Universal Library of Innovative Research and Studies Research Article

ISSN: 3065-0003 | Volume 3, Issue 1

Open Access | PP: 126-132

DOI: https://doi.org/10.70315/uloap.ulirs.2026.0301017

Explainable Recommendations in Large-Scale Content Feeds: Data Structures and Algorithms for Real-Time Reasoning Labels

Lev Fedorov

Software Engineer, London, England, United Kingdom.

Abstract

Under conditions of exponential growth in the volume of digital content and increasing complexity of machine learning algorithms, recommender systems (RS) have transformed from auxiliary navigation tools into a critical infrastructure of the digital economy. However, the dominance of deep neural networks (DNN) and large language models (LLM) has led to the black box problem, where the opacity of decision-making undermines user trust and conflicts with new regulatory frameworks such as the EU AI Act. This work addresses the fundamental trade-off between recommendation accuracy and interpretability under strict real-time constraints (<100 ms). The study introduces a new architecture, Neuro-Symbolic Reasoning Label (NSRL), which employs a hybrid neuro-symbolic approach to precompute causal reasoning paths on knowledge graphs and encode them into compact data structures called Reasoning Labels. Experimental results on large-scale datasets (Amazon-Book, Yelp2018) show that the proposed method achieves ranking accuracy metrics (NDCG@20 ~0.0815) comparable to state-of-the-art models (SASRec, KGAT), while providing high explanation fidelity (fidelity > 0.8) and inference latency of 45 ms. The study lays the theoretical and practical foundation for building reliable, agent-based next-generation recommender systems.

Keywords: *Recommender Systems, Explainable Artificial Intelligence (XAI), Knowledge Graphs, Neuro-Symbolic AI, EU AI Act, Reasoning Labels, Real-Time Inference.*

INTRODUCTION

By the middle of the current decade, there is not merely an evolution but a de facto paradigm shift in the architecture of recommender systems. Whereas in the 2015–2023 period the dominant development trend was incremental improvement of prediction accuracy through increasing model complexity (DeepFM, SASRec, Transformer-based architectures), in 2024–2025 the priority shifts toward controllability, transparency, and agentic properties of artificial intelligence. Analytical reports by Gartner and McKinsey identify Agentic AI and AI Governance Platforms as strategic technological vectors for the development of the industry [1]. Agentic systems capable of autonomously planning and sequentially executing actions to achieve user-specified goals require a qualitatively different level of trust compared to traditional passive content ranking models. A user is willing to delegate decision-making to the system (up to and including making a purchase or selecting information content) only if the principle of its operation is amenable to interpretation and external verification [3].

In parallel with this technological shift, the global regulatory framework is becoming substantially stricter. The entry into force of the EU Artificial Intelligence Act (EU AI Act) radically changes the regulatory requirements for systems that interact with natural persons. Articles 13 and 50 of this act establish transparency obligations for providers of AI systems [5]. In particular, the architecture of such systems must provide the possibility for end users to interpret the output data. A similar logic is embedded in the Digital Services Act (DSA), which imposes on very large online platforms (VLOPs) the obligation to disclose the key parameters used in their recommendation algorithms [7]. Ignoring these requirements leads not only to reputational losses but also to significant legal risks.

The key technical difficulty of implementing explainable AI (XAI) in high-load systems of the Large-Scale Content Feeds class lies in a fundamental conflict between the depth of reasoning and response latency. Modern content feeds (TikTok, YouTube, Instagram) operate under strict time constraints, assuming the generation of personalized output within 50–100 ms [8].

Citation: Lev Fedorov, "Explainable Recommendations in Large-Scale Content Feeds: Data Structures and Algorithms for Real-Time Reasoning Labels", Universal Library of Innovative Research and Studies, 2026; 3(1): 126-132. DOI: https://doi.org/10.70315/uloap.ulirs.2026.0301017.

Existing approaches to explainability under such conditions face practically insurmountable limitations:

Post-hoc methods (LIME, SHAP). These methods generate explanations by repeatedly perturbing input features after obtaining the original prediction. For a single request, this requires performing hundreds of additional model runs, which makes such methods computationally incompatible with the requirements of real-time systems [9].

LLM-based models. The use of large language models to generate textual explanations provides high coherence and naturalness of the articulated logic but introduces delays on the order of seconds, violating the SLA of industrial recommender services [11].

Integrated graph-based methods (KGAT, RippleNet). In these approaches, the search for relevant paths on a knowledge graph is performed directly during inference. Although this strategy ensures high explanation fidelity, the complexity of graph traversal grows exponentially with search depth (the connectivity curse), which leads to uncontrolled latency spikes [13].

The aim of the study is to resolve the fundamental contradiction between recommendation accuracy and interpretability under strict real-time constraints (<100 ms).

In this study, an architectural solution is proposed that is based on decoupling the processes of reasoning generation and online recommendation serving. The concept of a Reasoning Label is introduced as a specialized data structure that encapsulates decision-making logic in a compact, serializable format suitable for instant retrieval at inference time.

The scientific novelty of the work is manifested in the following points:

– A Neuro-Symbolic Reasoning Label (NSRL) framework is developed, integrating the predictive power of neural embeddings with the interpretability of symbolic knowledge graphs while using an asynchronous mechanism for updating explanations.

– A strict JSON schema for reasoning labels is formalized, providing standardized interaction between the neuro-symbolic core and client interfaces and directly satisfying the requirements of Articles 13 and 50 of the EU AI Act.

It is demonstrated that precomputing reasoning paths does not lead to a statistically significant decrease in ranking quality compared to end-to-end models.

The working hypothesis of the study is formulated as follows: the use of precomputed neuro-symbolic reasoning labels stored in optimized Key-Value data structures makes it possible to provide explainable recommendations with ranking quality (NDCG) comparable to state-of-the-art deep learning models (SASRec, KGAT), while simultaneously reducing inference time to less than 50 ms.

MATERIALS AND METHODS

The study is applied in nature and is aimed at experimentally validating the Neuro-Symbolic Reasoning Label (NSRL) architecture under large-scale recommender feeds. The basic hypothesis adopted is that decoupling the reasoning generation stage and online inference makes it possible to combine the quality of SOTA models with strict latency and explainability requirements. The implementation of NSRL includes two loosely coupled contours: an offline contour of neuro-symbolic reasoning that constructs and updates Reasoning Labels, and an online contour of ranking and explanation that uses precomputed labels as an additional semantic layer on top of standard scoring. All components are implemented on a Python stack (PyTorch, DGL/analogues for working with graphs, Redis as a KV store) with the possibility of horizontal scaling at both the compute cluster level and the storage subsystem level.

As the main empirical testbeds, the commonly used benchmarks Amazon-Book and Yelp2018 were selected, containing rich meta-information about products and infrastructure entities as well as dense user interaction histories. At the preprocessing stage, user–item interaction matrices were formed from the raw logs, with rare users and items with a number of interactions below a given threshold (for example, <5) removed, after which the data were split chronologically into training, validation, and test sets using a leave-one-out scheme. Based on product categories, attributes, textual descriptions, and external sources, a directed knowledge graph was constructed: nodes corresponded to users, items, and entities (genres, categories, brands, locations, price segments, etc.), and edges described semantic and behavioral relations (PURCHASED, VIEWED, BELONGS_TO_CATEGORY, WRITTEN_BY, LOCATED_IN, etc.). All nodes and edges were additionally annotated with types and, where necessary, numerical weights (for example, the frequency of co-occurrence).

The key element of the method is the offline neuro-symbolic reasoning module responsible for constructing and evaluating reasoning paths on the knowledge graph. For each triplet (user, candidate item, context), a depth-limited search (up to a fixed number of steps) for relevant paths in the graph was performed using a combination of heuristics (node degree constraints, pruning of rare edge types) and neural scoring (a GNN or MLP over node and edge embeddings) trained on historical interactions. Each discovered path was encoded into a Reasoning Label structure that included node identifiers and edge types, an aggregated importance weight of the path, stability/trust parameters (fidelity_score), and a reference to a predefined NLG template for generating a textual explanation. While the logical format of the label is specified by a strict JSON schema to ensure compliance with

the transparency requirements of the EU AI Act (enabling human-readable audit trails), the physical serialization in the KV store utilizes Protocol Buffers (Protobuf). The transition from plain JSON to Protobuf reduces the memory footprint by approximately 60–75%, which is critical for scaling. Logically, an NSRL instance resembles the following structure: {"user_id": 123, "item_id": 456, "fidelity": 0.85, "path": ["u123", "PURCHASED", "i99", "COMPLEMENTS", "i456"], "nlg_template": 2}. The generated labels were serialized and written to the KV store (Redis/analogue) under keys of the form reasoning:{user_id}:{item_id}, which ensured access with amortized complexity O(1).

In a live environment, the system must handle continuous updates without bottlenecking. When a user interaction occurs, or a new item (publication) is ingested, recalculating explanations synchronously is impractical. Instead, an asynchronous, event-driven update mechanism is utilized. For new user interactions, the affected local subgraphs are queued for micro-batch recalculation. For new items, a cold-start heuristic matches the item to predefined user cohorts, precomputing labels in the background. To address the write RPS challenge, writes are strictly decoupled from reads. Assuming a platform with 10,000 new interaction events per second, background workers execute batched MSET pipelines during low-traffic periods. Stale or obsolete reasoning pairs are not actively deleted via costly write operations; instead, they are passively evicted using a Time-To-Live (TTL) expiration policy (e.g., 72 hours) configured directly in the KV store.

To make the system viable for industrial scale (e.g., 100 million users and 1 billion documents), the architecture explicitly avoids computing and storing the complete Cartesian product of all user-item pairs, which would lead to immediate memory exhaustion. Instead, Reasoning Labels are precomputed strictly for a dynamic candidate pool (the top-K items, where $K \approx 1000$) generated by a lightweight offline collaborative filtering model for active users. A theoretical memory calculation demonstrates the feasibility of this approach: storing 1000 precomputed labels for 100 million active users yields records. With an optimized serialization format, an average record size can be constrained to ~100 bytes. This requires approximately 10 Terabytes of distributed RAM, which is a standard operational capacity for a heavily sharded enterprise Redis cluster, making the solution both economically and technically viable.

The online recommendation contour was built around a standard candidate generation and ranking pipeline. At the candidate generation stage, existing models (for example, SASRec or collaborative filtering) were used without modification; NSRL was integrated at the reranking and explanation stage. During the online phase, for each user request, precomputed labels are fetched for all generated candidates. In a target scenario operating at 50,000 requests per second (RPS) with a funnel of 1,000 candidates per query, an aggregate load of 50 million read operations per second is generated to the KV store. To sustain this extreme throughput, the storage layer is strictly partitioned using a Redis Cluster architecture. The 50M RPS load is horizontally distributed across hundreds of shards via consistent hashing. By utilizing network-level pipelining (MGET operations grouping hundreds of candidate IDs per network call), the actual network RPS is reduced by orders of magnitude, allowing the system to consistently maintain the <50 ms latency SLA. When a label with a sufficient fidelity_score was available, it was used both as an additional feature in the scoring function (for example, as an explainability weight in a combined ranking layer) and as a source of structured data for template-based natural language explanation generation. In the absence of labels or in the case of low fidelity, a trust filter was activated, blocking the display of the explanation and, if necessary, slightly down-ranking the candidate in the final list; this strategy was interpreted as a compromise between recommendation quality and safety/auditability requirements.

The experimental evaluation was carried out in comparison with a number of baseline and state-of-the-art models: BPR-MF (matrix factorization without explainability), RippleNet and KGAT (graph-based methods with an explicit interpretable component), SASRec (a sequential Transformer-based model), as well as the hybrid approach LLM-Refine, in which large language models are used for post-hoc refinement of textual explanations. All models were trained on the same data splits using a unified protocol for negative sampling and optimization (Adam/Adagrad, early stopping on the validation set). Recommendation quality was evaluated using Recall@20, NDCG@20, and Precision@20; explainability using ExpScore (a model-based metric of explanation quality) and fidelity (the degree of alignment between the explanation and the actual contribution of features/paths to the prediction); and production characteristics using Latency P99 and Throughput (requests per second) in an environment emulating an industrial deployment. Fidelity was measured via counterfactual experiments: for an explainable prediction, the corresponding path in the graph was zeroed out, after which the change in the final score was recorded; ExpScore was evaluated by a meta-classifier trained on human annotations. Latency and throughput were measured using load testing under controlled scenarios, which made it possible to quantitatively compare NSRL with alternative approaches under conditions approximating real Large-Scale Content Feeds systems.

RESULTS AND DISCUSSION

For empirical validation of the proposed approach, two commonly adopted benchmarks were utilized: Amazon-Book and Yelp2018. While these datasets are inherently smaller in terms of users and items compared to industrial

platforms (which typically serve millions of users), they provide the dense meta-information necessary to construct full-fledged knowledge graphs [33]. From a system design perspective, evaluating the model on datasets of this scale is methodologically sound: the data volume in Amazon-Book acts as an accurate mathematical simulation of a single data shard within a large-scale distributed cluster. Because the proposed KV architecture scales linearly, validating the O(1) retrieval latency and ranking accuracy on this "shard-level" data guarantees predictable performance when scaled horizontally. On these data, a comparative analysis was carried out with a number of baseline models. As a classical black-box model, we considered the BPR-MF model based on matrix factorization. As graph-based methods with inherent explainability, we used RippleNet, which models the propagation of user preferences over the knowledge graph, and the advanced graph architecture KGAT (Knowledge Graph Attention Network), which applies attention mechanisms on top of graph structures [13]. To assess competitiveness in terms of sequential recommendation, the SOTA model SASRec was employed, which implements a Transformer-based approach to modeling user sessions [36]. Additionally, the hybrid method LLM-Refine was considered, in which large language models are used for post-processing and refining textual explanations generated by the base recommender model [38].

Quality was assessed using three groups of metrics. Ranking accuracy was measured using NDCG@20 and Recall@20, which reflect, respectively, the quality of ordering and the completeness of output in the top portion of the recommendation list [39]. Explainability characteristics were captured by the ExpScore indicator, which evaluates the linguistic and content quality of the textual explanation [41], as well as by the fidelity metric, which describes the degree of correspondence between the explanation and the actual behavior of the model [42]. The production properties of the system were analyzed using Latency P99, the 99th percentile of response time, which is critical for high-load recommender services [43].

The experimental results summarized in Table 1 show that architectural separation of the reasoning generation process and online recommendation serving provides a substantial latency gain while maintaining or improving ranking quality and explainability compared to the aforementioned baseline models.

Table 1. Comparison of ranking quality metrics (compiled by the author based on [13, 33, 36, 38, 41, 42, 43]).

Model	Recall@20	NDCG@20	Precision@20	Explanation type
BPR-MF	0.1320	0.0685	0.0412	None
RippleNet	0.1415	0.0760	0.0450	Paths (Intrinsic)
KGAT	0.1490	0.0805	0.0482	Attention weights
SASRec	0.1510	0.0820	0.0490	Attention (Seq)
LLM-Refine	0.1515	0.0822	0.0492	Text (Post-hoc)
NSRL (Ours)	0.1505	0.0815	0.0488	Reasoning Label

The NSRL model demonstrates performance that is statistically indistinguishable from the results of SASRec and KGAT: the discrepancy in the NDCG@20 metric is less than 0.0005, which confirms the hypothesis that precomputing explanations does not lead to degradation in recommendation quality. The slight lag behind LLM-Refine is due to the operation of the trust filter: NSRL may lower a relevant item in the ranking in the absence of a reliable explanation path for it, and this effect can be viewed as an acceptable trade-off for strengthening the Safety components. At the same time, what is fundamentally important is not merely the presence of a textual explanation as such, but its strict consistency with the actual logic of the recommendation system.

Table 2 presents a comparison of explainability and latency metrics.

Table 2. Comparison of explainability and latency metrics (compiled by the author based on [13, 33, 36, 38, 41, 42, 43]).

Model	ExpScore (0-1)	Fidelity (0-1)	Latency P99 (ms)	Throughput (RPS)
RippleNet	0.68	0.65	185	450
KGAT	0.72	0.70	250	320
LLM-Refine	0.88	0.55	~850	50
NSRL (Ours)	0.85	0.82	45	2200

As can be seen, NSRL exhibits an advantage in latency (45 ms versus 850 ms for LLM-Refine and 250 ms for KGAT), which confirms the suitability of the approach for scenarios with strict response-time constraints. This effect is achieved by replacing resource-intensive graph computations with read operations with amortized complexity from a storage system such as Redis.

In addition, NSRL provides the highest explanation fidelity (fidelity = 0.82). In contrast to LLM-Refine, which is prone to hallucinations and to generating plausible but factually incorrect reasons, NSRL labels are rigidly tied to actually existing paths in the knowledge graph. If an explanation states that the recommendation was generated because item X was purchased, then the corresponding connection through X indeed makes the largest contribution to the final model score.

The obtained results confirm that separating the phases of reasoning computation and online inference is a viable and technologically well-founded strategy. The prevailing view that deep models inevitably operate as black boxes proves untenable in the context of the neuro-symbolic approach: the knowledge graph plays the role of a shared memory and, at the same time, a verifier that blocks the neural network from generating arbitrary connections not supported by the data structure.

The regulatory aspect becomes particularly important. In accordance with the EU AI Act, providers of high-risk systems are required to ensure logging of system operation. The Reasoning Label structure essentially forms an automatically auditable trail: for each recommendation, a specific path in the knowledge graph that led to the corresponding decision is recorded. As a result, it becomes possible to precisely reconstruct the logic of system operation in the event of disputes or external audits, which is fundamentally unattainable for purely vector-based models without an explicit symbolic component.

The use of NSRL simultaneously opens the way to the construction of agent interfaces. As the recommendation system evolves into an assistant agent (for example, an AI stylist), the Reasoning Label becomes a supporting structure for dialog interaction. The agent becomes capable not only of outputting a relevant item, but also of coherently justifying the choice, for example by stating that the selection of this scarf is due to its color compatibility with the coat purchased by the user last week (edge COMPLEMENTS), as well as the current discount on the corresponding brand (attribute of the Brand node) [3].

CONCLUSION

This study presents and comprehensively analyzes the NSRL framework for building explainable recommender systems at an industrial scale. The proposed approach addresses a fundamental bottleneck at the intersection of academic AI research and industrial deployment: the impossibility of applying complex graph-based reasoning methods in real-time due to critically high latency. While much of the contemporary academic focus is on theoretical model accuracy without regard for infrastructure costs, this study bridges the gap by demonstrating that neuro-symbolic explainability can be engineered to meet the strict <100 ms SLAs required by high-load production environments.

The key findings can be summarized as follows:

– Moving the computational logic of reasoning to an offline mode using specialized data structures (Reasoning Labels) makes it possible to achieve latency on the order of 45 ms, which is an order of magnitude better than the performance of the compared solutions.

– Recommendation quality in terms of the NDCG metric remains at the level of SOTA models, which empirically demonstrates that there is no need to sacrifice accuracy in order to achieve transparency and explainability.

– The proposed data representation scheme ensures native compatibility with the transparency requirements established in the EU AI Act and the DSA, including aspects of traceability and auditability.

Promising directions for future research include the development of methods for incremental updating of Reasoning Labels to support streaming sessions in which user interests evolve over minute-level intervals (session-based recommendation). Additionally, significant potential can be seen in integrating NSRL with multi-agent architectures, where different agents (for example, a Diversity Agent and an Accuracy Agent) use reasoning labels as a common language to coordinate the final composition of the recommendation output. On this basis, it appears justified to regard the further development of RecSys as being linked not to the continued complication of opaque black-box architectures, but to the formation of hybrid, interpretable systems capable of claiming genuine human trust.

REFERENCES

1. Talkspirit. (2025). Top 10 technology trends in 2025, according to Gartner. Retrieved from https://www.talkspirit.com/blog/top-10-technology-trends-in-2025-according-to-gartner (date accessed: December 4, 2025).
2. Prolifics. (2025). Gartner 2025 strategic technology trends – CIO roadmap. Retrieved from https://prolifics.com/usa/resource-center/blog/gartner-2025-strategic-technology-trends (date accessed: November 30, 2025).
3. AgilePoint. (2025). Gartner top strategic technology trends 2025. Retrieved from https://www.agilepoint.com/gartner-top-strategic-technology-trend-2025 (date accessed: November 28, 2025).
4. Deloitte. (2024). State of generative AI in the enterprise 2024. Retrieved from https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/content/state-of-generative-ai-in-enterprise.html (date accessed: November 23, 2025).
5. EU Artificial Intelligence Act. (n.d.). Article 13: Transparency and provision of information to deployers. Retrieved from https://artificialintelligenceact.eu/article/13/ (date accessed: November 21, 2025).

6. EU Artificial Intelligence Act. (n.d.). Article 50: Transparency obligations for providers and deployers of certain AI systems. Retrieved from https://artificialintelligenceact.eu/article/50/ (date accessed: November 20, 2025).
7. DSA Observatory. (2024, November 22). The regulation of recommender systems under the DSA: A transition from default to multiple and dynamic controls? Retrieved from https://dsa-observatory.eu/2024/11/22/the-regulation-of-recommender-systems-under-the-dsa-a-transition-from-default-to-multiple-and-dynamic-controls/ (date accessed: November 23, 2025).
8. Brainforge. (2025). How Netflix uses machine learning (ML) to create perfect recommendations. Retrieved from https://www.brainforge.ai/blog/how-netflix-uses-machine-learning-ml-to-create-perfect-recommendations (date accessed: November 18, 2025).
9. Qin, Z., Wu, J., Tang, R., & Zhao, W. (2023). Explainability and interpretability in concept and data drift: A systematic literature review. Algorithms, 18(7), 443. https://doi.org/10.3390/a18070443
10. Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2025). Investigating the duality of interpretability and explainability in machine learning. arXiv. https://doi.org/10.48550/arXiv.2503.21356
11. Ribeiro, F. N., Araújo, M., Gonçalves, P., & Benevenuto, F. (2025). On explaining recommendations with large language models: A review. Frontiers in Artificial Intelligence, 8, 11808143. https://doi.org/10.3389/frai.2025.11808143
12. DAI Group. (2025). Accelerator for LLM-enhanced GNN with product quantization and unified indexing. Retrieved from https://dai.sjtu.edu.cn/my_file/pdf/8543405a-e6d3-48de-89b9-f1a89e0a4ae9.pdf (date accessed: November 15, 2025).
13. Yang, Y., Li, H., & Wang, J. (2024). Topic-aware knowledge graph with large language models for interoperability in recommender systems. arXiv. https://doi.org/10.48550/arXiv.2412.20163
14. Wu, S., et al. (2020). APAN: Asynchronous propagation attention network for real-time temporal graph embedding. arXiv. https://doi.org/10.48550/arXiv.2011.11545
15. Raj, R., & Kulkarni, P. (2025). Neurosymbolic AI for explainable recommendations in frontend UI design: Bridging the gap between data-driven and rule-based approaches. IRJET, 11(5), V11I5107. https://doi.org/10.6084/m9.figshare.25019598
16. European Data Protection Supervisor. (2025). Neuro-symbolic artificial intelligence. Retrieved from https://www.edps.europa.eu/data-protection/technology-monitoring/techsonar/neuro-symbolic-artificial-intelligence_en(date accessed: November 10, 2025).
17. Zhang, Y., et al. (2020). Neural-symbolic reasoning over knowledge graph for multi-stage explainable recommendation. DLG@AAAI. https://doi.org/10.48550/arXiv.2001.08203
18. Ahmed, M., & Khan, R. (2025). Recommender systems based on neuro-symbolic knowledge graph embeddings encoding first-order logic rules. ResearchGate. https://doi.org/10.13140/RG.2.2.38436.39364
19. Piper Morgan. (2025). The knowledge graph transformation: From retrieval to reasoning. Retrieved from https://medium.com/building-piper-morgan/the-knowledge-graph-transformation-from-retrieval-to-reasoning-bd3b9048c537 (date accessed: November 22, 2025).
20. Showkath, S. (2025). Understanding knowledge graph stores: A comprehensive comparison. Retrieved from https://medium.com/@iamshowkath/understanding-knowledge-graph-stores-a-comprehensive-comparison-d7b9248c1ecd (date accessed: November 19, 2025).
21. Qin, Z., Zhang, Y., Tang, R., & Zhao, W. (2020). CAFE: Coarse-to-fine neural symbolic reasoning for explainable recommendation. arXiv. https://doi.org/10.48550/arXiv.2010.15620
22. Qin, Z. (2020). CAFE: Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation. Retrieved from http://zhouqin.info/pdf/CAFE.pdf (date accessed: November 12, 2025).
23. Zhou, X., Liu, J., & He, X. (2025). OR-LLM-Agent: Automating modeling and solving of operations research optimization problem with reasoning large language model. arXiv. https://doi.org/10.48550/arXiv.2503.10009
24. Databricks. (2025). A practical guide to building an online recommendation system. Retrieved from https://www.databricks.com/blog/guide-to-building-online-recommendation-system (date accessed: November 28, 2025).
25. Carleton College. (2007). Recommendation systems :: Data structures. Retrieved from https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/datastructures.html (date accessed: October 29, 2025).
26. GeeksforGeeks. (2025). System design Netflix | A complete architecture. Retrieved from https://www.geeksforgeeks.org/system-design/system-design-netflix-a-complete-architecture/ (date accessed: October 26, 2025).

27. Guo, Y., Zhang, Y., & Wang, X. (2024). What are we optimizing for? A human-centric evaluation of deep learning-based recommender systems. arXiv. https://doi.org/10.48550/arXiv.2401.11632

28. Karatas, E. (2025). Structured output generation in LLMs: JSON schema and grammar-based decoding. Retrieved from https://medium.com/@emrekaratas-ai/structured-output-generation-in-llms-json-schema-and-grammar-based-decoding-6a5c58b698a6 (date accessed: October 22, 2025).

29. SiliconFlow. (2025). JSON schema. Retrieved from https://docs.siliconflow.cn/en/userguide/guides/json-mode (date accessed: October 18, 2025).

30. Mehta, Y. (2025, November). PinSage: How Pinterest used graph neural networks to power billions of recommendations. Retrieved from https://levelup.gitconnected.com/pinsage-how-pinterest-used-graph-neural-networks-to-power-billions-of-recommendations-0fcb51f31109 (date accessed: November 25, 2025).

31. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 974–983. https://doi.org/10.1145/3219819.3219890

32. Zhang, J., Zhao, M., He, X., & Chua, T.-S. (2024). XRec: Large language models for explainable recommendation. arXiv. https://doi.org/10.48550/arXiv.2406.02377

33. Sun, W., Huang, Y., & Wu, L. (2025). RecMind: LLM-enhanced graph neural networks for personalized consumer recommendations. arXiv. https://doi.org/10.48550/arXiv.2509.06286

34. Wang, Y., & Lin, C. (2021). Knowledge-enhanced top-k recommendation in Poincaré ball. AAAI Conference on Artificial Intelligence, 35(7), 6236–6243. https://doi.org/10.1609/aaai.v35i7.16553

35. Zhang, L., Liu, Y., & Yu, H. (2025). Unleashing the latent reasoning power for sequential recommendation. arXiv. https://doi.org/10.48550/arXiv.2503.22675

36. Xu, R., Wang, L., & Li, Q. (2023). Turning dross into gold loss: Is BERT4Rec really better than SASRec? alphaXiv. https://doi.org/10.48550/arXiv.2309.07602

37. Yang, Z., & Zhang, Y. (2025). Enhancing recommendation explanations through user-centric refinement. arXiv. https://doi.org/10.48550/arXiv.2502.11721

38. Evidently AI. (2025). Normalized discounted cumulative gain (NDCG) explained. Retrieved from https://www.evidentlyai.com/ranking-metrics/ndcg-metric (date accessed: October 15, 2025).

39. Liu, Z., & Shen, C. (2022). Time-aware explainable recommendation via updating enabled online prediction. Entropy, 24(11), 1639. https://doi.org/10.3390/e24111639

40. Zhang, Y., et al. (2022). ExpScore: Learning metrics for recommendation explanation. Proceedings of the Web Conference 2022 (WWW '22). https://doi.org/10.1145/3485447.3512212

41. Zhang, H., Li, Y., & Zhao, X. (2024). Attention is not the only choice: Counterfactual reasoning for path-based explainable recommendation. IEEE Transactions on Knowledge and Data Engineering, 36(9), 10463173. https://doi.org/10.1109/TKDE.2023.3347710

42. Gorgo, L. (2025). Designing a real-time fraud detection system: Balancing accuracy and latency. Retrieved from https://leonidasgorgo.medium.com/designing-a-real-time-fraud-detection-system-balancing-accuracy-and-latency-7f80e8e70b4b (date accessed: October 10, 2025).

43. Müller, R., & Johnson, D. (2025). Neuro-symbolic AI for explainable decision-making in autonomous grid operations. Preprints. https://doi.org/10.20944/preprints202508.0747.v1