Avoiding Over-Personalization with Rule-Guided Knowledge Graph Adaptation for LLM Recommendations

Fernando Spadea[1,*], Oshani Seneviratne[2]

[1]Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

We present a lightweight neuro-symbolic framework to mitigate over-personalization in LLM-based recommender systems by adapting user-side Knowledge Graphs (KGs) at inference time. Instead of retraining models or relying on opaque heuristics, our method restructures a user's Personalized Knowledge Graph (PKG) to suppress feature co-occurrence patterns that reinforce Personalized Information Environments (PIEs), i.e., algorithmically induced filter bubbles that constrain content diversity. These adapted PKGs are used to construct structured prompts that steer the language model toward more diverse, *Out-PIE recommendations* while preserving topical relevance. We introduce a family of symbolic adaptation strategies, including soft reweighting, hard inversion, and targeted removal of biased triples, and a client-side learning algorithm that optimizes their application per user. Experiments on a recipe recommendation benchmark show that personalized PKG adaptations significantly increase content novelty while maintaining recommendation quality, outperforming global adaptation and naive prompt-based methods.

Keywords

Personalized Knowledge Graphs, Large Language Models, Recommendation Systems, Over-Personalization, Filter Bubbles, Neuro-Symbolic AI, Knowledge Graph Adaptation, User-Centric AI, Recommendation Diversity

1. Introduction

Overly tailored recommendations lead to Personalized Information Environments (PIEs), which are algorithmically reinforced content silos, akin to filter bubbles, where new information is repeatedly filtered through prior user preferences [1]. Although PIEs can initially enhance relevance, they often narrow exposure to diverse content, reduce user agency, and inhibit discovery [2, 3]. Without transparency or control mechanisms, users struggle to diversify their content landscape [4, 5, 6]. Many early approaches to filter bubble mitigation rely on KG embeddings [7] or influence-based retraining [8], which can be effective but often require retraining, complex domain modeling, or hardcoded logic.

Building on our prior work [9], we introduce a lightweight inference-time method for escaping PIEs without retraining the underlying language model. By adapting a user's Personalized Knowledge Graph (PKG), which is a structured, editable representation of preferences, we enable interpretable, symbolic control over the recommendation process. Our rule-based PKG adaptation selectively down-weights overrepresented features in the user's profile, steering large language models (LLMs) to generate more novel recommendations while preserving relevance.

Motivating Example: Consider a user whose preferences are captured by a PKG. Over time, their interactions with a recommender system create biased associations, forming a PIE. For instance, as shown in Figure 1 *Base PKG*, a user may consistently give high ratings to Italian dishes containing tomatoes, such as "Tomato Pasta," "Lasagna," and "Margherita Pizza." Consequently, the user's PKG becomes dominated by a strong association between "Italian" cuisine and the feature "Tomato." Although initially beneficial for relevance, this over-personalization ultimately limits novelty. Ideally, when seeking new Italian recipes, the user would appreciate recommendations that introduce variety, such as "Pesto Pasta,"

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
spadef@rpi.edu (F. Spadea); senevo@rpi.edu (O. Seneviratne)
0009-0006-4278-3666 (F. Spadea); 0000-0001-8518-917X (O. Seneviratne)

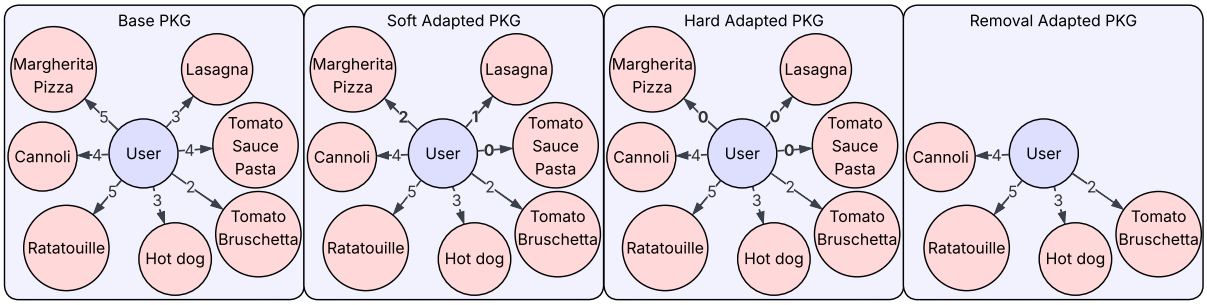

Figure 1: Motivating example. The *Base PKG* shows a strong association between Italian cuisine and tomato-based dishes, forming a PIE (filter bubble). The three adaptation strategies modify this bias differently: *Soft* reduces ratings while preserving order, *Hard* inverts ratings more aggressively, and *Removal* deletes PIE-aligned items, such as "Margherita Pizza," without modifying unrelated or already disliked items.

which align with their preference for Italian dishes but break the entrenched link with tomato-based recipes.

2. Methodology

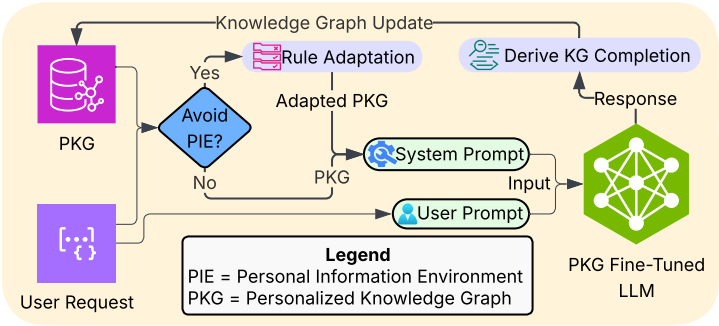

Figure 2: System overview.

Figure 2 provides an overview of our recommendation pipeline. When a recommendation query is issued (e.g., "Suggest an Italian dish"), the system first checks whether the query risks reinforcing a known over-personalization, or PIE. If so, a soft/hard/removal adaptation is applied to adapt the PKG, selectively modifying it to reduce the influence of overrepresented feature pairs (such as "Italian + tomato"). The adapted PKG is used to construct a structured prompt that guides the LLM. This prompt reflects the user's preferences but intentionally avoids PIE-aligned content. The LLM then generates novel, relevant recommendations that satisfy the user's query.

2.1. Prompt Construction

The prompt consists of two parts: a system message and a user message. The system message instructs the model to perform KG completion and provides the PKG. It is formatted as follows:

System Message

```
You perform Knowledge Graph Completion. You will recommend a new triple to add to
the user's knowledge graph with a tail entity that isn't already in their knowledge
graph. The user's entity is represented by {User ID}. Use this knowledge graph when
responding to their queries: {Knowledge Graph}
```

The user message requests a recommendation with a specific trait. For our train and test dataset, it is formatted as:

User Message

```
Recommend a recipe with trait of {Relation Type} -> {Trait Value}.
```

In the user message, `Relation Type` is either `hasIngredient` (for ingredients) or `hasTag` (for Food.com tags), and `Trait Value` corresponds to a specific ingredient or tag.

2.2. Detecting PIEs

Our system addresses over-personalization by adapting the user's PKG that contains preferences derived from prior interactions (e.g., ratings of recipes annotated with ingredients or tags).

We define a PIE as a user-specific bias toward certain co-occurring pairs of features. Given a pair $(F_{\text{given}}, F_{\text{bias}})$, we detect a PIE if a user consistently assigns significantly higher or lower ratings to items containing both features compared to items containing only F_{given}. For example, as illustrated by the *Base PKG* in Figure 1, the pair (F_{given} = `Italian`, F_{bias} = `tomato`) forms a positively biased feature association, creating a "tomato-centric" PIE. Our objective is to detect these biases and adapt the PKG accordingly to generate recommendations that remain relevant but introduce greater diversity. We quantify the strength of a PIE using a feature-pair bias score, q_{bias}:

$$q_{\text{bias}}(F_{\text{given}}, F_{\text{bias}}) = \frac{1}{\mu_{\text{neutral}} \cdot |\mathcal{O}_{F_{\text{given}}}|} \sum_{o \in \mathcal{O}_{F_{\text{given}}, F_{\text{bias}}}} (R_u(o) - \mu_{\text{neutral}}) \tag{1}$$

Here, μ_{neutral} serves as a baseline for interpreting user preferences (e.g., 2.5 on a 0–5 scale), and $R_u(o)$ is the rating that user u assigns to item o. The set $\mathcal{O}_{F_{\text{given}}}$ includes all items in the user's PKG that contain the feature F_{given}, while $\mathcal{O}_{F_{\text{given}}, F_{\text{bias}}}$ further restricts this to items that also include F_{bias}.

The bias score q_{bias} captures how much the user's ratings for items with both features deviate from the neutral point, relative to the size of the user's PKG. A high positive score indicates user preference amplification for feature pairs (e.g., Italian dishes with tomato), while a strong negative score indicates underrating. When q_{bias} exceeds a set threshold (e.g., ± 0.5), we consider the feature pair to be PIE-inducing and subject to adaptation.

2.3. Adapting PKGs

Upon detecting a PIE, we apply one of three **symbolic adaptation strategies** to selectively modify the PKG before passing it to the LLM, as illustrated in Figure 1:

- **Soft Adaptation:** Adjusts ratings of PIE-aligned items by symmetrically inverting their strength around a neutral midpoint, preserving relative preference order. For instance, highly rated tomato-based Italian dishes like "Margherita Pizza" (rating 5) become somewhat lightly rated (rating 2), while "Tomato Sauce Pasta" (rating 4) gets a harsher rating (rating 1), gently nudging recommendations away from tomato dishes while maintaining the user's cuisine preference.
- **Hard Adaptation:** Aggressively assigns the extreme opposite ratings to PIE-aligned items. For example, dishes previously highly favored by the user, such as "Margherita Pizza" (rating 5) and "Tomato Sauce Pasta" (rating 4), receive the lowest possible rating (rating 0), strongly discouraging their recommendation.
- **Removal Adaptation:** Completely eliminates PIE-aligned triples from the PKG. For example, recipes like "Margherita Pizza," which explicitly links Italian cuisine and tomato, are entirely removed, forcing the recommender system to explore alternative items.

PIE Characterization:

- **Out-PIE:** contains F_{given} but not F_{bias}; relevant to the user's stated interest but breaks the learned over-personalization. *Example:* Cannoli (Italian, no tomato) **[preferred outcome]**
- **In-PIE:** contains both F_{given} and F_{bias}; reinforces the over-personalized association the user is trying to avoid. *Example:* Margherita pizza (Italian, tomato)
- **Invalid:** does not contain F_{given}; fails to satisfy the user's original intent or query (e.g., recommending a non-Italian dish when the user asked for Italian). *Example:* Ratatouille (French, tomato)

Tuning the PKG Adaptation Proportion: To control the extent of symbolic intervention, we introduce a user-specific parameter called `adaptProportion`, which determines the fraction of PIE-aligned triples in the PKG that should be adapted before inference. A low `adaptProportion` results in minimal intervention, while a high `adaptProportion` aggressively steers the recommendation away from the PIE.

We learn a personalized `adaptProportion` for each user using a feedback-driven tuning algorithm, which simulates PIE-avoidance scenarios using synthetic data points, and incrementally adjusts `adaptProportion` based on their outcomes. If the adapted PKG still yields `In-PIE` recommendations, the proportion is increased. If it produces `Invalid` results, the proportion is decreased. Successful `Out-PIE` results leave the parameter unchanged. This iterative procedure converges on a personalized adaptation strength tailored to each user's PKG structure and feature biases.

2.4. Model Fine-Tuning

We fine-tune Qwen3-0.6B [10] using Hugging Face's `KTOTrainer` [11], which implements Kahneman-Tversky Optimization (KTO) [12]. Each training data point consists of a *(prompt, completion, label)* triplet: the prompt includes user context and a query, the completion is a potential response to the prompt, and the label is a binary indicator of the completion's quality (positive or negative).

Training data is derived from a customized version of the *Food.com Recipes and Interactions dataset* [13], which includes user ratings (on a 0–5 star scale), ingredients, and categorical tags for recipes. From this corpus, we construct the PKG capturing individual user preferences as a set of rated recipes.

3. Experimental Results

We evaluate our PIE avoidance framework using PKGs derived from the *Food.com Recipes and Interactions dataset* [13]. We randomly select 20 user PKGs and, for each, sample 50 PIE-inducing feature pairs. These are split 80/20 into 40 training and 10 evaluation PIEs per user. The training set is used to learn a personalized `adaptProportion` for each user via the tuning algorithm in Section 2.3. For comparison, we also compute a single global `adaptProportion` across all 800 training PIEs. A learning rate of 0.05 is used in both cases.

Table 1: Performance of PKG adaptation strategies. ↑ better for **Out-PIE**; ↓ better for **In-PIE**, and **Invalid**. The best values are highlighted in green, and the worst values are highlighted in red.

Strategy	Out-PIE ↑	In-PIE ↓	Invalid ↓
Soft (personalized)	0.3237	0.2158	0.4604
Soft (global)	0.2517	0.2583	0.4901
Hard (personalized)	0.2848	0.2152	0.5000
Hard (global)	0.2768	0.1977	0.5254
Removal (personalized)	0.3020	0.2416	0.4564
Removal (global)	0.3277	0.2203	0.4520
Prompt-Based Adaptation	0.1925	0.1863	0.6211
No Adaptation	0.2517	0.2583	0.4901

During evaluation, we generate 10 test queries per user (200 total) and categorize each model-generated recommendation into `Out-PIE`, `In-PIE`, or `Invalid` classes. We aggregate and normalize the counts to obtain proportions summing to one.

We benchmark three PKG adaptation strategies (soft, hard, removal), each with both personalized and global `adaptProportion` settings, against two baselines: (1) a prompt-based method using natural language instructions to avoid PIEs, and (2) a no-adaptation baseline.

As shown in Table 1, soft adaptation with personalized tuning achieves the best overall performance. It increases `Out-PIE` recommendations from 25.2% (global `adaptProportion` baseline) to 32.4%, while simultaneously reducing the `Invalid` rate from 49.0% to 46.0%, avoiding over-personalization without decreasing recommendation quality. In contrast, the prompt-based approach, which attempts PIE avoidance via plain-text instructions, performs poorly, with the lowest `Out-PIE` rate (19.3%) and the

highest `Invalid` rate (62.1%). This underscores the limitations of relying solely on natural language prompting and demonstrates the effectiveness of symbolic PKG adaptations.

4. Conclusion

We introduced a novel, neuro-symbolic framework for enhancing personalization and user agency in recommender systems by adapting PKGs rather than modifying model internals. Unlike conventional approaches that require model retraining or rely on brittle heuristics, our method operates entirely at the user-side knowledge representation layer, enabling efficient, interpretable, and privacy-preserving control over recommendation behavior.

Our key contributions include: (1) a formalization of PIEs as measurable feature-pair biases within PKGs, (2) a suite of symbolic PKG adaptation strategies (Soft, Hard, and Removal) that steer LLMs toward more diverse, Out-PIE content, and (3) a client-side learning algorithm for optimizing user-specific adaptation policies. Through an evaluation on a real-world recipe dataset, we show that personalized PKG adaptation consistently outperforms both global adaptation and natural-language prompting in reducing over-personalization without compromising recommendation quality.

These findings point toward a broader paradigm shift: from tuning black-box models to shaping the symbolic structures that guide them. Our work demonstrates that adapting structured user knowledge offers a powerful, generalizable mechanism for embedding user intent and improving controllability in LLM-powered systems, laying the groundwork for safer, more personalized, and user-aligned AI for diverse recommendations.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Gemini and Grammarly in order to rephrase some of the sentences and also to fix grammar and spelling issues. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Supplemental Materials

All research artifacts, including source code, dataset construction scripts, and result generation pipelines, are available in our GitHub repository. All external datasets and software dependencies used in this work are documented and linked in the repository's README.
https://github.com/brains-group/KGAdaptation.

References

[1] Y. Xi, M. Weng, W. Chen, C. Yi, D. Chen, G. Guo, M. Zhang, J. Wu, Y. Jiang, Q. Liu, et al., Bursting filter bubble: Enhancing serendipity recommendations with aligned large language models, arXiv preprint arXiv:2502.13539 (2025).

[2] E. Pariser, The filter bubble: What the Internet is hiding from you, penguin UK, 2011.

[3] J. Burrell, How the machine 'thinks': Understanding opacity in machine learning algorithms, Big data & society 3 (2016) 2053951715622512.

[4] T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, J. A. Konstan, Exploring the filter bubble: the effect of using recommender systems on content diversity, in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 677–686.

[5] S. Milano, M. Taddeo, L. Floridi, Recommender systems and their ethical challenges, Ai & Society 35 (2020) 957–967.

[6] Y. Zhang, X. Chen, et al., Explainable recommendation: A survey and new perspectives, Foundations and Trends® in Information Retrieval 14 (2020) 1–101.
[7] T. Donkers, J. Ziegler, The dual echo chamber: Modeling social media polarization for interventional recommending, in: Proceedings of the 15th ACM conference on recommender systems, 2021, pp. 12–22.
[8] V. Anand, M. Yang, Z. Zhao, Mitigating filter bubbles within deep recommender systems, arXiv preprint arXiv:2209.08180 (2022).
[9] F. Spadea, O. Seneviratne, Bursting the Filter Bubble with Knowledge Graph Inversion, in: Companion Proceedings of the ACM on Web Science Conference 2025, Association for Computing Machinery, New York, NY, USA, 2025, pp. 39–43. URL: https://doi.org/10.1145/3720554.3736182. doi:10.1145/3720554.3736182.
[10] Hugging Face, Qwen3-0.6b, https://huggingface.co/Qwen/Qwen3-0.6B, 2024.
[11] Hugging Face, KTOTrainer, https://huggingface.co/docs/trl/main/en/kto_trainer#trl.KTOTrainer.ref_model, 2024.
[12] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, D. Kiela, Kto: Model alignment as prospect theoretic optimization, 2024.
[13] S. Li, Food.com recipes and interactions, 2019. URL: https://www.kaggle.com/dsv/783630. doi:10.34740/KAGGLE/DSV/783630.