes. This modelling cycle works best when time-sensitive decisions must be made like in personal customer service or live marketing. By adjusting model input instead of the structure, the system generates unpredictable results. Keeping outputs consistent and of good quality throughout all needed scales still presents difficulties that researchers from Henrickson & Meroño-Peñuela (2023) continue to address. Prompt engineering helps with many different projects, but it does not let you inspect the inner

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

workings of the model. DIY approaches become hard to use in industries like medicine where specific content accuracy needs to meet strict guidelines.

➢ *Case 2: Google DeepMind's Med-PaLM*

DeepMind Med-PaLM sets a precedent by training an existing model to handle sensitive healthcare tasks effectively. Med-PaLM gained its expertise through being trained on medical data and doctor-written answers, enabling it to match or even beat human doctors during medical certification exams. Medical experts at DeepMind carefully trained and refined their model with high-quality medical sources while GPT-4 enables users to create specific prompts. Due to its speciality Med-PaLM prevents frequent AI issues like making things up and understanding technical medical terms. DeepMind enhanced the model's accuracy by adding medical data and training it with medical experts' resources, making Med-PaLM better suited for medical diagnostic and clinical assistance.

Using fine-tuning creates several significant problems for users. You need much labelled medical data and specialized knowledge from healthcare professionals. The method needs significant resources so organizations or researchers with small budgets or restricted resources might be unable to implement it. Customizing the model with updated medical information takes time and money. When fine-tuning occurs excessive association with training samples might decrease model usefulness beyond specific medical scenarios [15].

Med-PaLM tuning produces more reliable results than prompt engineering techniques. This method enables exact management of the model outputs which is useful when safety and accuracy are crucial to an industry. The method requires less speed and flexibility than prompt engineering despite delivering better control and dependability.

➢ *Case 3: Meta's LLaMA – Open-Source Access*

Instead of adding its tuning method Meta opts to empower the development community to fine-tune LLaMA products. Meta grants AI researchers worldwide access to its pre-trained model weights to support their experiments with dataset tuning on different projects. Meta promotes community optimization by using open-source techniques which allows the platform to receive unique model training guidance from various sources.

LLaMA's design lets researchers tailor the model to serve specific areas of business or language requirements which helps spread AI technology across multiple fields. The community-led fine-tuning process helps LLaMA models adjust to various regions, language varieties and professional domains which extends their impact across regular applications and remote communities [16]. Many users can work together to advance tuning methods while they search for effective ways to improve the model performance.

The community-based tuning approach makes it easier to reduce biased outcomes. A wide range of contributors meets regularly to find and fix existing biases in LLaMA's initial training model. The LLaMA community successfully develops text processing tools with less prejudice while making them sensitive to various cultures [17]. The level of community involvement brings unpredictable variations in the output quality. When several contributors adjust the model without oversight, it leads to inconsistent results in its different versions.

➢ *Comparative Outcomes and Practitioner Insights*

Study results prove that engineers use prompt engineering alongside model training as separate yet helpful actions. Each method delivers unique benefits, but it is accompanied by specific problems that depend on the project's needs.

- Med-PaLM produces the best reliable outputs especially when handling medical reasoning tasks. Through GPT-4 prompt engineering users can achieve diverse outcomes but the system might generate inconsistent results at times. The LLaMA models effectively work across different subject areas but produce less reliable results each time.
- Fine-tuning Med-PaLM and LLaMA models offers more formal ways to manage bias as it allows select teams to process different datasets while community developers help to eliminate bias in these models. Controlling bias through prompt engineering depends heavily on proper prompt construction instead of system modifications. The systematic removal of bias remains hard to achieve when our data collection process lacks control or comes from different sources.
- Through OpenAI's Playground and Meta's open-source strategy users across different backgrounds gain easier access to work with AI models and design prompts. Anyone can use prompt engineering tools to test AI even without technical experience thanks to their accessibility and LLaMA opens its model source to everyone for customization. The training process used for Med-PaLM remains challenging because it requires specific resources and medical datasets.

When it comes to AI applications that require precise outcomes Med-PaLM's model tuning produces better and reliable results faster than prompt engineering. Experts predict AI developers will use both prompt engineering and model tuning techniques together because they complement each other by promoting fast testing with customization and precise domain-specific results [9][21].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

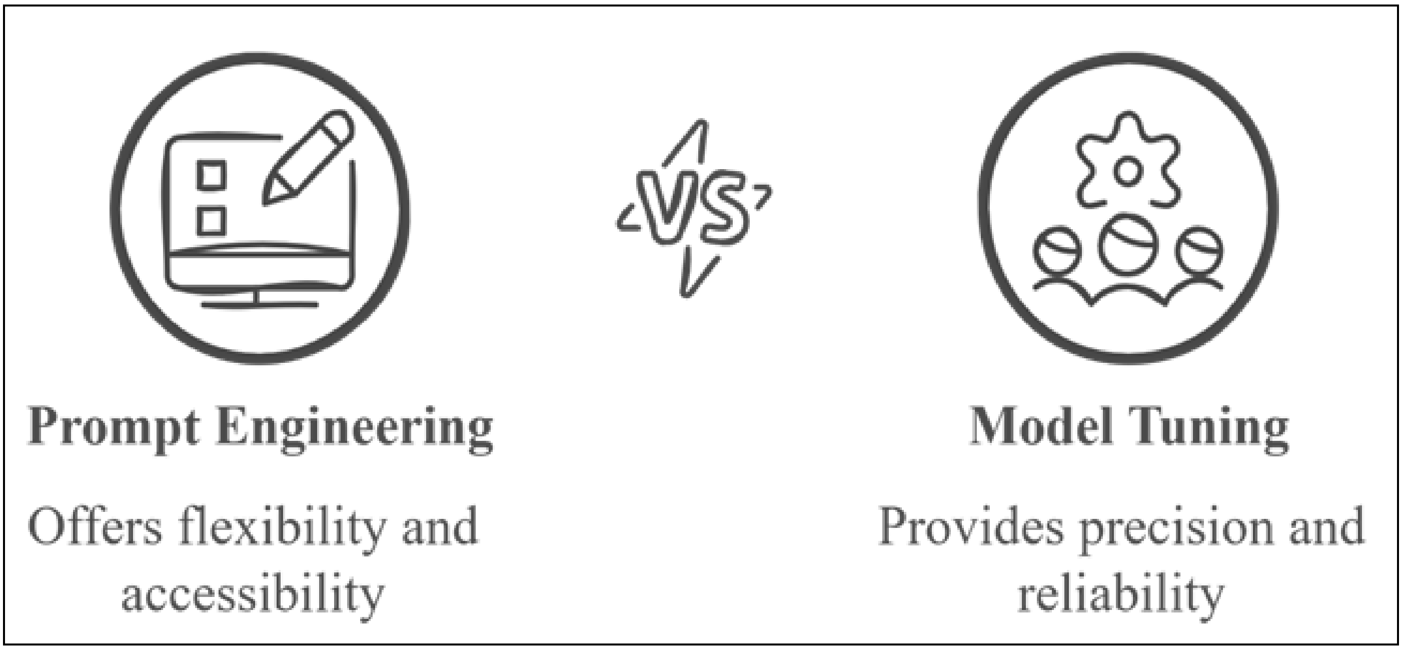

Fig 2 Choosing AI techniques for specific project needs

- *Case 4: A Combined Strategy*

Typically, the avenues of prompt engineering and model tuning have been looked at as separate strategies. However, a gathering agreement between AI experts appears that the mix of these two procedures would improve effectiveness and presentation [8][21]. In this thesis, prompt engineering is sequentially applied, followed by model tuning, which leverages the advantages of both approaches: to first use flexible and low-cost iteration to determine performance gaps and then data-driven homing in through fine tuning to stay within the limits of the model. One documented experimental scenario was developing a sentiment analysis system using a general-purpose large language model (LLM). The first utilization of the model's baseline performance was in evaluating zero-shot and few-shot prompt engineering. Iterative prompt refinement on the developer side improved the consistency and accuracy of responses to questions regarding abstract tasks, prompt phrasing, and what examples to include [9]. Although, these efforts were not enough as the model was still unable to capture subtle emotional tonality and context specific sentiment (which is an outcome of end of the road for prompt only strategies).

To bridge this gap, a fine-tuning phase targeting the incorrect utterance was introduced. During the prompt engineering phase, annotated samples were directly drawn from the observations of user interaction, and the sample was highly relevant and tailored to observed weaknesses [17][19]. With this alignment, the model could learn subtleties of sentiment expression that were not learned when prompting as there were still several dimensions that the model could not generalize. With this, a more robust and context-sensitive model was produced that could provide accuracy and coherence across different test scenarios.

These implications are very important. The first use case involves the role of prompt engineering as a valuable diagnostic and data collection phase to identify specific edge cases and user needs that can then be fed back as ground truth for terminal resource intensive fine tuning [7]. Rooting the process in real world interactions is then done to lower the risk of overfitting or misalignment while tuning. Second, this allows for a cost-effective development cycle in which tuning is saved for those areas where prompting is insufficient to optimize resources. Aside from this, this strategy also supports the idea that ADT and tuning are not 'antagonistic' methodologies but alternative gadgets inside the general AI plan equipment pack [13]. Together, they provide a pathway to iterate on the development of the system that is exploratory at prompt engineering and structural at model tuning. In addition, this combined method fits the responsible AI development principles by the adaptive, human-in-the-loop feedback processes before the model is retrained [2][23][26].

Finally, our conclusions from Case 4 can be summed up as: Prompt and tune interventions represent a practical and strategic way in order to develop intelligent systems. Instead of selecting one of the two, developers and researchers may be most benefited by combining the two: introducing prompt-based exploration and validation followed by precision tuning for durable and context specific optimization.

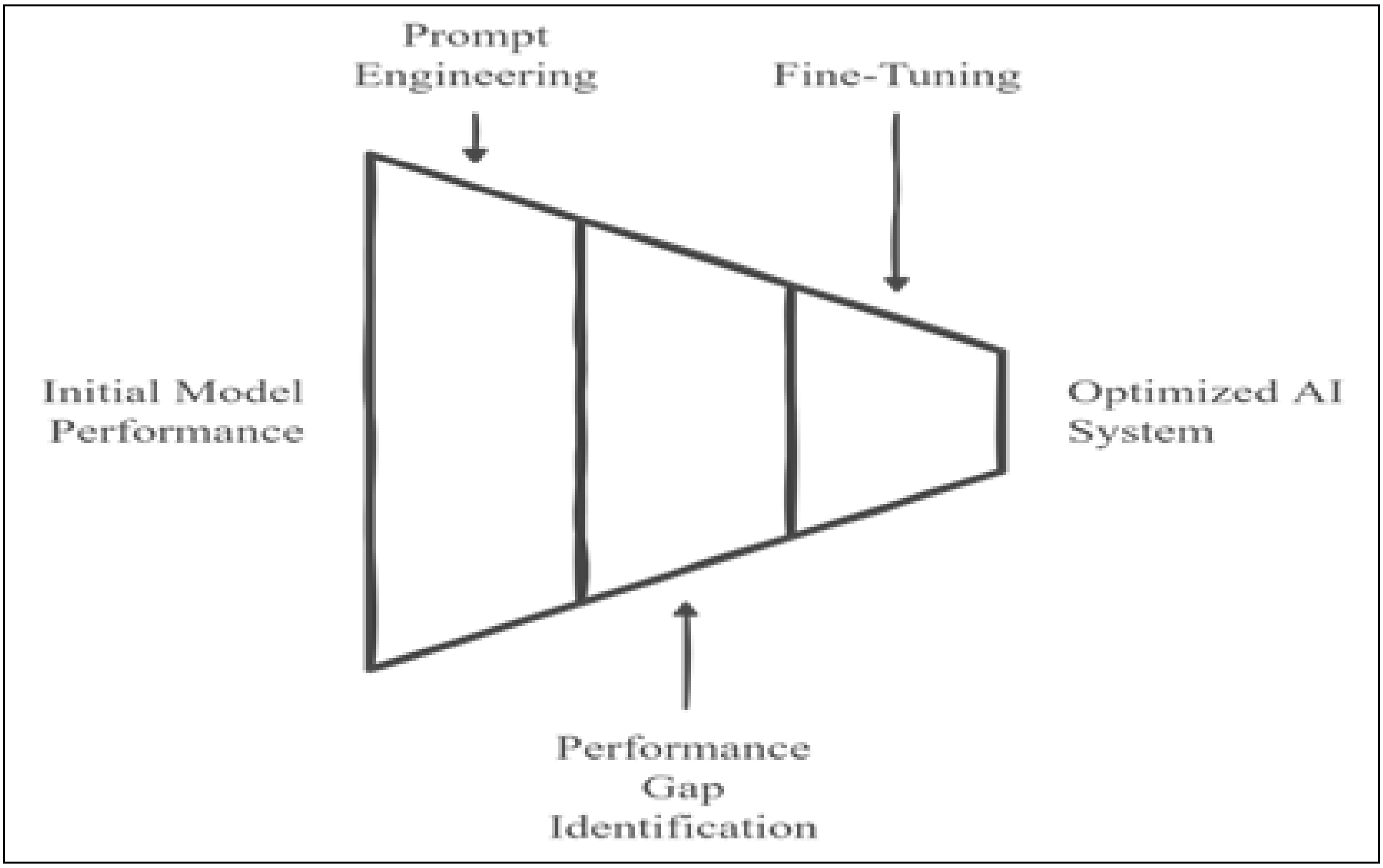

Fig 3 Enhancing AI Systems through Combined Strategies

VI. POWER DYNAMICS: WHO'S REALLY IN CONTROL?

How much power a prompt engineer or model tuner retains when developing AI systems depends on different views of control in artificial intelligence processes. Control means the power to lead outputs toward specific desired outcomes of AI models. People interpret control according to whether human power stays unified under leadership or is distributed among many users. This matches the discussion of humans versus artificial intelligence self-rule. Developers can make needed changes to both the automatic responses and the model settings that create them.

Main System(command) users oversee the AI system by changing what goes into it through the prompt entry. They create adequate instructions that specify what an AI model must generate each time. A prompt engineer uses model capabilities to develop ideal formats for model responses to given requests. Although they do not directly edit the model, logic prompts engineers to control what the system produces at its core level [2].

Model tuners serve as internal designers in this structure. These professionals adjust multiple layers of the model through parameters to improve training datasets making fundamental changes to the system's operation. Changing model behaviour model tuners enable their product to handle various tasks better in all situations. Tuners make long-lasting changes to a model when they handle parameters and learning rate settings plus modify activation functions [1]. The ability to control system components does not have to function like a fixed amount between individuals. Everyone must take part in this task. From a system-wide viewpoint both prompt engineers and model tuners create unique parts of the model's efficiency. The model achieves good results when prompt engineers direct it towards suitable responses and model tuners build a strong and flexible base. The model's performance shows what both external and internal changes do [8].

How systems handle ownership between proprietary and open-source platforms affects who can influence the operating methods. System owners at OpenAI and Google DeepMind strongly control how their AI network can be tuned, even for their engineers. Meta's LLaMA open-source platform lets multiple members of the AI community direct its development process [25]. These platforms give end-users the tools to work with model training and adjusting tools directly and remove control from single entities to distribute it across many participants.

This evaluation finds multiple sides in controlling AI advancement. Preset engineers guide model outcomes by sending organized inputs live while model programmers modify the technical rules that the model uses to interpret inputs. The two roles of prompt engineers and model tuners support AI development but vary in their abilities and control authority based on proprietary and open-source platforms. To create balanced AI systems organizations must blend the expertise of both prompt engineers and model tuners.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

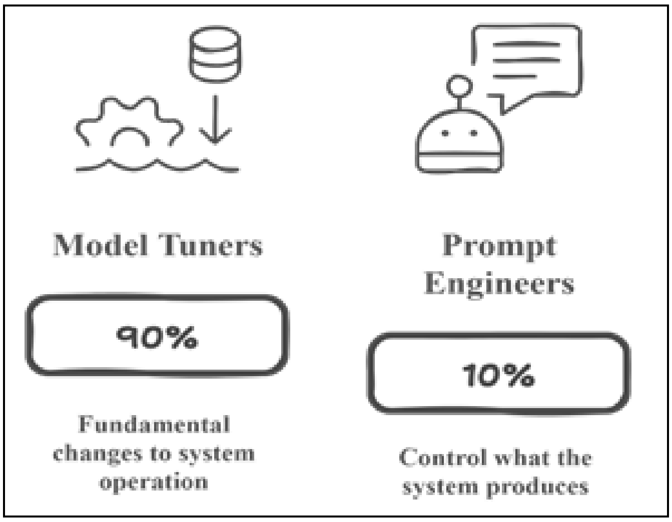

Fig 4 AI System Control

VII. ETHICAL AND GOVERNANCE IMPLICATIONS

AI model ethics and governance directly connect the methods used for controlling system behaviors between model tuning and prompt engineering techniques.

Through model tuning organizations possess an effective means to adjust AI system responses therefore achieving compatibility with institutional values together with ethical standards and domain-specific objectives [27]. The process of controlling AI systems through several developers produces a situation where developers along with organizations serve as gatekeepers determining what the system can and cannot perform [16]. The need for safety requires these measures yet creates issues regarding transparent operations and diminished diversity in perspective allocation. The developing technology of prompt engineering represents a more open system for direct user control. The natural language interface enables end users to steer model outputs thus lowering the obstacle to powerful model access [9]. The influence of educational tools along with legal assistants and creative applications requires no prior technical expertise. Contrary to its advantages prompt accessibility exposes systems to dangers including dangerous output generation and misused safety features [23].

The deployment of these systems encounters difficulties when maintaining ethical guidelines. Implementing fine-tuning models introduces potential unintended bias, which persists because of problems with training data or suboptimal optimization targets. The application of prompt-based systems can lead to the production of harmful stereotypes alongside malicious threats through prompt injection attacks [2].

A broad agreement exists between experts for creating regulatory structures that enable transparency and audit abilities to address these risks. The requirement for better disclosure emerges about tuning protocols, how input data is tracked and how system prompts are monitored should be established. A governance system comprising developers, users, policymakers and ethicists needs implementation to guarantee that both tuning operations and prompting methods match societal norms and human rights standards [13].

VIII. FUTURE OUTLOOK

The development of artificial intelligence systems leads to the advancement of methods used for their control and guidance systems. The primary AI trend involves systems which produce optimized prompts to aid other AI models through meta-prompting. Through recursive prompt engineering practice, developers can potentially create automated solutions for complex operations such as document summarizing and code creation while optimizing queries in domains that need adaptive context capability [4][7]. The trend includes the growing adoption of hybrid approaches, which integrate prompt engineering methods with fine-tuning approaches. Combining prompt tuning and parameter-efficient adapters enables developers to direct large models without performing expensive, complete retraining operations. In order to offer domain-specific performance and improved stability these methods directly embed prompts and tuning vectors within model architectures [16][20]. Controllable AI systems are replacing the traditional methods due to this new development.

The use of low-code as well as no-code AI platforms makes advanced AI technology available to users who lack programming skills. Educators alongside other domain experts who lack technical expertise can produce operational AI instruments through low-code and no-code platforms which simplify model tuning and prompt construction [8][12]. Making AI technologies accessible matches current developments in AI technology accessibility that are gaining public acceptance.

AI UX establishes itself as an independent discipline through which the field progresses. The core aspects of AI UX concentrate on human interactions with AI models exceeding accuracy needs which include transparency, explainability alongside responsiveness and trust factors. Greater mainstream application of large language models demands human-centred design implementation for AI workflow development [2][17].

AI control strategies in the future will emerge as a coordinated system since prompt engineering and model tuning interact via accessibility, efficiency, and usability requirements.

IX. CONCLUSION

Research demonstrates prompt engineering functions separately from model tuning, two distinct capabilities for controlling AI output results. This accessible approach to prompt engineering enables various users to operate AI systems using descriptive questions while leveraging contextual hints. Model tuning provides users with an extensive control mechanism because they can directly modify model parameters while requiring more resources but delivering improved consistency and ongoing adaptability.

The specific purposes along with performance limits and deployment situations within OpenAI, Google DeepMind and Meta respectively determine which strategy emerges most

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

suitable for each AI system implementation. The strengths of prompt engineering involve quick prototyping and democratization, but model tuning delivers better performance capabilities joined with domain-specific reliability.

This paper shows how the two approaches operate in unison instead of being considered separate paradigms. AI systems deliver better trustworthiness, equity, and response capabilities through the combined efforts of these approaches. People who work in and develop critical sectors incorporating AI must advocate for prompt engineering and model-tuning strategies with responsible and knowledgeable usage to guarantee the ethical and effective use of AI technology.

REFERENCES

[1]. A. Ajagekar, N. S. Mattson, and F. You, "Energy-efficient AI-based control of semi-closed greenhouses leveraging robust optimization in deep reinforcement learning," *Advances in Applied Energy*, vol. 9, p. 100119, Feb. 2023, doi: 10.1016/j.adapen.2022.100119.

[2]. P. Brusilovsky, "AI in Education, Learner Control, and Human-AI Collaboration," *International Journal of Artificial Intelligence in Education*, vol. 34, no. 1, pp. 122–135, Aug. 2023, doi: 10.1007/s40593-023-00356-z.

[3]. H. Zhang and M. O. Shafiq, "Survey of transformers and towards ensemble learning using transformers for natural language processing," Journal of Big Data, vol. 11, no. 1, Feb. 2024, doi: 10.1186/s40537-023-00842-0.

[4]. L. Giray, "Prompt Engineering with ChatGPT: A Guide for Academic Writers," *Annals of Biomedical Engineering*, vol. 51, no. 12, pp. 2629–2633, Jun. 2023, doi: 10.1007/s10439-023-03272-4.

[5]. D. Han, J. Lee, J. Im, S. Sim, S. Lee, and H. Han, "A Novel Framework of Detecting Convective Initiation Combining Automated Sampling, Machine Learning, and Repeated Model Tuning from Geostationary Satellite Data," *Remote Sensing*, vol. 11, no. 12, p. 1454, Jun. 2019, doi: 10.3390/rs11121454.

[6]. S. M. Shaffi, "Enhancing Customer Journey Intelligence: a unified framework for 360 - degree analytics using generative AI," International Journal of Science and Research (IJSR), vol. 14, no. 2, pp. 635–640, Feb. 2025, doi: 10.21275/sr25210113419.

[7]. L. Henrickson and A. Meroño-Peñuela, "Prompting meaning: a hermeneutic approach to optimising prompt engineering with ChatGPT," *AI & Society*, Sep. 2023, doi: 10.1007/s00146-023-01752-8.

[8]. E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, Mar. 2023, doi: 10.1016/j.lindif.2023.102274.

[9]. P. Korzynski, G. Mazurek, P. Krzypkowska, and A. Kurasinski, "Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT," *Entrepreneurial Business and Economics Review*, vol. 11, no. 3, pp. 25–37, Jan. 2023, doi: 10.15678/eber.2023.110302.

[10]. K. Lee, "A Systematic Review on Social Sustainability of Artificial Intelligence in Product Design," *Sustainability*, vol. 13, no. 5, p. 2668, Mar. 2021, doi: 10.3390/su13052668.

[11]. H. Zheng et al., "Learning from models beyond fine-tuning," Nature Machine Intelligence, Jan. 2025, doi: 10.1038/s42256-024-00961-0.

[12]. C. Liu, "Artificial Intelligence Interactive Design system based on digital multimedia technology," *Advances in Multimedia*, vol. 2022, pp. 1–12, Jan. 2022, doi: 10.1155/2022/4679066.

[13]. S. Makridakis, F. Petropoulos, and Y. Kang, "Large language models: their success and impact," *Forecasting*, vol. 5, no. 3, pp. 536–549, Aug. 2023, doi: 10.3390/forecast5030030.

[14]. B. Meskó, "Prompt engineering as an important emerging skill for medical professionals: tutorial," *Journal of Medical Internet Research*, vol. 25, p. e50638, Sep. 2023, doi: 10.2196/50638.

[15]. C. Monday, M. S. Zaghloul, D. Krishnamurthy, and G. Achari, "A Review of AI-Driven Control Strategies in the Activated Sludge Process with Emphasis on Aeration Control," *Water*, vol. 16, no. 2, p. 305, Jan. 2024, doi: 10.3390/w16020305.

[16]. S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and Knowledge Graphs: A Roadmap," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3580–3599, Jan. 2024, doi: 10.1109/tkde.2024.3352100.

[17]. M. Perkins, "Academic Integrity considerations of AI Large Language Models in the post-pandemic era: ChatGPT and beyond," *Journal of University Teaching and Learning Practice*, vol. 20, no. 2, Jan. 2023, doi: 10.53761/1.20.02.07.

[18]. Y. Zhang, X. Wang, L. Wu, and J. Wang, "Enhancing chain of thought prompting in large language models via reasoning patterns," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 24, pp. 25985–25993, Apr. 2025, doi: 10.1609/aaai.v39i24.34793.

[19]. U. U. Rehman, S.-B. Park, and S. Lee, "Secure Health FOG: a novel framework for personalized recommendations based on adaptive model tuning," *IEEE Access*, vol. 9, pp. 108373–108391, Jan. 2021, doi: 10.1109/access.2021.3101308.

[20]. J. Ruggaber, K. Ahmic, J. Brembeck, D. Baumgartner, and J. Tobolář, "AI-For-Mobility—A new research platform for AI-Based control methods," *Applied Sciences*, vol. 13, no. 5, p. 2879, Feb. 2023, doi: 10.3390/app13052879.

[21]. C. E. Short and J. C. Short, "The artificially intelligent entrepreneur: ChatGPT, prompt engineering, and entrepreneurial rhetoric creation," *Journal of Business Venturing Insights*, vol. 19, p. e00388, Mar. 2023, doi: 10.1016/j.jbvi.2023.e00388.

[22]. M. Suzuki and S. Yahagi, "Yaw-Rate Controller Tuning for Autonomous Driving: Virtual Internal Model Tuning approach," *Journal of Robotics and*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may255

Mechatronics, vol. 35, no. 2, pp. 308–316, Apr. 2023, doi: 10.20965/jrm.2023.p0308.

[23]. S. Tian *et al.*, “Opportunities and challenges for ChatGPT and large language models in biomedicine and health,” *Briefings in Bioinformatics*, vol. 25, no. 1, Nov. 2023, doi: 10.1093/bib/bbad493.

[24]. M. Wang, M. Wang, X. Xu, L. Yang, D. Cai, and M. Yin, “Unleashing ChatGPT’s power: A case study on optimizing information retrieval in flipped classrooms via prompt engineering,” *IEEE Transactions on Learning Technologies*, vol. 17, pp. 629–641, Oct. 2023, doi: 10.1109/tlt.2023.3324714.

[25]. L. Yan *et al.*, “Practical and ethical challenges of large language models in education: A systematic scoping review,” *British Journal of Educational Technology*, vol. 55, no. 1, pp. 90–112, Aug. 2023, doi: 10.1111/bjet.13370.

[26]. C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, “Large language models for human–robot interaction: A review,” *Biomimetic Intelligence and Robotics*, vol. 3, no. 4, p. 100131, Oct. 2023, doi: 10.1016/j.birob.2023.100131.

[27]. J. Zhang, Y. Shu, and H. Yu, “Fairness in Design: A framework for facilitating ethical artificial intelligence designs,” *International Journal of Crowd Science*, vol. 7, no. 1, pp. 32–39, Mar. 2023, doi: 10.26599/ijcs.2022.9100033.