Evaluating the Impact of Adapter-Based Fine-Tuning on Structured Parsing Performance in Large Language Models

Ratomir Karlović[1] **and Luka Sever**[2]

[1]**Address of first author**
[2]**Address of second author**

ABSTRACT

Recent advances in large language models (LLMs) highlight two dominant strategies for performance improvement: prompt engineering and fine-tuning. While prompt design can significantly influence model output, it remains uncertain whether lightweight fine-tuning methods,such as adapter-based training, offer meaningful advantages for structured, domain-specific tasks. This study builds on prior research comparing three prompting strategies for natural-language command parsing into JSON schemas. Expanding that framework, the current work investigates how adapter-based fine-tuning, where most model parameters are frozen and only small adapter modules are trained, affects model accuracy, consistency, and robustness. The experiment uses the same controlled shopping-cart parsing task and dataset of 1000 synthetic commands to ensure direct comparability. Results will quantify the trade-off between computational cost and performance gains, offering evidence-based insights into whether fine-tuning is a justified investment compared to advanced prompt engineering. The expected contribution is a clear, empirical framework for deciding when fine-tuning meaningfully enhances LLM utility in applied natural-language understanding.

Keywords: LLM, Adapters, PEFT

1 INTRODUCTION

The rapid evolution of Large Language Models (LLMs) has fundamentally transformed the landscape of Natural Language Processing (NLP), enabling advanced capabilities in reasoning, content generation, and semantic understanding (Ling et al., 2025). While early applications focused primarily on open-ended text generation, modern production environments increasingly require LLMs to function as reliable semantic parsers capable of mapping unstructured user intent into rigorous, machine-readable formats such as JSON or SQL (Han et al., 2024). This capability is particularly critical in domain-specific applications, such as e-commerce controllers, where valid schema adherence is a prerequisite for system functionality. A primary challenge in deploying these systems is the ”optimization dilemma”: choosing between inference-time strategies and model training. As highlighted in our previous survey (Karlović and Lorencin, 2025), while prompt engineering offers a lightweight ”black-box” approach to adaptation, it often struggles with robustness when constrained to rigid schema definitions. Conversely, full-parameter fine-tuning yields high precision but incurs prohibitive computational costs and memory requirements (Mundra et al., 2024). To address this, Parameter-Efficient Fine-Tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA), have emerged as a middle ground, freezing pre-trained weights and injecting trainable rank-decomposition matrices to approximate full fine-tuning performance with a fraction of the resources (Hu et al., 2023). To address these challenges, this study evaluates the cost-benefit ratio of fine-tuning against prompting for high-fidelity semantic parsing. Building upon the theoretical frameworks established in our prior work (Karlović and Lorencin, 2025), we implement a controlled experimental design using the LLaMA-3-8B model. The system compares two distinct optimization paradigms: Advanced Prompt Engineering and Adapter-Based Fine-Tuning, utilizing LoRA and its quantized variant (QLoRA) to update the model’s internal representations (Patil et al., 2025). This setup allows for a direct comparison of how each method handles the rigorous constraints of mapping natural language commands to structured JSON outputs. The main aim of this research is to identify the ”break-even point” where the computational

investment of training adapters yields a statistically significant advantage over advanced prompting. The main contribution of this study is a structured empirical framework for benchmarking LLaMA-3 on structured parsing tasks, providing evidence-based insights into whether the resource consumption of fine-tuning is justified by gains in accuracy and robustness. The evaluation reveals that while prompting is sufficient for simple intents, adapter-based methods significantly outperform in handling complex, multi-intent schemas, particularly when targeting all linear layers with optimized hyperparameters.

2 BACKGROUND AND RELATED WORKS

The rapid growth of large language models (LLMs) has necessitated strategies for adapting them efficiently to downstream tasks. Zhiqiang Hu et al. (Hu et al., 2023) introduced LLM-Adapters, a framework for parameter-efficient fine-tuning (PEFT) that integrates Series adapters, Parallel adapters, Prefix-Tuning, and LoRA. Their study demonstrated that smaller LLaMA models (7B and 13B) equipped with optimized adapters could match or surpass the performance of much larger models such as ChatGPT and GPT-3.5, particularly on structured reasoning tasks using datasets like Math10K and Commonsense170K. These findings suggest that adapter-based PEFT can improve structured task accuracy compared to advanced prompting strategies.

Ding et al. (Ding et al., 2023) proposed a unified framework called delta-tuning, which adjusts only a small fraction of model parameters. Through extensive evaluation on over 100 NLP tasks, they found that delta-tuning achieved performance comparable to full fine-tuning while significantly reducing GPU memory usage and accelerating backpropagation. Larger models not only benefited more from minimal parameter updates but also converged faster, reinforcing the potential of adapter-based methods to enhance task-specific performance efficiently.

Esmaeili et al. (Esmaeili et al., 2026) conducted empirical studies on PEFT for code-specific LLMs, examining LoRA, Compacter, and IA3. Their findings showed that LoRA consistently outperformed other adapters in code summarization and generation, occasionally exceeding full fine-tuning on low-resource languages like R. The study highlighted that the number of trainable parameters influenced functional correctness more than the adapter architecture itself, further supporting the effectiveness of PEFT in improving structured outputs in specialized domains.

Razuvayevskaya et al. (Razuvayevskaya et al., 2024) compared parameter-efficient techniques and full fine-tuning for multilingual news classification. Using XLM-RoBERTa Large, they showed that LoRA and bottleneck adapters could reduce trainable parameters by 140–280 times while maintaining competitive accuracy. Their results suggested that PEFT strategies can outperform traditional prompting approaches, especially when sufficient source data is available.

Shin et al. (Shin et al., 2025) assessed the trade-offs between prompt engineering and fine-tuning for automated code tasks. Their evaluation of GPT-4 against seventeen fine-tuned baselines revealed that while task-specific prompting sometimes outperformed automated methods in code summarization, fine-tuned models were generally superior in code generation. Human-in-the-loop conversational prompting further enhanced outcomes, demonstrating that adapter-based fine-tuning can stabilize and improve output quality over prompting alone.

Ming Gong et al. (Gong et al., 2025) developed structure-learnable adapters, which introduced differentiable gating and sparsity control to dynamically optimize adapter placement and activation paths. Experiments on the Multi-Task NLU Benchmark indicated that this approach not only achieved high accuracy with minimal parameters but also reduced output variability and maintained robustness under noisy conditions, supporting the notion that adapter fine-tuning can yield more consistent outputs than prompting.

Fang and Ye (Fang and Ye, 2025) proposed RFedLR, a robust PEFT framework for federated learning. By selectively updating noise-sensitive parameters and dynamically weighting client updates, RFedLR achieved up to 10.15% accuracy improvements over state-of-the-art baselines under high-noise conditions while using only 0.1754% of the trainable parameters. These results reinforce that adapter-based fine-tuning provides more stable and reliable outputs than prompt-based methods, even in decentralized and noisy environments.

Shuo Chen et al. (Chen et al., 2023) benchmarked eleven adaptation methods on vision-language models under textual and visual corruptions. They found that parameter-efficient adapters exhibited higher resilience to input perturbations than full fine-tuning or prompting, although no single method dominated

across all datasets. Increasing adaptation data or parameter size did not guarantee robustness, highlighting the importance of careful PEFT design to enhance model stability.

Chen et al. (Chen et al., 2025) evaluated prompt engineering for industrial document processing tasks and found that few-shot learning improved reasoning capabilities over zero-shot prompts. However, adapter-based PEFT consistently offered more reliable and less variable outputs, particularly in complex scenarios with noisy OCR inputs.

Kaijie Zhu et al. (Zhu et al., 2023) introduced PromptRobust, a benchmark for adversarial prompt evaluation across multiple LLMs. Their results showed that word-level adversarial prompts caused a 39% average performance drop, and few-shot prompts exhibited better robustness than zero-shot prompting. The study illustrates that adapter fine-tuning can improve resilience to prompt-based perturbations.

Jaehyung Kim et al. (Kim et al., 2023) presented ROAST, combining adversarial perturbations with selective training to improve robustness across sentiment classification and entailment tasks. ROAST yielded average improvements of 18.39% and 7.63% over state-of-the-art baselines, demonstrating that adapter fine-tuning can mitigate input noise effects more effectively than prompting.

Pei et al. (Pei et al., 2025) proposed SelfPrompt, which autonomously evaluates LLM robustness using domain-constrained knowledge graphs and adversarial prompt generation. Their findings indicated that domain-specific adapter fine-tuning enhances robustness in specialized fields like medicine and biology, outperforming prompting strategies alone.

Lin Mu et al. (Mu et al., 2025) introduced Robustness of Prompting (RoP), a parameter-free method for improving LLM resilience through error correction and guidance prompts. While effective, experiments confirmed that adapter fine-tuning provides superior stability and transferability across architectures, emphasizing the practical value of PEFT in noisy real-world environments.

Sajjadi Mohammadabadi et al. (Sajjadi Mohammadabadi et al., 2025) surveyed LLM architectures and adaptation methods, emphasizing the efficiency of PEFT, instruction tuning, and RLHF. They noted that adapter-based fine-tuning can deliver significant performance gains relative to prompting, justifying its computational cost in applied settings.

Trad and Chehab (Trad and Chehab, 2024) studied phishing detection LLMs and found that while refined prompting improved performance, fine-tuned models achieved higher F1-scores and superior reliability, demonstrating that the cost of PEFT is justified when high precision is required.

Pornprasit and Tantithamthavorn (Pornprasit and Tantithamthavorn, 2024) evaluated LLM-based code review, showing that fine-tuning GPT-3.5 achieved 73–74% higher Exact Match rates than prompting approaches. Their study confirmed that adapter-based fine-tuning can deliver meaningful gains even with modest training data.

Wang et al. (Wang et al., 2025) introduced COSMOS, a framework for predicting adaptation outcomes with minimal trials. Their results indicated that fine-tuning consistently outperformed in-context learning in medium to high-cost scenarios, supporting the view that adapter PEFT justifies its resource investment by providing predictable and superior performance.

Based on the cumulative evidence from these studies, the overarching goal of this work is to investigate how adapter-based parameter-efficient fine-tuning (PEFT) compares to advanced prompting in terms of structured task accuracy, output stability, robustness, and cost-effectiveness. Accordingly, the following research hypotheses are formulated:

- **H1:** Adapter-based PEFT improves structured task performance relative to advanced prompting.
- **H2:** Adapter fine-tuning reduces output variability compared to prompting approaches.
- **H3:** Adapter fine-tuning enhances robustness to input noise, adversarial perturbations, and paraphrasing relative to prompting.
- **H4:** Performance gains achieved via adapter-based PEFT justify the computational and resource cost compared to advanced prompting methods.

3 METHODOLOGY

This section details the experimental framework designed to evaluate the trade-offs between parameter-efficient fine-tuning (PEFT) and advanced in-context learning (ICL). The primary objective is to determine the performance ceiling, robustness, and structural integrity of various Large Language Models (LLMs)

when tasked with converting unstructured natural language into strictly formatted JSON objects. This methodology extends the comparative benchmarks established in (Karlović and Lorencin, 2025), applying them to a specialized domain of structured parsing.

3.1 Dataset and Task Definition

The experimental task focuses on the "Natural Language to Structured Command" conversion, a critical component in automated inventory and commerce systems. The dataset consists of 10,000 synthetically generated yet linguistically diverse shopping-cart instructions. Each instance requires the model to extract or infer three specific attributes:

1. **Action:** A classification task mapping verbs to a binary set of $\{add,\ remove\}$.
2. **Product:** A named entity recognition (NER) task to identify the subject item.
3. **Quantity:** A numerical extraction task where the model must default to an integer value of 1 if no quantifier is explicitly mentioned in the prompt.

The corpus was split into a **Training Set** of 8,000 examples, utilized for the LoRA adapter training phase, and a **Test Set** of 2,000 examples. The test set remained unseen by the adapters to ensure unbiased evaluation against the prompting strategies.

3.2 Adapter Training and Evaluation Data

To rigorously evaluate the effectiveness of adapter-based fine-tuning, we constructed a controlled dataset specifically tailored for structured command parsing. The corpus comprises **10,000 annotated natural-language shopping instructions**, each paired with a ground-truth structured representation containing three labeled attributes: *action*, *product*, and *quantity*.

All samples were synthetically generated but linguistically diversified to emulate real-world user interactions in e-commerce environments. Variation was introduced through paraphrasing, synonym substitution, implicit quantity expressions, and differing grammatical constructions to ensure that models could not rely on surface-form memorization alone.

3.2.1 Training–Validation Split

From the full corpus, **8,000 examples** were allocated to the adapter training phase. During fine-tuning, this subset was further divided into:

- **Training Set:** 8,000 examples used for gradient updates of adapter parameters.
- **Validation Set:** 2,000 held-out examples used exclusively for intermediate evaluation, hyperparameter monitoring, and early-stopping criteria.

The validation subset enabled continuous assessment of convergence behavior, overfitting risk, and structural output fidelity during adapter optimization.

3.2.2 Test Set for Generalization Assessment

To measure true generalization performance, we constructed an additional **independent test set of 2,000 examples**. These samples were fully disjoint from the adapter training and validation data, containing novel sentence structures, paraphrasing patterns, and lexical combinations not observed during fine-tuning.

This separation ensures that evaluation reflects the adapters' ability to generalize learned structural mappings rather than memorize training distributions.

3.2.3 Input Characteristics

Each input instance represents a free-form shopping command, such as product additions, removals, or quantity adjustments. Linguistic diversity includes:

- Explicit quantities (e.g., "Add 3 apples")
- Implicit quantities (e.g., "Remove milk" $\rightarrow$ quantity defaults to 1)
- Multi-word paraphrases (e.g., "Put two bottles of water into the cart")
- Verb variation (add, insert, delete, remove, take out)

Despite surface variability, all commands map deterministically to a fixed JSON schema, enabling objective evaluation of structural and semantic correctness.

3.2.4 Annotation Schema

Ground-truth labels were normalized to enforce schema consistency:

- **Action:** Binary label $\{add, remove\}$
- **Product:** Canonical single-entity product name
- **Quantity:** Integer value with a default fallback of 1

This normalization ensures that evaluation metrics reflect parsing capability rather than lexical formatting differences.

3.2.5 Rationale for Dataset Scale

The selected dataset size balances computational feasibility with sufficient diversity for adapter specialization. Prior PEFT studies indicate that adapter modules can achieve strong task alignment with relatively modest data volumes, provided the schema is well-defined and supervision signals are consistent. The 8k/2k/2k split therefore provides adequate capacity for training, validation monitoring, and unbiased performance benchmarking.

3.3 Model Selection and Inference Environment

We selected a heterogeneous suite of models to analyze how architectural scale and specialized training (e.g., reasoning-distilled models) influence parsing performance. The models were deployed via the Ollama framework to ensure local reproducibility and consistent inference parameters (e.g., temperature set to 0 to minimize stochastic variance).

The model suite is categorized by parameter count and family:

- **Large-Scale:** Llama 3.3 (70B).
- **Mid-Range:** DeepSeek-R1 (14B), Phi-4 (14B), Llama 3.1 (8B), Qwen 3 (8B), and Granite 3.3 (8B).
- **Small-Scale (SLMs):** Mistral (7B), Qwen 3 (4B), Gemma 3 (4B), Llama 3.2 (3B), and DeepSeek-R1 (1.5B).

3.4 Adapter-Based Fine-Tuning (LoRA)

For the fine-tuning paradigm, we employed **Low-Rank Adaptation (LoRA)**. This technique was chosen for its ability to maintain the underlying general knowledge of the base model while specializing its output structure for JSON parsing with minimal computational overhead.

The LoRA approach modifies the pre-trained weight matrices $W_0 \in \mathbb{R}^{d\times k}$ by adding a low-rank decomposition BA, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. The forward pass for a given input x is represented as:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{1}$$

In this study, we targeted all linear modules within the transformer architecture (specifically `q_proj`, `v_proj`, `k_proj`, `o_proj`, and `MLP` layers) to ensure the adapter captured the necessary structural constraints for JSON generation. The specific hyperparameters utilized in this study include:

- **Rank (r):** [Insert Value]
- **Alpha (α):** [Insert Value]
- **Target Modules:** All linear layers (comprehensive adaptation).
- **Precision:** 4-bit (QLoRA) or 16-bit, depending on the hardware constraints at runtime.

3.5 Prompt Engineering Strategies

To establish a baseline for In-Context Learning (ICL), we implemented three escalating levels of prompt complexity:

1. **Minimal Instruction (Zero-Shot):** A concise prompt defining the role of the model and the required JSON schema.
2. **Extended Prompt (Instructional Weight):** This strategy includes detailed edge-case instructions, emphasizing the default quantity logic and strict "no-prose" constraints to prevent conversational "chatter."
3. **Few-Shot Prompting:** Building upon the Extended Prompt, this version includes [Insert Number] high-quality examples of natural language inputs and their corresponding JSON outputs, providing a semantic and structural blueprint for the model.

3.6 Evaluation Metrics

The models are evaluated on two primary dimensions: **Syntactic Integrity** and **Semantic Accuracy**.

3.6.1 JSON Validity Rate

Before measuring accuracy, we assess if the output is a "well-formed" JSON object. Any output that fails standard library parsing ($json.loads()$) is categorized as a failure.

3.6.2 Structured F1 Score

For all valid JSON outputs, we calculate the F1 score. This requires an exact match between the predicted value and the ground truth for each of the three fields. The field-level F1 score is defined as the harmonic mean of precision (P) and recall (R):

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2}$$

We report the macro-averaged F1 across all fields to provide a singular measure of parsing performance.

4 RESULTS AND DISCUSSION

5 CONCLUSION

6 ACKNOWLEDGMENTS

This research is (partly) supported by SPIN projects "INFOBIP Konverzacijski Order Management (IP.1.1.03.0120)", "Projektiranje i razvoj nove generacije laboratorijskog informacijskog sustava (iLIS)" (IP.1.1.03.0158), "Istraživanje i razvoj inovativnog sustava preporuka za napredno gostoprimstvo u turizmu (InnovateStay)" (IP.1.1.03.0039), "EDIH ADRIA – European Digital Innovation Hub Adria 2.0 (project no. 101256325)" and the FIPU project "Sustav za modeliranje i provedbu poslovnih procesa u heterogenom i decentraliziranom računalnom sustavu".

REFERENCES

Chen, L.-C., Weng, H.-T., Pardeshi, M. S., Chen, C.-M., Sheu, R.-K., and Pai, K.-C. (2025). Evaluation of prompt engineering on the performance of a large language model in document information extraction. *Electronics*, 14(11):2145.

Chen, S., Gu, J., Han, Z., Ma, Y., Torr, P., and Tresp, V. (2023). Benchmarking robustness of adaptation methods on pre-trained vision-language models. *Advances in Neural Information Processing Systems*, 36:51758–51777.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Esmaeili, A., Saberi, I., and Fard, F. (2026). Empirical studies of parameter efficient methods for large language models of code and knowledge transfer to r. *Empirical Software Engineering*, 31(2):30.

Fang, X. and Ye, M. (2025). Towards robust parameter-efficient fine-tuning for federated learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Gong, M., Deng, Y., Qi, N., Zou, Y., Xue, Z., and Zi, Y. (2025). Structure-learnable adapter fine-tuning for parameter-efficient large language models. In *International Conference on Artificial Intelligence and Computational Engineering (AICE 2025)*, volume 2025, pages 225–230. IET.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., and Lee, R. (2023). Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276.

Karlović, R. and Lorencin, I. (2025). Large language models as retail cart assistants: A prompt-based evaluation. *Human Systems Engineering and Design (IHSED2025): Future Trends and Applications*, 198.

Kim, J., Mao, Y., Hou, R., Yu, H., Liang, D., Fung, P., Wang, Q., Feng, F., Huang, L., and Khabsa, M. (2023). Roast: Robustifying language models via adversarial perturbation with selective training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3412–3444.

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., et al. (2025). Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 58(3):1–39.

Mu, L., Chu, G., Ni, L., Sang, L., Wu, Z., Jin, P., and Zhang, Y. (2025). Robustness of prompting: Enhancing robustness of large language models against prompting attacks. *arXiv preprint arXiv:2506.03627*.

Mundra, N., Doddapaneni, S., Dabre, R., Kunchukuttan, A., Puduppully, R., and Khapra, M. M. (2024). A comprehensive analysis of adapter efficiency. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, pages 136–154.

Patil, R., Khot, P., and Gudivada, V. (2025). Analyzing llama3 performance on classification task using lora and qlora techniques. *Applied Sciences*, 15(6):3087.

Pei, A., Yang, Z., Zhu, S., Cheng, R., and Jia, J. (2025). Selfprompt: Autonomously evaluating llm robustness via domain-constrained knowledge guidelines and refined adversarial prompts. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6840–6854.

Pornprasit, C. and Tantithamthavorn, C. (2024). Fine-tuning and prompt engineering for large language models-based code review automation. *Information and Software Technology*, 175:107523.

Razuvayevskaya, O., Wu, B., Leite, J. A., Heppell, F., Srba, I., Scarton, C., Bontcheva, K., and Song, X. (2024). Comparison between parameter-efficient techniques and full fine-tuning: A case study on multilingual news article classification. *Plos one*, 19(5):e0301738.

Sajjadi Mohammadabadi, S. M., Kara, B. C., Eyupoglu, C., Uzay, C., Tosun, M. S., and Karakuş, O. (2025). A survey of large language models: evolution, architectures, adaptation, benchmarking, applications, challenges, and societal implications. *Electronics*, 14(18).

Shin, J., Tang, C., Mohati, T., Nayebi, M., Wang, S., and Hemmati, H. (2025). Prompt engineering or fine-tuning: An empirical assessment of llms for code. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 490–502. IEEE.

Trad, F. and Chehab, A. (2024). Prompt engineering or fine-tuning? a case study on phishing detection with large language models. *Machine Learning and Knowledge Extraction*, 6(1):367–384.

Wang, J., Albarghouthi, A., and Sala, F. (2025). Cosmos: Predictable and cost-effective adaptation of llms. *arXiv preprint arXiv:2505.01449*.

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Zhang, Y., Gong, N., et al. (2023). Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM workshop on large AI systems and models with privacy and safety analysis*, pages 57–68.