Encoding Domain Expertise in Agents: Lessons from NFL Fantasy AI

Michael Butler (✉)[1[0009-0009-8946-9158]], Henry Wang[1[0009-0001-0911-4560]], Jake Lee[1[0009-0001-7131-5427]], Kenton Blacutt[1[0009-0004-5821-5511]], Dan Volk[1[0009-0008-5068-7468]], Mike Band[2[0009-0003-7182-4293]], Diego Socolinsky[1[0009-0002-7583-6366]]

[1] Amazon Web Services, Seattle, USA
[2] National Football League, New York, USA
mbutler@amazon.com

Abstract. Agentic AI systems can access vast data but struggle to apply domain expertise, namely the contextual understanding of how to use specialized information. This paper presents a practical framework for encoding such expertise, demonstrated with the National Football League (NFL) through NFL Fantasy AI, a production system delivering analyst-grade fantasy football advice, as assessed by NFL Pro analysts. We introduce a three-step encoding method: (1) analyst-sourced reasoning guidance, encoding analytical patterns as generalized guidance rather than enumerated rules; (2) category-level semantic framing, where experts describe data usage rather than definitions; and (3) LLM-optimized semantic interfaces, using model-to-model iteration for field naming and tool design. Deployed in eight weeks, the system achieved over 90% analyst agreement on response quality, sub-5-second response times, and zero policy violations across more than 10,000 production queries.

Keywords: National Football League · Next Gen Stats · Agentic Systems · Domain Expertise · Sports Analytics

1 Introduction

Fantasy football is the most popular category among the over 60 million fantasy sports players in North America [6], driving fan engagement [5] and generating subscription revenue [10] for the National Football League (NFL). The league recognized an opportunity to differentiate NFL Pro, its advanced analytics experience available through NFL+ Premium, by leveraging NFL Next Gen Stats (NGS), a real-time player and ball tracking system. This proprietary data represented untapped value, yet the challenge extended beyond data access: how could an AI system learn not just what the data means, but when and how do analysts use it to make expert recommendations?

Fantasy football analysts and language models can access the same data, yet only analysts know that a receiver's snap share matters more than yards-per-catch when evaluating floor versus ceiling (i.e., minimum expected output versus maximum upside). This gap between data access and contextual application defines the challenge of encoding domain expertise into agentic AI systems.

This paper presents a practical framework for encoding domain expertise into agentic AI systems under aggressive time constraints. Through NFL Fantasy AI, we demonstrate how semantic data dictionaries, analyst-sourced reasoning guidance, and Large Language Model (LLM)-driven data field and tool optimization transform raw statistics into contextual expert judgment, offering an alternative to traditional fine-tuning [8], Retrieval-Augmented Generation (RAG) [9], and knowledge graph approaches [11].

Deployed in eight weeks, the system achieved over 90% analyst agreement on recommendation quality. The framework proved transferable: analysts now use the system to draft content, and the architecture serves as a foundation for conversational access to NGS data beyond fantasy football.

2 Background and Related Work

Recent work establishes that LLMs can function as reasoning engines orchestrating external tools [12, 16, 18], with protocols like Model Context Protocol (MCP) [2] providing structured interfaces to external data. This shifts the central challenge from retrieval to application: can the model interpret results with domain-appropriate judgment?

Established approaches carry tradeoffs limiting their applicability to judgment-intensive tasks. Fine-tuning approaches, including parameter-efficient methods [8], require retraining as knowledge evolves and struggle to encode reasoning that depends on query intent. RAG [9] assumes retrieved passages contain sufficient interpretive context, yet expert judgment often resides in how information is weighed rather than what exists. Knowledge graphs [11] excel at entity relationships but do not naturally capture conditional reasoning such as "use this metric for floor, that metric for ceiling."

Sports analytics applications of LLMs have addressed system design [4], natural language querying [13], content retrieval [14], and media search [15]. These approaches address what data to retrieve, how to structure relationships, or how to adapt model weights. None addresses how an agent should interpret and apply domain data once retrieved. Our work targets this gap: we encode domain expertise into the agent's reasoning guidance and its semantic interface to data, teaching the agent not what statistics mean, but when and how analysts use them. The framing is prescriptive: not 'snap share measures percentage of offensive plays' but 'snap share indicates opportunity volume; use to assess workload floor and role stability, not scoring upside.'

3 Solution Overview

NFL Fantasy AI delivers analyst-grade fantasy advice through three architectural layers (Fig. 1): delivery infrastructure handling authentication and response streaming via Amazon EKS; agent orchestration using Strands Agents on Amazon Bedrock [1]; and a semantic data layer exposing NFL and third-party statistics through MCP servers [2].

We chose an agentic architecture over deterministic workflows because fantasy questions are inherently unpredictable. Player comparisons, trade evaluations, waiver

recommendations, and matchup analyses arrive in any combination and sequence. The agent must dynamically determine which data to retrieve and how to synthesize it.

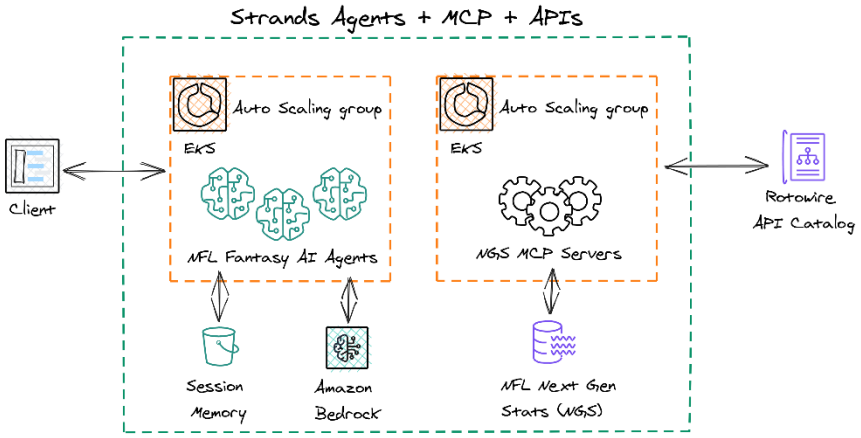

Fig. 1. NFL Fantasy AI solution architecture

The semantic data layer ingests NGS and third-party data (RotoWire injury reports, news, projections) through MCP, chosen over embedded tools for separation of concerns, independent scaling, and reusability. Critically, MCP's tool interface provided the surface area for encoding domain expertise, detailed in Section 4.

Production deployment required solving three additional engineering challenges: cost efficiency, resilience under peak load, and topic control. We achieved cost efficiency through three mechanisms: (1) semantic field curation and LLM-optimized tool docstrings; (2) strategic cache allocation for large MCP tool results; and (3) tool output structure optimized for deeply nested statistical data. Collectively, these reduced token consumption by 70%, doubled throughput, and cut inference costs by 45%. We achieved resilience through automatic fallback to secondary models under throttling conditions. Guardrails constrain the agent to fantasy football topics through layered defenses including prompt-based topic control and input/output classification, avoiding responses that could create brand or liability risk.

4 The Domain Expertise Encoding Framework

We developed a three-layer framework (Fig. 2) addressing the semantic gap between raw data and expert reasoning. Each layer encodes one aspect: reasoning guidance establishes how to think, category framing defines when to use data, and LLM optimization ensures the agent understands what the data means and how to obtain it efficiently.

4.1 Raw Domain Data

NGS data arrives via NFL-hosted API endpoints in JSON format. The NFL publishes API specifications listing available fields and data types, but not semantic definitions:

field names `recRtPG` (receiving routes run per game) and `ptPct` (percentage of plays participated in) offer no interpretive guidance. We supplement NGS with RotoWire data (news, projections, depth charts) for situational context. The encoding framework described below addresses how we transformed undefined, abbreviated NGS fields into semantically accessible data.

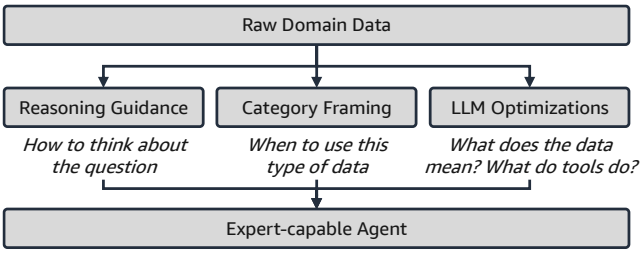

Fig. 2. The domain expertise encoding framework operates at three levels of abstraction: reasoning guidance, category framing, and LLM optimization of fields and tools.

4.2 Analyst-Sourced Reasoning Guidance

The first layer establishes the analytical approach guiding the agent's reasoning. Working with NFL Pro analysts, we articulated their reasoning process at a level of abstraction that generalizes across scenarios, capturing the structure of expert analysis [17]: what factors to consider, how to weigh competing evidence, how to frame recommendations, rather than enumerating specific rules. We embedded this guidance directly in the agent's system prompt, organized from analytical philosophy down to procedural steps. For example, we encoded temporal reasoning patterns, as seen in this excerpt:

```
"Fantasy Season (September to December): ALWAYS start with
recent news... PRIORITIZE recent performance trends that may
lead to future success... CONSIDER upcoming matchups in short-
term prognostication."
```

At a procedural level, we encoded the sequence analysts follow, such as checking news and depth chart position before retrieving statistics (a backup's volume projection depends entirely on the starter's health status). This distinction (reasoning sequence, not conclusions) proved critical: overly specific rules (e.g., "recommend players with >15 targets per game") degraded performance on edge cases, while appropriately abstract guidance enabled robust generalization.

4.3 Category-Level Semantic Framing

When asked to document statistics, analysts instinctively described how to use them rather than defining them. Extending dataset documentation practices [7] beyond descriptive metadata, we formalized these insights into a semantic data dictionary organized around analytical categories rather than data structures.

Working with analysts, we developed 21 category descriptions covering player and team scenarios by position. For running backs, we describe participation: *"Snap share, touches, and route metrics that reveal weekly workload floor and role size; use to gauge sustainability and future volume*", guiding the agent to retrieve participation data when

assessing opportunity rather than past performance. For quarterbacks: *"Next Gen Stats efficiency and pressure metrics that reveal context behind traditional numbers"*, directing the agent toward advanced metrics when box scores require deeper interpretation.

4.4 LLM-Optimized Semantic Interfaces

Category framing addressed when to retrieve data, but two gaps remained: whether the agent could understand and apply what it received (field semantics), and whether it could efficiently obtain it (tool design). Both required effective semantic descriptions for LLM comprehension and application. Given aggressive delivery timelines, we adopted an approach we call model-to-model iteration: (1) prompt Claude Sonnet with source (field name, descriptor, tool docstring) and ask it to explain the element's analytical purpose; (2) identify failures where interpretation diverged from ground truth; (3) prompt Claude Opus with the actual purpose alongside Sonnet's misinterpretation, requesting compressed alternatives; (4) validate by re-testing Sonnet's comprehension. In practice, convergence typically required three iterations.

For field semantics, raw NGS field names like `recRtPG` carried no semantic signal. For example, `recRouteParticipationPct` ("percentage of receiving routes participated") became `recRtPartPct` ("route participation %"). Applied across hundreds of fields and combined with category-level framing, the LLM-optimized field descriptors yielded significant token reduction with equal or better comprehension.

For tool design, our original approach provided discrete tools for each use case (`get_weekly_projections`, `get_season_projections`, etc.), with well-documented parameters and usage examples. This produced substantial token overhead in the tool specification and a large number of tool calls to answer sample questions. Using the same protocol, we consolidated nine projection tools into a single `get_projections` tool [3] and compressed parameter names to their minimum LLM-effective form (`positions` to `pos`, `players` to `plyrs`). Notably, docstrings shifted from describing mechanical function to encoding analyst behavior. For example, a human-readable description like *"Team code(s) - filters player results and required for defense projections"* became *"team(s) - for matchup analysis, include both teams."* The resulting consolidation from 29 tools to 10 significantly reduced tool specification overhead, while the LLM-optimized parameter design reduced typical tool calls per query, as the agent could express nuanced intent in a single tool call.

5 Evaluation

5.1 Quality Assessment

Fantasy recommendations lack verifiable ground truth. Correct decisions depend on future outcomes. We evaluated whether agent reasoning aligned with expert standards through structured analyst review [19]. NFL Pro analysts authored 50 test questions spanning player comparisons, start/sit decisions, waiver pickups, trade evaluations, and matchup analyses. The agent processed each without human intervention. Analysts rated responses on three tiers: acceptable (substantively similar advice), acceptable with

comments (directionally correct, requiring refinement), and unacceptable (different recommendations or flawed reasoning). After three evaluation rounds incorporating feedback, over 90% of responses received acceptable ratings. Evaluation was not blinded; analysts knew they were assessing AI-generated responses. Disagreements typically involved edge cases where reasonable analysts might differ, such as borderline start/sit decisions involving players with uncertain game-time injury status.

Additional validation emerged organically: the NFL Pro content team began using the agent to draft player insights, and in informal testing, analysts could not reliably distinguish AI-generated content from human-written content.

5.2 System Performance

The system achieved sub-5-second time to first token with complete responses under 30 seconds, stable across 10x traffic spikes on Sunday mornings. With over 10,000 production queries, guardrails prevented all off-topic and policy-violating responses.

5.3 Limitations

The encoding approach requires ongoing maintenance as domain knowledge evolves, demanding sustained expert collaboration. Analyst agreement measures reasoning alignment but cannot validate actual fantasy outcomes. For domains with stable patterns and large labeled datasets, fine-tuning may outperform runtime encoding [8]. We did not perform component-level ablation due to production timeline constraints. Production and proprietary constraints limit the implementation detail that can be disclosed.

5.4 Generalizability

Practitioners should assess whether: (1) queries are unpredictable or enumerable; (2) experts can articulate reasoning patterns, not just conclusions; (3) interpretation is context-dependent; (4) domain knowledge evolves frequently. When these conditions hold, our approach may offer a viable path to production-quality reasoning.

6 Conclusion

This paper presented a practical framework for encoding domain expertise into agentic AI systems, demonstrated through NFL Fantasy AI. By encoding expertise through analyst-sourced reasoning guidance, category-level framing, and LLM-optimized semantic interfaces, we achieved reasoning that NFL Pro analysts assessed as analyst-grade. The techniques may extend to other domains characterized by specialized terminology, contextual interpretation, and implicit expert reasoning.

Acknowledgments. The authors thank the NFL Next Gen Stats analytics team for their participation in evaluation and domain expertise sessions.

Disclosure of Interests. The system described in this paper was developed by Amazon Web Services (AWS) for the NFL and uses AWS commercial products.

References

1. Amazon Web Services: Amazon Bedrock User Guide. https://docs.aws.amazon.com/bedrock/latest/userguide/, last accessed 2026/01/29
2. Anthropic: Model Context Protocol Specification. https://modelcontextprotocol.io, last accessed 2026/01/29
3. Anthropic: Tool Use Best Practices. https://platform.claude.com/docs/en/agents-and-tools/tool-use/implement-tool-use, last accessed 2026/01/29
4. Davis, J., et al.: Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned. Mach. Learn. 113, 6977–7010 (2024). doi: 10.1007/s10994-024-06585-0
5. Drayer, J., Shapiro, S.L., Dwyer, B., Morse, A.L., White, J.: The effects of fantasy football participation on NFL consumption: A qualitative analysis. Sport Management Review 13(2), 129–141 (2010). doi: 10.1016/j.smr.2009.02.001
6. Fantasy Sports & Gaming Association: Industry Demographics. https://thefsga.org/fantasy-sports-gaming-industry/, last accessed 2026/01/29
7. Gebru, T., et al.: Datasheets for Datasets. Communications of the ACM 64(12), 86–92 (2021). doi: 10.1145/3458723
8. Hu, E.J., et al.: LoRA: Low-rank adaptation of large language models. In: Tenth International Conference on Learning Representations. OpenReview (2022). doi: 10.48550/arXiv.2106.09685
9. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems, vol. 33. Curran Associates (2020). doi: 10.48550/arXiv.2005.11401
10. Nesbit, T.M., King, K.A.: The impact of fantasy football participation on NFL attendance. Atlantic Economic Journal 38(1), 95–108 (2010). doi: 10.1007/s11293-009-9202-x
11. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE TKDE (2024). doi: 10.1109/TKDE.2024.3352100
12. Schick, T., et al.: Toolformer: Language models can teach themselves to use tools. In: Advances in Neural Information Processing Systems, vol. 36. Curran Associates (2023). doi: 10.48550/arXiv.2302.04761
13. Schilling, A., et al.: Querying football matches for event data: Towards using large language models. In: ISACE 2024, pp. 216–227. Springer. doi: 10.1007/978-3-031-69073-0_19
14. Strand, A.T., Gautam, S., Midoglu, C., Halvorsen, P.: SoccerRAG: Multimodal soccer information retrieval via natural queries. In: CBMI 2024, pp. 1–7. IEEE (2024). doi: 10.48550/arXiv.2406.01273
15. Wang, H., Salekin, M.S., Lee, J., Claytor, R., Zhang, S., Chi, M.: Agentic Generative AI for Media Content Discovery at the National Football League. In: ISACE 2025. Springer (2025). doi: 10.48550/arXiv.2510.07297
16. Wang, L., et al.: A Survey on Large Language Model based Autonomous Agents. Frontiers of Computer Science 18(6), 186345 (2024). doi: 10.1007/s11704-024-40231-1
17. Wei, J., et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Advances in Neural Information Processing Systems, vol. 35. Curran Associates (2022). doi: 10.48550/arXiv.2201.11903
18. Yao, S., et al.: ReAct: Synergizing reasoning and acting in language models. In: Eleventh International Conference on Learning Representations. OpenReview (2023). doi: 10.48550/arXiv.2210.03629
19. Zheng, L., et al.: Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: Advances in Neural Information Processing Systems, vol. 36. Curran Associates (2023). doi: 10.48550/arXiv.2306.05685