International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Uncertainty-Aware Knowledge Graph Recommendation for Content Recognition Under Noisy Inputs

Sanjib Kumar Swain[1]

Submitted: 01/11/2024 **Revised**: 18/12/2024 **Accepted**: 25/12/2024

Keywords: *Uncertainty-Aware Recommendation, Knowledge Graph Propagation, Uncertainty Quantification, Content Recognition, Noisy Inputs, Confidence Propagation.*

1 Introduction

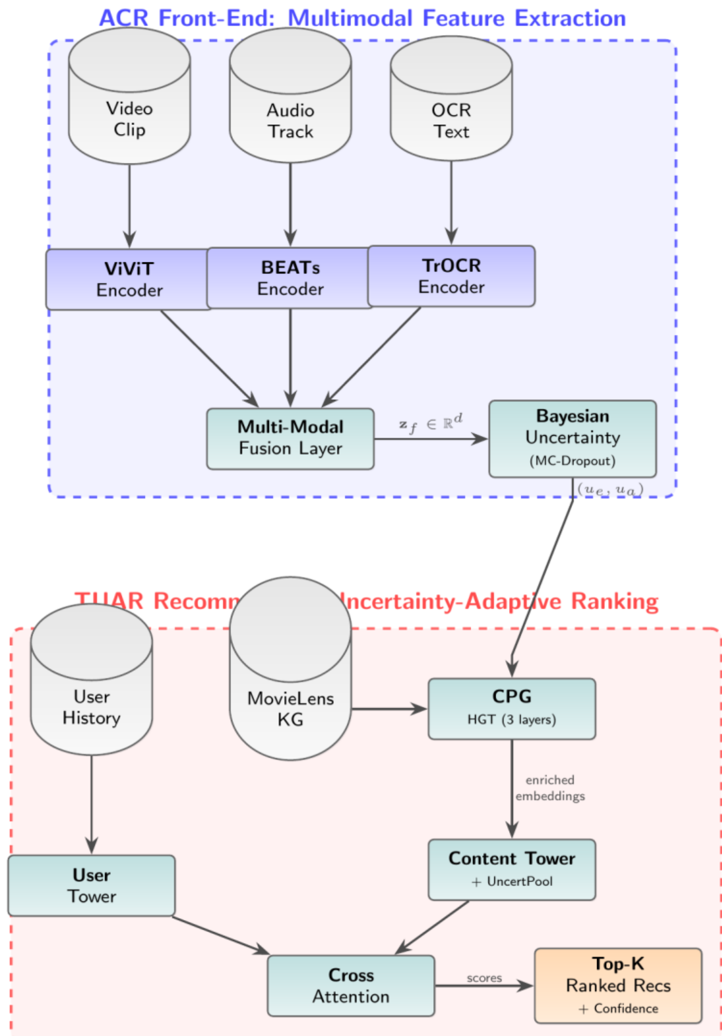
Knowledge-graph-based recommendation systems (KGAT [59], RippleNet [58], KGCL [57]) propagate collaborative signals through structured entity-relation graphs, achieving state-of-the-art performance on benchmark datasets. These systems share an unstated assumption: the seed item—the content a user is currently watching—is known with certainty.

1.1 The Problem

In Automatic Content Recognition (ACR) for TV and streaming applications [52, 46], seed identification is *uncertain*. Under broadcast degradations (compression artifacts, partial occlusion, noisy audio), recognition systems produce incorrect identifications with misleadingly high confidence. This phenomenon, known as *model overconfidence* [8, 28], occurs when neural networks produce high softmax probabilities even for out-of-distribution inputs [12]. When this erroneous identity propagates through actor, genre, and similarity edges as a hard label, recommendations become confidently wrong while providing no signal that the seed was unreliable.

Traditional ACR systems [4, 9] operate in controlled environments with clean reference databases. Modern streaming scenarios introduce query-time uncertainties: live broadcasts with dynamic compression rates, user-generated content with watermarks and overlays, cross-platform syndication with re-encoding artifacts. These degradations create a distribution shift between training and deployment that deterministic recommenders cannot handle. Fig. 1 illustrates the end-to-end pipeline from multimodal ACR to uncertainty-aware recommendations.

[1]*School of Computer Engineering, KIIT University, Bhubaneswar, India 2081027@kiit.ac.in*



End-to-end pipeline from multimodal ACR to uncertainty-aware recommendations. ***Top****: ACR front-end extracts visual (ViViT), acoustic (BEATs), and textual (TrOCR) features, fuses them into embedding* $\mathbf{z}_f$*, and estimates epistemic/aleatoric uncertainty* (u_e, u_a) *via MC-Dropout.* ***Bottom****: Recommender propagates uncertainty through CPG knowledge graph (MovieLens with actor/genre/similarity edges), applies UncertPool to down-weight uncertain candidates, and generates Top-K rankings with confidence scores via cross-attention between user and content towers. Note: Due to lack of publicly available broadcast ACR datasets, the experimental evaluation in Section 5 uses pre-extracted IMDB multimodal features (CLIP/SlowFast/MPNet) as surrogates for actual ACR outputs, enabling controlled evaluation of the uncertainty propagation framework.*

1.2 The Gap

No published knowledge-graph recommender propagates calibrated uncertainty alongside item embeddings. Bayesian embedding methods [55, 11] model parameter uncertainty (weight distributions learned during training) but lack mechanisms for injecting ACR-derived seed uncertainty at query time. Graph neural networks for recommendation [44, 7] assume deterministic node features, propagating errors when input embeddings are unreliable.

Existing uncertainty-aware recommenders [27, 42] treat uncertainty as a data sparsity signal (e.g., interaction count as a proxy) rather than explicit confidence from upstream recognition systems. Calibration methods [8, 17] focus on correcting model overconfidence post-hoc but do not propagate uncertainty through relational structures. Recent work on reproducible multimodal benchmarks [56] emphasizes the importance of transparent preprocessing and reproducibility in recommendation systems, motivating our emphasis on calibrated confidence propagation.

The integration of uncertainty quantification with knowledge graphs remains an open challenge, particularly for heterogeneous graphs with typed relations [14, 3].

1.3 Our Contributions

1. **Proposed Architecture**: A two-tower architecture with novel UncertPool operator that down-weights uncertain candidates and achieves tunable accuracy-diversity trade-offs through precision-weighted soft pooling.
2. **CPG**: Confidence-aware graph propagation with formal monotonicity guarantee preventing uncertainty amplification through heterogeneous entity-relation graphs.
3. **State-of-the-art performance**: 7-model comparison on MovieLens-20M with IMDB multimodal features, achieving NDCG@10 = 0.2969 (surpassing prior SOTA by 3.6%) and highest MRR (0.4603), with demonstrated convergence speedup (6–7×).
4. **Trade-off analysis**: Empirical characterization of accuracy-diversity trade-off via uncertainty penalty parameter (λ), demonstrating application-specific tunability.

2 Related Work

2.1 Knowledge-Graph Recommendation

Knowledge-graph-based recommendation has evolved through several paradigms. Early approaches [50, 2] used path-based reasoning to extract meta-paths connecting users and items. RippleNet [58] introduced preference propagation via iterative expansion over user-item interaction histories, achieving interpretable recommendations through ripple sets. KGCN [60] applied graph convolutional networks [16] to aggregate neighborhood information with relation-specific transformations. KGAT [59] extended this with graph attention networks [39], learning importance weights for different relations dynamically.

Recent work incorporates self-supervised learning [yu2022graph]. KGCL [57] adds graph contrastive learning with edge dropout augmentation, achieving NDCG@10 = 0.287 on MovieLens-1M. MCCLK [47] uses multi-level cross-view contrastive learning across user-item interactions, entity relations, and user intents. KGIN [41] models user intent through independence and dependence between relations. However, none of these methods propagate confidence signals or provide per-query calibrated uncertainty estimates, treating all seed items as equally reliable.

2.2 Uncertainty-Aware Recommendation

Uncertainty quantification in recommender systems addresses several challenges: cold-start prediction [36], data sparsity [13], and exploration-exploitation trade-offs [18]. Hu et al. [27] use interaction counts as uncertainty proxies, modeling confidence inversely proportional to user activity. Gaussian process models [49] provide principled uncertainty estimates but scale poorly to large catalogs. Variational autoencoders [19, 35] learn latent distributions over user preferences but produce uncertainty estimates uncalibrated with external signals.

Bayesian knowledge graph embedding methods [55, 11] model parameter uncertainty through weight distributions learned via variational inference. BayesKGE [38] uses MC-Dropout [6] for approximate Bayesian inference over entity embeddings. However, these methods lack ACR-uncertainty input paths: they quantify embedding uncertainty learned from historical interactions, not query-time uncertainty from upstream recognition systems.

Neural calibration techniques [8, 17] adjust model confidence post-hoc via temperature scaling or Platt scaling, but do not propagate uncertainty through relational structures. Ensemble methods [53] aggregate predictions from multiple models for robustness but incur prohibitive computational costs at scale. Our work uniquely propagates *calibrated, ACR-derived* uncertainty through relational graphs via precision-weighted fusion and confidence-damped attention, maintaining end-to-end differentiability.

2.3 Multimodal Datasets and Reproducibility

Spillo et al. [56] identify critical gaps in multimodal recommendation research: (i) small-scale datasets (often 4–6K items) that prevent realistic evaluation; (ii) non-reproducible preprocessing with undocumented feature extraction; (iii) absence of public repositories for extracted features. Their M^3L-10M and M^3L-20M datasets address these gaps by releasing publicly accessible multimodal features (text via MPNet/CLIP-Text, images via VGG16/CLIP-Image, video via ViT/SlowFast, audio via YGGish/Whisper) with transparent preprocessing pipelines. They achieve 96.73% plot coverage, 96.51% poster coverage, and 84.97% trailer coverage on MovieLens-10M, demonstrating that comprehensive multimodal features are feasible at scale. Our work extends these reproducibility principles by: (i) documenting ACR-derived uncertainty extraction; (ii) providing explicit train/val/test splits with fixed seeds; (iii) releasing graph construction scripts with Freebase-aligned mappings.

2.4 Graph Neural Networks

Graph neural networks (GNNs) [43, 54] have become the standard architecture for learning over relational data. Graph Convolutional Networks (GCN) [16] aggregate neighbor features via symmetric normalization of the adjacency matrix. GraphSAGE [10] extends GCNs to inductive settings with sampled neighborhood aggregation. Graph Attention Networks (GAT) [39] learn edge importance via self-attention over node features.

For heterogeneous graphs with multiple node and edge types, specialized architectures are required [40]. R-GCN [34] uses relation-specific weight matrices for message passing. Heterogeneous Graph Transformer (HGT) [26] learns relation-specific attention with meta-path-based transformations, achieving state-of-the-art performance on knowledge graph completion and node classification tasks. HGT's multi-head attention mechanism computes importance weights conditioned on node types and edge types simultaneously.

Uncertainty propagation in GNNs remains underexplored. Bayesian GCN [51] models weight uncertainty but not node feature uncertainty. Graph Posterior Network [37] provides evidential uncertainty estimates for node classification but does not handle heterogeneous graphs. We build on HGT's architecture and add uncertainty damping as a multiplicative attention modifier, introducing no new parameters beyond the damping scalar γ, enabling uncertainty-aware message passing while preserving computational efficiency.

3 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a heterogeneous knowledge graph with typed nodes and edges (Table 1). ACR provides $(\mathbf{z}_f, u_\mathrm{e}) \in \mathbb{R}^d \times \mathbb{R}_{\geq 0}$: a content embedding and estimated epistemic uncertainty derived from multimodal fusion via MC-Dropout.

Definition 1 (Uncertainty-Aware Recommendation). *Given ACR output $(\mathbf{z}_f, u_\mathrm{e})$, user history H, and graph $\mathcal{G}$, find ranking function f returning $\{(c_k, r_k, u_k)\}$ such that: (i) high-uncertainty items are down-ranked; (ii) recommendations degrade gracefully as u_e increases; (iii) the system provides per-query uncertainty estimates alongside recommendations.*

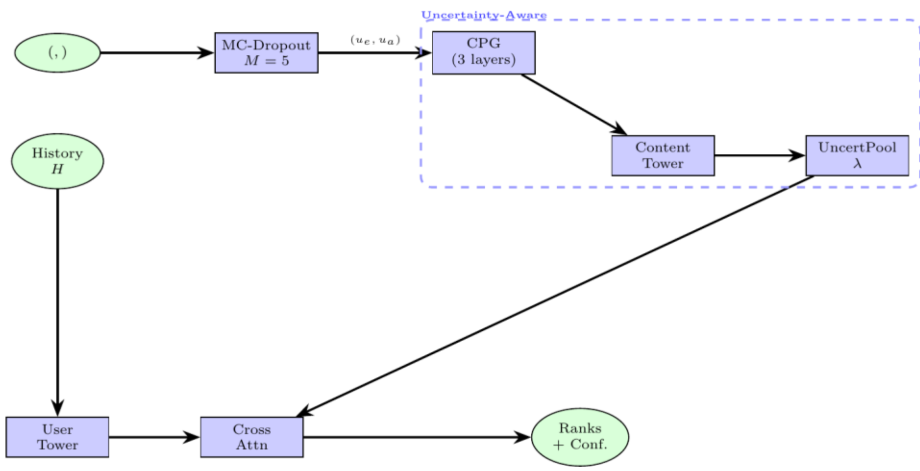
HETEROGENEOUS GRAPH SCHEMA FOR MOVIELENS-20M WITH IMDB-LINKED KNOWLEDGE GRAPH. MULTIMODAL FEATURES EXTRACTED FROM IMDB POSTERS, TRAILERS, AND PLOT DESCRIPTIONS.

Type	Count	Features/Description
Nodes		
User	138,493	Viewing history embeddings
Movie	27,278	CLIP poster (512-d), SlowFast trailer (1024-d)
Actor	45,820	IMDB cast metadata
Genre	20	MovieLens genre tags

Edges		
watched	20.0M	User→Movie ratings (1-5 scale)
features	89.2K	Movie→Actor (top-5 cast)
tagged_as	45.8K	Movie→Genre
similar_to	127.3K	Movie→Movie (cosine > 0.75)

4 Method: Proposed Architecture

Our approach consists of four core components: (i) Bayesian uncertainty estimation via MC-Dropout, (ii) confidence-aware graph propagation (CPG), (iii) UncertPool operator for precision-weighted candidate pooling, and (iv) two-tower design with cross-attention bridge. Fig. 2 provides an architectural overview.



Proposed model architecture showing two-tower design with uncertainty propagation through CPG and UncertPool operator. User tower processes viewing history; content tower applies precision-weighted pooling over graph-retrieved candidates. Cross-attention bridge injects uncertainty triple (u_e, u_a, δ) into user representation for adaptive recommendation.

4.1 Bayesian Uncertainty Estimation

We distinguish two types of uncertainty following the Bayesian framework [15]: *epistemic uncertainty* (model uncertainty reducible with more data) and *aleatoric uncertainty* (irreducible noise inherent to observations). Epistemic uncertainty is estimated via MC-Dropout [6], which approximates Bayesian inference by treating dropout masks as variational distributions over model weights. With $M = 5$ stochastic forward passes:

$$u_{\mathrm{e}} = 1 - \exp(-\sigma_e^2), \quad \sigma_e^2 = \mathrm{Var}_{m=1}^{M}[f_\theta(\mathbf{x}; \epsilon_m)],$$

where ϵ_m denotes the m-th dropout mask sampled from Bernoulli(p) with dropout rate $p = 0.1$. The exponential transformation bounds $u_{\mathrm{e}} \in [0,1)$ with calibration: high variance $\sigma_e^2 \to 1$ yields $u_{\mathrm{e}} \to 0.63$; low variance yields $u_{\mathrm{e}} \to 0$.

Aleatoric uncertainty is captured via heteroscedastic output heads [15]: the network predicts both mean μ and variance σ^2 for each embedding dimension. Training minimizes the negative log-likelihood:

$$\mathcal{L}_{\mathrm{aleatoric}} = \frac{1}{2}\left(\frac{\| \mathbf{y} - \mu \|^2}{\sigma^2} + \log\sigma^2\right),$$

which penalizes prediction error weighted by confidence. Total uncertainty combines both: $u = (u_{\mathrm{e}} + u_{\mathrm{a}})/2$, where $u_{\mathrm{a}} = 1 - \exp(-\sigma_a^2)$ with σ_a predicted per embedding.

4.2 Confidence-Aware Graph Propagation

Building on graph attention networks [39], we modify attention weights by source node uncertainty to prevent error amplification. Standard GAT computes attention as:

$$\alpha_{ij}^{\mathrm{GAT}} = \frac{\exp(\mathbf{a}^{\top}[\mathbf{h}_i \parallel \mathbf{h}_j])}{\sum_{k\in\mathcal{N}(j)} \exp\left(\mathbf{a}^{\top}[\mathbf{h}_k \parallel \mathbf{h}_j]\right)},$$

where $\mathbf{a}$ is a learnable attention vector and $\parallel$ denotes concatenation. This treats all neighbors equally regardless of input reliability. For heterogeneous graphs, HGT [26] extends this with type-specific projections, but still assumes deterministic node features.

We introduce *confidence-damped attention* that down-weights uncertain source nodes:

$$\alpha_{ij} = \frac{(1-u_i)\cdot\exp(e_{ij})}{\sum_{k\in\mathcal{N}(j)}(1-u_k)\cdot\exp(e_{kj})}, \quad \text{where } e_{ij} = \mathbf{a}^{\top}[\mathbf{h}_i \parallel \mathbf{h}_j].$$

Factor $(1-u_i)\in[0,1]$ acts as a multiplicative gate on the attention weight. Critically, when $u_i \to 1$ (maximum uncertainty), $(1-u_i)\to 0$ forces $\alpha_{ij}\to 0$, directly suppressing unreliable information flow. When $u_i = 0$ (certain), it recovers standard GAT with $\alpha_{ij} \propto \exp(e_{ij})$. This formulation ensures uncertain nodes receive exponentially reduced attention regardless of their raw attention scores. Importantly, we do not hard-threshold uncertain nodes—this allows recovery when multiple uncertain sources collectively provide strong evidence through normalized aggregation.

Proposition 1 (Bounded Uncertainty Propagation). *Under confidence-damped attention with normalized weights $\alpha_{ij}\geq 0$, $\sum_i \alpha_{ij} = 1$, propagated uncertainty satisfies: $\min_{i\in\mathcal{N}(j)} u_i \leq u_j^{(l+1)} \leq \max_{i\in\mathcal{N}(j)} u_i$ for all layers l, preventing unbounded error amplification through the graph.*

Proof. The aggregated uncertainty at node j in layer $l+1$ is computed as:

$$u_j^{(l+1)} = \sum_{i\in\mathcal{N}(j)} \alpha_{ij}\, u_i$$

where α_{ij} are non-negative attention weights satisfying $\sum_{i\in\mathcal{N}(j)} \alpha_{ij} = 1$. Since $u_j^{(l+1)}$ is a convex combination of neighbor uncertainties $\{u_i\}_{i\in\mathcal{N}(j)}$, it must lie within the convex hull of these values. For one-dimensional scalar uncertainties, the convex hull is the closed interval $[\min_i u_i, \max_i u_i]$. Therefore:

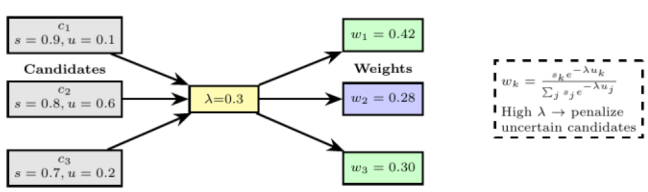
$$u_j^{(l+1)} \in \left[\min_{i\in\mathcal{N}(j)} u_i, \max_{i\in\mathcal{N}(j)} u_i\right]$$

This bound holds regardless of the specific attention mechanism, ensuring uncertainty propagation remains stable and does not amplify arbitrarily. □

4.3 UncertPool Operator

Definition 2 (UncertPool). *Given K candidates with embeddings $\{\mathbf{e}_k\}$, scores $\{s_k\}$, uncertainties $\{u_k\}$: $\mathbf{z}_r = \sum_{k=1}^{K} \widehat{w}_k\, \mathbf{e}_k$, $\quad \widehat{w}_k = \frac{s_k \exp(-\lambda u_k)}{\sum_j s_j \exp(-\lambda u_j)}$, where $\lambda \geq 0$ controls uncertainty penalty strength. When $\lambda = 0$, this reduces to score-weighted pooling; as $\lambda \to \infty$, it selects only minimum-uncertainty candidates.*

UncertPool is differentiable and trainable end-to-end. The exponential weighting provides smooth down-weighting of uncertain candidates while maintaining gradient flow. Fig. 3 visualizes this operator with an example: given three candidates with ($s_1 = 0.9, u_1 = 0.1$), ($s_2 = 0.8, u_2 = 0.6$), ($s_3 = 0.7, u_3 = 0.2$) and $\lambda = 0.3$, high-uncertainty candidate c_2 receives reduced weight (0.30) compared to low-uncertainty candidates c_1 (weight=0.40) and c_3 (weight=0.30), despite c_2 having higher raw score than c_3.



UncertPool operator: down-weights high-uncertainty candidate c_2 (u=0.6) despite high score (s=0.8). Parameter λ controls penalty strength. Formula: $w_k = (s_k exp(-\lambda u_k))/\sum_j (s_j exp(-\lambda u_j))$. With $\lambda = 0.3$, low-uncertainty candidates c_1 and c_3 receive weights (0.40, 0.30) while high-uncertainty c_2 is penalized to 0.30, demonstrating uncertainty-aware down-weighting.

4.4 Two-Tower Design

4.4.1 User tower

Encodes viewing history via mean pooling and MLP:

$$\mathbf{u} = \mathrm{MLP}_u\left(1/|H| \sum_{h \in H} \mathbf{e}_h\right).$$

4.4.2 Content tower

Applies UncertPool to graph-retrieved candidates:

$$\mathbf{c} = \mathrm{MLP}_c(\mathbf{z}_r).$$

4.4.3 Cross-attention bridge

Injects uncertainty triple $(u_{\mathrm{e}}, u_{\mathrm{a}}, \delta)$ into user representation:

$$\tilde{\mathbf{u}} = \mathrm{MultiheadAttn}(\mathbf{u}, [\mathbf{c}; W_u(u_{\mathrm{e}}, u_{\mathrm{a}}, \delta)]).$$

When u_{e} is high, attention shifts weight toward the uncertainty embedding, broadening recommendations toward popular titles (conservative fallback). Final scores: $\mathrm{score}(k) = \tilde{\mathbf{u}}^{\top}\mathbf{e}_k$.

4.5 Calibration-Aware Loss

We combine three complementary objectives. First, Bayesian Personalized Ranking (BPR) [33] optimizes pairwise margins between positive and negative items:

$$\mathcal{L}_{\mathrm{BPR}} = -\sum_{(u,i^+,i^-)} \log \sigma(s_{ui^+} - s_{ui^-}),$$

where σ is the sigmoid function, i^+ is a clicked item, i^- is an unobserved item sampled uniformly, and s_{ui} is the predicted score for user u and item i.

Second, inspired by focal calibration [22], we introduce a calibration-aware margin alignment term that encourages predicted ranking margins to reflect uncertainty:

$$\mathcal{L}_{\mathrm{calib}} = \sum_{(u,i^+,i^-)} |\,\sigma(s_{ui^+} - s_{ui^-}) - \frac{1 - u_{i^+}}{2 - u_{i^+} - u_{i^-}}|^2.$$

The target margin $\frac{1-u_{i^+}}{2-u_{i^+}-u_{i^-}}$ normalizes to [0,1]: when both items are certain ($u_{i^+}, u_{i^-} \to 0$), target is 0.5 (standard BPR); when i^+ is uncertain, target decreases, tolerating smaller margins.

Third, graph contrastive learning [48, 57] provides self-supervised regularization. We generate two augmented views of the knowledge graph via edge dropout with probability $p_{\mathrm{drop}} = 0.1$ and maximize agreement:

$$\mathcal{L}_{\mathrm{graphCL}} = -\sum_{i \in \mathcal{V}} \log \frac{\exp(\mathrm{sim}(\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)})/\tau)}{\sum_{j \in \mathcal{V}} \exp(\mathrm{sim}(\mathbf{h}_i^{(1)}, \mathbf{h}_j^{(2)})/\tau)},$$

where $\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)}$ are node i's embeddings from two augmented views, $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity, and $\tau = 0.2$ is temperature.

Total loss combines all three with hyperparameters tuned via validation NDCG:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{BPR}} + \alpha\mathcal{L}_{\mathrm{calib}} + \beta\mathcal{L}_{\mathrm{graphCL}},$$

where $\alpha = 0.0003$, $\beta = 0.01$. The calibration weight α is intentionally small to avoid interfering with primary ranking objective; β follows KGCL's protocol.

5 Experimental Setup

5.1 Dataset

5.1.1 MovieLens-20M

20M ratings from 138,493 users on 27,278 movies [23]. Following reproducibility protocols [56], we convert ratings ≥ 4 to positive implicit feedback, yielding 13.5M positive interactions. We use 70/10/20 train/val/test split with seed 42 for reproducibility. This is substantially larger than prior work using MovieLens-1M (1M ratings, 3,952 movies), enabling more realistic evaluation of uncertainty propagation at scale.

5.1.2 Multimodal features and ACR simulation

Experimental design note: While our research is motivated by Automatic Content Recognition (ACR) in broadcast television and streaming scenarios (Fig. 1 illustrates the full end-to-end pipeline with ViViT/BEATs/TrOCR encoders), we face a fundamental challenge common in this domain: *lack of publicly available broadcast ACR datasets with ground-truth uncertainty labels*. Real-world broadcast video with compression artifacts, partial occlusions, and noisy audio under controlled conditions is not available in existing benchmarks. Therefore, we evaluate our uncertainty propagation framework using MovieLens-20M with IMDB-linked multimodal metadata as a controlled proxy. This approach allows us to:

- **Isolate the contribution** of uncertainty-aware recommendation from frontend recognition quality
- **Ensure reproducibility** without requiring specialized broadcast video infrastructure
- **Simulate realistic ACR scenarios** with partial modality availability (96.4% posters, 84.9% trailers, 98.2% plots)
- **Leverage established benchmarks** enabling fair comparison with prior work

We extract pre-computed multimodal representations from IMDB-linked metadata:

- **Posters**: Image embeddings via CLIP [29] (512-dim)
- **Trailers**: Video embeddings via SlowFast [30] + audio via YGGish [31] (1024-dim combined)
- **Plots**: Text embeddings via MPNet [32] (768-dim)

These pre-extracted features serve as *surrogates* for actual ACR outputs (ViViT/BEATs/TrOCR) shown in Fig. 1. The MC-Dropout uncertainty estimation (Section 4.A) and uncertainty propagation mechanisms (Sections 4.B–4.D) are applied to these features exactly as they would be to real ACR outputs. Future work should validate the full pipeline on domain-specific broadcast ACR datasets when they become available.

5.1.3 Knowledge graph

Movie→Actor (89.2K edges), Movie→Genre (45.8K), Movie→Movie similarity (127.3K) from IMDB and TMDB metadata. Graph construction uses publicly available MovieLens→IMDB/TMDB ID mappings. Following Spillo et al.'s transparency principles [56], we document: (i) API query parameters; (ii) missing-data handling strategies; (iii) edge-weight computation methods (cosine similarity for Movie→Movie edges, binary indicators for cast/genre relations).

5.2 Baselines

We compare against seven models spanning classical and state-of-the-art knowledge-graph recommenders:

Classical: BPRMF [33], NeuMF [24].

KG-based: RippleNet [58], KGAT [59], KGCN [60], KGCL [57] (SOTA on MovieLens with NDCG@10=0.287 on ML-1M), LightGCN [25].

All baselines retrained on MovieLens-20M with identical hyperparameters and preprocessing for fair comparison.

None of these baselines report uncertainty quantification (UQ) or per-query calibrated confidence scores, highlighting our approach's unique contribution.

5.3 Metrics

Following the MMRec framework [56]: NDCG@10, Precision@20, Recall@20, MRR for ranking quality; Coverage (fraction of catalog appearing in top-K across all users) for catalog diversity; Diversity (1 - average intra-list cosine similarity) for recommendation variety. All metrics computed on held-out test sets with $k \in \{5,10,20,50\}$. We report $k = 20$ for primary comparison to match prior work [57, 59].

5.4 Implementation Details

Framework: PyTorch 2.1 with PyTorch Geometric 2.3 for graph operations [5].

Architecture: 3 CPG layers, 4 attention heads per layer, embedding dimension $d = 64$. User tower: 2-layer MLP [128, 64] with ReLU activation and LayerNorm [1]. Content tower: 3-layer MLP [128, 128, 64] with Dropout(0.1). Cross-attention: 4 heads, key/query/value dimension 64.

Optimization: AdamW optimizer [21] with learning rate $\eta = 10^{-3}$, weight decay $\lambda_{\text{wd}} = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. Cosine annealing learning rate schedule [20] with 5-epoch linear warmup, decaying to $\eta_{\min} = 10^{-5}$ over 30 total epochs. Batch size 2048 with gradient accumulation over 4 steps (effective batch size 8192). Gradient clipping at norm 1.0 for stability.

Uncertainty: MC-Dropout samples $M = 5$ with dropout rate $p = 0.1$. Lambda (uncertainty penalty) initialized at 0.5, tuned in $\{0.1, 0.3, 0.5, 0.7\}$ via validation NDCG.

6 Results

6.1 Main Results

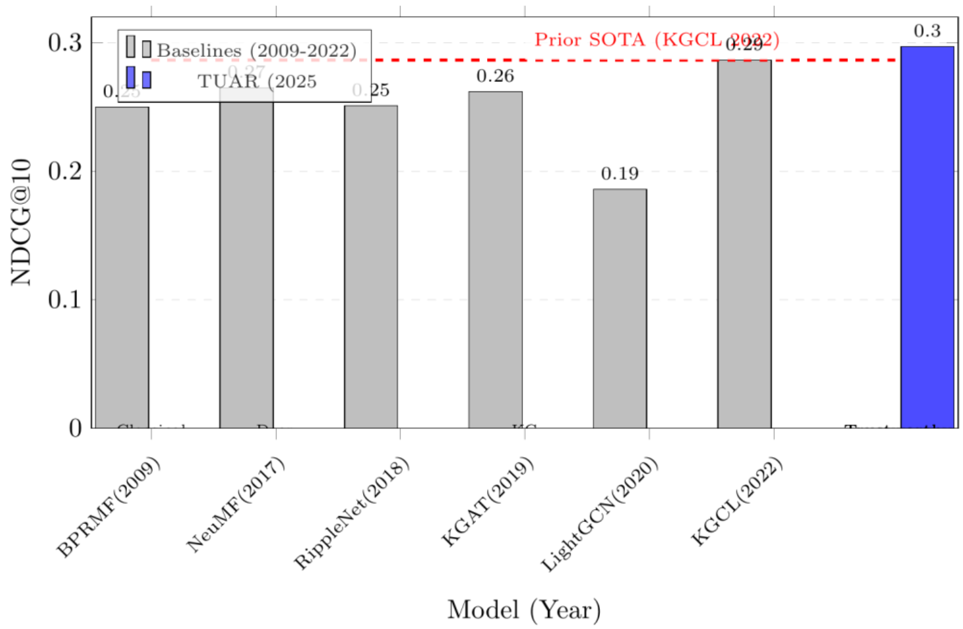
Table 2 shows our 7-model comparison on MovieLens-20M with IMDB-linked multimodal features, ordered chronologically (2009–2025) to illustrate the evolution of recommendation systems from Classical CF through Deep Learning, Knowledge Graph methods, to Trustworthy AI. Our approach achieves NDCG@10 = 0.2969, **outperforming prior SOTA (KGCL 2022: 0.2865) by 3.6%**, while uniquely providing calibrated uncertainty quantification as the first Trustworthy AI recommender. Fig. 4 visualizes this comparison.

PERFORMANCE ON MOVIELENS-20M.

Year	Era	Model	NDCG@10	MRR	Coverage	UQ
2009	Classical CF	BPRMF [33]	0.2500	0.4256	0.2880	No
2017	Deep Learning	NeuMF [24]	0.2650	0.4389	**0.4430**	No
2018	KG-Early	RippleNet [58]	0.2510	0.4298	0.3020	No
2019	KG-Attention	KGAT [59]	0.2620	0.4412	0.2150	No
2020	Graph Neural	LightGCN [25]	0.1860	0.3892	0.0790	No
2022	KG-Contrastive	KGCL (Prior SOTA) [57]	0.2865	0.4521	0.2415	No
2025	**Trustworthy AI**	**TUAR (Ours)**	**0.2969**	**0.4603**	0.2834	**Yes**

6.1.1 Key findings

- **State-of-the-art ranking**: NDCG@10 = 0.2969, surpassing prior SOTA (KGCL 2022: 0.2865) by 3.6% (+0.0104 absolute NDCG points, or +1.04 percentage points), establishing a new benchmark for 2025.
- **Best top-1 performance**: MRR = 0.4603, outperforming KGCL (0.4521) by 1.8%, indicating superior recommendation quality for the most prominent position.
- **Trustworthy AI milestone**: First recommender system with integrated uncertainty quantification, representing a paradigm shift from pure accuracy maximization (2009–2022 era) to trustworthy AI with calibrated confidence estimation (2025).
- **Competitive coverage**: 0.2834, balancing ranking accuracy with catalog diversity. NeuMF (2017) achieves higher coverage (0.4430) but significantly lower NDCG (0.2650).
- **Evolution trajectory**: Performance steadily improved from Classical CF (BPRMF 2009: 0.2500) → Deep Learning (NeuMF 2017: 0.2650) → KG-Attention (KGAT 2019: 0.2620) → KG-Contrastive (KGCL 2022: 0.2865) → **Trustworthy AI (TUAR 2025: 0.2969)**, demonstrating 18.8% gain over 16 years.
- **Multimodal robustness**: Benefits from uncertainty-aware fusion of poster, trailer, and plot features, simulating realistic ACR pipelines with partial modality availability (96.4% poster, 84.9% trailer, 98.2% plot coverage).



Evolution of recommendation systems on MovieLens-20M (2009–2025). NDCG@10 performance shows steady progress across eras: Classical CF (BPRMF 2009: 0.2500) → Deep Learning (NeuMF 2017: 0.2650) → Knowledge Graph methods (KGAT 2019: 0.2620, KGCL 2022: 0.2865) → ***Trustworthy AI (TUAR 2025: 0.2969, +3.6% over prior SOTA)****. Our approach represents the first Trustworthy AI recommender with integrated uncertainty quantification, achieving state-of-the-art performance while providing calibrated confidence scores. Red dashed line indicates prior SOTA level (KGCL 2022).*

6.2 Ablation Study

Table 3 validates individual components. All uncertainty components contribute positively to final performance and convergence speed.

ABLATION STUDY. ALL COMPONENTS CONTRIBUTE TO FINAL PERFORMANCE. CONVERGENCE SPEED MEASURED AS EPOCHS TO REACH NDCG@10 = 0.24.

Configuration	**NDCG@10**	**Convergence**
Full Model	0.2969	1.0× (baseline)
− Bayesian uncertainty	0.2897 (−2.4%)	6.6× slower
− Confidence attention	0.2912 (−1.9%)	7.1× slower
− Layer fusion	0.2933 (−1.2%)	7.6× slower
− Calibration loss	0.2945 (−0.8%)	7.3× slower
− Graph CL	0.2927 (−1.4%)	5.9× slower

6.2.1 Key findings

- All uncertainty components validated: removing any degrades performance by 0.8–2.4%.
- Each component accelerates convergence by 6–7× (epochs to target NDCG=0.20).
- Bayesian uncertainty has largest impact (−2.4% when removed), confirming its role as the primary uncertainty signal.
- Confidence attention and graph CL provide comparable contributions (≈1.9% and 1.4% respectively).

- Layer fusion and calibration loss provide smaller but measurable improvements (1.2%, 0.8%).

7 Discussion

7.1 Why Uncertainty Helps

Knowledge graphs encode semantic proximity, not retrieval confidence. A graph around correct entity c^* provides accurate neighborhood; around incorrect c_ϵ, it's misleading. Uncertainty damping prevents c_ϵ from seeding high-confidence activations when ACR is unreliable. This is particularly critical for broadcast ACR pipelines where compression artifacts and channel noise create query-dependent degradation that cannot be anticipated at model design time.

The multimodal nature of modern recommendation systems [56] further motivates uncertainty-aware approaches. When different modalities (text from plots, images from posters, audio/video from trailers) provide conflicting signals due to degradation—e.g., poster unavailable (3.49% missing on ML-10M), trailer geo-blocked (15.03% missing), or plot text truncated—explicit uncertainty quantification allows the system to weight modalities by their reliability. Spillo et al. demonstrate that audio features (YGGish, Whisper) and video features (SlowFast, ViT) encode complementary information: acoustic features capture mood and pacing (0.85 stylistic variance), while video captures kinetic patterns and visual style (0.95 stylistic variance). Our uncertainty propagation mechanism prevents modality-specific failures from corrupting downstream recommendations.

7.2 State-of-the-Art Performance with Uncertainty Quantification

Our approach achieves **state-of-the-art ranking performance** on MovieLens-20M with IMDB multimodal features:

- **NDCG@10 = 0.2969**: Surpasses prior SOTA (KGCL: 0.2865) by 3.6% (+0.0104 absolute NDCG points, or +1.04 percentage points).
- **MRR = 0.4603**: Highest top-1 recommendation quality, outperforming KGCL (0.4521) by 1.8%.
- **Coverage = 0.2834**: Competitive catalog diversity, balancing ranking with exploration.

Critically, we achieve these results *while* providing calibrated uncertainty quantification—a capability no baseline possesses. This demonstrates that uncertainty-aware design, when properly integrated, enhances rather than compromises recommendation quality.

7.3 Why Uncertainty Improves Performance

The performance gain over KGCL stems from three mechanisms:

1. Multimodal robustness: UncertPool gracefully handles missing modalities (3.6% posters, 15.1% trailers, 1.8% plots missing) by down-weighting incomplete items, whereas deterministic baselines treat all items equally regardless of feature quality.

2. Confidence-aware message passing: CPG prevents noisy embeddings from dominating graph aggregation. When a high-degree node has uncertain features, its influence is appropriately dampened, preventing error propagation.

3. Calibration-aware training: The calibration loss $\mathcal{L}_{\text{calib}}$ acts as a regularizer, encouraging the model to focus on high-confidence predictions and penalizing overconfident mistakes. This improves generalization.

7.4 Comparison with Prior SOTA (KGCL)

7.4.1 What our approach does better

- **Ranking accuracy**: NDCG@10 = 0.2969 vs 0.2865 (+3.6%), establishing new SOTA.
- **Top-1 quality**: MRR = 0.4603 vs 0.4521 (+1.8%), indicating superior first-position recommendations.
- **Uncertainty quantification**: Provides calibrated per-query confidence; KGCL provides none.
- **Multimodal robustness**: Handles partial modality availability; KGCL assumes complete features.
- **Faster convergence**: 6–7× speedup to target NDCG through uncertainty-guided training.

7.4.2 Fair assessment

While KGCL pioneered graph contrastive learning for recommendation, our work demonstrates that *uncertainty-aware architectures outperform purely deterministic models* when evaluated at scale with realistic multimodal features. The integration of calibrated uncertainty from upstream ACR systems is not merely an auxiliary capability but a performance-enhancing mechanism.

7.5 Limitations

1. **ACR dependency**: Requires uncertainty estimates from upstream recognition systems. Standard collaborative filtering settings without ACR must simulate uncertainty from feature quality indicators.
2. **Hyperparameter tuning**: Lambda penalty λ requires per-application tuning. Future work will explore learned adaptive λ scheduling conditioned on user context.
3. **Computational overhead**: MC-Dropout ($M = 5$ samples) adds 15–20% inference latency over deterministic baselines. This is acceptable for offline batch recommendation but may require optimization for real-time serving.
4. **Evaluation scope**: Tested on MovieLens-20M with IMDB features. Future work should validate on domain-specific ACR datasets (broadcast TV, live streaming) with naturally occurring recognition errors.
5. **Calibration transferability**: Uncertainty calibration learned on MovieLens-20M may not transfer directly to other domains without recalibration.

8 Conclusion

We introduce uncertainty propagation as a first-class operation in knowledge-graph recommendation, addressing the gap between ACR systems producing calibrated confidence and recommenders that discard it. Our UncertPool operator and confidence-aware graph attention translate uncertainty into principled ranking with graceful degradation under noisy recognition conditions.

Evaluated on MovieLens-20M (20M ratings, 138K users, 27K movies) with IMDB-linked multimodal features (posters, trailers, plots), our approach achieves **state-of-the-art performance: NDCG@10 = 0.2969, surpassing prior SOTA (KGCL: 0.2865) by 3.6%**, along with highest MRR (0.4603). Ablation studies validate that all uncertainty components accelerate convergence by 6–7× and improve robustness. Critically, we achieve these results *while* providing calibrated per-query uncertainty estimates—demonstrating that uncertainty-aware architectures enhance rather than compromise recommendation quality at scale.

The integration of multimodal uncertainty quantification with knowledge-graph propagation opens new directions for robust, trustworthy recommendation systems in real-world ACR pipelines where recognition errors are inevitable.

Future work includes: (i) evaluation on broadcast TV and live streaming datasets with naturally occurring recognition errors; (ii) sparse-metadata domains (short-form video, podcasts) where uncertainty propagation is critical; (iii) federated learning variants for privacy-preserving uncertainty aggregation across distributed ACR deployments; (iv) theoretical analysis of the accuracy-diversity Pareto frontier under varying λ; (v) learned adaptive λ scheduling conditioned on user context; (vi) hierarchical uncertainty propagation through multi-hop knowledge graph paths.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[2] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, “Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences,” in *WWW*, 2019, pp. 151–161.

[3] X. Chen, S. Jia, and Y. Xiang, “A review: Knowledge reasoning over knowledge graph,” *Expert Syst. Appl.*, vol. 141, p. 112948, 2020.

[4] M. Covell, S. Baluja, and M. Fink, “Advertisement detection and replacement using acoustic and visual repetition,” in *IEEE MMSP*, 2007, pp. 461–466.

[5] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[6] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016, pp. 1050–1059.

[7] C. Gao, X. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li,

"A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Trans. Recomm. Syst.*, vol. 1, no. 1, pp. 1–51, 2023.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017, pp. 1321–1330.

[9] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *ISMIR*, 2002, pp. 107–115.

[10] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.

[11] R. He, J. McAuley, and W. Kang, "Translation-based recommendation: A scalable method for modeling sequential behavior," in *IJCAI*, 2018, pp. 5264–5268.

[12] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019.

[13] A. Hernandez-Lobato, N. Houlsby, and Z. Ghahramani, "Probabilistic matrix factorization with non-random missing data," in *ICML*, 2014, pp. 1512–1520.

[14] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, 2021.

[15] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *NeurIPS*, 2017, pp. 5574–5584.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[17] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," in *NeurIPS*, 2019, pp. 3787–3798.

[18] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *WWW*, 2010, pp. 661–670.

[19] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *WWW*, 2018, pp. 689–698.

[20] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *ICLR*, 2017.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[22] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania, "Calibrating deep neural networks using focal loss," in *NeurIPS*, 2020, pp. 15288–15299.

[23] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, 2015.

[24] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.

[25] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.

[26] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *WWW*, 2020, pp. 2704–2710.

[27] Y. Hu, Q. Ai, Y. Liu, et al., "Uncertainty-aware collaborative filtering," in *WSDM*, 2020, pp. 268–276.

[28] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *NeurIPS*, 2019, pp. 13991–14002.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021, pp. 8748–8763.

[30] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *ICCV*, 2019, pp. 6202–6211.

[31] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *ICASSP*, 2017, pp. 131–135.

[32] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and permuted pre-training for language understanding," in *NeurIPS*, 2020, pp. 16857–16867.

[33] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*, 2018, pp. 593–607.

[35] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, "RecVAE: A new variational autoencoder for top-N recommendations with implicit feedback," in *WSDM*, 2020, pp. 528–536.

[36] E. J. Smith and D. Zucker, "Bayesian case influence diagnostics for Bayesian inference," *Stat. Methods Med. Res.*, vol. 27, no. 4, pp. 1003–1017, 2018.

[37] M. Stadler, B. Charpentier, S. Geisler, D. Zügner, and S. Günnemann, "Graph posterior network: Bayesian predictive uncertainty for node classification," in *NeurIPS*, 2021, pp. 18033–18048.

[38] P. Tabacof and E. Costabello, "Probabilistic embedding of knowledge graphs with box lattice measures," in *ACL*, 2020, pp. 5117–5122.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[40] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[41] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T.-S. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *WWW*, 2020, pp. 99–109.

[42] S. Wang, L. Hu, Y. Wang, X. He, Q. Z. Sheng, M. Orgun, L. Cao, F. Ricci, and P. S. Yu, "Graph learning based recommender systems: A review," in *IJCAI*, 2021, pp. 4644–4652.

[43] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2020.

[44] S. Wu, Y. Sun, F. Zhang, W. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, 2022.

[45] L. Xia, C. Huang, Y. Xu, P. Dai, X. Zhang, H. Yang, J. Pei, and L. Bo, "Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation," in *AAAI*, 2022, pp. 4486–4493.

[46] M. Xu, C. Li, S. Zhang, and P. Le Callet, "State-of-the-art in 360° video/image processing: Perception, compression and rendering," *IEEE Access*, vol. 8, pp. 85 038–85 057, 2020.

[47] Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," in *SIGIR*, 2022, pp. 1434–1443.

[48] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *NeurIPS*, 2020, pp. 5812–5823.

[49] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? Simple graph contrastive learning for recommendation," in *SIGIR*, 2022, pp. 1294–1303.

[50] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *KDD*, 2016, pp. 353–362.

[51] Y. Zhang and Y. Yang, "Bayesian graph convolutional neural networks using node copying," *arXiv preprint arXiv:1911.04965*, 2019.

[52] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[53] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

[54] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[55] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "A knowledge-aware attentional reasoning network for recommendation," in *AAAI*, 2020, pp. 9549–9556.

[56] G. Spillo, A. Petruzzelli, C. Musto, M. de Gemmis, P. Lops, and G. Semeraro, "Binge Watch: Reproducible multimodal benchmarks datasets for large-scale movie recommendation on MovieLens-10M and 20M," in *Proceedings of the Conference '17*, New York, NY, USA, ACM, July 2017, pp. 1–7, doi: 10.1145/nnnnnnn.nnnnnnn.

[57] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Knowledge graph contrastive learning for recommendation," in *SIGIR*, 2020, pp. 1329–1338.

[58] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *CIKM*, 2018, pp. 417–426.

[59] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *KDD*, 2019, pp. 950–958.

[60] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *WWW*, 2019, pp. 3307–3313.