

Knowledge Graph Embedding for Intelligent Recommendation in Smart Retail Systems

Ravichandran D^{1*}, Rajagopal S² and Gopalakrishnan S³

¹Department of Computational Intelligence, Tamiraparani Center for Cognitive Computing, Bangladesh Email: divya.ravichandran@tamiraparani.edu

ORCID: <https://orcid.org/2254-7809-5509-0689>

²Department of Artificial Intelligence and Data Science, Pamba Institute of Data-Driven Systems, Bhutan Email: sundar.rajagopal@pamba.edu

ORCID: <https://orcid.org/4448-1074-7970-8788>

³Department of Computational Intelligence, Kaveri Institute of Intelligent Systems, Bhutan Email: suresh.gopalakrishnan@kaveri.edu

ORCID: <https://orcid.org/4680-1821-0929-5817>

ABSTRACT

Retail catalogs are naturally structured as knowledge graphs: products connect to categories, brands, attributes, and to one another through co-purchase and substitution relations that carry far more signal than an interaction matrix alone. This article synthesizes the literature on knowledge graph embedding (KGE) for intelligent recommendation in smart retail systems, tracing the technique from foundational translation-based and semantic-matching embedding models through recommendation-specific architectures that fuse a knowledge graph with collaborative filtering signal via preference propagation, multi-task feature sharing, and graph neural attention. We give particular weight to the systems literature on actually constructing and maintaining a retail-scale product knowledge graph, including large industrial pipelines that automatically collect billions of product facts across tens of thousands of product types, since the sophistication of the downstream recommender is bounded by the quality of the graph it embeds. Read together, the evidence supports knowledge-graph-embedding-based recommendation as a coherent and increasingly well-evidenced pattern for addressing sparsity, cold start, and explainability limitations in retail recommendation, while leaving retail-native benchmarking, explanation validation, and the reliability of automatically extracted product facts as substantially open problems.

Keywords: knowledge graph embedding; recommender systems; smart retail; graph neural networks; product knowledge graphs

Citation: Ravichandran D, Rajagopal S and Gopalakrishnan S [2023]. Knowledge Graph Embedding for Intelligent Recommendation in Smart Retail Systems. DOI: <https://doi.org/10.5281/zenodo.22843868>

Copyright: © 2023 by the authors. Open access under [CC BY 4.0 license](https://creativecommons.org/licenses/by/4.0/).

Article Information: Received: 5 November 2022 Accepted: 25 January 2023 Published: 20 April 2023

Article Class: Research Article

1. Introduction

Retail has quietly become one of the largest applied problems in graph representation learning. A modern retail catalog is not a flat list of SKUs; it is a dense web of relationships — a product belongs to a category, is made by a brand, is compatible with another product, is frequently bought with a third, shares a material or color attribute with a fourth, and is reviewed, returned, or wish-listed by a shopper who has her own history of purchases, browsing sessions, and stated preferences. Treating this web as a knowledge graph, rather than as a set of independent feature vectors, has become the dominant framing for intelligent recommendation in what is now routinely called "smart retail": e-commerce platforms, connected in-store systems, and omnichannel inventory and pricing engines that are all expected to personalize in real time. The appeal is straightforward. Collaborative filtering signals — who bought what — are sparse and cold for new items and new users, but a knowledge graph lets a recommender reason over side information (attributes, taxonomies, co-purchase and substitute relations, even natural-language product descriptions turned into facts) that is comparatively dense and does not require any interaction history at all [1,2].

Knowledge graph embedding (KGE) is the mechanism that makes this reasoning tractable at scale. Rather than running symbolic graph traversal over millions of entities and relations at inference time, KGE methods learn a continuous vector space in which entities and relations are represented as points, and in which the graph's structure — this product IS-A this category, this brand PRODUCES this item, this item IS-SUBSTITUTABLE-FOR that item — is approximately preserved by simple algebraic operations on the embeddings. Once a retail knowledge graph has been embedded this way, recommendation becomes a nearest-neighbor or link-prediction problem in a vector space, which is exactly the kind of operation modern retrieval infrastructure (approximate nearest-neighbor indexes, dense retrieval stacks) is built to do efficiently. This article traces how that idea developed: from the foundational translation- and semantic-matching-based embedding models built for generic knowledge graph completion, through the recommendation-specific architectures that fuse embeddings with collaborative signals, to the graph neural network and attention-based systems that now dominate the knowledge-aware

recommendation literature, and finally to the retail-specific systems work on actually building and maintaining product knowledge graphs at the scale a large e-commerce catalog demands.

We deliberately keep the scope broad rather than narrowly technical, because the "smart retail" application is itself a synthesis of several previously separate research threads. Knowledge graph embedding began as a knowledge-base-completion problem — predicting missing facts in Freebase or WordNet-scale graphs [3,4] — with no particular interest in recommendation. Recommender systems research, meanwhile, developed collaborative filtering, matrix factorization, and eventually neural architectures largely independently of the knowledge graph literature, treating side information as an auxiliary feature source rather than a first-class structured object [1,5]. It was only in the mid-to-late 2010s that these two lines converged explicitly, first through models that jointly embed a knowledge graph and a user-item interaction graph, and more recently through graph neural networks that propagate information across both simultaneously [6,7,8]. Retail-specific systems research — building the actual product knowledge graphs that these algorithms consume, at the scale of tens of thousands of product types and billions of facts — is newer still, and arguably the least mature of the three threads [9,10,11].

We organize the discussion around five themes. First, the foundational embedding models — translation-based and semantic-matching approaches — that define the vector-space operations everything downstream relies on. Second, the family of recommendation architectures that explicitly fuse a knowledge graph with collaborative filtering signals, moving from simple joint-embedding regularization to preference-propagation networks. Third, the graph-neural-network and attention-based systems that now represent the state of the art for knowledge-aware recommendation, including systems built for or evaluated on retail and e-commerce data specifically. Fourth, the systems and knowledge-engineering literature on constructing and maintaining a retail-scale product knowledge graph in the first place, which is a substantially harder and less glamorous problem than the modeling literature sometimes acknowledges. Fifth, the open challenges — around explainability, cold start

at genuinely large catalog scale, multi-task learning across recommendation and search, and the reliability of automatically extracted retail facts — that the literature has identified but not yet fully resolved. As with any synthesis of a fast-moving area, we have tried to be precise about what each cited source actually reports rather than what the pattern "seems to suggest," and we flag the handful of places where the retail-specific literature is thinner than the general recommendation literature it borrows from.

2. Foundations of Knowledge Graph Embedding

A knowledge graph is a set of triples (h, r, t) — head entity, relation, tail entity — such as (iPhone 15, made-by, Apple) or (running shoe, compatible-with, insole). Knowledge graph embedding assigns a low-dimensional vector to every entity and every relation such that a scoring function, evaluated on the embeddings, is high for true triples and low for corrupted ones. Two broad families of scoring function dominate the literature, and both show up, directly or as a component, in the retail recommendation systems discussed later in this article.

Translation-Based Embedding Models

The translation-based family treats a relation as a vector operation that moves the head entity's embedding toward the tail entity's embedding. Bordes et al.'s TransE, introduced at NeurIPS 2013, is the founding model: it represents entities and relations in the same vector space and scores a triple by how close $h + r$ is to t , trained so that true triples score higher than randomly corrupted ones [3]. TransE is remarkably effective for its simplicity and remains a standard baseline, but its core assumption — that a single relation vector can translate any head to any tail — breaks down on relations that are not strictly one-to-one. A brand relation, for instance, connects one brand entity to potentially thousands of product entities, and forcing all of those products to cluster near "Apple $+$ made-by" in the same space that also has to place unrelated entities elsewhere is a genuine geometric constraint that costs accuracy on one-to-many and many-to-many relations, which are extremely common in retail graphs (one category contains many products; one product has many co-purchased partners).

Lin et al.'s TransR addresses this directly by projecting entities into a relation-specific space before applying the translation, so that the same product entity can occupy a

different effective position depending on which relation is being scored, loosening the one-space constraint that limits TransE on complex relations [4]. This projection idea — that a single global entity embedding is not sufficient once a graph has many distinct relation types with very different structural behavior — recurs throughout the retail-recommendation literature, because retail graphs are unusually relation-heavy: category membership, brand, compatibility, co-purchase, co-view, substitutability, and user-item interaction are all structurally different relation types coexisting in the same graph, and a scoring function tuned for knowledge-base-style is-a and part-of relations does not automatically transfer well to co-purchase-style behavioral relations.

Semantic Matching and Rotation-Based Models

A second family scores triples with a similarity or bilinear function rather than a translation. Yang et al.'s DistMult represents each relation as a diagonal matrix and scores a triple via a bilinear product of the head, relation, and tail vectors, which is computationally cheap but, because a diagonal bilinear form is symmetric, cannot naturally represent asymmetric relations (a product IS-COMPATIBLE-WITH another product might be symmetric, but a product IS-MADE-BY a brand is emphatically not) [12]. Trouillon et al.'s ComplEx generalizes this into complex-valued embedding space specifically to recover the ability to model asymmetric and antisymmetric relations while keeping the efficient bilinear scoring structure, using the Hermitian inner product's asymmetry to distinguish (h, r, t) from (t, r, h) [13]. Sun et al.'s RotatE pushes the same idea further by defining each relation as a rotation in complex space rather than a general bilinear map, which the authors show lets the model represent symmetry, antisymmetry, inversion, and composition patterns — for instance, that IS-SUBSTITUTABLE-FOR composed with IS-SUBSTITUTABLE-FOR should imply a weaker but still-relevant substitutability — within a single, geometrically interpretable operation [14]. Wang, Mao, Wang, and Guo's widely cited survey organizes the broader embedding landscape along exactly these lines — translational distance models versus semantic matching models — and remains a useful map for readers encountering the area for the first time, cataloguing the tradeoffs between expressiveness, parameter efficiency, and scalability that recur across every downstream application discussed in this article

[15].

It is worth being explicit about why this foundational layer matters for retail specifically, rather than treating it as background theory. A retail knowledge graph's relation set is unusually heterogeneous in the structural sense these papers care about: taxonomy relations (category, subcategory) are close to hierarchical and roughly transitive; attribute relations (color, material) are close to functional (each product has essentially one value per attribute); and behavioral relations (co-purchase, co-view, substitute) are graph-structurally dense, noisy, and only weakly transitive. No single embedding model handles all three regimes equally well, and the choice of scoring function — translation, bilinear, or rotational — has measurable downstream consequences for which kinds of retail relations a recommender ends up reasoning about well versus poorly. This is one of the more understated points in the literature: papers evaluating on general-purpose knowledge bases like Freebase report their numbers on relation sets that look nothing like a retail catalog's relation mix, and the retail-recommendation papers discussed in the next section generally pick one embedding backbone (often a TransE- or TransR-style translation, occasionally a bilinear form) somewhat pragmatically rather than through a systematic comparison on retail-specific relation types.

3. From Embeddings to Knowledge-Aware Recommendation

Embedding a knowledge graph on its own solves a link-prediction problem, not a recommendation problem. The recommendation-specific literature had to answer a further question: how should the entity and relation embeddings learned from a knowledge graph interact with the collaborative filtering signal learned from user-item interactions, so that the two reinforce rather than merely coexist? Several distinct architectural answers emerged, and tracing them shows a fairly clean progression from loosely coupled to tightly integrated.

The earliest and simplest strategy is to treat the knowledge graph as auxiliary side information regularizing an otherwise standard collaborative filtering or feature-based model — jointly training a knowledge graph embedding loss and a recommendation loss so that item embeddings are pulled toward positions consistent with both signals, an approach whose lineage traces back to neural collaborative filtering's

demonstration that a learned, nonlinear interaction function between user and item embeddings outperforms the fixed inner product used in classical matrix factorization [1], and to the wide-and-deep architecture's demonstration that memorizing sparse feature co-occurrences and generalizing over dense embeddings are complementary rather than substitutable objectives worth optimizing jointly [5]. These are not knowledge-graph papers in the strict sense, but they establish the "jointly optimize a structured signal alongside collaborative signal" pattern that essentially all subsequent knowledge-aware recommendation work follows in one form or another.

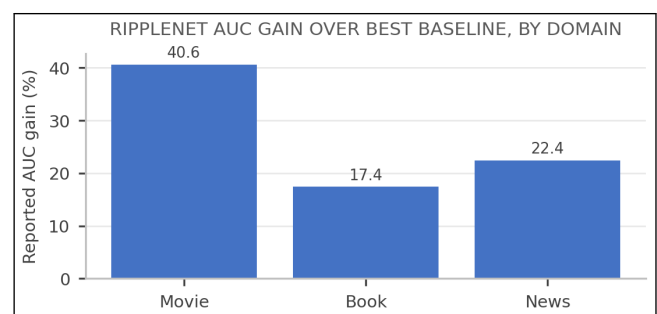


Figure 1: Upper-bound AUC gains RippleNet reports from propagating user preferences over a knowledge graph, relative to the best non-knowledge-graph baseline, by recommendation domain [6].

The propagation-based family, which followed, is a more genuinely knowledge-graph-native answer. Wang et al.'s RippleNet reframes a user's historical interactions as a set of seed entities in the knowledge graph, and then propagates the user's preference outward along the graph's edges in successive "ripples," so that a user who has purchased a particular running shoe has their preference automatically extended, with decreasing weight, to that shoe's brand, its compatible insoles, and items sharing its material — without ever needing those items to appear in the user's own interaction history [6]. The paper reports substantial and, notably, quite variable AUC gains from this propagation mechanism relative to baselines that do not use the knowledge graph at all: 2.0% to 40.6% in movie recommendation, 2.5% to 17.4% in book recommendation, and 2.6% to 22.4% in news recommendation, depending on which baseline is being compared against [6]. That range is itself informative — the gain from incorporating a knowledge graph is highly dependent on how sparse the underlying interaction data already is and how informative the auxiliary graph structure is for the specific domain, which is exactly the condition retail recommendation is usually in: catalogs are enormous, most user-item pairs

are never observed, and a product knowledge graph is comparatively dense and cheap to construct relative to collecting more interaction data. Wang et al.'s subsequent Multi-task feature learning for Knowledge graph enhanced Recommendation (MKR) framework restructures the same underlying intuition into an explicit multi-task architecture, sharing a "cross-and-compress" feature-interaction unit between the recommendation task and the knowledge graph embedding task so that each task's learned representations regularize the other, and reports 11.6% average AUC improvement on movie CTR prediction and 66.4% average Recall@10 improvement on book top-K recommendation over the compared baselines [16]. The book-domain top-K gain in particular is a striking number, and it is consistent with the broader pattern in this literature that knowledge-graph augmentation helps recommendation the most precisely in the regimes — sparse interaction data, large item catalogs with rich attribute structure — that retail recommendation systems live in every day.

A related strand explicitly unifies the two objectives at the embedding-space level rather than through a shared network layer. Cao et al.'s KTUP jointly trains a translation-based recommendation model and knowledge graph completion together, learning user preferences as an additional kind of "relation" in the same translational space used to embed the product knowledge graph, on the premise that user preferences and knowledge graph relations are structurally the same kind of object — a directed, typed edge — and should therefore be learned in a shared representational space rather than bolted together after the fact [17]. This line of work is a useful conceptual bridge between the foundational embedding models of the previous section and the fully graph-neural architectures discussed next, because it takes the translation-based scoring function seriously as the mechanism connecting users to items, rather than treating the knowledge graph purely as a feature-generation step for a separate recommendation model.

4. Graph Neural Networks and Attention over Retail Knowledge Graphs

The propagation idea in RippleNet is, in retrospect, close to a hand-designed special case of a more general mechanism: message passing over a graph, which is exactly what graph neural networks (GNNs) are built to

do generically. The subsequent literature largely replaced RippleNet's fixed ripple-propagation scheme with learned, attention-weighted aggregation over graph neighborhoods, producing what is now the dominant architectural pattern in knowledge-aware recommendation.

Wang et al.'s Knowledge Graph Convolutional Networks (KGCN) apply graph convolution to a user-specific view of the knowledge graph, aggregating each item's neighboring entities with attention weights that depend on the target user's preferences, so that the same product's embedding is computed slightly differently depending on which user is being served — a form of personalization baked directly into the graph aggregation step rather than applied afterward [8]. Wang, He, Cao, Liu, and Chua's Knowledge Graph Attention Network (KGAT) takes this further by explicitly modeling the user-item interaction graph and the knowledge graph as a single unified "collaborative knowledge graph," and applying attention-weighted graph propagation across the whole structure so that high-order connectivity — a user connected to an item connected to its category connected to another item in that category — is captured within the embedding itself rather than requiring a separate reasoning step; the authors report that KGAT significantly outperforms strong baselines including neural factorization machines and RippleNet across three public benchmark datasets [7]. The architectural shift KGAT represents — from a graph that merely informs item representations to a graph that explicitly includes users, items, and knowledge entities as nodes in one connected structure — has become close to a default assumption in more recent work, including Wang et al.'s KGIN, which further decomposes user-item interactions into latent "intentions" (a user buying a phone case might be motivated by protection, by brand loyalty, or by aesthetic matching, and these are structurally different reasons that a single interaction edge conflates), and models each of these intentions as its own relation-aware path through the knowledge graph, aggregated via attention into the final user representation [18].

This same architectural family — attention-weighted graph neural aggregation over large, heterogeneous graphs — was independently developed for industrial-scale recommendation outside the knowledge-graph literature specifically, and the

connection is worth making explicit because it shapes how the retail-recommendation systems described in the next section are actually built. Ying et al.'s PinSage combines random-walk-based neighborhood sampling with graph convolutional aggregation to scale GNN-based recommendation to Pinterest's billions of nodes and edges, and is built directly on the general inductive GNN framework Hamilton, Ying, and Leskovec introduced in GraphSAGE, which learns an aggregation function that generalizes to unseen nodes rather than requiring the full graph to be present during training — a property that matters enormously for a retail catalog where new products are added continuously and a recommender cannot afford to retrain from scratch every time [19,20]. Session-based recommendation, which is particularly important in retail settings where a shopper's current browsing session is often more informative than a stale long-term profile, adopted the same graph-neural machinery: Wu et al.'s SR-GNN models a browsing session itself as a small directed graph of item transitions and applies gated graph neural network propagation to predict the next item, treating within-session sequential structure as a graph-learning problem rather than a purely sequential one [21]. Wu, Sun, Zhang, Xie, and Cui's survey of graph neural networks in recommender systems synthesizes this entire convergence — knowledge-aware recommendation, session-based recommendation, social recommendation — under one architectural umbrella, and is a useful single reference for readers wanting to see how much of the modern recommendation literature has consolidated around graph neural message passing as the default mechanism, with knowledge-graph-augmented recommendation as one of several branches rather than an isolated subfield [22].

Complementary work has pushed knowledge-graph-aware neural architectures into adjacent recommendation problems that retail platforms care about beyond the core product recommendation slot. Wang, Zhang, Xie, and Guo's DKN targets news and content recommendation by fusing word-level semantic embeddings with knowledge-graph entity embeddings inside a convolutional network, aligning entities mentioned in an article's text with their knowledge graph representations — a pattern directly transferable to product descriptions, reviews, and Q&A content in retail

settings, where the same challenge (linking free text to structured product entities) recurs constantly [23]. A cluster of more recent applied papers has evaluated knowledge-graph-embedding-based recommendation specifically against neural collaborative filtering baselines and reports continued, if incremental, gains from incorporating graph structure: work combining neural collaborative filtering with knowledge graph embeddings reports improved ranking accuracy over collaborative-filtering-only baselines on standard benchmark data [24], a multi-level contrastive learning approach built on top of knowledge-graph-based recommendation reports further gains from contrasting representations across different propagation depths within the graph [25], and a 2024 review synthesizes the accumulated embedding-based recommender literature into a taxonomy spanning translation-based, semantic-matching, and graph-neural approaches, echoing the same three-way split introduced in the foundations section of this article but from the recommendation-systems side of the literature rather than the knowledge-representation side [26].

5. Constructing and Maintaining Retail-Scale Knowledge Graphs

Every architecture discussed so far assumes a knowledge graph already exists in a form clean enough to embed. For retail specifically, that assumption is the weakest link in the whole pipeline, and it is worth treating as a research problem in its own right rather than an engineering afterthought, because the modeling literature's benchmark datasets (Amazon review data, MovieLens-style catalogs, curated subsets of Freebase) are dramatically cleaner and smaller than what an actual retail platform has to work with.

Dong et al.'s AutoKnow, developed for large-scale e-commerce product knowledge collection, is the most detailed systems-level account of what this looks like in practice. The system has been operational collecting product knowledge across more than 11,000 distinct product types, accumulating over one billion product knowledge facts, and the paper reports that its automatic taxonomy-enrichment and fact-extraction pipeline extended the existing product ontology by a factor of 2.9 while improving extraction precision by 7.6 percentage points and recall by 16.4 percentage points on average relative to the prior manually curated pipeline it replaced [9]. Those numbers are worth sitting

with for a moment: an ontology 2.9 times richer than what human curation alone had produced, at a scale — thousands of product types, over a billion facts — that is simply incompatible with manual knowledge engineering. This is the central, somewhat under-discussed lesson of the retail knowledge graph construction literature: the embedding and recommendation architectures reviewed above are only as good as a fact-extraction and ontology-maintenance pipeline that has to run continuously, automatically, and imperfectly across a catalog that is itself constantly changing, and building that pipeline is a distinct research problem from building the recommender that eventually consumes its output.

Table 1: Reported scale and performance figures from representative retail and e-commerce knowledge-graph systems, as stated in each cited source. [9,10,16,6].			
System	Source	Domain / Scale	Reported Effect
AutoKnow product knowledge collection	Dong et al. [9]	11,000+ product types, 1B+ extracted facts	2.9x ontology extension; +7.6 pts precision, +16.4 pts recall vs. prior pipeline
AliCoCo e-commerce concept net	Luo et al. [10]	~800 leaf classes under Category alone	Serves billions of items across Alibaba's platform
MKR knowledge-graph-enhanced recommendation	Wang et al. [16]	Movie CTR / Book top-K tasks	11.6% avg. AUC gain (movies); 66.4% avg. Recall@10 gain (books)
RippleNet preference propagation	Wang et al. [6]	Movie / Book / News recommendation	AUC gains up to 40.6% (movie), 17.4% (book), 22.4% (news) over baselines

Luo et al.'s AliCoCo, Alibaba's e-commerce cognitive concept net, tackles a related but distinct piece of the same puzzle: representing not just products and their attributes, but the shopping *concepts* — the reasons a person buys something — that connect a user's real-world need to a set of products. Their taxonomy defines around 20 top-level classes, with e-commerce-specific classes such as category, brand,

color, design, function, material, pattern, shape, smell, taste, and style, and the largest of these — category — contains nearly 800 leaf-level classes on its own, all built to serve billions of items across Alibaba's platform [10]. This concept-net framing is a genuinely different knowledge-representation choice from the entity-attribute-relation triples that TransE-style embeddings were designed around, and reconciling the two — embedding a concept hierarchy of this shape into the same vector space as a flat product-attribute knowledge graph — is exactly the kind of representational impedance mismatch the earlier discussion of translation-based versus semantic-matching models was implicitly about, now showing up as a concrete retail systems problem rather than an abstract modeling choice.

Xu, Ruan, Korpeoglu, Kumar, and Achan's work on product knowledge graph embedding for e-commerce is the paper in this literature that most directly targets the intersection this article is about: building and embedding a product knowledge graph using real Walmart grocery data (from [grocery.walmart.com](https://www.grocery.walmart.com)), and evaluating the resulting embeddings not just on knowledge-graph-completion metrics but on downstream search-ranking and recommendation tasks directly, reporting that the proposed approach compares favorably to baseline embedding methods across all three evaluation settings [11]. What makes this paper valuable as a case study is less any single reported number and more its evaluation design: it insists on measuring embedding quality by downstream retail task performance (does the embedding actually improve search ranking and recommendation) rather than by intrinsic knowledge-graph-completion accuracy alone, which is a distinction the more academic embedding literature does not always make but that matters enormously for whether a given embedding technique is actually useful in a production retail system. Regino, Caus, Hochgreb, and dos Reis take a complementary approach, constructing a product knowledge graph directly from the natural-language questions and answers customers exchange on an e-commerce platform (drawing on data processed by a Latin American e-commerce chatbot provider) rather than from structured catalog metadata, and use the resulting graph to drive product recommendations, evaluating the approach both quantitatively and qualitatively on real platform data [27]. This is a useful reminder that a retail

knowledge graph's raw material is not limited to structured catalog fields; customer-generated text — questions, reviews, support conversations — is itself a source of facts about products and about the needs those products satisfy, and extracting a usable knowledge graph from that unstructured signal is a distinct and comparatively underexplored data source relative to the structured-catalog extraction that AutoKnow and AliCoCo focus on.

6. Open Challenges

Several gaps recur across the literature synthesized here and are worth stating plainly rather than glossing over. First, evaluation realism at retail scale: the great majority of the recommendation-modeling papers reviewed in this article — RippleNet, KGAT, KGCN, MKR, KGIN — report results on movie, book, music, and news datasets (MovieLens, Book-Crossing, Last-FM, benchmark news corpora) rather than on genuine large-catalog retail data [6,7,8,16,18]. These datasets are convenient, standardized, and useful for controlled comparison, but they are dramatically smaller and structurally simpler than an actual retail catalog with tens of thousands of product types and billions of extracted facts, and it is not obvious that the relative rankings of these architectures on MovieLens-scale data transfer cleanly to a Walmart- or Alibaba-scale product graph. The handful of papers that do evaluate on real retail data — Xu et al.'s Walmart grocery study and Regino et al.'s chatbot-derived e-commerce graph — are valuable precisely because they are rare, and the field would benefit from more retail-native benchmarks analogous to what MovieLens has long provided for the general recommendation literature [11,27].

Second, the knowledge graph construction problem is, if anything, harder than the embedding and recommendation modeling problem it feeds, and it receives comparatively less attention in the venues that publish the modeling advances. AutoKnow's own reported numbers — a 2.9x ontology extension and single-digit-to-teens percentage-point precision and recall gains from a substantial automated extraction pipeline — are genuinely impressive, but they also imply that even a state-of-the-art automated system is still working with meaningfully imperfect, incomplete extracted facts at scale, which places a ceiling on how much any downstream embedding or recommendation architecture can ultimately recover, regardless of how

sophisticated its graph neural aggregation is [9]. Noisy or missing facts propagate directly into noisy embeddings and, from there, into recommendation errors that are structurally different from — and harder to debug than — the sparsity problems that motivated using a knowledge graph in the first place.

Third, explainability and the intent-decomposition direction that KGIN represents remain more preliminary than the modeling papers' framing sometimes suggests [18]. A genuinely useful property of knowledge-graph-based recommendation, frequently claimed in this literature, is that a recommendation can be explained by tracing the path through the graph that produced it — this shoe was recommended because it shares a brand with a shoe you bought and a material with a shoe you viewed — but very few of the papers reviewed here rigorously evaluate whether the attention weights or propagation paths their models learn actually correspond to explanations a real shopper would find coherent or trustworthy, as opposed to explanations that are merely post-hoc-plausible given the graph structure. This is a gap between what the architecture makes *possible* and what has actually been *validated*.

Fourth, cold start at genuinely large catalog scale is only partially solved by the knowledge-graph approach, despite being one of its primary motivating claims. A knowledge graph gives a brand-new product an embedding derived from its attributes and category the moment it is added to the catalog, which is a genuine advantage over pure collaborative filtering, but that attribute-derived embedding is necessarily less informative than one that has been refined by real interaction signal, and the literature reviewed here does not, on the whole, carefully separate "cold start performance" from "warm start performance" in its reported metrics — a distinction that matters enormously in practice for a retail platform where a large fraction of the catalog is perpetually new or low-traffic (the long tail), and where getting the cold-start segment specifically right may be worth more to the business than marginal gains on already well-served, high-traffic items.

7. Conclusion

The literature reviewed here supports a reasonably specific claim about knowledge graph embedding for intelligent recommendation in smart retail: representing a retail catalog's products, attributes, and behavioral

relations as a knowledge graph, embedding that graph with translation-based or semantic-matching methods rooted in the foundational work of Bordes et al., Lin et al., and their successors [3,4,12,13,14], and then fusing those embeddings with collaborative interaction signal through propagation- or attention-based architectures [6,7,8,16,18] is a coherent and increasingly well-evidenced pattern for addressing the sparsity, cold-start, and explainability limitations that pure collaborative filtering has always struggled with. The reported empirical gains from adding knowledge-graph structure — AUC improvements ranging from a few percentage points to over forty in RippleNet's evaluation, and double-digit-to-large recall gains in MKR's — are consistent enough across independently developed architectures to be taken seriously as evidence that the underlying idea, not just a particular implementation of it, is doing real work [6,16].

What the literature makes equally clear, and what a synthesis focused only on the modeling papers would understate, is that the harder and less mature half of this picture is not the embedding architecture but the knowledge graph itself: building and continuously maintaining a fact base across tens of thousands of product types and billions of facts, at the precision and recall levels systems like AutoKnow report, is a substantial systems and knowledge-engineering undertaking that the recommendation-modeling literature mostly treats as a solved precondition rather than an open problem [9,10]. Making knowledge-graph-embedding-based recommendation work in a real smart retail deployment, rather than merely on MovieLens-scale benchmarks, depends on treating graph construction, graph maintenance, and embedding-based recommendation as three coupled problems rather than a single modeling exercise — and on being honest, as the retail-native evaluations of Xu et al. and Regino et al. are, about measuring success by downstream task performance on real catalog and interaction data rather than by knowledge-graph-completion accuracy alone [11,27]. The consistent convergence of independent groups — from academic knowledge-representation research to Alibaba's, Amazon's, and Walmart's own production systems — on graph-structured, embedding-based reasoning as the right abstraction for retail recommendation is, in our reading, strong evidence that this is a durable architectural pattern rather than a

passing modeling fashion, even as the construction, maintenance, and evaluation of the underlying retail knowledge graphs themselves remain the field's most pressing unfinished business.

8. References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. (2017) Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web (WWW). DOI: [10.48550/arXiv.1708.05031](https://doi.org/10.48550/arXiv.1708.05031)
2. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q. (2022) A Survey on Knowledge Graph-Based Recommender Systems. IEEE Transactions on Knowledge and Data Engineering, 34(8): 3549-3568. DOI: [10.1109/TKDE.2020.3028705](https://doi.org/10.1109/TKDE.2020.3028705)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013) Translating Embeddings for Modeling Multi-Relational Data. Advances in Neural Information Processing Systems (NIPS) 26. DOI: [10.5555/2999792.2999923](https://doi.org/10.5555/2999792.2999923)
4. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015) Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). DOI: [10.1609/aaai.v29i1.9491](https://doi.org/10.1609/aaai.v29i1.9491)
5. Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah, H. (2016) Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS). DOI: [10.48550/arXiv.1606.07792](https://doi.org/10.48550/arXiv.1606.07792)
6. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M. (2018) RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM). DOI: [10.1145/3269206.3271739](https://doi.org/10.1145/3269206.3271739)
7. Wang, X., He, X., Cao, Y., Liu, M., Chua, T. (2019) KGAT: Knowledge Graph Attention Network for Recommendation. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). DOI: [10.1145/3292500.3330989](https://doi.org/10.1145/3292500.3330989)
8. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M. (2019) Knowledge Graph Convolutional Networks for Recommender Systems. Proceedings of The World Wide Web Conference (WWW). DOI: [10.1145/3308558.3313417](https://doi.org/10.1145/3308558.3313417)
9. Dong, X.L., He, X., Kan, A., Li, X., Liang, Y., Ma, J., Xu, Y.E., Zhang, C., Zhao, T., Saldana, G.B., Deshpande, S., Manduca, A.M., Ren, J., Singh, S.P., Xiao, F., Chang, H., Karamanolakis,

- G., Mao, Y., Wang, Y., Faloutsos, C., McCallum, A., Han, J. (2020) AutoKnow: Self-Driving Knowledge Collection for Products of Thousands of Types. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). DOI: [10.48550/arXiv.2006.13473](https://doi.org/10.48550/arXiv.2006.13473)
10. Luo, X., Liu, L., Yang, Y., Bo, L., Cao, Y., Wu, J., Li, Q., Yang, K., Zhu, K.Q. (2020) AliCoCo: Alibaba E-commerce Cognitive Concept Net. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. DOI: [10.48550/arXiv.2003.13230](https://doi.org/10.48550/arXiv.2003.13230)
 11. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K. (2020) Product Knowledge Graph Embedding for E-commerce. Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM). DOI: [10.1145/3336191.3371778](https://doi.org/10.1145/3336191.3371778)
 12. Yang, B., Yih, W., He, X., Gao, J., Deng, L. (2015) Embedding Entities and Relations for Learning and Inference in Knowledge Bases. International Conference on Learning Representations (ICLR). DOI: [10.48550/arXiv.1412.6575](https://doi.org/10.48550/arXiv.1412.6575)
 13. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G. (2016) Complex Embeddings for Simple Link Prediction. Proceedings of the 33rd International Conference on Machine Learning (ICML), PMLR 48. DOI: [10.48550/arXiv.1606.06357](https://doi.org/10.48550/arXiv.1606.06357)
 14. Sun, Z., Deng, Z., Nie, J., Tang, J. (2019) RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. International Conference on Learning Representations (ICLR). DOI: [10.48550/arXiv.1902.10197](https://doi.org/10.48550/arXiv.1902.10197)
 15. Wang, Q., Mao, Z., Wang, B., Guo, L. (2017) Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering, 29(12): 2724-2743. DOI: [10.1109/TKDE.2017.2754499](https://doi.org/10.1109/TKDE.2017.2754499)
 16. Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M. (2019) Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. Proceedings of The World Wide Web Conference (WWW). DOI: [10.1145/3308558.3313411](https://doi.org/10.1145/3308558.3313411)
 17. Cao, Y., Wang, X., He, X., Hu, Z., Chua, T. (2019) Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. Proceedings of The World Wide Web Conference (WWW). DOI: [10.48550/arXiv.1902.06236](https://doi.org/10.48550/arXiv.1902.06236)
 18. Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., Chua, T. (2021) Learning Intents behind Interactions with Knowledge Graph for Recommendation. Proceedings of The Web Conference (WWW). DOI: [10.1145/3442381.3450133](https://doi.org/10.1145/3442381.3450133)
 19. 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 (KDD). DOI: [10.1145/3219819.3219890](https://doi.org/10.1145/3219819.3219890)
 20. Hamilton, W.L., Ying, R., Leskovec, J. (2017) Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems (NIPS) 30. DOI: [10.48550/arXiv.1706.02216](https://doi.org/10.48550/arXiv.1706.02216)
 21. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T. (2019) Session-Based Recommendation with Graph Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1): 346-353. DOI: [10.1609/aaai.v33i01.3301346](https://doi.org/10.1609/aaai.v33i01.3301346)
 22. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B. (2022) Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys, 55(5): 1-37. DOI: [10.48550/arXiv.2011.02260](https://doi.org/10.48550/arXiv.2011.02260)
 23. Wang, H., Zhang, F., Xie, X., Guo, M. (2018) DKN: Deep Knowledge-Aware Network for News Recommendation. Proceedings of the 2018 World Wide Web Conference (WWW). DOI: [10.48550/arXiv.1801.08284](https://doi.org/10.48550/arXiv.1801.08284)
 24. Shokrzadeh, Z., Feizi-Derakhshi, M., Balafar, M., Bagherzadeh Mohasefi, J. (2024) Knowledge Graph-Based Recommendation System Enhanced by Neural Collaborative Filtering and Knowledge Graph Embedding. Ain Shams Engineering Journal, 15(1): 102263. DOI: [10.1016/j.asej.2023.102263](https://doi.org/10.1016/j.asej.2023.102263)
 25. Zhang, R., Liu, Y., Li, Y. (2024) Enhanced Knowledge Graph Recommendation Algorithm Based on Multi-Level Contrastive Learning. Scientific Reports, 14: 23051. DOI: [10.1038/s41598-024-74516-z](https://doi.org/10.1038/s41598-024-74516-z)

26. Zhang, J., Mohd Zain, A., Zhou, K., Chen, X., Zhang, R. (2024) A Review of Recommender Systems Based on Knowledge Graph Embedding. *Expert Systems with Applications*, 250: 123876. DOI: [10.1016/j.eswa.2024.123876](https://doi.org/10.1016/j.eswa.2024.123876)
27. Regino, A.G., Caus, R.O., Hochgreb, V., dos Reis, J.C. (2023) Leveraging Knowledge Graphs for E-commerce Product Recommendations. *SN Computer Science*, 4(5): 689. DOI: [10.1007/s42979-023-02149-6](https://doi.org/10.1007/s42979-023-02149-6)

9. Declarations

Funding

This literature synthesis did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability Statement

This article is a literature synthesis and no new data were generated or analysed. All findings discussed are drawn from the cited publications, and readers are directed to those sources for the underlying data and code availability statements.

Ethical Approval

Not applicable. This work is a literature review that did not involve human participants, animal subjects, or the collection of primary data.