Universiteit Leiden
The Netherlands

Computer Science - Artificial Intelligence

Optimizing Hero Selection in Dota 2 with LLMs: Leveraging Llama 3 for Enhanced Strategy and Decision-Making

Christos Tsirogiannis

Supervisors:
Mike Preuss & Matthias Müller-Brockhausen

MASTER THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

18/06/2025

Abstract

This thesis investigates whether Large Language Models (LLMs) can effectively address the complexity of hero selection in Dota 2. It explores if a small-parameter model like Llama-3.1-8B can be prompted and fine-tuned to provide accurate recommendations, with outputs evaluated by a custom strategic quality metric based on historical match data. The study compares multiple prompt engineering strategies (Zero-shot, Few-shot, Chain-of-Thought) against Supervised Fine-Tuning (SFT) with QLoRA. Key findings demonstrate that structured prompting is crucial. A distinct trade-off was identified between the high-quality, yet slow, reasoning of Chain-of-Thought (CoT) and the fast, reliable outputs of Few-shot prompting. Most notably, the experiment revealed a critical negative result: Supervised Fine-Tuning (SFT) severely degraded model performance, a likely consequence of catastrophic forgetting on the narrow domain dataset.

The thesis concludes that for complex reasoning tasks, invoking a model's existing knowledge is more effective than attempting to embed it through narrow training, establishing prompt engineering as the superior strategy. The findings highlight a clear trade-off between prompt quality and response latency, offering a methodological framework for developing and evaluating similar LLM-based decision-support tools on accessible hardware.

Contents

Chapter 1

Introduction

1.1 The Evolving Landscape of Gaming AI and the Dota 2 Challenge

Large Language Models (LLMs) are demonstrating significant potential in gaming, serving both as tools for procedural content generation, creating dynamic quests and environments [1], and also as intelligent assistants providing strategic guidance to players [2]. We use the video game Dota 2 as a testing ground for these models, focusing on the algorithmic challenge of hero selection. As a leading multiplayer online battle arena (MOBA) game, Dota 2 is recognized for its steep learning curve and strategic depth, where team composition is a major factor in match outcomes [3, 4]. A central component of this strategy is the hero selection phase, where players draft a team from an extensive pool of over 120 heroes. This process involves a complex interplay of hero counters, synergistic abilities, and overall team structure [3]. Despite the importance of this phase, existing in-game tools provide limited real-time support, often failing to take into account draft dynamics or player preferences [5]. This gap presents an opportunity for research into AI systems with improved contextual understanding capable of navigating this complexity [6]. The core challenge lies in developing an AI assistant that can process the combinatorial complexity of hero interactions and provide accurate suggestions within the time constraints of a competitive match.

1.2 Developing a Specialized Model for Enhanced Hero Selection

To address this challenge, this thesis proposes the development and evaluation of a specialized LLM assistant, engineered to enhance decision-making during the hero selection phase in Dota 2. Leveraging the capabilities of the Llama 3 architecture [7], the goal is to produce accurate, reliable, and actionable hero recommendations. The methodology of this research is analyzed in the following main components:

- **Comprehensive Data Foundation:** A comprehensive Dota 2 dataset was collected and prepared, which includes hero statistics, detailed matchup data, historical match records, and curated expert knowledge. This dataset forms the knowledge base for the LLM.
- **Systematic Prompt Engineering:** A systematic comparison of various prompt engineering strategies, including Zero-shot [8], Few-shot [9], and Chain-of-Thought (CoT) prompting [10], was conducted to optimize the LLM's ability to understand context and generate relevant counter-picking advice. This also involved the development of tailored system prompts and an exploration of prompt tuning principles [11].
- **Advanced Model Specialization:** The base LLM was adapted to the Dota 2 domain through domain-specific fine-tuning. This included the application of Parameter-Efficient

Fine-Tuning (PEFT) techniques like LoRA [12] and QLoRA [13], enabling efficient model adaptation and the incorporation of up-to-date domain knowledge, such as mechanics from recent game patches.

- **Data-Driven Validation:** A custom-developed evaluation framework was employed, centered around a custom Draft Evaluator metric. This evaluator is designed to objectively quantify the strategic effectiveness of hero suggestions by considering multiple factors like direct hero matchups, mechanical advantages, and team synergy indicators, informed by general principles of evaluating AI agents in games.

Through these integrated techniques, the study aims to produce a model that is not only strategically adept but also accessible for deployment on consumer-grade GPUs, making it a practical tool for players.

1.3 Research Focus, Questions, and Contributions

This research investigates the encoding of domain-specific Dota 2 knowledge into an LLM, the optimization of its outputs through structured prompting and fine-tuning [14], and the challenges of applying such models to real-time decision-making tasks [15]. The primary research question guiding this thesis is:

Can a large language model be effectively used during the hero selection phase in Dota 2?

To address this central question, the following sub-research questions are investigated:

1. To what extent can prompt engineering strategies influence an LLM's ability to provide accurate hero recommendations in Dota 2?
2. How effectively can domain-specific knowledge, such as hero attributes and patch changes, be integrated into an LLM to provide better hero recommendations?
3. What are the key limitations and challenges of using large language models for real-time strategic tasks, such as hero selection in Dota 2, particularly concerning response latency and computational cost?

The findings of this thesis are expected to contribute to the advancement of AI applications in complex strategy games by demonstrating the potential of appropriately adapted LLMs for real-time decision support [16, 17, 18]. Furthermore, it aims to provide insights into methodologies for developing specialized LLMs that operate with defined performance metrics with accessible computational resources.

Chapter 2

Related Work: LLMs for Strategic Decision Support in Complex Games

2.1 The Evolving Landscape of AI in Complex Strategy Games

For years, complex strategy games have served as a challenging frontier for artificial intelligence research. Today, the field is exploring new directions. Alongside the continued development of highly specialized, superhuman agents, an alternative approach is emerging: leveraging the general-purpose reasoning of Large Language Models (LLMs). This development introduces new possibilities for creating AI that can act as strategic partners, assisting players rather than simply aiming to outperform them.

Before the recent rise of LLMs, the field saw incredible breakthroughs from custom-built AIs designed to master some of the world's most challenging competitive games. Systems like DeepMind's AlphaStar for StarCraft II and OpenAI Five for Dota 2 proved that an AI could handle making countless decisions with incomplete information, just like a human player [19, 20]. While these projects demonstrated that AI could conquer these games, their success came at a cost. They required massive amounts of computing power and were tailor-made for a single game, which meant only a few well-funded labs could attempt such research.

The arrival of powerful, pre-trained LLMs has introduced new possibilities to this landscape. As detailed in recent surveys, LLMs present a novel opportunity to build game agents with faculties for reasoning, planning, and reflection [21, 18]. This paradigm offers a new research direction, shifting the focus for some researchers from the difficult task of learning complex strategies from scratch to the more tractable problem of leveraging the knowledge and reasoning structures already present in a pre-trained model. This thesis, by using a publicly available LLM to provide strategic advice for *Dota 2*, is a direct product of this more accessible approach to research.

2.2 Architectural Paradigms for LLM-based Game Agents

As researchers have begun to integrate LLMs into games, several architectural patterns have taken shape to capitalize on their strengths while addressing their limitations. The field has moved beyond simple "prompt-in, action-out" models, converging on hybrid designs that separate the high-level, deliberative reasoning of LLMs from the low-level, rapid execution needed for real-time play.

A common strategy involves architectures that assign the LLM the role of a high-level strategist, while a more efficient module handles the immediate in-game actions. The SwarmBrain agent for *StarCraft II* is a prime example. It employs an LLM-powered "Overmind Intelligence Matrix" for broad strategic planning and a separate, fast "Swarm ReflexNet" for tactical control

[16]. This hybrid design preserves the strategic depth an LLM can provide while meeting the strict performance demands of a real-time environment.

The design of this thesis is focused exclusively on the pre-game hero selection phase. This is a deliberate choice made in light of these architectural challenges. The drafting phase in *Dota 2* is turn-based and time-limited but does not demand millisecond responses. This setting permits the direct use of LLM inference for decision support without the engineering overhead of a hybrid system. This choice effectively isolates the core research question: how well can an LLM perform the strategic reasoning task itself, separated from the confounding problem of real-time agent embodiment?

2.3 Methodologies for Invoking Strategic Reasoning

Unlocking the latent reasoning abilities of a general-purpose LLM for a specialized task is a significant challenge. Prompt engineering has become the primary method for guiding an LLM's thought process without altering the model itself. The experimental comparison of Zero-shot, Few-shot, and Chain-of-Thought prompting in this thesis follows a classic and robust design for measuring the performance gains from more structured instructions [9, 8]. This approach is mirrored in large-scale empirical studies in other complex fields, such as software engineering, where researchers conduct similar comparisons to find the best prompting technique for tasks like code generation and debugging [22]. This parallel validates the experimental framework of this thesis as a standard scientific method for evaluating LLM performance.

Among the many advanced techniques, Chain-of-Thought (CoT) prompting has been particularly effective for tasks requiring multi-step reasoning [10]. The core idea is to instruct the model to first articulate the intermediate steps of its reasoning before providing a final answer. An important trade-off, observed in this thesis and elsewhere, is that while CoT often gives the best results, it is also substantially slower and more verbose. It appears to be a fundamental principle of prompt engineering that techniques invoking more explicit reasoning consistently produce higher quality results, but at a significantly higher computational cost [22].

2.4 The Challenge of Domain-Specific Knowledge Adaptation

Applying a general-purpose LLM to a specialized domain requires adapting its knowledge, a task made more difficult in dynamic environments like *Dota 2* with its frequent patches. While Supervised Fine-Tuning (SFT) is a standard technique for this, the literature documents a severe risk: "catastrophic forgetting" (CF), the tendency for a model to lose its broad, generalized reasoning capabilities when it is fine-tuned on new, specialized information [23].

The results of this thesis provide a clear empirical case study of this issue. The finding that SFT significantly degraded model performance is a real-world demonstration of CF in action. This negative result is therefore one of the most important scientific contributions of this thesis, serving as a stark warning about the risks of applying SFT to complex reasoning domains, even when using modern Parameter-Efficient Fine-Tuning (PEFT) techniques like LoRA and QLoRA [12, 13].

Given the drawbacks of SFT, Retrieval-Augmented Generation (RAG) has become a leading alternative for injecting domain-specific, dynamic knowledge into an LLM [24, 25]. In a RAG system, the LLM retrieves information from an external, up-to-date knowledge base at the time of a query and uses that information in the prompt. This allows the model to reason with the latest information without altering its internal parameters, thereby avoiding the risk of catastrophic forgetting. The clear failure of SFT in this thesis, contrasted with the promise of RAG, provides a strong justification for the proposed future work.

Chapter 3

Methodology

This research investigates the effectiveness of a Large Language Model (LLM), specifically Llama 3, in providing strategic hero selection recommendations for Dota 2. The study explores how various enhancement techniques—from prompt engineering and tuning to supervised fine-tuning—can improve the quality and relevance of the model's outputs. This chapter outlines the conceptual design of this work, covering the data acquisition and preparation, the strategies for instructing the LLM, the model specialization procedures, and the framework for performance validation. Specific technical implementations are detailed in Chapter 4.

3.1 Data Foundation for Hero Recommendation

The foundation of this research is a comprehensive Dota 2 dataset, created to provide the model with the necessary information about hero mechanics, matchups, and prevailing game strategies. Data was acquired by extracting information from publicly accessible statistics platforms like OpenDota and Dotabuff [26, 27]. These sources offer extensive data, including general hero performance metrics, specific hero matchup data, and historical match records.

3.1.1 Curation of Expert Knowledge and Test Scenarios

While raw statistics provide a broad overview, they are insufficient for defining robust counter-relationships or for establishing precise evaluation benchmarks. A more targeted approach was taken by computing hero counters based on a combination of statistical win-rate advantages and significant mechanical impacts. A curated set of test scenarios was constructed, defining specific enemy sub-compositions (e.g., 1v1 and 2v2 matchups typical of the early-game laning phase) and enumerating a corresponding list of "known good heroes", which are considered to be effective counters. These "known good heroes" were selected through a multi-faceted approach combining domain expertise, analysis of high-level gameplay patterns, and insights from the statistically generated counter data.

3.1.2 Synthetic Data Generation and Preprocessing

To enhance the model's performance by incorporating more extensive domain knowledge for supervised fine-tuning (SFT), a custom dataset was curated. Foundational knowledge on hero attributes, abilities, and core game mechanics was first programmatically scraped from the Dota 2 Fandom wiki [28]. This structured information formed the basis of the knowledge base, which was then used to complement an existing instruct-prompt dataset for Dota 2 [29]. To update this base with the latest game dynamics (patch 7.37) and to fill potential knowledge gaps, the dataset was further augmented with approximately 1,000 new, synthetically generated question-answer pairs using an external large language model (Gemini 2.0 Pro). The goal of creating this

enhanced dataset was to provide the model with a more comprehensive and recent knowledge base for the fine-tuning task.

3.2 Language Model and Interaction Strategies

3.2.1 Core Language Model Selection Rationale

This study utilizes a pre-trained Large Language Model from the Llama 3 family [7]. A smaller 8-billion parameter, instruction-tuned variant was specifically chosen for two reasons. First, its instruction-tuning makes it adept at following complex directions and generating structured outputs. Second, its smaller size makes it a suitable candidate for investigating the feasibility of deploying such models in resource-constrained applications, such as real-time in-game assistance tools. This focus on accessibility aligns with the significant engineering challenges of bringing LLMs from research into production environments, where latency and computational cost are primary concerns [25].

3.2.2 Prompt Engineering Approaches

To investigate the impact of prompt design on the model's reasoning, three distinct prompting strategies were compared. Each was used with a standardized **System Prompt** that defined the model's persona and output requirements.

1. **Zero-Shot Prompting:** As a baseline, the model receives a request for counter-suggestions against a set of enemy heroes with minimal context and no examples. This approach tests the modelś inherent ability to generate relevant responses based solely on its pre-trained knowledge, without additional context [8].

2. **Few-Shot Prompting:** This technique includes several examples of high-quality requests and responses in the prompt, priming the model with successful patterns. This approach builds on the finding that large models can learn from a handful of examples provided directly in the prompt [9].

3. **Chain-of-Thought (CoT) Prompting:** This strategy guides the model to "think step by step," explicitly asking it to break down the problem and generate intermediate reasoning before the final answer. This method has been shown to dramatically improve performance on complex reasoning tasks [10].

3.3 Model Optimization Techniques

3.3.1 Prompt Tuning Concepts

Following the initial prompt engineering comparison, the focus was on systematically optimizing prompt performance. The methodology involved a form of discrete prompt tuning, where a programmatic search was conducted over a space of text-based prompts built from modular components. This approach uses principles related to Prompt Ensembling, but instead of combining outputs, the goal was to **select** the single best-performing template from the generated set. This differs from the *"original"* prompt tuning which learns continuous *"soft prompts"* [11], but it allows for a robust, quantitative comparison of different instructional phrasings and structures.

3.3.2 Supervised Fine-Tuning (SFT) and PEFT with QLoRA

The final optimization phase aimed to specialize the LLM for Dota 2 using Supervised Fine-Tuning (SFT). The objective was to adapt the model's internal parameters to the specific domain

of draft analysis. This involved training the model on a task-specific dataset of input-output pairs, such as player queries and expert-validated counter recommendations.

As full fine-tuning is computationally prohibitive, a Parameter-Efficient Fine-Tuning (PEFT) approach was employed. Specifically, the method is built on **Low-Rank Adaptation (LoRA)**, which freezes the base model's weights and injects small, trainable low-rank matrices into its layers [12]. To further enhance efficiency, **QLoRA** was employed, an optimization that uses 4-bit quantization to dramatically reduce memory usage by training the LoRA adapters on a frozen, quantized version of the base model [13]. This combined PEFT approach makes training feasible on more constrained hardware by updating only a small fraction of the model's total parameters.

3.4 Validation and Evaluation Framework

Assessing the LLM's recommendations required a quantitative and objective evaluation framework. Evaluating strategic advice in a game as complex as Dota 2 demands a specialized metric, as generic scores for text similarity are insufficient to capture strategic quality, a noted challenge in the field of LLMs and games [21]. The framework is centered around a custom-developed **Draft Evaluator** and a suite of performance metrics, applied uniformly across all experiments.

3.4.1 The Draft Evaluator: Conceptual Overview

Central to the evaluation is the Draft Evaluator, a tool designed to quantify the predicted strategic effectiveness of a suggested hero draft. The evaluator's scoring mechanism is rooted in domain-specific Dota 2 knowledge and statistical data. It combines information from four strategic areas, normalizing the score for each component before aggregating them into a final weighted score:

- **Direct Matchup Performance:** Assessing the historical win rate of each suggested hero against each enemy hero.
- **Mechanical Advantage Score:** Evaluating if suggested heroes possess abilities that directly counter enemy strengths or exploit key weaknesses.
- **Overall Team Win Rate Comparison:** Comparing the average historical win rates of the suggested team against the enemy team.
- **Hero Grade/Tier Relevance:** Comparing the general tier classification (e.g., S, A, B) of the suggested heroes versus the enemy heroes.

3.4.2 Key Performance Metrics and Benchmarking

The analysis relies on several key metrics to provide a holistic view of performance. These include the *Average Evaluation Score* from the Draft Evaluator, the number of *Best Performances ("Wins")*, the *Error Rate*, the *Score Difference vs. Known Good* heroes, and the *Hero Match Percentage*, which quantifies the overlap between model suggestions and expert selections. Finally, a *Qualitative Assessment of Reasoning* is used to examine the coherence and logic of the model's justifications. This comprehensive evaluation strategy allows for a robust comparison of the different model enhancement techniques.

Chapter 4

Experimental Setup and Procedures

This chapter provides a detailed account of the technical implementations, configurations, and procedures used to conduct the research outlined conceptually in Chapter 3. It links the research design to its outcomes, specifying how the data was collected, the models were configured, and the evaluations were performed.

4.1 Dataset Collection and Preparation

This section details the implementation of the data acquisition and curation strategy defined in Section 3.1. The process involved gathering raw data, computing strategic relationships from it, and constructing the specific scenarios used for model evaluation.

4.1.1 Acquisition of Hero Statistics and Matchup Data

The data collection began by extracting information from publicly accessible Dota 2 statistics platforms, primarily OpenDota and Dotabuff [26, 27]. The collected data included aggregated statistics like hero win rates, granular matchup data detailing inter-hero performance, and historical records from past matches. A custom web service was developed to orchestrate and automate this process, handling data retrieval, the mapping of hero identifiers to names, and data normalization.

4.1.2 Development of Curated Counter Data and Test Scenarios

Using the raw data, a more targeted set of data for analysis and evaluation benchmarks was created. A custom script processed the matchup data to compute a "counter score" for each hero pairing based on win rates and mechanical disadvantages. In parallel, a series of curated test scenarios were defined within the experimental codebase. These scenarios specified enemy compositions (e.g., 1v1, 2v2) and a corresponding list of "known good heroes," further categorized by strategic concepts such as "Anti-Healing" or "Break" to ensure a diverse evaluation.

4.1.3 Data Preprocessing Procedures

All collected data underwent a preprocessing pipeline to ensure quality and consistency. This involved standard procedures for cleaning (addressing missing values and inconsistencies), structuring (organizing data into consistent formats and standardizing nomenclature), and formatting (ensuring correct data types and schema conformance).

4.1.4 Test Scenario Specification for Evaluation

The evaluation framework was built upon a specific, curated set of controlled scenarios. The final dataset for the experiments was composed of data on all Dota 2 heroes and a custom set of test scenarios focusing on 1v1 and 2v2 configurations. This dataset included 109 unique 1v1 matchups and 100 unique 2v2 matchups. For each scenario, a list of **Known Good Heroes** was defined, representing expert-curated counter-selections chosen based on a combination of high win rates and specific mechanical advantages as derived from the collected data.

4.2 Language Model Configuration

In line with the model selection rationale, this study utilized the **Meta-Llama-3.1-8B-Instruct** model [7], an 8-billion parameter LLM chosen as it is designed for instruction-following tasks [7]. Unless otherwise specified, all experiments were conducted using a standard inference configuration with a temperature of `0.6`, a `top_p` of `0.9`, and a maximum generation length of `1500` tokens to balance creativity with coherence.

4.3 Prompt Engineering Experiments

This section details the execution of the comparative analysis between Zero-Shot, Few-Shot, and Chain-of-Thought prompting, as conceptually defined in Section 3.2.2.

4.3.1 Prompt Design and Examples

The three prompting strategies were implemented using a standardized System Prompt that defined the model's persona, outlined analysis considerations, and enforced a structured JSON output. For the Few-Shot prompts, representative examples were included in the prompt context to guide the model's reasoning style, as shown in Listing 4.1.

```
{
    "user": "The enemy team has Storm Spirit and Luna, what should I
       pick?",
    "assistant": {
      "my_heroes": ["Anti-Mage", "Disruptor"],
      "enemy_heroes": ["Storm Spirit", "Luna"],
      "reasoning": "Anti-Mage's Mana Break depletes Storm Spirit's mana
          pool... Disruptor's Static Storm silences both heroes..."
    }
}
```

Listing 4.1: Example of a Few-Shot Prompt Entry (JSON format).

The Chain-of-Thought (CoT) prompts included a core instruction designed to invoke a step-by-step reasoning process before the final answer, detailed in Listing 4.2.

```
Let's think through this step by step:

1. First, analyze each enemy hero's strengths and weaknesses.
2. Consider which hero abilities would be effective against these
   strengths.
3. Identify heroes with those counter abilities.
4. Evaluate how well these potential counter heroes work together.
5. Select the {num_heroes_to_pick} best counter {hero_term} based on
   this analysis.
```

Listing 4.2: Core Instruction for Chain-of-Thought Prompting.

4.3.2 Experimental Procedure for Prompt Engineering

A systematic evaluation pipeline was executed to compare the prompting techniques:

1. The Dota 2 hero data, Draft Evaluator, and a client interface to the LLM were initialized.
2. The test scenarios were loaded and grouped by type (e.g., 1v1, 2v2).
3. For each scenario, each of the three prompting techniques was applied by sending the constructed prompt to the LLM.
4. The raw response was received and parsed, with error handling to manage malformed outputs.
5. The suggested heroes were evaluated using the Draft Evaluator, with the known good heroes also evaluated to provide a benchmark.
6. Results, including suggestions, reasoning, scores, and errors, were recorded.
7. Finally, summary statistics and visualizations were generated to compare performance across techniques.

4.4 Prompt Tuning Experiments

To optimize the prompts, an experiment was designed to systematically generate and evaluate different prompt templates. This process used the same Llama 3 model and a consistent inference configuration, partitioning the test scenarios into a 90% training set and a 10% validation set.

4.4.1 Prompt Component Library and Template Generation

A repository of granular text snippets—the prompt component library—was utilized, which included different phrasing for system instructions, user questions, few-shot examples, and reasoning frameworks. An iterative function programmatically generated new templates by randomly combining, replacing, or reordering these components.

4.4.2 Experimental Procedure for Prompt Tuning

The tuning procedure was executed as follows:

1. The scenario dataset was loaded and split into training and validation sets.
2. A set of initial baseline prompt templates were evaluated on the *training set*.
3. An iterative tuning loop was run for a predefined number of iterations, generating and evaluating a new template against the *training set* in each pass. The average score was recorded, and the best-performing template was tracked.
4. The final "best template" from the training phase was then evaluated on the held-out *validation set*.
5. Validation scores were compared against the baseline scores to measure improvement.

4.5 Supervised Fine-Tuning (SFT) Experiments

This section details the practical implementation of the SFT phase, introduced conceptually in Section 3.3.2.

4.5.1 Fine-Tuning Dataset and Configuration

The fine-tuning process utilized a collaborator's dataset, which was supplemented with data from Dota 2 patch 7.37, including updated hero stats and new mechanics. The dataset consisted of natural language queries paired with structured data outputs containing counter-picks and reasoning. To make training feasible, the QLoRA PEFT strategy was implemented, configuring the model for 4-bit quantization and specifying LoRA parameters for rank, alpha scaling, and dropout, targeting standard transformer modules. Training was managed using the Hugging Face Transformers and TRL libraries, with hyperparameters set for a multi-epoch run using a cosine learning rate scheduler and FP16 mixed precision.

4.5.2 Fine-Tuning and Evaluation Procedure

The end-to-end fine-tuning was implemented using a dedicated pipeline that loaded the dataset, initialized the quantized base model and tokenizer, applied the LoRA configuration, and invoked the TRL trainer to run the SFT process. Upon completion, the trained PEFT adapters were saved to disk.

To assess the impact of fine-tuning, the specialized model was compared against the original base model. Both models were evaluated on the curated test scenarios using a standardized few-shot prompt to ensure a fair comparison. The suggestions from each model were parsed and scored with the Draft Evaluator, and the final scores were recorded for analysis.

4.6 Evaluation Metric Implementation

This section details the technical implementation of the custom Draft Evaluator, which was conceptually described in Section 3.4.

4.6.1 Technical Design and Scoring Components

The Draft Evaluator is a custom software component designed to quantify the strategic effectiveness of a draft. It assesses several components using domain-specific Dota 2 data, including a **Matchup Winrate Score**, a statistical **Disadvantage Score**, an overall **Winrate Comparison**, and a **Grade Score** based on hero tiers.

4.6.2 Final Score Calculation and Weights

The normalized component scores are aggregated into a single Final Score (F_s). This score is calculated as the weighted sum of the four previously mentioned metrics (M_i), where each metric is assigned a specific weight (W_i) designed to prioritize direct matchup performance. The formula is defined as:

$$F_s = \sum_{i=1}^{4} M_i \cdot W_i$$

The four metrics and their corresponding weights are defined as:

- **Winrate Per Hero (M_1)** with weight $W_1 = 1.0$.
- **Disadvantage (M_2)** with weight $W_2 = 0.6$.
- **Winrate Comparison (M_3)** with weight $W_3 = 0.2$.
- **Grade Score (M_4)** with weight $W_4 = 0.1$.

4.6.3 Metric Validation and Scope of Application

An early step was to validate the Draft Evaluator's efficacy. It was tested against a large dataset of 22,300 historical 5v5 matches, where it achieved an accuracy of 52.8% in predicting match outcomes from the draft alone. However, further analysis revealed a weak correlation (r=0.064) between the metric's score and the actual outcome. This finding led to the conclusion that the Draft Evaluator is not an accurate absolute predictor of match outcomes. The complexity of a full Dota 2 match, which includes in-game execution and dynamic strategy, cannot be reliably determined by a static pre-game evaluation alone, a challenge consistent with broader discussions of evaluation in game AI [21].

Given this limitation, the research strategy was adapted. The Draft Evaluator's primary utility was redefined as a **consistent and reproducible comparative benchmark**. Its purpose is not to predict winners, but to serve as a stable benchmark for determining which LLM approach produces a relatively more strategically sound draft according to a fixed, domain-specific logic. Consequently, the research focus was shifted to more controlled 1v1 and 2v2 scenarios. In these curated tests, the metric's alignment improved, correctly favoring the "known good hero" draft 60.25% of the time in 1v1 scenarios and 58.3% in 2v2 scenarios, establishing the baseline for the comparative experiments.

4.6.4 Hero Match Percentage Calculation

To measure the direct similarity between the model's suggestions and the expert-defined answers, the hero match percentage was calculated using the Jaccard similarity coefficient. This metric quantifies the overlap between the set of heroes suggested by the model and the set of known good heroes for a given scenario.

Chapter 5

Results

This chapter presents the empirical results of the experiments outlined in the preceding methodology chapter. The findings are presented in a logical sequence, beginning with the construction and validation of the foundational components: the dataset and the custom evaluation metric. Subsequent sections detail the outcomes of the prompt engineering and fine-tuning experiments, allowing for a direct comparison of the various techniques investigated.

5.1 Foundational Framework: Data and Metric Validation

A robust evaluation framework is a requirement for an empirical study. This section details the outcomes of developing two primary components: the domain-specific dataset used to inform the model, and the custom evaluation metric, the Draft Evaluator, used to quantify strategic reasoning.

5.1.1 Synthetic Data Generation for Enhanced Domain Knowledge

To ensure the LLM possessed a granular and structured understanding of game mechanics beyond what could be gleaned from statistical matchup data alone, a supplementary instruction-answer dataset was created. Data on hero attributes, abilities, and game mechanics was programmatically scraped from the Dota 2 Fandom wiki. This raw data was then processed using large language models (GPT-3.5 Turbo and Gemini Pro) to generate a structured dataset of questions and corresponding answers. This process yielded a dataset that encapsulates explicit domain knowledge in a conversational format. For example, it contains entries that detail specific hero attribute gains, the mechanics of ability upgrades, and the function of unique hero abilities, as shown in Listing 5.1 for the hero Abaddon.

```
instruction: What are Abaddon's base attributes for Strength, Agility,
             and Intelligence, and their respective attribute gains?
answer:      - Strength: 22 + 2.2
             - Agility: 23 + 1.3
             - Intelligence: 19 + 1.6
```

Listing 5.1: Example of a synthetically generated instruction-answer pair.

This synthetically generated dataset, alongside an existing dataset enhanced with data from the recent Patch 7.37, forms the knowledge base intended to ground the LLM's strategic reasoning in factual, up-to-date game information.

5.1.2 The Draft Evaluator: Methodological Correction and Validation

The main tool for quantitative analysis in this thesis is the Draft Evaluator, a custom metric designed to score the strategic quality of a hero draft based on a weighted sum of four compo-

nents, as detailed in Section 4.6. An early phase of the research involved its technical validation and refinement. An initial implementation flaw that could cause the final score to fall outside its intended [-1, 1] bounds was resolved by normalizing the component weights to sum to 1.0 while preserving their relative importance. This correction established the mathematical integrity for all subsequent experiments.

With the metric corrected, a large-scale validation was conducted on 22,300 historical matches to assess its ability to predict outcomes based solely on the pre-game draft. The primary finding was that the metric achieved an *overall accuracy of 52.8%*. While this is marginally above a random baseline, a deeper analysis revealed a weak correlation ($r = 0.064$) between the metric's draft score and the actual match outcome. This result demonstrates that the Draft Evaluator, in its current form, is *not an absolute predictor of match outcomes*, as the complex strategy of Dota 2 cannot be reliably predicted from the draft alone. However, this finding does not invalidate its utility. The metric's primary function is to serve as a *consistent and reproducible benchmark for comparison*. By applying the same complex, domain-specific logic uniformly to the outputs of different LLM techniques, the Draft Evaluator can reliably determine which approach produces a draft with a higher score according to the metric. Therefore, for the remainder of this thesis, the Draft Evaluator is used not to claim predictive power over game results, but as a tool for validation to facilitate the comparative analysis of LLM-generated strategies.

5.2 Prompt Engineering Performance Analysis

This section presents the results from the experimental run comparing Zero-shot, Few-shot, and Chain-of-Thought (CoT) prompting, as described in Section 4.3. The analysis covers 109 unique 1v1 scenarios and 100 unique 2v2 scenarios. These findings should be contextualized within the capabilities of the evaluation tool; the performance scores reflect the model's ability to align with the Draft Evaluator metric, which itself has a demonstrated 52.8% predictive accuracy on real-world match outcomes.

5.2.1 Performance in 1v1 Scenarios

In the simpler 1v1 matchup scenarios, the three prompting techniques showed clear differences in performance, reliability, and cost. As shown in Figure 5.1, all techniques scored lower than the benchmark set by the "Known Good" heroes. The Zero-shot technique achieved the highest average score (0.067), while Few-shot registered a negative average score (-0.030). This apparent discrepancy is clarified when considering reliability and consistency. Figure 5.2 shows that Chain-of-Thought secured the most "wins" (37), suggesting it was more frequently the top performer. Furthermore, Figure 5.3 highlights Zero-shot's high error rate: a 55.0% error rate. In contrast, Few-shot was the most reliable with an 18.3% error rate, followed closely by CoT at 21.1%. This context clarifies that Zero-shot's high average score is based on a small number of valid responses, making it an impractical strategy.

5.2.2 Performance in 2v2 Scenarios: The Impact of Complexity

When strategic complexity increased in 2v2 scenarios, the strengths and weaknesses of each technique became more distinct. In this more complex setting, Chain-of-Thought achieved the highest number of "wins". As seen in Figure 5.5, CoT's lead in "wins" increased from 37 (in 1v1) to 49, indicating its structured reasoning is more effective at handling synergistic considerations. The reliability of the techniques diverged. As seen in Figure 5.6, Zero-shot became non-functional, with an 80.0% error rate. In contrast, both Chain-of-Thought and Few-shot demonstrated high reliability, with error rates decreasing to 5.0% and 6.0%, respectively. This difference in outcomes demonstrates that structured prompting is required for navigating complex strategic tasks.

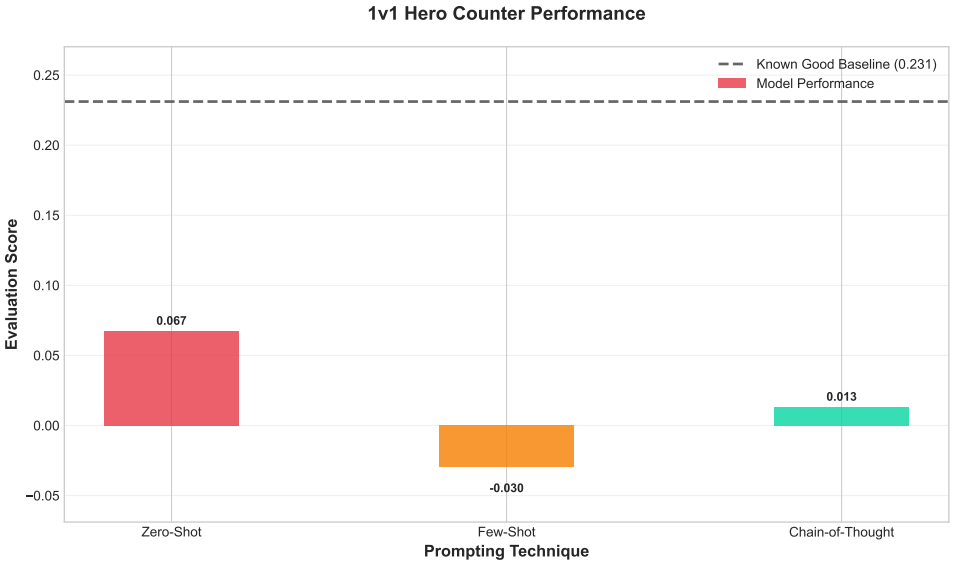

Figure 5.1: Average evaluation scores for 1v1 scenarios. All techniques performed significantly below the Known Good Baseline (0.231).

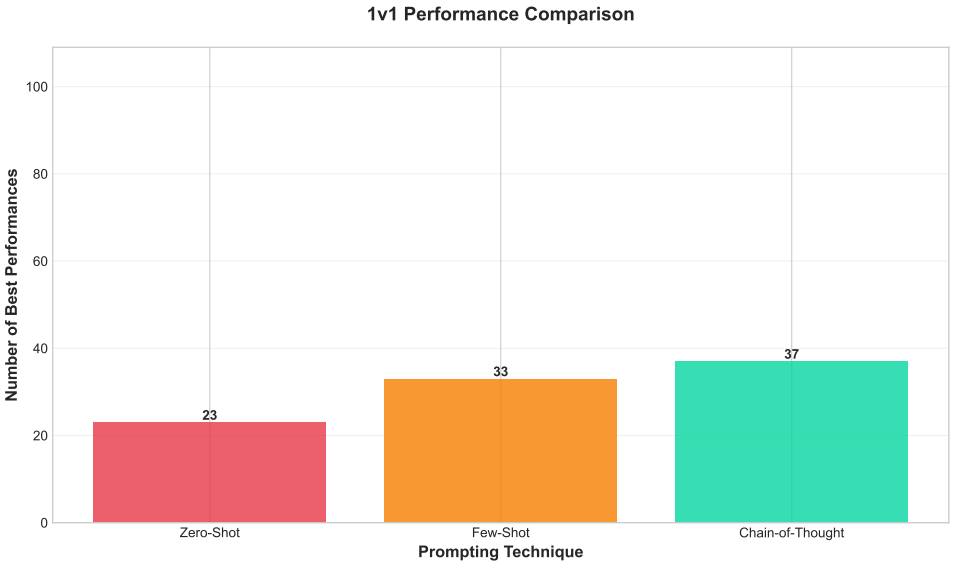

Figure 5.2: Number of best performances ("wins") in 1v1 scenarios. Chain-of-Thought secured the most wins, despite not having the highest average score.

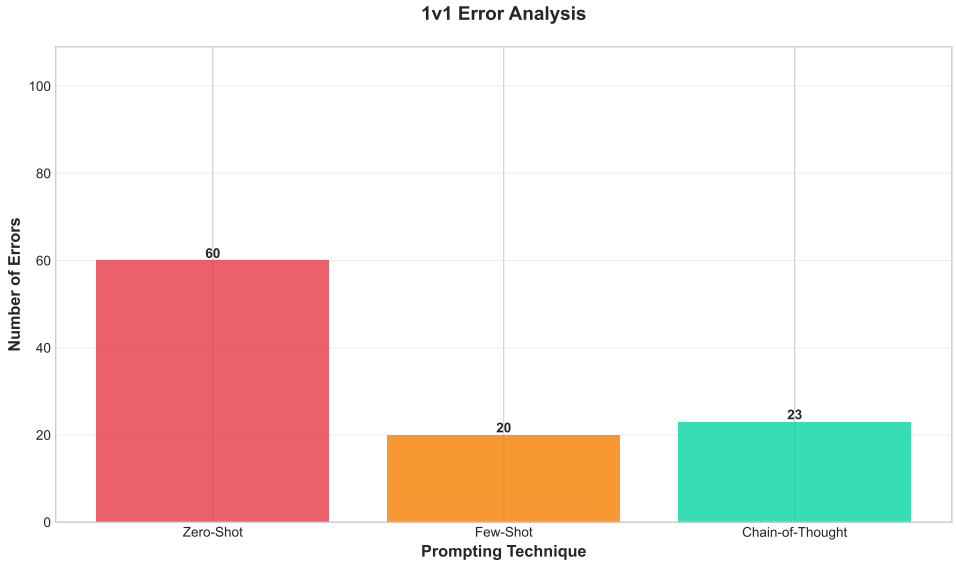

Figure 5.3: Number of errors in 1v1 scenarios. Zero-shot's high error rate undermines its scoring performance.

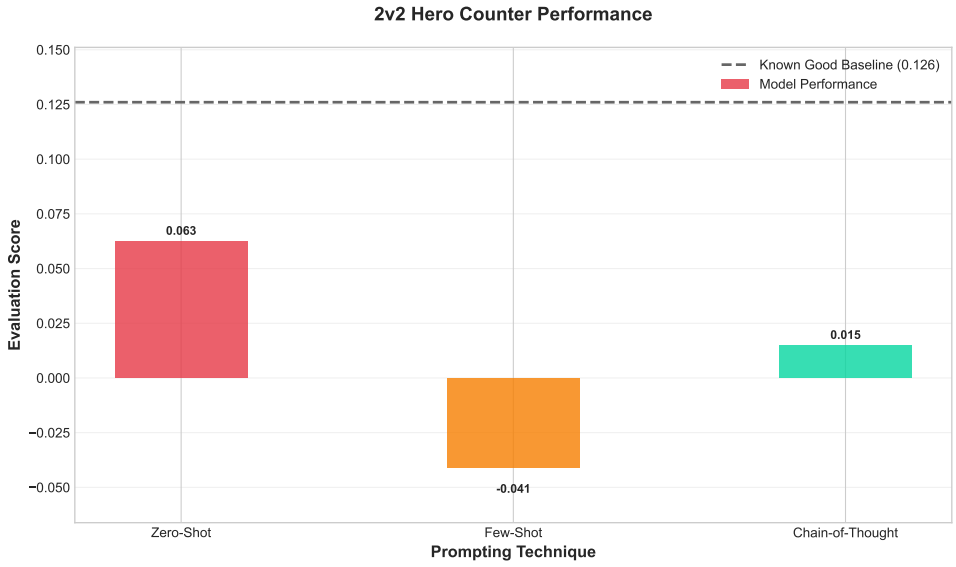

Figure 5.4: Average evaluation scores for 2v2 scenarios. The performance gap between techniques and the Known Good Baseline (0.126) remains large.

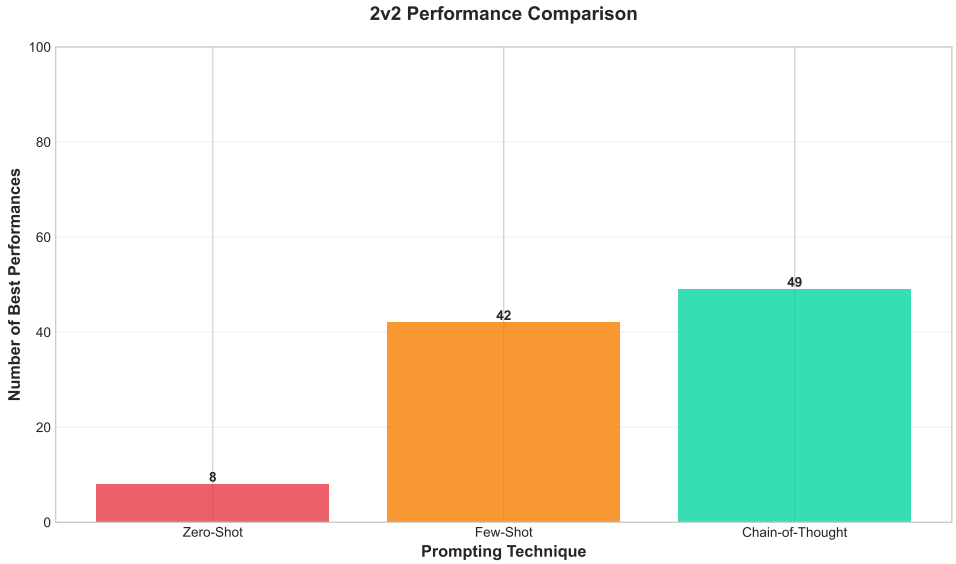

Figure 5.5: Number of best performances ("wins") in 2v2 scenarios. CoT's lead in performance quality grew with complexity, while Zero-shot's fell sharply.

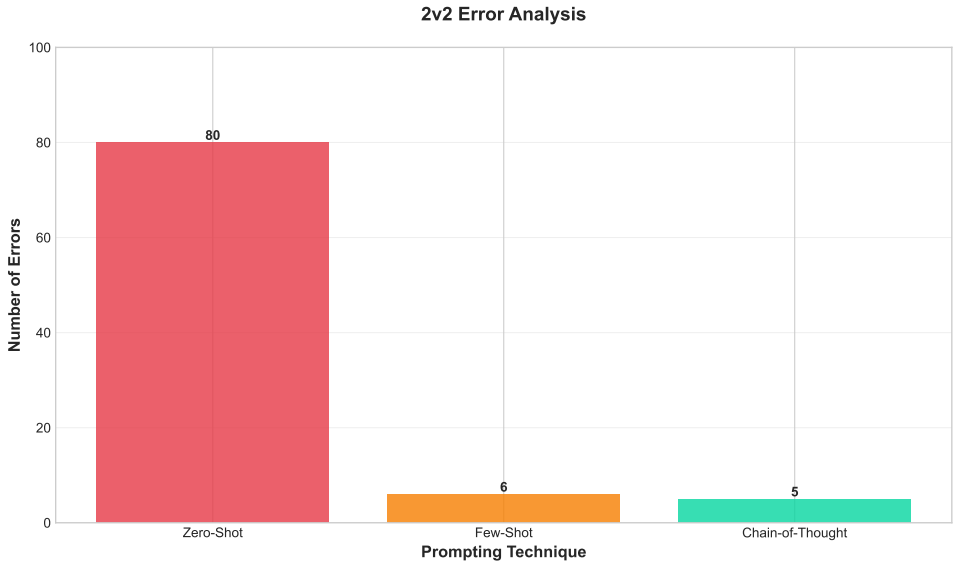

Figure 5.6: Number of errors in 2v2 scenarios. The error rate for structured prompts (Few-shot, CoT) decreased dramatically, while Zero-shot's failure rate surged.

5.2.3 Response Characteristics and Computational Cost

The higher number of "wins" from Chain-of-Thought comes at a measurable cost. The verbosity of its reasoning process results in responses that were approximately ten times longer, as shown in Figures 5.7 and 5.8, than the concise responses from Few-shot. This increased length has a direct impact on generation time. As illustrated in Figures 5.9 and 5.10, the Few-shot technique averaged 2.5-3.4 seconds per response, while CoT was 6-8 times slower, requiring 19.3-23.2 seconds. This presents a direct trade-off between the highest number of "wins" and practical usability in time-sensitive applications.

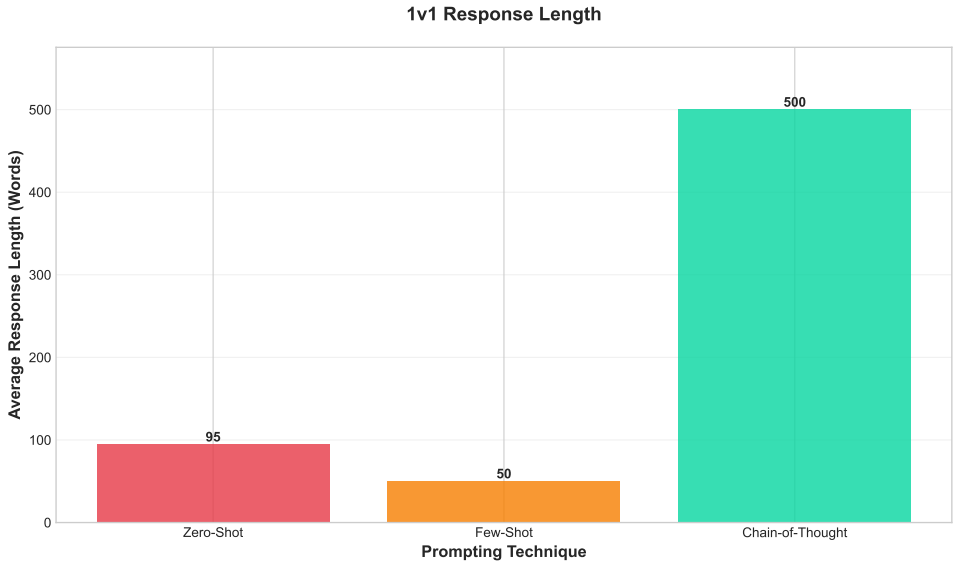

Figure 5.7: Average response length in words for 1v1 scenarios. CoT is significantly more verbose than other methods.

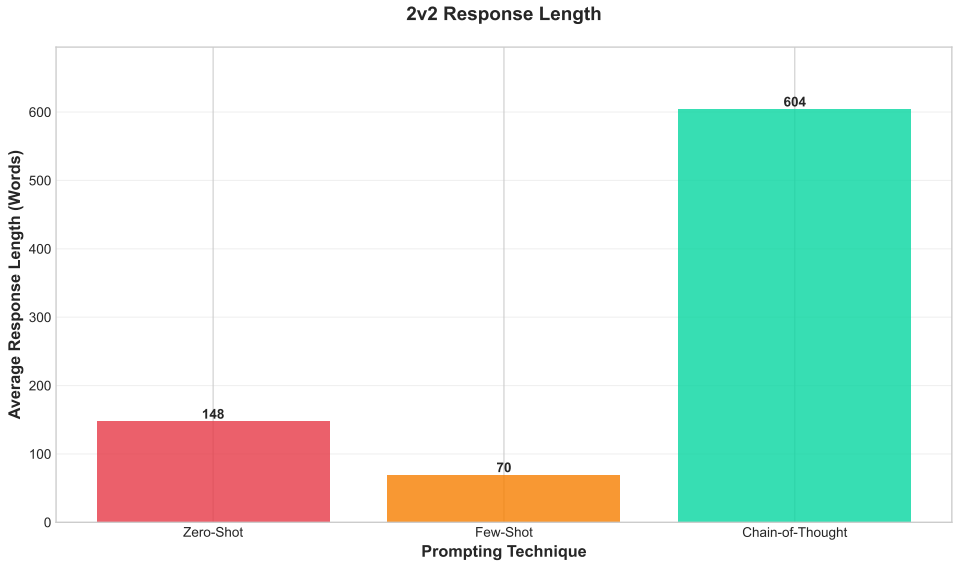

Figure 5.8: Average response length in words for 2v2 scenarios. The length of CoT responses grows with complexity.

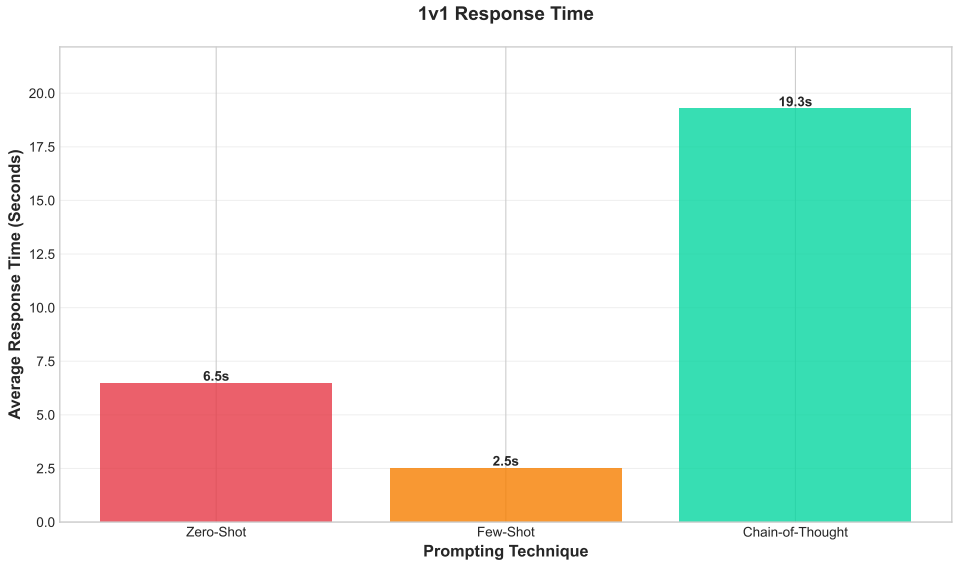

Figure 5.9: Average response time in seconds for 1v1 scenarios. The time cost of CoT is substantial.

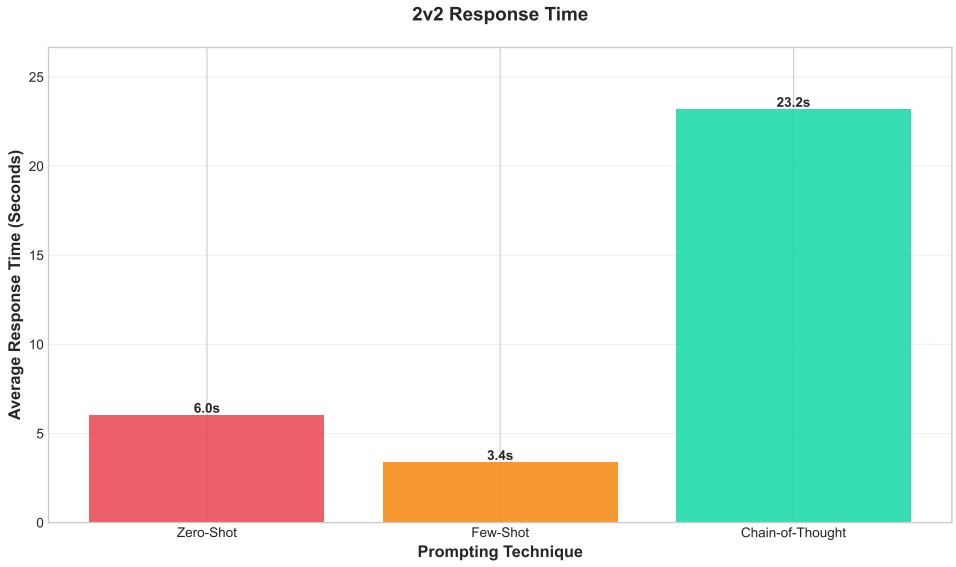

Figure 5.10: Average response time in seconds for 2v2 scenarios, highlighting the efficiency of Few-shot prompting.

5.2.4 Comparative Analysis and Discussion

The experimental results show clear trade-offs between prompting strategies. **Chain-of-Thought** consistently produced the highest number of the highest-scoring recommendations ("wins"), and its performance scaled positively with task complexity, making it the most effective technique for recommendation quality, albeit at a high computational cost. In contrast, the **Few-Shot** technique demonstrated high reliability and processing speed. Its low error rate and rapid response time make it a suitable candidate for production systems where speed and consistency are important, even if it does not always produce the highest-scoring recommendation. Finally, the **Zero-shot** technique is not a viable strategy due to its high error rates. In conclusion, these findings show that structured prompting is a required component for this domain, and the choice between techniques is a trade-off between the recommendation quality of CoT and the practical efficiency of Few-shot prompting.

5.3 Evaluating Prompt Tuning via Ensemble Analysis

To further refine the understanding of prompt engineering, a second experiment, as described in Section 4.4, was conducted to evaluate an ensemble of six distinct prompt templates—two for each category of Zero-shot, Few-shot, and CoT—across 200 scenarios.

5.3.1 Overall Performance and Template Ranking

The analysis reveals that template choice has a measurable impact on performance. The Few-shot category, on average, outperformed the others with a mean score of 0.128, as shown in Figure 5.11. A detailed ranking of the individual templates, presented in Figure 5.12, identifies the highest-performing template: the basic few-shot template achieved the highest mean evaluation score of 0.142. Its more detailed counterpart was the second-best performer, demonstrating that specific phrasing and structure matter even within the best-performing category.

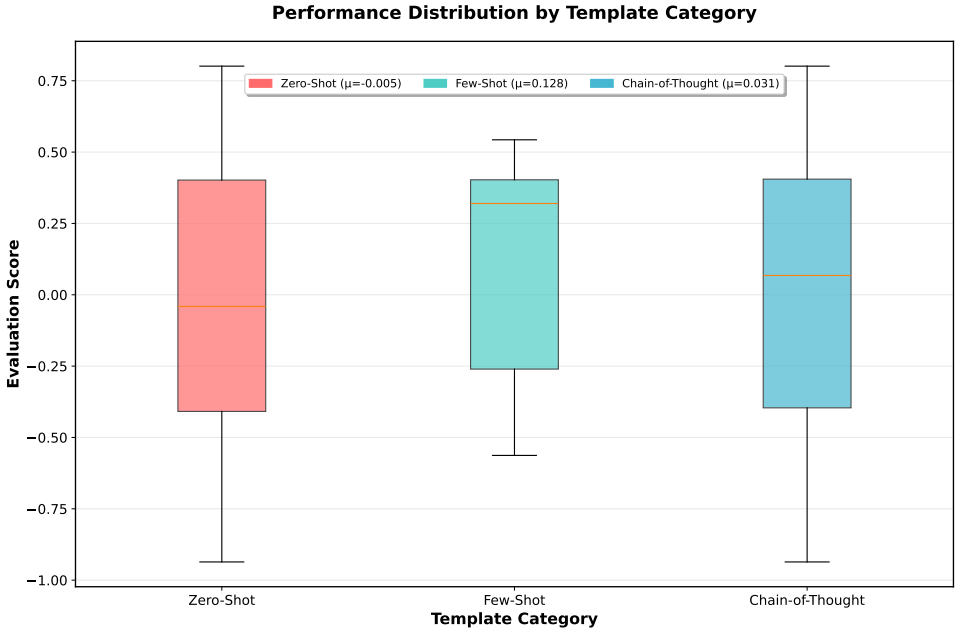

Figure 5.11: Performance distribution by template category. The Few-shot category shows the highest mean evaluation score.

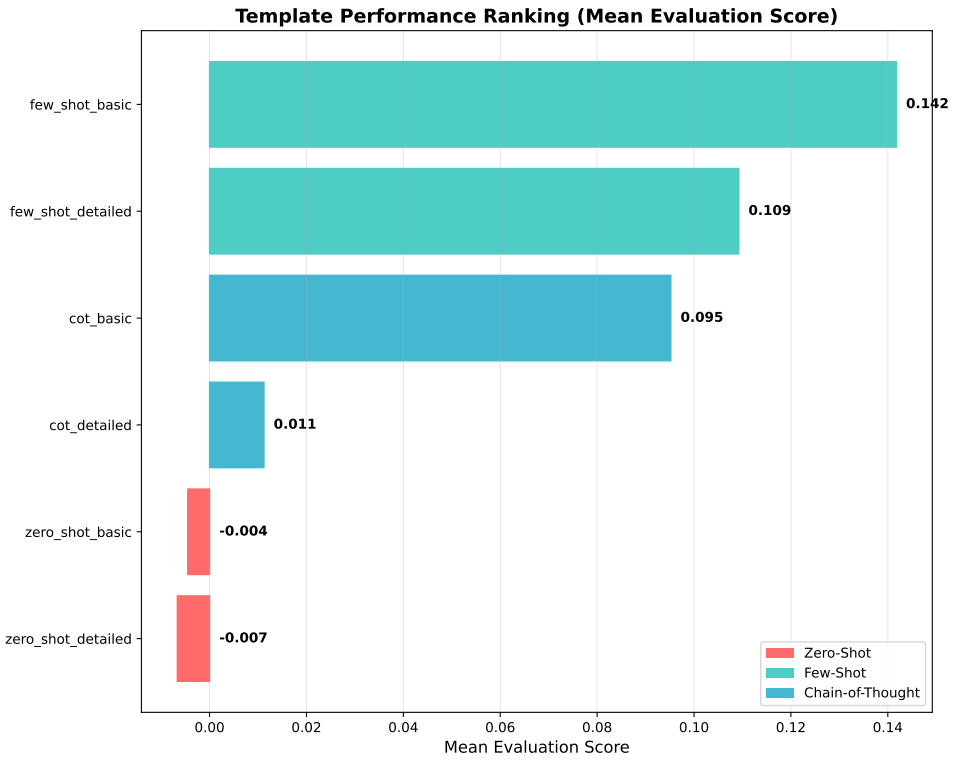

Figure 5.12: Template performance ranking by mean evaluation score. The basic few-shot template was the top-performing prompt across all experiments.

5.3.2 Impact of Complexity and Metric Validation

The ensemble experiment revealed a nuanced trend regarding complexity. While the overall mean score between 1v1 (0.016) and 2v2 (0.024) scenarios was similar, this masked internal variations: the Few-shot category's score degraded by 47.3% with complexity, whereas the CoT category's performance improved. This experiment also provided further validation for the Draft Evaluator. Although the model's suggestions consistently scored lower than the expert-curated baseline (Figure 5.14), a statistically significant positive correlation was found between the model's scores and the known good scores (Pearson $r = 0.380$, $p < 0.001$). This indicates that the metric correctly identifies that suggestions considered superior by the baseline receive higher scores.

5.3.3 Discussion and Implications

The prompt tuning experiment shows that prompt design is not monolithic; variations in structure lead to measurable performance differences. The identification of the basic few-shot template as the top performer provides a concrete, optimized prompt for future use. The experiment also reinforces the conclusions regarding complexity, with CoT templates proving more suitable for complex reasoning tasks. This analysis validates prompt tuning as a method for optimizing LLM performance for specialized tasks.

5.4 The Impact of Supervised Fine-Tuning

Following the analysis of prompt engineering, the next phase of research investigated the impact of domain-specific supervised fine-tuning, as detailed in Section 4.5. The experiment aimed to adapt the base Llama 3 model to the Dota 2 domain, with the hypothesis that this specialization

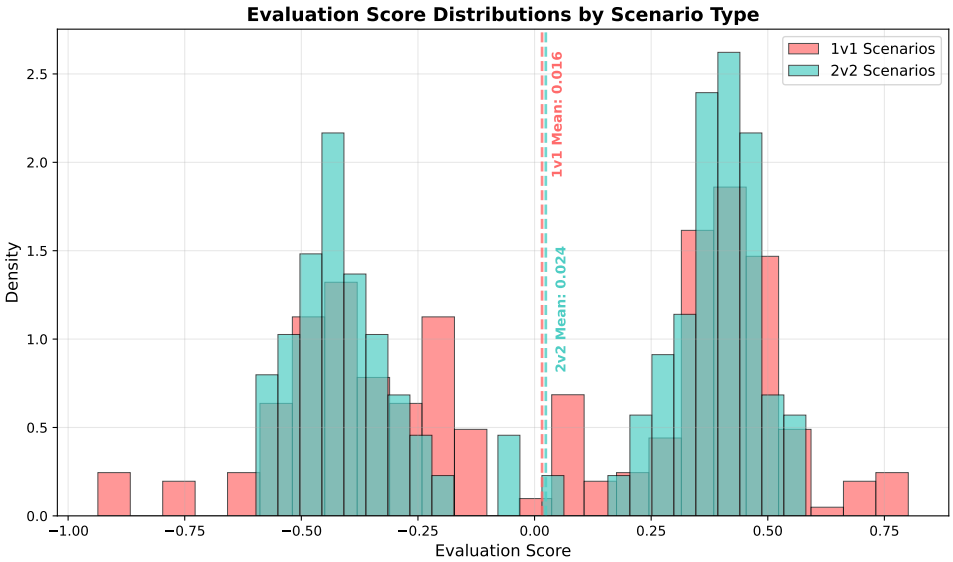

Figure 5.13: Evaluation score distributions by scenario type. The mean scores for 1v1 and 2v2 scenarios were closely aligned.

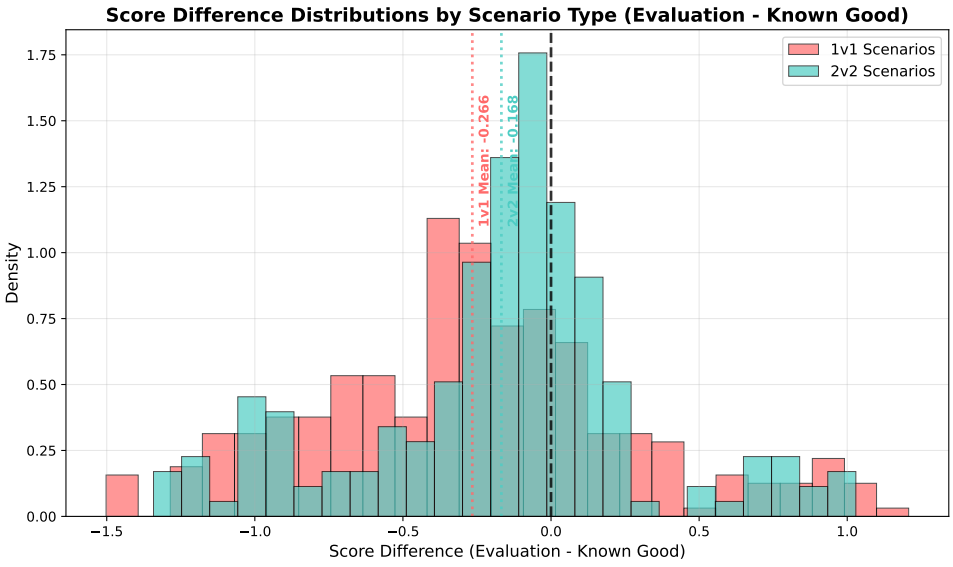

Figure 5.14: Score difference distributions (Evaluation - Known Good). The negative mean values indicate model performance is consistently below the expert baseline.

would enhance its strategic recommendation capabilities.

5.4.1 Quantitative Performance Comparison

The results of the fine-tuning experiment were contrary to the initial hypothesis. Instead of improving, the fine-tuned model exhibited a consistent degradation in performance compared to the original baseline model across both 1v1 and 2v2 scenarios, as shown in Figure 5.15. In 1v1 scenarios, the baseline model achieved a score of 0.458, whereas the fine-tuned model scored only 0.238. The trend continued in 2v2 scenarios, where the baseline scored 0.269 and the fine-tuned model's score decreased to 0.089. In both cases, the fine-tuning process resulted in a model that scored lower at generating strategically sound recommendations according to the Draft Evaluator.

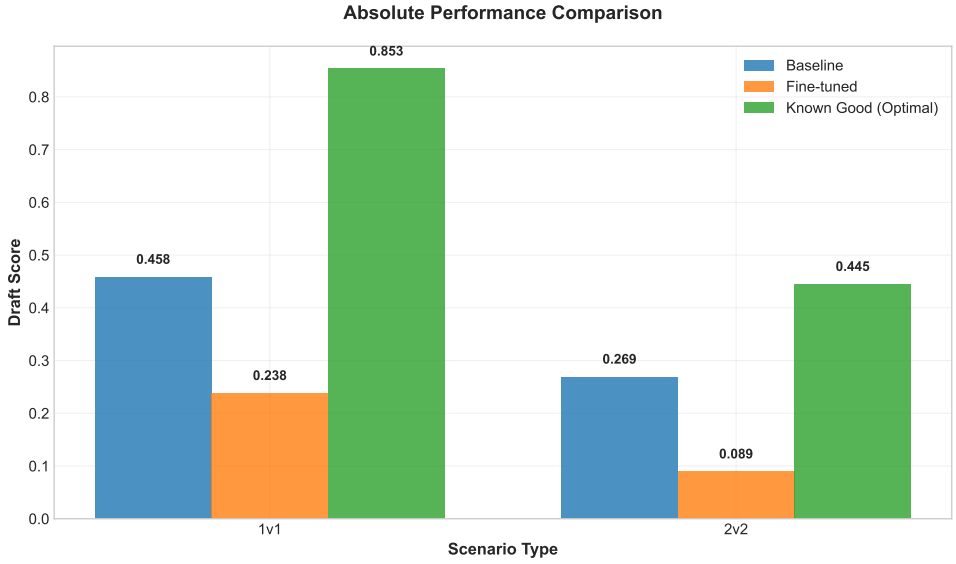

Figure 5.15: Absolute performance comparison between the baseline model, the fine-tuned model, and the optimal "Known Good" hero selections. The fine-tuned model consistently scored lower than the baseline.

5.4.2 Analysis of Performance Degradation

To quantify this negative impact, the percentage drop in performance from the baseline to the fine-tuned model was calculated. The analysis revealed not only that the degradation was substantial, but that it was exacerbated by task complexity. As visualized in Figure 5.16, the fine-tuning process led to an average performance drop of 33.1%. This degradation was unequal: the simpler 1v1 scenarios saw a performance drop of 25.8%, while the more complex 2v2 scenarios experienced a greater degradation of 40.4%. This suggests that the fine-tuning was detrimental to the model's ability to handle multi-hero synergistic reasoning.

5.4.3 Discussion of Negative Results

These negative results provide insights into the challenges of domain adaptation for LLMs. The performance regression is likely due to a combination of factors. **Catastrophic forgetting** is a likely cause; in specializing on the relatively small and narrow dataset, the model may have lost the generalized reasoning capabilities from its pre-training, replacing them with overfitting to the training data, resulting in poor generalization. Additionally, the outcome may stem from *data and hyperparameter limitations*, as the dataset might have lacked sufficient diversity and

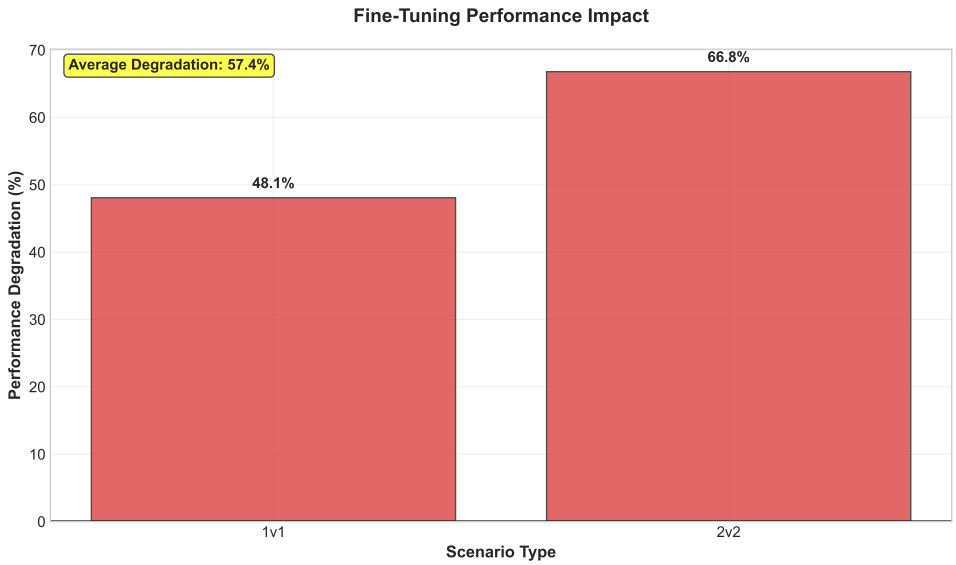

Figure 5.16: Performance degradation after fine-tuning. This chart visualizes the percentage drop in score from the baseline to the fine-tuned model, showing a greater negative impact in more complex scenarios.

the training configuration may have been suboptimal. The relationship between task complexity and performance degradation is the most notable finding. The fine-tuning experiment, therefore, highlights the risks of direct domain specialization. These findings steer future research away from simple fine-tuning and toward more nuanced approaches like advanced prompt engineering and Retrieval-Augmented Generation (RAG), which can add domain knowledge without risking the base model's foundational reasoning abilities.

Chapter 6

Discussion

6.1 Limitations of the Study

While this thesis provides insights, it is important to acknowledge its limitations. First, the evaluation was primarily conducted on 1v1 and 2v2 scenarios. Although this was a necessary simplification to establish a controlled experimental environment, it does not capture the full complexity of a 5v5 draft. Second, the research was limited to a single LLM architecture, Llama 3 (8B). While this model is capable, results may differ with other architectures or model sizes. Third, the supervised fine-tuning experiment explored only one configuration of data and hyperparameters; a more exhaustive search may have yielded different results. Finally, the Draft Evaluator metric, while effective for comparison, is itself an approximation of strategic value and does not account for all the unquantifiable factors that influence a real Dota 2 match, such as player skill and in-game execution.

6.2 Answer to Research Questions

This section provides direct responses to the primary research question and its sub-questions, synthesizing the findings from the experimental evaluation.

Primary Research Question

Can a large language model be effectively used during the hero selection phase in Dota 2?

Answer: The findings confirm that a large language model can serve as a tool for hero selection in Dota 2, provided that appropriate prompting strategies are applied. Among the tested methods, Few-shot prompting was the most suitable at generating fast and consistent responses. It is important to note, however, that even the best-performing model did not match the accuracy of an expert baseline. Moreover, the study found that supervised fine-tuning (SFT) had a negative effect on the model's capabilities in this context. The conclusion is that structured prompting is a viable method, whereas simple fine-tuning is not a suitable path for performance enhancement in this specific application.

Sub-Research Question 1

To what extent can prompt engineering strategies influence an LLM's ability to provide accurate hero recommendations in Dota 2?

Answer: Prompt engineering strategies influence an LLM's performance in Dota 2 hero recommendations.

- Chain-of-Thought (CoT) prompting consistently produced the highest number of successful recommendations ("wins") and its performance scaled positively with increasing task

complexity (e.g., 2v2 scenarios), which was the most effective technique for recommendation quality. CoT also showed high reliability with low error rates (21.1% in 1v1, 5.0% in 2v2).

- Prompt tuning further reinforced the importance of prompt design, with the basic few-shot template emerging as the best-performing template across experiments, demonstrating that specific phrasing and structure within prompts can yield observable performance differences. Overall, structured prompting is necessary for this domain, requiring a trade-off between the higher quality of CoT recommendations and Few-shot's practical efficiency.

Sub-Research Question 2

How effectively can domain-specific knowledge, such as hero attributes and patch changes, be integrated into an LLM to provide better hero recommendations?

Answer: Domain knowledge was incorporated through two primary methods:

- Synthetic Data Generation: A supplementary instruction-answer dataset was programmatically created from Dota 2 wiki data and processed by other LLMs to provide structured data on hero attributes, abilities, and game mechanics. This dataset, along with a collaborator dataset enhanced with Patch 7.37 data, formed the knowledge base for the LLM.

- Supervised Fine-Tuning (SFT): Contrary to the hypothesis, the SFT process to adapt the Llama 3 model to the Dota 2 domain, particularly incorporating recent patch knowledge, resulted in a consistent degradation in performance. The fine-tuned model scored worse than the baseline, with an average performance drop of 33.1%, which increased with task complexity (40.4% drop in 2v2 scenarios). This negative outcome is attributed to factors like "catastrophic forgetting" of generalized reasoning capabilities and potential limitations in the fine-tuning dataset or hyperparameters. These findings highlight the risks associated with this method of domain specialization through simple fine-tuning and suggest that alternative approaches, such as advanced prompt engineering and Retrieval-Augmented Generation (RAG), may be more suitable for integrating domain knowledge without compromising the base model's generalized reasoning abilities.

Sub-Research Question 3

What are the key limitations and challenges of using large language models for real-time strategic tasks, such as hero selection in Dota 2, particularly concerning response latency and computational cost?

Answer: The research identified several limitations and challenges for using LLMs in real-time Dota 2 hero selection:

- **Response Latency:** While Few-shot prompting offered low response times (2.5-3.4 seconds), Chain-of-Thought (CoT), despite its superior quality, was slower (19.3-23.2 seconds) due to its verbosity. This presents a direct trade-off between recommendation quality and practical usability in time-sensitive, real-time applications like hero drafting.

- **Computational Requirements:** Although a smaller 8-billion parameter Llama 3 model was selected and Parameter-Efficient Fine-Tuning (PEFT) with QLoRA was employed to make fine-tuning feasible on constrained hardware, computational resources remain a challenge for deploying and running such models for real-time decision-making. The increased response length of CoT also contributes to higher computational cost.

- **Data Availability and Generalization:** Challenges include acquiring sufficient and diverse domain-specific data, especially for nuanced scenarios. The unexpected performance

degradation from fine-tuning also suggests difficulties in effectively generalizing domain knowledge without compromising the model's generalized capabilities.

- **Evaluation Complexity:** The Draft Evaluator, while a consistent comparative benchmark, was found not to be a reliable absolute predictor of match outcomes in complex 5v5 scenarios, indicating the inherent difficulty in quantitatively assessing strategic value in a game with multi-faceted dynamics. The study therefore focused on more controlled 1v1 and 2v2 scenarios for evaluation.

Chapter 7

Conclusion and Future Work

This thesis set out to investigate the application of a large language model, Llama 3, for the complex strategic task of hero selection in Dota 2. By systematically exploring a range of techniques from prompt engineering to supervised fine-tuning, the research aimed to evaluate the model's ability to provide accurate, reliable, and actionable recommendations. This final chapter synthesizes the findings of the study, directly addresses the research questions posed in the introduction, discusses the broader implications and limitations of the work, and outlines potential directions for future research.

The investigation began by establishing a data and evaluation foundation. A dataset was created by combining statistical data from public platforms with synthetically generated knowledge on game mechanics, which was further enhanced with data from the latest game patch. The core of the evaluation framework was a custom Draft Evaluator, a metric designed to score the strategic quality of a hero draft. While validation showed that this evaluator is not a reliable predictor of absolute match outcomes in a game as complex as Dota 2, it proved to be a consistent and effective benchmark for the comparative analysis of different LLM techniques.

The experimental results revealed a clear hierarchy of effectiveness among prompt engineering strategies. Unstructured, Zero-shot prompts were found to be unreliable and impractical due to excessive error rates. In contrast, structured methods were found to be more effective. Few-shot prompting was an efficient and reliable method, delivering consistent outputs with low latency. Chain-of-Thought (CoT) prompting, while computationally more expensive, consistently produced the highest quality strategic recommendations, particularly as task complexity increased from 1v1 to 2v2 scenarios. The prompt tuning experiments further demonstrated that even minor variations in prompt design can lead to observable performance differences.

Finally, the research explored domain specialization through supervised fine-tuning. Contrary to the initial hypothesis, the results showed that the fine-tuning process led to a consistent degradation in the model's performance. This outcome suggests that this form of specialization on a narrow dataset can be detrimental, potentially causing "catastrophic forgetting" of the model's generalized reasoning abilities.

7.1 Future Directions

The findings and limitations of this study open several avenues for future research. A logical next step would be to expand the evaluation framework to full 5v5 draft scenarios. This would require a more sophisticated evaluation metric, perhaps one that can assess team synergy and role composition more directly.

Another direction for research is to explore more advanced methods for integrating domain knowledge. Given the negative results from simple SFT, investigating Retrieval-Augmented Generation (RAG) is a priority. A RAG-based system could dynamically retrieve up-to-date information about hero statistics, patch changes, and specific matchup data at inference time,

potentially providing the benefits of domain knowledge without the risks of catastrophic forgetting.

Further research could also involve a broader comparative study of different LLM architectures and sizes to understand how these factors influence strategic reasoning capabilities. Finally, there is potential in developing interactive, human-in-the-loop systems. An AI assistant that can have a dialogue with the player, understand their playstyle preferences, and collaboratively build a draft strategy could be a future direction for AI-driven decision support in complex games like Dota 2[24, 25].

References

[1] Shyam Sudhakaran et al. *MarioGPT: Open-Ended Text2Level Generation through Large Language Models*. 2023. arXiv: 2302.05981 [cs.AI]. URL: https://arxiv.org/abs/2302.05981.

[2] Alicia Vidler and Toby Walsh. "Playing games with Large language models: Randomness and strategy". In: (2025). arXiv: 2503.02582 [cs.AI]. URL: https://arxiv.org/abs/2503.02582.

[3] Zhan Gong. "Dota 2 Hero Selection Analysis". PhD thesis. CUNY Academic Works, 2021. URL: https://academicworks.cuny.edu/gc_etds/4415.

[4] Cornell University INFO 2040 Students. *The Game Theory of Drafting Heroes in Dota 2*. Blog Post. URL: https://blogs.cornell.edu/info2040/2020/09/23/the-game-theory-of-drafting-heroes-in-dota-2/. Sept. 2020.

[5] Kevin Conley and Daniel Perry. "How Does He Saw Me ? A Recommendation Engine for Picking Heroes in Dota 2". In: 2013. URL: https://api.semanticscholar.org/CorpusID:15859842.

[6] Alec Sapienza, Nidia Rodriguez, and Grace Sotomayor. "Draft-Analysis of the Ancients: Predicting Draft Picks in DotA 2 using Machine Learning". In: *Proceedings of the Future Technologies Conference (FTC) 2022, Volume 1*. Springer, 2022.

[7] Meta AI. *Introducing Meta Llama 3: The most capable openly available LLM to date*. Blog Post. URL: https://ai.meta.com/blog/meta-llama-3/. Apr. 2024.

[8] Takeshi Kojima et al. "Large Language Models are Zero-Shot Reasoners". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2022, pp. 10037–10057. arXiv: 2205.11916. URL: https://arxiv.org/abs/2205.11916.

[9] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[10] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[11] Brian Lester, Rami Al-Rfou, and Noah Constant. *The Power of Scale for Parameter-Efficient Prompt Tuning*. 2021. arXiv: 2104.08691 [cs.CL]. URL: https://arxiv.org/abs/2104.08691.

[12] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations (ICLR)*. 2022. arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[13] Tim Dettmers et al. "QLoRA: Efficient Finetuning of Quantized LLMs". In: 2023. arXiv: 2305.14314 [cs.LG]. URL: https://arxiv.org/abs/2305.14314.

[14] Shangheng Du et al. "A Survey on the Optimization of Large Language Model-based Agents". In: (2025). arXiv: 2503.12434 [cs.AI]. URL: https://arxiv.org/abs/2503.12434.

[15] Zexuan Li et al. "Efficient LLM Serving on Hybrid Real-time and Best-effort Requests". In: (2025). arXiv: 2504.09590 [cs.DC]. URL: https://arxiv.org/abs/2504.09590.

[16] Xiao Shao et al. *SwarmBrain: Embodied agent for real-time strategy game StarCraft II via large language models*. 2024. arXiv: 2401.17749 [cs.AI]. URL: https://arxiv.org/abs/2401.17749.

[17] Timothée Anne et al. "Harnessing Language for Coordination: A Framework and Benchmark for LLM-Driven Multi-Agent Control". In: *IEEE Transactions on Games* (2025), pp. 1–25. ISSN: 2475-1510. DOI: 10.1109/tg.2025.3564042. URL: http://dx.doi.org/10.1109/TG.2025.3564042.

[18] Jialu Liu et al. "A Survey on Large Language Model based Game Agents". In: (2024). arXiv: 2404.02039 [cs.AI]. URL: https://arxiv.org/abs/2404.02039.

[19] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575 (2019), pp. 350–354. URL: https://api.semanticscholar.org/CorpusID:204972004.

[20] OpenAI. "Dota 2 with Large Scale Deep Reinforcement Learning". In: *arXiv preprint arXiv:1912.06680* (2019). URL: https://arxiv.org/abs/1912.06680.

[21] Roberto Gallotta et al. "Large Language Models and Games: A Survey and Roadmap". In: *IEEE Transactions on Games* (2024), pp. 1–18. ISSN: 2475-1510. DOI: 10.1109/tg.2024.3461510. URL: http://dx.doi.org/10.1109/TG.2024.3461510.

[22] Zhenyu Hou et al. "Which Prompting Technique Should I Use? An Empirical Investigation of Prompting Techniques for Software Engineering Tasks". In: *arXiv preprint arXiv:2506.05614* (2025). URL: https://arxiv.org/pdf/2506.05614.

[23] Yun Luo et al. "An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning". In: (2025). arXiv: 2308.08747 [cs.CL]. URL: https://arxiv.org/abs/2308.08747.

[24] Ruixiang Zhang et al. "Bring Your Own Knowledge: A Survey of Methods for LLM Knowledge Expansion". In: (2025). arXiv: 2502.12598 [cs.CL]. URL: https://arxiv.org/abs/2502.12598.

[25] Louis-Francois Bouchard and Louie Peters. *Building LLMs for Production*. Towards AI, Oct. 2024.

[26] The OpenDota Contributors. *OpenDota*. Website. URL: https://www.opendota.com. 2025.

[27] Elo Entertainment, LLC. *Dotabuff*. Website. URL: https://www.dotabuff.com. 2025.

[28] Dota 2 Fandom Wiki Contributors. *Dota 2 Wiki*. Website. URL: https://dota2.fandom.com/wiki/Dota_2_Wiki. 2025.

[29] Aiden07. *dota2_instruct_prompt*. Hugging Face Dataset. URL: https://huggingface.co/datasets/Aiden07/dota2_instruct_prompt. 2024.