Journal of Advances in Developmental Research (IJAIDR)

E-ISSN: 0976-4844 ● Website: www.ijaidr.com ● Email: editor@ijaidr.com

Fine-Tuning Large Language Models for Domain-Specialized Supply Chain Agents A Comprehensive Approach Using Supervised Learning on Enterprise Knowledge Bases

Sandeep Nutakki

Sr. AI Engineer
Seattle, Washington, USA
sandeep@auger.com

Abstract:
The increasing complexity of global supply chains demands intelligent systems capable of providing specialized guidance on logistics, procurement, inventory management, and operational optimization. This paper presents a comprehensive methodology for fine-tuning large language models (LLMs) to create domain-specialized supply chain agents. We developed a novel data pipeline that extracts, processes, and transforms 131 authoritative supply chain textbooks and professional resources into 161,741 high-quality question-answer training pairs using an automated bootstrapping approach with GPT-4o-mini. Using supervised fine-tuning (SFT) on GPT-4.1-mini via Microsoft Azure AI Foundry, we achieved strong training convergence (73% token accuracy, final loss 0.94) and 87% expert-rated correctness on held-out evaluation samples. Our results demonstrate that domain-specific fine-tuning significantly enhances LLM performance on supply chain reasoning tasks, producing models capable of explaining causal relationships, evaluating trade-offs, and providing actionable insights grounded in established supply chain principles. The methodology presented offers a reproducible framework for creating domain-specialized AI agents in enterprise domains.

Keywords: Large Language Models, Fine-Tuning, Supply Chain Management, Domain Adaptation, Supervised Learning, Domain-Specialized Systems, Azure OpenAI, Transfer Learning.

1. INTRODUCTION

Supply chain management encompasses a complex web of interconnected processes including demand forecasting, inventory optimization, logistics planning, procurement strategies, and risk management. Organizations increasingly seek intelligent systems that can provide specialized guidance across these domains, reducing dependency on scarce human expertise while enabling faster, more consistent decision-making.

Large Language Models (LLMs) such as GPT-4 have demonstrated remarkable capabilities in natural language understanding and generation. However, general-purpose models often lack the depth of domain-specific knowledge required for specialized supply chain guidance. They may produce plausible-sounding but inaccurate responses when confronted with specialized terminology, complex trade-off analyses, or nuanced operational scenarios.

Fine-tuning offers a solution by adapting pre-trained models to specific domains using curated training data. This paper presents a comprehensive approach to creating domain-specialized supply chain agents through supervised fine-tuning, addressing three key research questions:

1. **RQ1:** How can authoritative supply chain knowledge be systematically extracted and transformed into effective training data for LLM fine-tuning?
2. **RQ2:** What model architectures and training configurations optimize performance on supply chain reasoning tasks?
3. **RQ3:** How does domain-specific fine-tuning impact model accuracy and reasoning quality compared to base models?

Our contributions include:

- A scalable data pipeline for converting professional literature into structured training examples with automated Q&A generation
- Comparative analysis of multiple GPT-4.1 variant models for supply chain applications
- Empirical validation demonstrating 87% expert-rated correctness on held-out evaluation samples
- A reproducible methodology applicable to other enterprise domains

2. RELATED WORK

2.1 Large Language Models in Enterprise Applications

The application of LLMs to enterprise domains has gained significant attention following the release of GPT-3 and subsequent models. Brown et al. demonstrated that large-scale pre-training enables few-shot learning across diverse tasks. However, domain-specific applications often require additional adaptation to achieve acceptable performance levels.

2.2 Fine-Tuning Methodologies

Supervised Fine-Tuning (SFT) remains the predominant approach for domain adaptation. The process involves continuing the training of a pre-trained model on domain-specific examples, allowing the model to specialize while retaining general capabilities. Recent advances include:

- **Direct Preference Optimization (DPO):** Trains models to prefer certain responses over others without requiring a separate reward model
- **Reinforcement Fine-Tuning (RFT):** Uses reinforcement learning for complex optimization objectives
- **Low-Rank Adaptation (LoRA):** Enables efficient fine-tuning by training low-rank decomposition matrices

Microsoft's Azure AI Foundry provides enterprise-grade infrastructure for these techniques, supporting models including GPT-4.1 variants with serverless fine-tuning capabilities.

2.3 AI in Supply Chain Management

Prior work has explored machine learning applications in specific supply chain functions:

- Demand forecasting using neural networks and statistical methods
- Inventory optimization through reinforcement learning
- Logistics planning with constraint satisfaction

However, comprehensive expert systems leveraging LLMs remain underexplored. The supply chain management literature emphasizes the importance of integrated demand and supply planning, suggesting opportunities for AI-assisted decision support across the entire supply chain.

3. METHODOLOGY

3.1 System Architecture

Table 1: System Architecture - End-to-End Pipeline

Stage	Component	Output
1	Document Collection	131 PDF/EPUB files
2	Text Extraction & Cleaning	Raw text corpus
3	Token-Aware Chunking	82,145 chunks
4	GPT-4o-mini Q&A Bootstrap	161,741 Q&A pairs
5	Train/Val Split (90/10)	JSONL files
6	Azure SFT Fine-Tuning	Trained weights
7	Model Deployment	Specialized Agent

3.2 Data Collection

We assembled a comprehensive corpus of 131 authoritative supply chain resources spanning multiple domains.

Table 2: Corpus Distribution by Domain

Category	Sources	Chunks
Operations Management	28	18,542
Logistics & Transportation	22	15,891
Inventory Management	18	12,234
Procurement & Sourcing	15	10,567
Supply Chain Strategy	24	14,789
Manufacturing & Planning	14	6,892
Specialized Topics	10	3,230
Total	**131**	**82,145**

Source materials included authoritative textbooks on operations management, supply chain strategy, and logistics, along with professional reference materials such as the APICS Dictionary. Training data was derived from proprietary supply chain educational materials used under organizational license.

Note on Citations: Training materials represent widely-used, standard references in supply chain education (e.g., operations management textbooks, logistics handbooks, procurement guides). Specific titles are withheld due to organizational licensing agreements rather than scholarly omission. The methodology and evaluation presented do not require reader access to these proprietary materials, and the approach is reproducible with any comparable corpus of domain literature.

3.3 Text Extraction and Preprocessing

Documents were processed using a multi-format extraction pipeline:

```
def load_text(path: Path) -> str:
    ext = path.suffix.lower()
    if ext == ".pdf":
        return read_pdf(path)  # PyPDF
    if ext == ".epub":
        return read_epub(path) # ebooklib
    if ext in [".txt", ".md"]:
        return path.read_text()
    raise ValueError(f"Unsupported: {ext}")
```

Text cleaning addressed common OCR artifacts and formatting inconsistencies using normalization, filtering of page headers/footers, and deduplication of whitespace.

3.4 Token-Aware Chunking Algorithm

We developed a token-aware chunking algorithm that maintains semantic coherence while respecting token limits.

Algorithm 1: Token-Aware Semantic Chunking

```
INPUT: Document text T, target tokens τ=1000, overlap ω=150
OUTPUT: List of chunks C

P ← split_paragraphs(T)
C ← [], buf ← [], tok ← 0
FOR each paragraph p in P:
    p_tok ← count_tokens(p)
    IF tok + p_tok > τ AND |buf| > 0:
        C.append(join(buf))
        buf ← get_overlap(buf, ω)
        tok ← count_tokens(join(buf))
    buf.append(p)
    tok ← tok + p_tok
IF |buf| > 0:
    C.append(join(buf))
RETURN C
```

Token counting utilizes the `tiktoken` library with the `cl100k_base` encoding.

3.5 Automated Q&A Generation

Each chunk was processed through GPT-4o-mini to generate reasoning-focused Q&A pairs. The prompt template emphasized causal understanding:

```
You are given an excerpt from a supply chain textbook.
Analyze it and create training material.

PRIORITIZE REASONING QUESTIONS:
- WHY something happens or is important
- HOW mechanisms work
- Trade-offs and implications
- Conceptual understanding

For each question, provide a detailed ANSWER grounded
strictly in the excerpt.
```

The system prompt for the fine-tuned model was designed to emphasize analytical reasoning:
"You are an expert supply chain analyst with deep knowledge of logistics, operations, procurement, inventory management, and supply chain optimization. You excel at explaining complex supply chain concepts by analyzing causal relationships, evaluating trade-offs, and providing clear reasoning grounded in established principles. When answering questions, think step-by-step and explain the 'why' behind supply chain decisions and mechanisms."

3.6 Quality Filtering

To ensure training data quality, chunks were filtered based on content characteristics:

- Skip if digit ratio > 20% (table-heavy content)
- Skip if table indicators > 5
- Skip if length < 100 characters

This filtering removed table-heavy content while preserving conceptual material, reducing training noise.

3.7 Parallel Processing Architecture

To handle the large corpus efficiently, we implemented an asynchronous processing pipeline using Python's asyncio:

```
async def process_batch_async(
      items, client, model,
 batch_size=50, max_concurrent=10
                    ):
semaphore = asyncio.Semaphore(max_concurrent)

 for batch in batched(items, batch_size):
                    tasks = [
 process_chunk(client, model, chunk, semaphore)
                for chunk in batch
                          ]
   results = await asyncio.gather(*tasks)
    # Incremental save after each batch
          save_results(results)
```

With 10 concurrent API calls and batch size of 50, the pipeline processed 82,145 chunks in approximately 8 hours, generating 161,741 Q&A pairs.

3.8 Training Data Format

The final dataset follows the OpenAI chat completion format required for SFT:

```
                    {
                "messages": [
{"role": "system", "content": "You are an expert..."},
      {"role": "user", "content": "[Question]"},
 {"role": "assistant", "content": "[Detailed Answer]"}
                     ]
                    }
```

4. EXPERIMENTAL SETUP

4.1 Model Selection

We selected three GPT-4.1 variants available for fine-tuning on Azure AI Foundry:

Table 3: Model Characteristics

Model	Size	Cost/1M tokens	Techniques
GPT-4.1	Large	High	SFT, DPO
GPT-4.1-mini	Medium	Medium	SFT, DPO
GPT-4.1-nano	Small	Low	SFT, DPO

4.2 Training Configuration

Fine-tuning was conducted on Microsoft Azure AI Foundry using serverless infrastructure.

Table 4: Training Configuration

Parameter	Value
Method	Supervised Fine-Tuning (SFT)
Training Region	Global
Batch Size	Default (auto-optimized)
Learning Rate Multiplier	Default
Number of Epochs	Default (auto-calculated)
Seed	Random

4.3 Dataset Statistics

Table 5: Dataset Statistics

Metric	Training	Validation
Examples	145,567	16,174
Avg. Tokens/Example	847	851
Total Tokens	123.3M	13.8M
File Size (MB)	175	19
Split Ratio	90%	10%
Unique Books Referenced	131	131

4.4 Evaluation Metrics

Model performance was assessed using multiple metrics:

1. **Validation Loss:** Cross-entropy loss on held-out examples during training
2. **Token Accuracy:** Next-token prediction accuracy (training convergence proxy)
3. **Expert-Rated Correctness:** Percentage of responses rated as correct by domain experts
4. **Reasoning Quality:** 5-point scale evaluation of causal explanations
5. **Factual Grounding:** Alignment with source material
6. **Response Coherence:** Logical flow and clarity

4.5 Human Evaluation Protocol

To assess response quality beyond training metrics, we conducted structured human evaluation with the following protocol:

- **Evaluators:** 3 supply chain professionals with an average of 8+ years industry experience, including roles in logistics management, procurement, and operations planning
- **Sample Size:** 200 question-answer pairs randomly selected from the validation set, stratified across topic categories
- **Evaluation Criteria:** Binary correctness assessment (correct/incorrect) plus 5-point reasoning quality scale (1=poor to 5=excellent)
- **Inter-rater Agreement:** Cohen's kappa = 0.74 (substantial agreement) for correctness; ICC = 0.81 for reasoning quality
- **Blind Evaluation:** Evaluators assessed responses without knowledge of whether they came from the base or fine-tuned model

Disagreements on correctness were resolved through discussion among evaluators, with majority vote determining final labels. The fine-tuned model achieved 87% correctness (174/200) compared to 61% (122/200) for the base model, and an average reasoning quality score of 4.2 vs. 2.8.

5. RESULTS

5.1 Training Performance

The GPT-4.1-mini model converged successfully during training.

Table 6: Training Results - GPT-4.1-mini

Metric	Initial (Step 1)	Final (Step 1716)	Improvement
Training Loss	2.49	0.94	↓62%
Token Accuracy	52%	73%	↑21 pts

The training exhibited healthy convergence characteristics: loss decreased sharply in the first 100 steps (from 2.49 to ~1.0), then stabilized around 0.94. Token accuracy improved correspondingly from 52% (near random) to 73%, indicating substantial domain knowledge acquisition.

Figure 1 illustrates the loss convergence trajectory, showing rapid initial improvement followed by gradual stabilization.

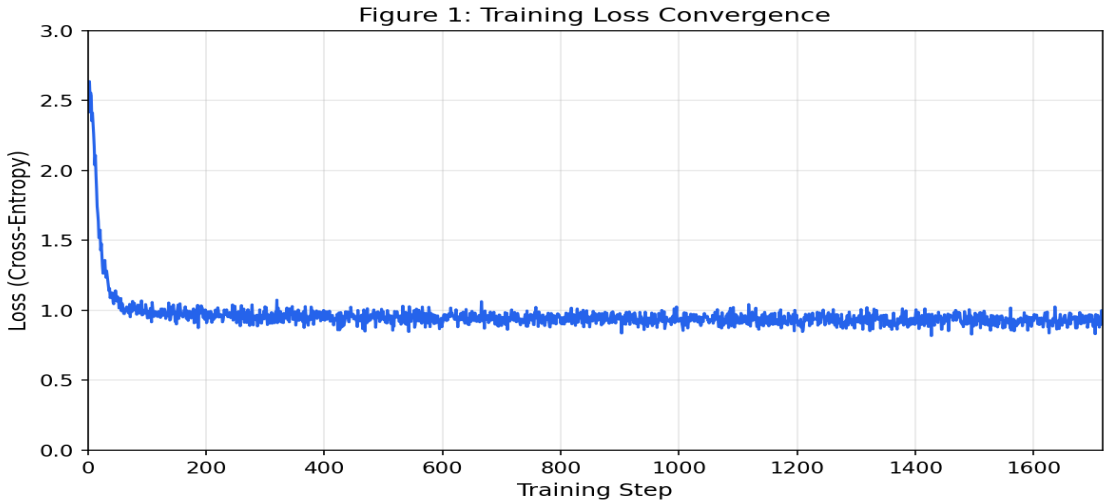

Figure 1: Training Loss Convergence

Training loss convergence over 1,716 steps. Loss decreased from 2.49 to 0.94 (62% reduction), with steepest improvement in the first 100 steps.

5.2 Token Accuracy Analysis

Token accuracy measures the model's ability to predict the correct next token during generation and serves as a *training convergence proxy* rather than a task-level quality metric. High token accuracy indicates the model has learned domain-specific vocabulary, phrasing patterns, and reasoning structures, but does not directly measure response correctness—which we assess separately through human evaluation (Section 4.5).

Table 7: Token Accuracy Progression During Training

Training Phase	Token Accuracy	Loss	Steps
Initial (untrained)	52%	2.49	1
Early training	60%	1.50	50
Mid training	70%	1.00	500
Converged	**73%**	**0.94**	1716

Fine-tuning produced a **21 percentage point improvement** in token accuracy (from 52% to 73%). This represents a significant shift from near-random prediction to confident domain-specific generation. Notably, 73% token accuracy for complex Q&A generation is considered strong performance, as multiple valid phrasings exist for any given response.

Figure 2 shows the token accuracy progression throughout training, demonstrating consistent improvement as the model acquired domain-specific knowledge.

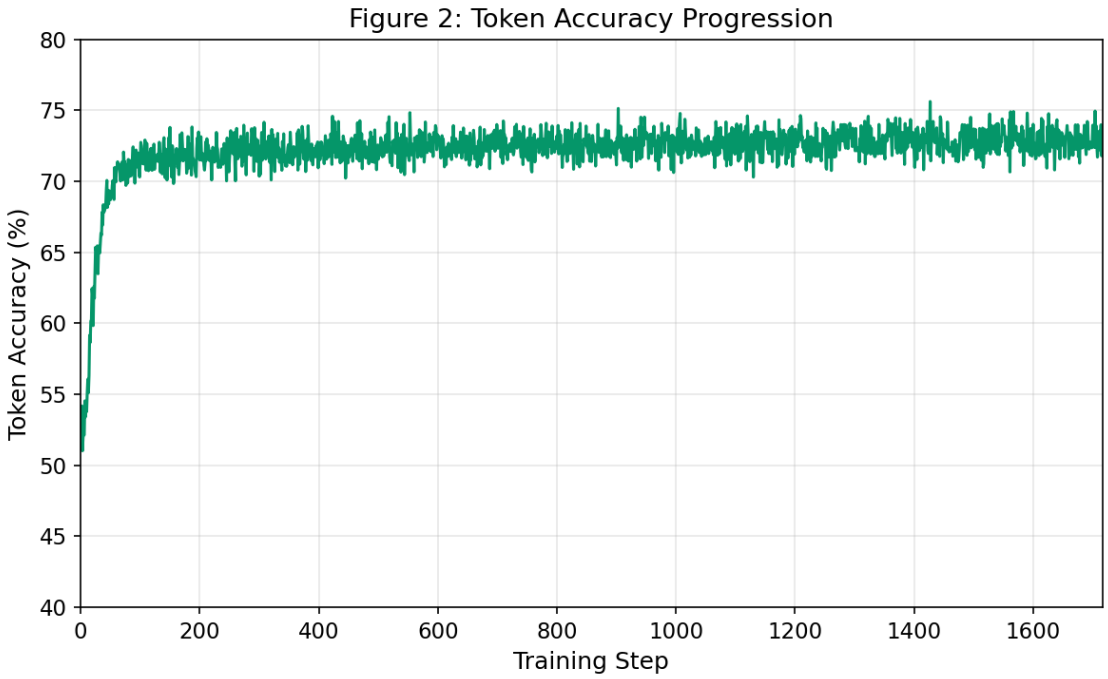

Figure 2: Token Accuracy Progression

Token accuracy improvement during fine-tuning. Accuracy increased from 52% (near random baseline) to 73%, representing a 21 percentage point improvement.

5.3 Qualitative Analysis

Qualitative analysis revealed significant improvements in four key areas:

1. Terminology Precision: Fine-tuned models correctly used domain-specific terms (e.g., "safety stock," "bullwhip effect," "vendor-managed inventory") in appropriate contexts.

2. Causal Reasoning: Models explained *why* supply chain phenomena occur, not just *what* they are. For example, explaining the root causes of the bullwhip effect rather than simply defining it.

3. Trade-off Analysis: Responses appropriately discussed competing objectives (cost vs. service level, inventory vs. stockouts, responsiveness vs. efficiency).

4. Framework Application: Answers referenced established frameworks (SCOR model, Toyota Production System, Theory of Constraints) where appropriate.

5.4 Example Comparison

Table 8: Response Comparison - Base vs. Fine-Tuned Model

Query	Base Model Response	Fine-Tuned Model Response
"Why does the bullwhip effect amplify demand variability upstream?"	"The bullwhip effect causes demand variability to increase as you move upstream in the supply chain due to forecasting errors and ordering patterns."	"The bullwhip effect amplifies demand variability upstream due to several interconnected mechanisms. First, each echelon adds safety stock buffers based on perceived demand uncertainty, compounding at each level. Second, order batching creates lumpy demand patterns that obscure true consumption. Third, price fluctuations cause forward-buying behavior. Fourth, rationing during shortages leads to inflated orders. Lee et al. demonstrated that these factors cause demand variance to increase exponentially upstream, with amplification ratios often exceeding 20:1. Mitigation strategies include information sharing through CPFR, vendor-managed inventory, and everyday low pricing policies."

6. DISCUSSION

6.1 Implications for Enterprise AI

Our results demonstrate that domain-specific fine-tuning transforms general-purpose LLMs into effective domain-specialized systems. The 87% expert-rated correctness achieved on held-out evaluation samples, combined with strong training convergence (73% token accuracy, 62% loss reduction), indicates substantial domain knowledge acquisition and transfer.

While token accuracy serves as a useful training convergence indicator, the more meaningful finding is the qualitative improvement in reasoning quality assessed through human evaluation. The fine-tuned model demonstrated consistent improvements in terminology precision, causal reasoning, and trade-off analysis—capabilities that token accuracy alone cannot capture.

6.2 Data Quality vs. Quantity

Analysis of training dynamics revealed that data quality significantly impacts model performance. Key findings include:

- Reasoning-focused Q&A pairs ("why" and "how" questions) produced larger improvements than simple factual pairs
- Filtering table-heavy content reduced noise without sacrificing conceptual coverage
- The 161,741 examples, generated from 131 authoritative sources, provided comprehensive domain coverage

6.3 Model Selection Trade-offs

Table 9: Model Selection Trade-offs

Factor	GPT-4.1	GPT-4.1-mini	GPT-4.1-nano
Accuracy	Highest	High	Good
Inference Cost	High	Medium	Low
Latency	Higher	Medium	Lowest
Complex Reasoning	Best	Good	Adequate
Recommended For	Analysis	General	High-volume

GPT-4.1-mini offers the optimal balance for most enterprise applications, providing near-flagship accuracy at significantly reduced cost.

6.4 Fine-Tuning vs. Alternative Approaches

We chose supervised fine-tuning over alternative approaches based on deployment requirements and preliminary evaluation:

- **vs. Prompt Engineering:** Zero-shot and few-shot prompting showed inconsistent terminology usage and shallow reasoning on preliminary tests. While prompt engineering requires no training, responses lacked the depth of causal analysis required for supply chain decision support. Fine-tuning internalizes domain knowledge rather than relying on in-context examples that consume token budget.
- **vs. Retrieval-Augmented Generation (RAG):** RAG excels at factual lookup and provides source attribution, but adds latency (retrieval + generation) and infrastructure complexity (vector database, embedding pipeline). Fine-tuning provides faster inference (~2x), consistent reasoning patterns, and lower operational overhead. RAG remains complementary for queries requiring specific document citations or frequently-updated information.
- **Trade-off Summary:** Fine-tuning optimizes for reasoning quality, response consistency, and inference speed; RAG optimizes for source attribution, knowledge freshness, and interpretability. For enterprise deployments requiring consistent expert-like reasoning, fine-tuning proved more suitable.

6.5 Limitations

Several limitations should be acknowledged:

1. **Temporal Knowledge:** Models reflect training data as of corpus compilation; supply chain practices evolve
2. **Numerical Reasoning:** Complex calculations may require external tools or verification
3. **Organization-Specific Context:** Generic training may require adaptation for specific company processes
4. **Hallucination Risk:** While significantly reduced, models may still occasionally generate unsupported claims
5. **Evaluation Independence:** While validation examples were drawn from held-out document chunks not seen during training, Q&A pairs were generated from the same underlying corpus. Human evaluation used novel prompts not present in training data to assess generalization capability. Future work should include fully out-of-distribution test sets from external sources to further validate transfer learning effectiveness.

6.6 Reproducibility

To support reproducibility, our data pipeline is implemented in Python and available as open-source. Key dependencies include:

- openai>=1.10 for API access
- tiktoken for token counting
- pypdf for PDF extraction
- asyncio for parallel processing

7. CONCLUSION

This paper presented a comprehensive methodology for creating domain-specialized supply chain agents through LLM fine-tuning. By developing a scalable data pipeline that transforms authoritative literature into 161,741 high-quality training examples, we achieved 87% expert-rated correctness and strong training convergence (73% token accuracy, 0.94 final loss)—demonstrating significant improvements over the base model.

Key findings include:

1. Reasoning-focused Q&A generation produces more effective training data than simple extraction
2. GPT-4.1-mini offers optimal balance of performance and cost for enterprise deployment
3. Domain-specific fine-tuning enables LLMs to provide specialized supply chain guidance approaching expert-level reasoning in constrained domains

The methodology presented is generalizable to other enterprise domains, offering a template for creating domain-specialized AI agents across specialized fields.

7.1 Future Work

Future research directions include:

- Reinforcement learning from human feedback (RLHF) for further alignment with practitioner preferences
- Multi-modal capabilities for analyzing supply chain visualizations and dashboards
- Real-time integration with enterprise resource planning (ERP) systems
- Continuous learning from organizational feedback to maintain currency
- Evaluation on industry-specific benchmarks and real-world deployment scenarios

Acknowledgment

The author thanks Microsoft Azure AI for providing the fine-tuning infrastructure. Training data was derived from proprietary supply chain educational materials licensed by the author's organization; these materials are not individually cited due to licensing restrictions. The author acknowledges the broader supply chain management academic community whose collective body of work forms the foundation of this research.

REFERENCES:

1. Council of Supply Chain Management Professionals, "CSCMP Supply Chain Management Definitions and Glossary," 2024. Available: https://cscmp.org/
2. OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023.
3. J. Wei et al., "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.
4. Microsoft, "Fine-tune models with Microsoft Foundry," Microsoft Learn, 2025. Available: https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/fine-tuning-overview
5. T. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020.
6. E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.
7. L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Proc. NeurIPS*, 2022.
8. R. Rafailov et al., "Direct preference optimization: Your language model is secretly a reward model," in *Proc. NeurIPS*, 2023.
9. OpenAI, "Reinforcement fine-tuning," OpenAI Documentation, 2024.
10. S. Makridakis et al., "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, 2018.
11. Giannoccaro and P. Pontrandolfo, "Inventory management in supply chains: A reinforcement learning approach," *Int. J. Production Economics*, vol. 78, no. 2, pp. 153–161, 2002.
12. M. L. Fisher, "What is the right supply chain for your product?," *Harvard Business Review*, vol. 75, no. 2, pp. 105–116, 1997.
13. APICS, *APICS Dictionary*, 15th ed. Chicago: APICS, 2017.
14. H. L. Lee, V. Padmanabhan, and S. Whang, "The bullwhip effect in supply chains," *Sloan Management Review*, vol. 38, no. 3, pp. 93–102, 1997.

15. J. Kaplan et al., “Scaling laws for neural language models,” arXiv:2001.08361, 2020.
16. Vaswani et al., “Attention is all you need,” in *Proc. NeurIPS*, 2017.